Scrutinizer Docs

Release 19.7.2

Plixer, LLC

May 28, 2026

VERSION 19.7.2

CHAPTER

ONE

DEPLOYMENT AND CONFIGURATION

Deployment

Deploy your hardware or virtual appliance

Scrutinizer **Appliance setup**

Initial setup, licensing, and SSL configuration

Basic configuration **Configuration**

Tailor Scrutinizer to your requirements

Configuration Guides **Upgrades**

Upgrade to the latest releases

Upgrades and updates

CHAPTER

TWO

EXPLORING SCRUTINIZER

Scrutinizer UI

Scrutinizer features and functionality

Features and Functionality **Use cases**

Common use cases and workflows

Use Cases **Admin functions**

Backup, update, and manage Scrutinizer

Administration and management **Plixer ML Engine**

Enable ML-driven monitoring and detection

Machine learning Click here to view documentation for the Scrutinizer Classic UI.

CHAPTER

THREE

EXPANDING CAPABILITIES

FlowPro

Expand visibility and enable advanced analytics

FlowPro **Replicator**

Replicate and distribute packet streams

Replicator **Endpoint Analytics**

Enable enhanced endpoint monitoring

Endpoint Analytics **Integrations**

Expand functionality through third-party integrations

Integrations

CHAPTER

FOUR

HELP AND REFERENCES

FAQ

Find answers to frequently asked questions

FAQ **Appendices**

Look up details for Scrutinizer's functional elements

Appendices **Changelogs**

Review version history and updates

Changelogs **Other resources**

Glossary and third-party attributions

Other References About Scrutinizer Scrutinizer is a network monitoring and analysis appliance that collects, interprets, and contextualizes data from every digital exchange to deliver network intelligence and security reports. The system consolidates network metadata from existing infrastructure into a unified database through dynamic correlation, scaling to process millions of flows per second via an intuitive web interface.

- **Operational efficiency** - Reduces NetOps and SecOps complexity with actionable insights from raw flow data via context-aware visualizations and real-time updates that quickly identify root causes.
- **Comprehensive monitoring** - Delivers accurate statistics covering bandwidth, application, and user utilization with proactive thresholds, alerts, and DDoS attack detection.
- **Intelligent analysis** - Leverages AI-backed ML Engine capabilities for threat and anomaly detection with open RESTful APIs for streamlined event response.
- **Advanced integration** - Seamlessly combines with other Plixer products to provide targeted functionality for specific environments.

For further questions, check out the *FAQ* or contact *Plixer Technical Support*.

4.1 Deployment Guides

Note

Prior to deploying Scrutinizer and other Plixer One platform products, ensure that firewall rules are correctly configured based on *this table*.

4.1.1 Scrutinizer

Scrutinizer

Deploy and set up your Scrutinizer appliance

Scrutinizer **Plixer ML Engine**

Deploy the Plixer ML Engine

Plixer ML Engine

4.1.2 Plixer One

FlowPro

License, register, and deploy FlowPro probes

FlowPro **Replicator**

Enable the local Replicator instance

Replicator

4.1.2.1 Scrutinizer

Scrutinizer virtual appliances can be deployed in local hypervisors, Amazon Web Services (as an AMI via the AWS Marketplace), Google Cloud Platform, Microsoft Azure, or Oracle Cloud Infrastructure. *Hardware appliances* are also available upon request.

Contact *Plixer Technical Support* or a local reseller for availability and licensing or visit www.plixer.com to learn more.

Note

Scrutinizer virtual appliance packages are also available for download from the Plixer Customer Portal.

On this page:

Virtual appliances	*Virtual appliance deployment*	Hardware appliances	*Hardware appliance deployment*
Basic configuration	*Basic configuration*		

Virtual appliance deployment

Table 1: Basic requirements for virtual appliances

Component	Minimum (for trial installations)	Recommended (for production environments)
Memory	16 GB	24 GB
Storage	100 GB	1+ TB 15K RAID 0 or 10 configuration
Processor	8 CPU cores, 2.0+ GHz	12 CPU cores, 2.0+ GHz

CPU cores and RAM based on flow rate and exporter count

Flow	Exporters							
	5	**25**	**50**	**100**	**200**	**300**	**400**	**500**
5k	8 CPU cores16 GB RAM	8 CPU cores16 GB RAM	10 CPU cores20 GB RAM	14 CPU cores28 GB RAM	20 CPU cores39 GB RAM	26 CPU cores52 GB RAM	32 CPU cores67 GB RAM	38 CPU cores82 GB RAM
10k	8 CPU cores16 GB RAM	8 CPU cores16 GB RAM	12 CPU cores24 GB RAM	18 CPU cores36 GB RAM	25 CPU cores50 GB RAM	32 CPU cores65 GB RAM	38 CPU cores81 GB RAM	43 CPU cores97 GB RAM
20k	16 CPU cores32 GB RAM	16 CPU cores32 GB RAM	16 CPU cores32 GB RAM	24 CPU cores48 GB RAM	32 CPU cores64 GB RAM	38 CPU cores80 GB RAM	43 CPU cores96 GB RAM	48 CPU cores112 GB RAM
50k	32 CPU cores64 GB RAM	32 CPU cores64 GB RAM	32 CPU cores64 GB RAM	32 CPU cores64 GB RAM	39 CPU cores80 GB RAM	44 CPU cores96 GB RAM	48 CPU cores112 GB RAM	52 CPU cores128 GB RAM
75k	46 CPU cores96 GB RAM	46 CPU cores96 GB RAM	46 CPU cores96 GB RAM	46 CPU cores96 GB RAM	46 CPU cores96 GB RAM	49 CPU cores112 GB RAM	52 CPU cores128 GB RAM	55 CPU cores144 GB RAM
100k	52 CPU cores128 GB RAM	52 CPU cores128 GB RAM	52 CPU cores128 GB RAM	52 CPU cores128 GB RAM	52 CPU cores128 GB RAM	52 CPU cores128 GB RAM	55 CPU cores144 GB RAM	58 CPU cores160 GB RAM
125k	58 CPU cores160 GB RAM	58 CPU cores160 GB RAM	58 CPU cores160 GB RAM	58 CPU cores160 GB RAM	58 CPU cores160 GB RAM	58 CPU cores160 GB RAM	58 CPU cores160 GB RAM	61 CPU cores176 GB RAM
150k	64 CPU cores192 GB RAM	64 CPU cores192 GB RAM	64 CPU cores192 GB RAM	64 CPU cores192 GB RAM	64 CPU cores192 GB RAM	64 CPU cores192 GB RAM	64 CPU cores192 GB RAM	64 CPU cores192 GB RAM

Note

- In clustered virtual environments, assign a static MAC address to the Scrutinizer NIC to avoid license key issues.
- Disk sizes can be expanded to support higher flow rates after deployment. A dedicated 15k RPM RAID 10 datastore is recommended for optimal performance.
- See *this guide* for further sizing recommendations.

Local Hypervisors

ESXi

Additional requirements:

- ESXi 6.7 U2+
- VMware vSphere or vCenter

Deploying the OVF template

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest VMware virtual appliance package.
2. Extract the contents of the package to a location on the ESXi server.
3. In vSphere or vCenter, deploy the appliance on a host using the *OVF template* option (this will require the OVF and VMDK files).
4. Select *Thick Provision* for the datastore disk format.
5. After selecting the network to be used by the virtual appliance, verify the configuration in the summary before starting the import operation.
6. After the template has been successfully imported (may take several minutes), assign a static MAC address to the Scrutinizer NIC for licensing purposes.
7. Power on the VM.

After the Scrutinizer virtual appliance completes booting, proceed with the *initial appliance setup*.

> **Note**
>
> To upgrade the virtual machine's hardware version to the latest ESXi version, select **Compatibility > Upgrade VM Compatibility** in vSphere or vCenter while the VM is powered off. When the VM is powered back on after the upgrade, it will boot up with the latest ESXi hardware version available.

Expanding database size

To allocate additional storage to the Scrutinizer database, follow these steps:

View instructions

1. Power off the Scrutinizer VM.
2. Add a new hard disk to the device.
3. Select the type of disk provisioning based on *these recommendations*.
4. Confirm to add the new disk.

Once the new disk has been added, power on the VM and follow *this guide* to make it available to Scrutinizer.

Hyper-V

Additional requirements:

- Generation 2 Hyper-V VM
- Hyper-V 2012
- Hyper-V Manager

Deploying the Hyper-V virtual appliance

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest Hyper-V virtual appliance package:
2. Extract the contents of the package to a location on the Hyper-V server.
3. In Hyper-V Manager, select the option to import a VM, and then select the **Scrutinizer Hyper-V** image.

4. After the image has been imported, provision the Scrutinizer VM based on the *recommended sizing* for the expected flow rate.
5. Select a network adapter and assign it to the appropriate virtual switch.
6. Assign a static MAC address to the VM.
7. Save the updated settings, and then start the VM.

After the Scrutinizer virtual appliance completes booting, connect to the VM and then proceed with the *initial appliance setup*.

Expanding database size

Depending on the volume of NetFlow data that will be forwarded to the Scrutinizer virtual appliance, it may be necessary to allocate *additional storage space* for its database.

To add a hard drive to the Scrutinizer virtual machine, follow these steps:

1. Power off the Scrutinizer VM.
2. In Hyper-V Manager, select the option to add a new virtual hard drive in the VM's settings.
3. Select *VHDX* as the disk format (supports expansion past 2 TB).
4. Configure the other disk settings as needed.

Once the new drive has been added, power the VM on and follow *this guide* to make it available to Scrutinizer.

KVM

Additional requirements:

- KVM 16 or higher

Deploying the KVM virtual appliance

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest KVM virtual appliance package.
2. Create a directory for the install:

```
mkdir /kvm/scrutinizer_vm/
```

3. Extract the contents of the package to the new directory:

```
sudo tar xvzf PACKAGE_FILENAME.tar.gz -C /kvm/scrutinizer_vm/
```

4. Run the installation script in the new directory:

```
cd /kvm/scrutinizer_vm/PACKAGE_FILENAME
sudo ./install-kvm-scrut.sh
```

5. Wait for the confirmation that the virtual machine has been created from the image.

After the VM starts up, access the console using `virsh console <VM_DOMAIN_OR_ID>` to proceed with the *initial appliance setup*.

Nutanix

Deploying the virtual appliance in Nutanix

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest qcow2 image file.
2. Log in to Prism Element and upload the image (as a disk) to any storage container (except *SelfServiceContainer*).
3. After the image becomes active, create a new VM with the following configuration:
   - *Resources*: *Recommended sizing* (minimum of 8 cores and 16 GB RAM, fewer CPUs with more cores is recommended)
   - *Boot configuration*: UEFI
   - *Operation*: Clone from image
   - *Bus type*: SATA (SCSI is not recommended due to known issues with Red Hat 9 systems)
   - *Image*: Image/disk uploaded in step 3
   - *Index*: Next available
4. Add a new NIC to the VM and assign it to the desired subnet.
5. Save the VM configuration, and then power on the VM.

After the Scrutinizer virtual appliance completes booting, launch the console to proceed with the *initial appliance setup*.

Proxmox

Note

- When attaching the imported disk (step 4), verify that its name matches what's displayed in the GUI.
- The syntax in the instructions below should be modified to match the actual VMID and disk names/numbers used.

Deploying the virtual appliance in Proxmox

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest qcow2 image file.
2. Create a new virtual machine in Proxmox with the following configuration:
   - BIOS: OVMF (UEFI)
   - SCSI controller: VMware PVSCSI
   - Network adapter: E1000
   - CPU/memory: *Recommended sizing*
   - Add a new EFI disk with default sizing
3. Import the disk via the CLI:

   ```
   qm importdisk VMID /var/lib/vz/template/Plixer_Scrutinizer.qcow2 ZFS_DISK_NAME
   ```

   Example:

   ```
   qm importdisk 100 /var/lib/vz/template/Plixer_Scrutinizer.qcow2 local-zfs
   ```

4. Attach the imported disk to the virtual machine:

   ```
   qm set VMID -scsi0 local-zfs:VM_DISK_NAME
   ```

   Example:

   ```
   qm set 100 -scsi0 local-zfs:vm-101-disk-1
   ```

5. Remove/delete the unused disk (the default disk created when the VM was added in Proxmox).
6. Start the VM.

After the VM starts up, access the console to proceed with the *initial appliance setup*.

Cloud platforms

Amazon Web Services AMI

Deploying the Scrutinizer AMI

After subscribing to the service via the AWS Marketplace product page, deploy the Scrutinizer AMI by creating/launching a new EC2 instance with the following configuration:

- *Names and tags*: Configure the name, resource types, and optional tags for the instance.
- *Application and OS images*: Select the Scrutinizer AMI from the **My AMIs** tab.
- *Instance type*: Select *C5.2xlarge* for flow rates up to 10,000 flows per second (contact *Plixer Technical Support* for assistance if the expected flow volume exceeds that).
- *Key pair*: Select or create a new key pair to assign to the instance.
- *Network settings*: Select the VPC, subnet, and security group to assign the instance to.

  > **Important**
  >
  > Because an active instance's primary private IP address cannot be released, we recommend deploying the AMI with two NICs and using the secondary as the collection interface.

- *Storage*: Leave the size of the root volume (`/dev/xvda/`) at the default 100 GB.
- *Advanced details*: Set *Shutdown behavior* to **Stop** and *Termination protection* to **Enabled**.

After the instance has been launched, access the Scrutinizer web interface via the instance's primary private or public IP address, and then proceed to *add a license*.

> **Note**
>
> - For AMI deployments, the default password for the web interface `admin` user is the AWS instance ID of the Scrutinizer instance, which can be copied from the **Instance Summary** view of the EC2 interface.
> - Use the following command to SSH to the server as the `plixer` user after the instance has been launched:
>
>   ```
>   ssh -i PATH_TO_KEY/key.pem plixer@SCRUTINIZER_IP
>   ```

Expanding database size

To expand the database size for a Scrutinizer AMI, create one or more additional EBS volumes in the same *availability zone* and attach them to the instance.

These volumes can then be made available to Scrutinizer by following *this guide*.

> **Note**
>
> `set partitions` (step 6 in the guide) will need to be run from the scrut_util prompt for each additional drive attached to the instance:
>
> ```
> SCRUTINIZER> set partitions <NEW_PARTITION>
> ```

Changing instance types

Follow these steps to change the Scrutinizer instance type to increase *CPU and RAM allocations*:

1. SSH to the instance as the `plixer` user and stop all services via *scrut_util*:

   ```
   SCRUTINIZER> services all stop
   ```

2. Power off the OS:

   ```
   shutdown -h now
   ```

3. Stop the instance. If an Elastic IP was assigned, note the instance ID and Elastic IP address beforehand.
4. Change the instance type and restart the instance following this guide.
5. Verify that a new public DNS (IPv4), Private DNS, and Private IPs have been assigned. The Elastic IP address should also be re-assigned to the instance ID if necessary.

After the instance has been reconfigured, SSH to the Scrutinizer IP address as the `plixer` user and run the following *scrut_util command* to re-tune the system:

```
SCRUTINIZER> set tuning
```

Google Cloud Platform

Additional requirements:

- A GCP project with *Billing*, *Compute Engine*, and *Migrate to Virtual Machines* enabled
- Permissions to create *Compute Engine images*, *Compute Engine VM instances*, and *Cloud Storage buckets* (if not using an existing bucket)
- A cloud storage bucket on the region intended for the VM (for staging the image)

Importing and deploying the Scrutinizer VM

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest VMware virtual appliance OVA package.
2. Upload the image to the staging bucket.
3. Select the option to import a **machine image** and use the following settings:
   - *Source*: Cloud Storage

  - *File*: Select the uploaded OVA
  - *Operating system*: RHEL 9

   This operation will create a reusable custom image and may take up to 15 minutes. The image must be successfully imported before the Scrutinizer VM can be created.

4. Create a new VM instance with the machine type most closely matching the *recommended resources* for the expected flow volume (*n4* or *c4* recommended).
5. Configure the OS and storage settings for the VM as follows:
   - *Boot disk*: The imported Scrutinizer image
   - *Disk type*: *Hyperdisk Balanced* (required for C4/N4 machine types)
   - *Disk size*: Adjust to match *storage requirements* (minimum of 100 GB)
6. Configure the networking settings for the VM as follows
   - Assign an external IPv4 address (ephemeral).
   - Enable HTTPS traffic through the firewall.
   - Add a network tag: *scrutinizer-https*.
   - Assign a hostname (optional but recommended).
7. Verify that all settings were configured correctly, and then create/launch the VM.

After the instance has been launched, connect to the VM via serial console (see below if not already enabled for the project) to proceed with the *initial appliance setup*.

Enabling serial console access

Serial console access (project-level setting) can be enabled for first boot validation and troubleshooting.

In the GCP console, edit the metadata settings for the Compute Engine to add the following:

- *Key*: `serial-port-enable`
- *Value*: `true`

The option to connect to the Scrutinizer VM via serial console will become available after the new key is saved.

Expanding database size

To expand the database size for a Scrutinizer appliance deployed on GCP, first add a new disk via the GCP console:

> **Note**
>
> A new disk can be added while the VM is running.

1. Select the option to edit the Scrutinizer VM in the GCP console.
2. Add a new disk with the following configuration.
   - *Disk type*: Select the same type as the boot disk.
   - *Disk size*: As needed
3. Save/create the new disk.

After the new disk has been added, follow *this guide* to make it available to Scrutinizer.

Microsoft Azure

Additional requirements:

- A Windows 10+ or Windows Server host with Internet access, at least 200 GB free disk space, and Hyper-V installed
- Administrator permissions (including PowerShell commands) on the Windows host
- Administrator credentials for the Azure account the Scrutinizer virtual appliance will be deployed on

Uploading and deploying the Scrutinizer VM

Important

Replace the file paths in step 3 and 6 below with the correct paths to the downloaded and converted files in your environment.

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest Hyper-V virtual appliance package on the Windows host.
2. Extract the VHD (`Scrutinizer.vhdx`) from the file.
3. Start a PowerShell session on the Windows host, and then convert the disk image to fixed size in Powershell:

```
Convert-VHD -Path 'C:\Users\User␣
↪Name\Downloads\Scrut-hyperv\Scrutinizer_Hyper-V\Virtual Hard␣
↪Disks\Scrutinizer.vhdx' -Destination 'C:\tmp\Scrutinizer.vhd' -VHDType Fixed
```

4. Install the **Az PowerShell module**:

```
Install-Module -Name Az
```

5. Authenticate the Windows PowerShell session with the Azure account to be used for deployment:

```
Connect-AzAccount
```

Note

If the connection fails after the correct Azure credentials are entered, run the following:

```
Set-ExecutionPolicy RemoteSigned
```

6. Upload the Scrutinizer VHD to Azure as a managed disk (replace `RESOURCE_GROUP`, `AZURE_REGION`, and `DISK_NAME` below with the correct details):

```
Add-AzVhd -LocalFilePath 'C:\tmp\Scrutinizer.vhd' -ResourceGroupName␣
↪RESOURCE_GROUP -Location AZURE_REGION -DiskName DISK_NAME -DiskHyperVGeneration␣
↪V2 -DiskOsType Linux
```

7. After the Scrutinizer VHD has been uploaded, deploy a new VM using the disk image from the Azure portal (note the IP address assigned to the VM as this will be required when setting up the appliance).
8. Launch/start the VM.

After the Scrutinizer VM completes booting, SSH to the IP address assigned as the `plixer` user to proceed with the *initial appliance setup*.

Oracle Cloud Infrastructure

Additional requirements:

- A cloud storage bucket (for staging the image)
- Gateway and netmask of the OCI VNC subnet that Scrutinizer will be deployed on

Importing and deploying the Scrutinizer VM

1. Log in to the Plixer Customer Portal or use the link provided by *Plixer Technical Support* to download the latest VMware virtual appliance package.
2. If necessary, extract the OVA (`Scrutinizer_Vmware_19.7.2-bios.ova`) from the file.
3. Upload the image to the storage bucket.
4. Create a new custom image by importing the uploaded file from the storage bucket with the following settings:
   - *Operating system*: Oracle Linux
   - *Image type*: VMDK
   - *Launch mode*: Emulated (required)
5. Create a new VM instance using the custom image and configure the following settings:
   - Select the custom image created in the previous step.
   - Select an image shape (e.g., VM.Standard.E5.Flex) and expand the CPU core count and memory allocation to match the *recommended resourcing* for the expected flow volume.
   - Enter a primary VNIC name (required for the Scrutinizer VM).
   - Manually assign a private IPv4 address to use as the static address for the Scrutinizer appliance (must be entered during appliance setup).
   - Add public or generated keys for SSH access.
   - Adjust the boot volume size based on *these storage recommendations* and keep VPU at the default value.
6. Save the instance configuration and start/launch the VM.

After obtaining the required details, SSH to the VM as the `plixer` user to proceed with the *initial appliance setup*.

Allocating additional storage

If the boot volume size defined when the VM instance was created was greater than 100 GB, make the additional storage available to Scrutinizer as follows:

Important

The steps below should be performed **after** the *initial appliance setup* has been completed.

1. SSH to the Scrutinizer VM as the `plixer` user and elevate to root:

```
su -
```

2. Run the following and enter `Fix` at the prompt that follows:

```
parted -l
```

3. Create a new partition:

   ```
   fdisk /dev/sda
   ```

   1. Enter `p` to verify the current partitions (`sda1`, `sda2`, and `sda3`).
   2. Enter `n` for Command, and then enter `4` for the partition number; afterwards, press **Enter** twice to keep the default values.
   3. Enter `t` for Command, `4` to select the partition, and then enter `30` to enable Linux LVM for the partition.
   4. Enter `w` to save the changes.

4. Restart the VM:

   ```
   reboot
   ```

5. Reconnect as the `plixer` user, elevate to root, and verify the previous changes:

   ```
   su -
   fdisk -l
   ```

6. Add the new partition:

   ```
   vgextend vg_scrut /dev/sda4
   ```

7. Allocate the storage (in excess of 100 GB) to the `root` and `db` logical volumes as needed (replace `X` and `Y` below with the desired storage allocations in GB):

   ```
   lvextend -L+XG /dev/vg_scrut/lv_db
   lvextend -L+YG /dev/vg_scrut/lv_root
   ```

8. Apply the changes:

   ```
   resize2fs /dev/vg_scrut/lv_db
   resize2fs /dev/vg_scrut/lv_root
   ```

9. Verify that the volume sizes have been expanded successfully:

   ```
   df -h | grep 'lv_'
   ```

When done, the additional storage will be available for use by the Scrutinizer VM/server.

Hardware appliance deployment

Scrutinizer hardware appliances support higher collection rates due to their dedicated resources and are strongly recommended for environments with extremely high flow volumes. They are available through *Plixer Technical Support*.

After removing the Scrutinizer hardware appliance from its packaging, verify that all accompanying accessories (rack-mount kit, appliance-locking bezel and keys, and power cord) are included. The appliance can be mounted in a standard 19-inch rack or cabinet.

Important

If your box arrives torn, dented, or otherwise damaged, the appliance itself seems damaged, or there are missing parts, *contact Plixer Technical Support* immediately and **do not attempt to install the unit**.

Hardware setup

1. Refer to the port labels to identify the ports to be used on the rear panel of the appliance:
   - iDRAC
   - Serial
   - VGA
   - USB Type-B x 2
   - 10GbE SFP x 2 (1 and 2)
   - 1GbE RJ45 x 2 (3 and 4)
   - Power supply x 2
2. Connect the power cable to one of the power supply sockets and plug the other end to a grounded AC outlet or UPS. To take advantage of the redundant PSUs, ensure that each socket is connected to an independent power source.
3. Depending on the bandwidth requirements of the environment, connect the appliance to the network using either RJ-45 or fiber optic cables. Unused ports may be left uncabled, but connecting both ports of either pair is recommended for high availability.
4. [Optional] Connect the iDRAC port to a remote access controller using an RJ-45 cable to enable remote console access for hardware management and monitoring. *Contact Plixer Technical Support* for help with configuring alerts for hardware-related events.
5. Using the additional ports provided, connect a monitor and keyboard to use during the appliance's initial setup.

Once the Scrutinizer hardware appliance has been set up and cabled, proceed with the *initial appliance setup*.

Note

- The Ethernet port pairs are configured for adapting load balancing (bonding mode 6).
- The iDRAC virtual console can also be used for the appliance's *initial setup*.

Basic configuration

After deploying and starting the appliance, follow the basic configuration steps below to prepare Scrutinizer for use.

Initial setup

After the Scrutinizer appliance completes its first boot sequence and a user logs in with the credentials `plixer:plixer`, it will perform a quick preliminary setup before rebooting itself.

After the reboot, log in again to start the initial setup script:

1. Provide the following information when prompted by the script:
   - Static IP address
   - Netmask

   - Gateway
   - FQDN
   - DNS IP address
   - NTP server IP address

2. Enter any additional information requested.
3. At the end of the script, press *Enter* and wait for the server to reboot again to apply the settings.

After the final appliance reboot, log in to the web interface at the IP address provided with the default `admin:admin` credentials and proceed to *add a license*.

> **Note**
>
> - The default password for the web interface `admin` account can be changed from the Admin > Users & Groups > User Accounts page.
> - The default self-signed certificate can be *replaced with a CA-signed certificate* if desired.

Adding a license

To add/register a Plixer One or Scrutinizer license key, navigate to Admin > Plixer > Scrutinizer Licensing in the web interface after completing the *initial appliance setup process*.

A license key can be obtained by contacting *Plixer Technical Support* and providing them with the *Machine ID* displayed on the licensing page. The key should then be pasted into the *License Key field* and saved.

Details for the current license (validity, appliance/server counts, etc.) will be displayed on the page after a key has been added.

Configuring SSL

SSL support is automatically enabled during the initial setup process for a Scrutinizer server. A self-signed SSL certificate with default values is created at the same time.

This self-signed certificate can later be replaced with a CA-signed certificate if desired.

> **Note**
>
> To learn more about additional certificate-related functions, see *this page*.

Installing a CA-signed SSL certificate

As long as the system is set to use the self-signed SSL certificate created during the initial setup process, browsers will return an untrusted certificate warning, which users must override to access the web interface.

To avoid this, an SSL certificate that has been signed by an internal or commercial Certificate Authority (CA) will need to be installed.

Generating a custom certificate signing request (CSR)

1. SSH to the primary reporter as the `plixer` user:

```
ssh plixer@PRIMARY_REPORTER_IP
```

2. [Optional] Create a new directory for the custom CSR, keys, and certificates:

   ```
   sudo mkdir /home/plixer/CustomCerts
   ```

   This will provide a static location for storing and managing future certificates.

3. Create a CSR config/details file:

   ```
   sudo touch /home/plixer/CustomCerts/csr_config.txt
   ```

   Tip

   - If the details for the CSR do not change from year to year, `csr_config.txt` can be re-used to create a new CSR when the old certificate expires.
   - When generating a new CSR, key, and certificate, including a date in the filename will help identify the correct files in case future changes (e.g., upgrades) overwrite the existing certificate.

4. Add the details for the CSR to `csr_config.txt` in the following format:

   ```
   [req]
   default_bits=2048
   prompt=no
   default_md=sha256
   req_extensions=req_ext
   distinguished_name=dn

   [dn]
   C=US
   ST=Maine
   L=Kennebunk
   O=Plixer, LLC
   OU=IT
   emailAddress=support@plixer.com
   CN=scrutinizer.plxr.local

   [req_ext]
   subjectAltName=@alt_names

   [alt_names]
   DNS.1=scrutinizer.plxr.local
   ```

   Note

   `[alt_names]` is now required. To specify multiple Subject Alternative Names (SANs), use one line for each entry, with incrementing DNS numbers (DNS.2=, DNS.3=, etc.).

5. Generate the new CSR and key:

```
cd /home/plixer/CustomCerts
sudo openssl req -new -sha256 -nodes -out newRequest.csr -newkey rsa:4096 -keyout␣
↪newCaKey.key -config csr_config.txt
```

The custom CSR (`/home/plixer/CustomCerts/newRequest.csr`) can then be sent to any preferred CA for signing.

Installing the signed certificate

Important

In some cases, Scrutinizer 19.5.x and Replicator 19.01 deployments will also have `localhost.crt` and `localhost.key` files in addition to `ca.crt` and `ca.key`. These files were generated during the deployment/upgrade process but should not be used.

The following steps will ensure that the correct certificates are in place and in use:

View instructions

1. Verify `localhost.crt` and `localhost.key` do not exist on the appliance:

   ```
   sudo ls /etc/pki/tls/certs/
   sudo ls /etc/pki/tls/private/
   ```

   If neither file exists, no further action is required.

2. If either of the previous commands discovers the corresponding `localhost` file, update the appliance to look for the correct files:

   ```
   sudo sed -i 's/localhost.crt/ca.crt/g' /etc/httpd/conf.d/ssl.conf
   sudo sed -i 's/localhost.key/ca.key/g' /etc/httpd/conf.d/ssl.conf
   sudo chmod 600 /etc/pki/tls/certs/ca.crt
   sudo chmod 600 /etc/pki/tls/private/ca.key
   sudo mv /etc/pki/tls/certs/localhost.crt /etc/pki/tls/certs.ca.crt
   sudo mv /etc/pki/tls/private/localhost.crt /etc/pki/tls/private/ca.key
   ```

3. Restart the httpd service:

   ```
   sudo systemctl restart httpd
   ```

After receiving the CA-signed certificate, follow these steps to install it:

1. Copy the new certificate to the `/home/plixer/CustomCerts` directory (or any temporary directory if `CustomCerts` was not previously created) on the Scrutinizer server.

2. Backup the current CA certificate and key:

   ```
   sudo cp /etc/pki/tls/certs/ca.crt /etc/pki/tls/certs/ca.crt.bak
   sudo cp /etc/pki/tls/private/ca.key /etc/pki/tls/private/ca.key.bak
   ```

3. Move the new certificate to the correct location:

   ```
   cp /home/plixer/CustomCerts/CA_CERT_FILENAME.crt /etc/pki/tls/certs/ca.crt
   ```

4. Move the new key generated with the CSR to the correct location:

```
sudo cp /home/plixer/CustomCerts/NEW_KEY_FILENAME.key /etc/pki/tls/private/ca.key
```

If the `CustomCerts` directory was not created/used, the key can be found in the same directory the CSR was generated in.

5) Restart the nginx service (httpd on pre-v19.7.0 Scrutinizer or pre-v20.0.0 Replicator deployments):

```
sudo systemctl restart nginx
```

To verify that the web interface is using the correct SSL certificate, use a browser to navigate to the login page using the FQDN specified in the CA-signed certificate. The browser should no longer return an untrusted certificate warning and the padlock icon in the address bar should be locked instead of open.

Note

The private key may need to be encrypted with the `/usr/bin/ask.sh` passphrase:

```
openssl rsa -in server.key -out server.key.new
```

Non-default CSR configurations

Certificate signing requests can also be generated with non-default configurations (stronger encryption, no email address, etc.) using the values in the `csr_config.txt` file in the *above instructions*.

After the desired configuration has been saved, continue to follow the same instructions to generate the CSR and install the CA-signed certificate.

4.1.2.2 Plixer ML Engine

After deploying Scrutinizer, follow the steps below to deploy the Plixer ML Engine (requires Plixer One Enterprise).

Pre-deployment preparations

The following preparatory steps should be completed before starting the deployment procedure for any type of Plixer ML Engine appliance.

Deploying the ML VM

Use the template obtained with the Plixer One Enterprise license to deploy the ML VM locally.

This VM will function as a separate deployment host and includes all prerequisite resources. The ML engine environment will be deployed and managed from this VM.

> **Note**
>
> When connecting to the VM via PuTTY, *VT100 line drawing even in UTF-8 mode* (under Settings > Window > Translation) must be enabled for the setup wizard to be displayed correctly. Requested details can be pasted into the prompts/dialogs using **Shift+Insert**.

KVM deployment

To deploy the ML VM image on a KVM host running **libvirt**, follow these steps:

View instructions

1. Download the KVM package:

```
wget --no-check-certificate https://
↪files.plixer.com/downloads/mlengine/19/MLEngine_KVM_19.7.0.tar.gz
```

2. Extract the contents of the package:

```
tar xvfz ML_KVM_19.7.0.tar.gz
```

3. Load the image (`MLEngine_KVM_Image.qcow2`):

```
virt-install \
--name MLEngine \
--memory 40960 \
--boot uefi \
--vcpus 8 \
--disk path=/PATH/TO/MLEngine_KVM_Image.qcow2,format=qcow2,bus=virtio \
--controller virtio-serial \
--os-variant ubuntu22.04 \
--network default \
--graphics vnc \
--import \
--noautoconsole
```

Once the image has been loaded, proceed to *deploying the appliance/cluster*.

Registering the ML engine

Before deploying an ML engine, the appliance must first be registered as follows:

1. In the Scrutinizer web interface, navigate to **Admin** > **Resources** > **ML Engines**.
2. Click the **+** button to add a new ML engine.
3. From the dropdown, select the type of ML engine paired with your Scrutinizer environment:
   - *Single VM*
   - *Amazon AWS*
   - *Azure*
   - *vSphere multi VM Cluster*
4. Enter a name to assign to the engine, and then click **Save**.

After returning to the main view, click the name of the new ML engine and save/copy the primary reporter address and authentication token shown in the tray. These will be required during the ML engine deployment process.

Confirming SSH credentials

To complete the appliance setup process, the ML engine will need to establish an SSH session with the primary reporter/server of the Scrutinizer environment. As such, the IP address of the primary reporter and the `plixer` user password will need to be provided.

If a private SSH key is required, verify that the public key is configured on the primary Scrutinizer reporter/server under `/home/plixer/.ssh/authorized_keys`. The private key should also be accessible from the machine hosting the ML engine virtual appliance, as the path to the key will need to be entered during the appliance setup process.

Deployment guides

Important

Scrutinizer 19.7.2 environments require v19.5+ of the Plixer ML Engine (v19.7 is recommended). For Scrutinizer 19.5.3 and below, see this deployment guide for v19.4 of the ML engine.

Note

When connecting to the ML VM via PuTTY, *VT100 line drawing even in UTF-8 mode* (under Settings > Window > Translation) must be enabled for the setup wizard to be displayed correctly. Requested details can be pasted into the prompts/dialogs using **Shift+Insert**.

Single node (local)

To deploy a local single-node Plixer ML Engine appliance, follow these steps after the *pre-deployment steps* have been completed:

1. Log in to the ML VM image using `plixer:plixer`.
2. Accept the EULA, and then configure network settings for the host.
3. SSH to the ML VM image using the `plixer` credentials set in step 2, and then wait for the setup wizard/scripts to start automatically.
4. Enter the following when prompted:
   - *Authentication token*
   - Primary reporter IP address
   - *Scrutinizer SSH credentials*

After the scripts complete running, navigate to Admin > Resources > ML Engines and wait for the engine to show as *Deployed* under its *Deploy Status*. Refresh the page if the status has not updated after a few minutes.

See *this guide* for configuration instructions and recommendations for the ML engine.

AWS (EKS)

Follow these steps to deploy the Plixer ML Engine multi-node cluster in AWS.

Additional prerequisites for AWS

- AWS IAM user secret access key ID and secret access key
- A VPC with two subnets for the deployment
- At least 60 IP addresses available in the VPC subnets for use by AWS (EKS)

Note

- The ML VM (the deployment host) deployed as part of the *pre-deployment preparations* will have all software prerequisites (Docker, Terraform, etc.) preinstalled.
- The setup scripts include an option to automatically set up a new VPC and will prompt the user to enter the necessary information.
- For existing VPCs, the following requirements must be met:
  - The VPC must have a DHCP option set with the option to use **AmazonProvidedDNS** for its domain name servers.
  - The VPC must have two private subnets on separate Availability Zones (AZs).
  - If the subnets cannot access the Internet (no NAT gateway attached), set `airgap_install` in `/home/plixer/common/kubernetes/aws.tfvars` to `TRUE`.
- For additional information on Amazon EKS VPC and subnet requirements and considerations, see this article.

Hybrid cloud deployments

When pairing an ML engine in AWS with an on-prem Scrutinizer environment, one of the following methods should be used to enable connectivity between the two before starting the deployment process.

AWS Site-to-Site VPN

Follow these instructions to create an AWS Site-to-Site VPN connection to allow communication between the two deployments.

Direct access via public IP

A public IP address can be used to allow external access to the on-prem Scrutinizer deployment. However, this will expose the Scrutinizer environment to the Internet via ports **5432**, **22**, and **443**.

The public IP address must be entered when prompted by the setup scripts. The Internet gateway IP must also be manually added to the Scrutinizer `pg_hba.conf` file to allow access to Postgres.

After the file has been modified, run the following command on the Scrutinizer server to reload the configuration:

```
psql -c "SELECT pg_reload_conf()"
```

Deploying the ML engine

Follow these instructions to set up the necessary infrastructure and deploy the ML engine:

1. Log in to the ML VM image using `plixer:plixer`.
2. Accept the EULA, and then configure network settings for the host.
3. SSH to the ML VM image using the `plixer` credentials set in step 2, and then wait for the setup wizard/scripts to start automatically.

4. Enter the *infrastructure deployment parameters* as prompted.

   > **Note**
   >
   > The requested details are automatically saved to `/home/plixer/common/kubernetes/aws.tfvars`, which also contains *other default parameters* for deploying the ML engine Kubernetes cluster. If there are issues with the infrastructure deployment, contact *Plixer Technical Support* for assistance before making changes to the file.

5. Wait as the Kubernetes cluster is deployed (may take several minutes), and then enter the Scrutinizer SSH credentials when prompted.

After the scripts complete running, navigate to Admin > Resources > ML Engines and wait for the engine to show as *Deployed* under its *Deploy Status*. Refresh the page if the status has not updated after a few minutes.

See *this guide* for configuration instructions and recommendations for the ML engine.

Terraform configuration

The following table lists all required and optional variables in `/home/plixer/common/kubernetes/aws.tfvars`, which are used when deploying the Kubernetes infrastructure for the ML engine.

> **Note**
>
> Contact *Plixer Technical Support* before making changes to this file.

Field name	Description
cluster_name	**REQUIRED**: Name to identify the ML engine cluster/deployment; can only contain the characters **a** to **z** (in lowercase), **0** to **9**, and **-** .
creator	**REQUIRED**: This is the name of the person creating these AWS resources, used as a tag in AWS to track utilization.
cost_center	**REQUIRED**: This is the cost center to use for these AWS resources, used as a tag in AWS to track utilization.
aws_certifica	**REQUIRED**: This is the name of an existing SSH certificate configured in your AWS environment. You can see a list of these in your AWS Console by navigating to **EC2 > Network > Security > Key Pairs**.
instance_type	**REQUIRED**: This is the AWS instance type to create for EKS worker nodes (i.e. t2.large).
fargate	**REQUIRED**: Use fargate instead of EKS nodes for applicable workloads. Setting the value to TRUE will allow using a smaller instance_type.
aws_region	**REQUIRED**: The AWS region to deploy infrastructure in.
airgap_install	OPTIONAL: If this is an airgapped install (i.e. the vpc_private_subnets don't have a route to a NAT gateway), then set this to TRUE.
create_ec2_end	OPTIONAL: If airgap_install = TRUE, this bool controls whether or not to create an EC2 endpoint in the VPC.
create_s3_endp	OPTIONAL: If airgap_install = TRUE, this bool controls whether or not to create an S3 endpoint in the VPC.
create_ecr_end	OPTIONAL: If airgap_install = TRUE, this bool controls whether or not to create an ECR endpoint in the VPC.
create_ssm_enc	OPTIONAL: If airgap_install = TRUE, this bool controls whether or not to create an SSM endpoint in the VPC.
new_vpc_ci	OPTIONAL: If you want to create a new VPC, then specify the IP address range in this field.
new_vpc_pu	OPTIONAL: If you want to create a new VPC, then specify the IP address range for the public subnet in the new VPC.
new_vpc_pr	OPTIONAL: If you want to create a new VPC, then specify the IP address range for the private subnet in the new VPC.
azs	OPTIONAL: Availability zones corresponding to the subnets you want created in new_vpc_public_cidr and new_vpc_private_cidr.
vpc_name	OPTIONAL: Existing vpc_name to create the EKS resources in.
vpc_private_	OPTIONAL: List of private subnet names to create the EKS resources in.
vpc_public_	OPTIONAL: List of public subnet names to create the EKS resources in.

Azure (AKS)

Follow these steps to deploy the Plixer ML Engine multi-node cluster in Azure after the *pre-deployment steps* have been completed:

Additional prerequisites for Azure

- Credentials for the Azure user account that will be used for deployment
- A VNet with one subnet for the deployment

Note

- The ML VM (the deployment host) deployed as part of the *pre-deployment preparations* will have all software prerequisites (Docker, Terraform, etc.) preinstalled.

- The Azure user account must be assigned the `owner` role to allow a role to be assigned to the AKS cluster user.
- VNet details for infrastructure deployment can be defined using the `vnet_addresses` and `new_subnet_cidr` fields in */home/plixer/common/kubernetes/azure.tfvars*.

Hybrid cloud deployments

When pairing an ML engine in Azure with an on-prem Scrutinizer environment, one of the following methods should be used to enable connectivity between the two before starting the deployment process.

Azure site-to-site (S2S) VPN

Follow these instructions to create a site-to-site VPN connection to allow communication between the two deployments.

Direct access via public IP

A public IP address can be used to allow external access to the on-prem Scrutinizer deployment. However, this will expose the Scrutinizer environment to the Internet via ports **5432**, **22**, and **443**.

The public IP address must be entered when prompted by the `01_azure_infrastructure.sh` and `setup.sh` scripts. The Internet gateway IP must also be manually added to the Scrutinizer `pg_hba.conf` file to allow access to Postgres.

After the file has been modified, run the following command on the Scrutinizer server to reload the configuration:

```
psql -c "SELECT pg_reload_conf()"
```

Deploying the Kubernetes infrastructure

1. Log in to the ML VM image using `plixer:plixer`.
2. Accept the EULA, and then configure network settings for the host.
3. SSH to the ML VM image using the `plixer` credentials set in step 2.
4. Exit the automated setup wizard by pressing **Ctrl + C**.
5. Start the Azure CLI and run the following to set up the client and log in:

   ```
   az login
   ```

6. Define the *infrastructure deployment parameters* in `/home/plixer/common/kubernetes/azure.tfvars` (as described in the file).

   > **Note**
   >
   > `azure.tfvars` may also include fields/variables with factory-defined values (e.g., `kube_version`) for deploying the ML engine Kubernetes cluster. Contact *Plixer Technical Support* for assistance before making changes to any default value.

7. Navigate to `/home/plixer/common/kubernetes` and run the Kubernetes cluster deployment script:

   ```
   ./01_azure_infrastructure.sh
   ```

8. Verify that the infrastructure was successfully deployed (may take several minutes):

```
kubectl get nodes
```

After confirming the Kubernetes cluster has been correctly deployed, proceed to deploying the ML engine.

Deploying the ML engine

Once the Kubernetes cluster has been deployed, follow these steps to deploy the ML engine:

1. Navigate to the `/home/plixer/ml` directory on the deployment host.
2. Run the ML engine deployment script and follow the prompts to set up the appliance:

   ```
   ./setup.sh
   ```

3. When prompted, enter the following Scrutinizer environment details:
   - *Authentication token*
   - Primary reporter IP address
   - *SSH credentials*

After the script completes running, navigate to Admin > Resources > ML Engines and wait for the engine to show as *Deployed* under its *Deploy Status*. Refresh the page if the status has not updated after a few minutes.

See *this guide* for configuration instructions and recommendations for the ML engine.

Terraform configuration

The following table lists all required and optional variables in `/home/plixer/common/kubernetes/azure.tfvars`, which are used when deploying the Kubernetes infrastructure for the ML engine.

Field name	Description
cluster_name	**REQUIRED**: Name to identify the ML engine cluster/deployment; can only contain the characters **a** to **z** (in lowercase), **0** to **9**, and **-** .
vm_type	**REQUIRED**: This is the Azure VM instance type to create for AKS worker nodes.
location	**REQUIRED**: This is the location to create the AKS worker nodes in (e.g. East US 2).
resource_gr	OPTIONAL: Name of existing resource group to use when deploying assets. If empty, a new resource group named *${var.cluster_name}-resource-group* will be created. resource_group_name must also be in the specified location field.
vnet_nam	OPTIONAL: Name of existing VNET to deploy AKS in.
vnet_subn	OPTIONAL: Name of existing subnet within vnet_name to deploy AKS in. Each subnet can only contain one AKS cluster.
vnet_addr	OPTIONAL: If vnet_name is not specified, then use this address space when creating the new VNET to place AKS in. By default, value is set to 172.18.0.0/16.
new_subn	OPTIONAL (required if vnet_subnet_name is not specified): If vnet_subnet_name is not specified, then use this address space when creating the new VNET subnet to place AKS in. Value must be within the address space of the specified VNET. Default value is set to 172.18.1.0/24.
public_node_	OPTIONAL: Whether or not to assign public IPs to AKS nodes. By default, value is set to FALSE.
service_cidr	OPTIONAL: Service CIDR space for internal k8s services. Must not conflict with the address space of the VNET being deployed to. By default, value is set to 172.19.1.0/24.
dns_servi	OPTIONAL: Service IP to assign to the k8s internal DNS service. Must be within the address space specified by service_cidr. By default, value is set to 172.19.1.5.

vSphere

Follow these steps to deploy the Plixer ML Engine multi-node cluster in vSphere after the *pre-deployment steps* have been completed:

Additional prerequisites for vSphere deployment

- The ML engine VM template must be available in vSphere. Note the path to the template as it will need to be entered when deploying the engine.
- The deployment process will require credentials for a vSphere user with permissions to create VMs and resource groups.

Note

The ML VM template includes all software prerequisites (Docker, Terraform, etc.).

Deploying the ML engine

Follow these instructions to set up the necessary infrastructure and deploy the ML engine:

1. Log in to the ML VM image using `plixer:plixer`.
2. Accept the EULA, and then configure network settings for the host.
3. SSH to the ML VM image using the `plixer` credentials set in step 2, and then wait for the setup wizard/scripts to start automatically.
4. Enter the *infrastructure deployment parameters* as prompted.

   Note

   The requested details are automatically saved to `/home/plixer/common/kubernetes/vsphere.tfvars`, which also contains *other default parameters* for deploying the ML engine Kubernetes cluster. If there are issues with the infrastructure deployment, contact *Plixer Technical Support* for assistance before making changes to the file.

5. Wait as the Kubernetes cluster is deployed (may take several minutes), and then enter the Scrutinizer SSH credentials when prompted.

After the scripts complete running, navigate to Admin > Resources > ML Engines and wait for the engine to show as *Deployed* under its *Deploy Status*. Refresh the page if the status has not updated after a few minutes.

See *this guide* for configuration instructions and recommendations for the ML engine.

Terraform configuration

The following table lists all required and optional variables in `/home/plixer/common/kubernetes/vsphere.tfvars`, which are used when deploying the Kubernetes infrastructure for the ML engine.

> **Note**
>
> Contact *Plixer Technical Support* before making changes to this file.

Field name	Description
create_hosts	Whether or not to create vSphere hosts. If FALSE, then the IPs in vm_master_ips should correspond to the VMs created using the VM template.
vm_master_	List of IPs to assign to Kubernetes nodes. This must be 1 or 3 hosts (can't be an even number of IPs).
vm_haproxy	The virtual IP address to assign to a VM running HAProxy.
vsphere_vce	The IP address of the vCenter host to deploy on.
vsphere_use	vSphere user to connect with.
vsphere_data	Datacenter in vSphere to deploy assets into.
vsphere_hos	Host within the specified datacenter to deploy assets into.
vsphere_reso	Resource pool to create for the VMs.
vm_folder	Folder name in vSphere to create the VMs in.
vm_datastor	The datastore name used to store the files of the VMs.
vm_network	The vSphere network name used by the VMs.
vm_gateway	The network gateway used by the VMs.
vm_dns	The DNS server used by the VMs.
vm_domain	The domain name used by the VMs.
vm_template	The vSphere template that the VM is based on.
vsphere_unv	Set to TRUE to bypass the vSphere host certificate verification.
offline_install	If set to TRUE, then it will be assumed that the template being used to create the VMs already has all assets it needs, and will skip downloading the assets.
rke2_airgap_	If set to TRUE and offline_install is also TRUE, then the script will attempt to proxy any downloads required for RKE2 Kubernetes setup through the host that `./01_vsphere_infrastructure.sh` is running on.

4.1.2.3 FlowPro

Before deploying a FlowPro probe, follow the steps below to license and register the appliance in Scrutinizer.

Adding a license

To add a FlowPro license:

1. Navigate to Admin > Plixer > FlowPro Licensing.
2. Contact *Plixer Technical Support* and provide them with your Customer ID and the Machine ID displayed.
3. Paste the license key in the provided field, and then click **Save**.

The page will indicate the maximum number of probes supported by the license as well as registered and deployed probe counts once an active license key has been saved.

Registering the probe

After adding a license, navigate to **Admin** > **Resources** > **FlowPro Probes** to register the probe(s) to be deployed.

See this page for instructions and additional information on FlowPro probe registration and management.

Deploying the probe/appliance

Once the probe(s) have been successfully registered, proceed with the deployment process as described in the FlowPro deployment guides.

4.1.2.4 Replicator

Replicator instances on Scrutinizer servers can forward or replicate packet streams from UDP sources/exporters to specified Scrutinizer collectors and other destinations.

This section contains guides and additional details for deploying Replicator as part of a Scrutinizer/Plixer One environment. Visit the Replicator online manual to learn more about Replicator-only deployments.

Local Replicator instance

The local Replicator instance on a Scrutinizer standalone appliance or primary reporter can be enabled at any time by entering a valid license key on the Admin > Plixer > Replicator page in the web interface.

After an active license key has been applied, the *Replicator* UI tab/page of the Scrutinizer web interface becomes accessible, and the local Replicator instance is automatically added to the Admin > Resources > Replicators management view. The local instance (named 'Local' by default) also functions as the admin/configuration instance for additional headless Replicator instances.

See *this section* for further information and instructions on setting up and configuring packet replication.

Note

A single Replicator instance can replicate packets at rates close to line speed with sufficient CPU provisioning. However, the interface configuration and number of destination collectors must also be taken into consideration. If all packets are being replicated to two destinations, the outbound bandwidth utilization will be twice the inbound volume.

Changed in version 19.7.2: Replicator instances are now bundled with all Plixer One license types. The local instance will be enabled by default and can be used for flow data replication (including *Auto Replicate*) to collectors in the same distributed cluster.

Support for additional instances and replication to external devices can enabled by adding a Replicator license key. Contact *Plixer Technical Support* for more information.

Headless instances

Additional Replicator instances (for greater replication capacity, *high availability*, etc.) can be deployed as "headless" instances to minimize their resource footprint. These deployments do not include the web interface component; they must be registered and managed from a remote administrative/configuration Replicator instance (Plixer One/Scrutinizer environment or standalone Replicator instance) and cannot be configured independently.

Note

- Replicator hardware appliances that are upgraded to v20.0.2 can be used as either standalone (default) or headless instances. See these instructions for more details.
- Additional Replicator instances must be supported by the current license key. Contact Plixer Technical Support for further details.

Registering a headless Replicator instance

Before deploying a headless Replicator instance, it must first be registered on the admin instance as follows:

View instructions

1. Navigate to Admin > Resources > Replicators in the web interface.
2. Click the **Add** button.
3. Enter a name to assign to the new Replicator instance, and then click **Save**.
4. Click on the name of the new instance in the main view and note the authentication token shown in the tray.

Once the new instance has been registered, proceed to deploying the headless Replicator instance.

Activating a headless Replicator instance on a Scrutinizer server

In *distributed clusters*, additional headless Replicator instances can be deployed by activating the Replicator service on the secondary reporter or any remote collector as follows:

View instructions

1. After a new headless Replicator instance has been registered on the admin instance, SSH to the Scrutinizer server as the `plixer` user and run:

   ```
   /user/share/replicator/util/setup.sh
   ```

2. Enter the following details when prompted:
   - IP address of the admin/configuration instance (typically the IP address of the Scrutinizer primary reporter or a standalone Replicator instance)
   - Authentication token generated when the headless instance was registered (see above)
   - Name given to the headless instance when it was registered
3. Return to the Admin > Resources > Replicators page in the Scrutinizer web interface and verify that the new instance has successfully self-registered with the correct IP address.

Once the headless instance on the Scrutinizer server has been successfully registered and deployed, it can be configured for standalone replication or used in a *high-availability pair* from the Scrutinizer web interface.

Deploying a headless Replicator instance

Follow these steps to deploy a headless Replicator instance after it has been registered on the admin instance:

View instructions

1. Download the latest headless Replicator VM package for your hypervisor from the Plixer Customer Portal.
2. Deploy the VM following the instructions here.
3. Complete the basic appliance configuration.
4. After it reboots, SSH to the appliance as the `plixer` user again.
5. Enter the following details when prompted:
   - IP address of the admin/configuration instance (typically the IP address of the Scrutinizer primary reporter or a standalone Replicator instance)

- Authentication token generated when the headless instance was registered (see above)
- Name given to the headless instance when it was registered

6. Return to the Admin > Resources > Replicators page in the Scrutinizer web interface and verify that the new instance has successfully self-registered with the correct IP address.

Once the headless Replicator instance has been successfully registered and deployed, it can be configured for standalone replication or used in a *high-availability pair* from the Scrutinizer web interface.

Changing IP addresses (headless instances only)

Follow these steps to change the IP address of a headless Replicator instance:

View instructions

1. SSH to the instance as the `plixer` user:

```
ssh plixer@HEADLESS_IP_ADDRESS
```

2. Stop the Replicator service:

```
sudo systemctl stop replicator
```

3. Update `IPADDR` in `/etc/sysconfig/network-scripts/ifcfg-eth0` (or `ifcfg-bond0`) with the new IP address (`IP_NEW`):

```
sudo sed -i 's/^IPADDR=.*/IPADDR=IP_NEW/' /
↪etc/sysconfig/network-scripts/ifcfg-eth0
```

or

```
sudo sed -i 's/^IPADDR=.*/IPADDR=IP_NEW/' /
↪etc/sysconfig/network-scripts/ifcfg-bond0
```

4. On the admin instance (Scrutinizer local instance or standalone Replicator instance) run the following queries to replace the previous IP address (`IP_PREV`) with the new IP address (`IP_NEW`) in the configuration:

```
psql plixer

BEGIN;
ALTER TABLE replicator.profiles DROP CONSTRAINT profiles_replicator_ip_fkey;
ALTER TABLE replicator.profiles ADD constraint profiles_replicator_ip_fkey␣
↪FOREIGN KEY(replicator_ip) REFERENCES replicator.deployments(replicator_ip) ON␣
↪UPDATE CASCADE ON DELETE CASCADE;

DELETE FROM replicator.collector_state  where replicator_ip='IP_PREV';
DELETE FROM replicator.exporter_state  where replicator_ip='IP_PREV';

UPDATE replicator.deployments SET replicator_ip='IP_NEW' WHERE replicator_ip=
↪'IP_PREV';
UPDATE replicator.deployments SET paired_ip='IP_NEW' WHERE paired_ip='IP_PREV';
COMMIT;
```

5. Restart the Replicator service:

```
sudo systemctl start replicator
```

When done, navigate to **Admin** > **Resources** > **Replicators** to verify that the IP address of the headless Replicator instance has been updated successfully.

Port restrictions

Replicator instances (both standalone/admin and headless) on Scrutinizer servers can only receive exporter streams on ports that are not already in use by Scrutinizer.

Scrutinizer uses *ports 2055, 2056, 2057, 4432, 4739, 9995, 9996, and 6343* for inbound traffic by default, but it can be configured to only listen on port 4739 (reserved for vitals/internal flows) to allow for additional ports for *Replicator exporters/sources*. This change can be made by editing the *Listener Port* list under Admin > Settings > Collector Settings.

If Scrutinizer is configured to use only port 4739 for inbound traffic, all Replicator *collector configurations* and *Auto Replicate profiles* must also be set to use port 4739.

> **Note**
>
> If exporter streams to Replicator can only be sent on port 4739, a separate appliance/VM can be deployed as a headless Replicator instance (but still managed from the same local/admin instance) to receive the streams.

Profile migrations

Profile configuration data from a Replicator 19.1.1 server can be migrated to a different Replicator 20.0.2 appliance/instance using the migration utility included with Plixer One/Scrutinizer deployments and standalone Replicator appliances.

See this section of the Replicator manual for further details and step-by-step instructions.

4.2 Configuration Guides

4.2.1 Getting started

Configuration checklist

Checklist for recommended configuration steps

Configuration checklist **Environment sizing**

Sizing instructions, guidelines, and tables

Environment sizing

4.2.2 Analytics and security

Alarms and events

Alarm policy and notification configuration

Alarms and events **Flow analytics**

FA algorithm configuration

Flow Analytics **Plixer ML Engine**

ML behavior and engine configuration

ML Engine

4.2.3 System architecture

Custom firewall rules

Add or edit custom firewall rules

Custom firewall rules **Distributed environments**

Distributed cluster setup guide

Distributed environments

4.2.4 Data and device management

Device groups

Device/resource grouping guidelines

Device groups **Importing data**

Import utility for entity/group definitions

Importing data

4.2.4.1 Alarms and events

Scrutinizer uses various technologies to recognize patterns in system activity and network traffic that may be of interest to NetOps and SecOps teams. These patterns are then reported as events via the *Alarm Monitor views*.

Combined, the Alarm Monitor interface and Scrutinizer's library of alarm policies allow for a highly configurable and comprehensive reporting interface that offers deep observability into an organization's network.

On this page:

Alarm life cycle *Alarm life cycle* Alarm Policy settings *Alarm Policy settings* Alarm notifications *Alarm notifications* Flow Analytics *Flow Analytics* Optimizing alarms *Optimizing alarms*

Alarm life cycle

Scrutinizer automatically manages alarm and event data based on the following life cycle:

1. Scrutinizer continuously monitors its environment for observations of system activity or network traffic that match preconfigured criteria.
2. Observations are aggregated and reported/managed as an event based on the alarm policy associated with the identified criteria.
3. The details of the event are reviewed under the corresponding alarm policy via the Alarm Monitor interface.
4. After investigation and/or resolution, the event is flagged as *acknowledged* by a user to clear it from all Alarm Monitor views.

Event data remains accessible for further review following the configured retention settings.

Global retention settings

The following global settings in the Admin > Settings > Data History tray can be used to change how the alarm and event data are managed:

Setting	Description
Alarm Retention Days	Sets the maximum number of days alarm and event data is retained before being deleted from the system
Alarm Retention Size	Sets the maximum amount of disk space that can be used for alarm and event data storage
Auto-Acknowledge Alarms	Sets the number of days before events are automatically tagged as *Acknowledged*(Can also be configured as a Notification Profile action)

Note

The alarm retention settings control automatic data deletion for both acknowledged and un-acknowledged events.

Alarm Policy settings

Individual alarm policy settings allow granular customization of what, when, and how alarms/events are reported.

The following settings can be accessed from the Admin > Alarm Monitor > Alarm Policies view:

Status

Sets the policy to one of three states:

Setting	Generates Events	Alarm Monitor	Stored in Database	Notifications by Profile(s)
Active	Yes	Yes	Yes	Yes
Store	Yes	No	Yes	Yes
Inactive	No	No	No	No

Hint

Setting nonessential policies to *Store* or *Inactive* can filter out events that do not require visibility. This can reduce the number of alarms being reported (and stored) in the Alarm Monitor views.

Weight

Assigns each event/violation under a policy a numerical weight for calculating the severity reported in the Alarm Monitor views

Event timeout

Sets the number of seconds the system will wait before another observation that meets the same criteria is considered a separate event

Refer to *this appendix* for individual alarm policy details, including default timeout settings.

Alarm notifications

Alarms/events in Scrutinizer can also be configured to trigger one or more notification actions when they are generated/observed.

Notification Profiles

Notification actions are assigned to individual alarm policies by way of notification profiles, each of which can be configured with one or more actions.

> **Note**
>
> Notification profiles can be used in conjunction with the *Store* alarm policy status to acknowledge, forward, and/or store the details of an event without them being reported in the Alarm Monitor views. An alarm policy can only be assigned one notification profile at a time.

Flow Analytics

Scrutinizer uses a collection of Flow Analytics (FA) algorithms to monitor collected flow data for specific traffic patterns and/or behavior typically associated with threats to a network.

Because FA algorithms rely on associated alarm policies for reporting, the *initial configuration and regular tuning of FA-based functions* are integral to optimizing alarms and events.

For additional information, see the *Flow Analytics configuration guide*.

Optimizing alarms

When correctly configured, the Scrutinizer Alarm Monitor is capable of reporting information that is accurate, relevant, and uniquely tailored to the organization or team using it.

To achieve this, the following configuration steps related to alarms and events should be completed as part of deploying Scrutinizer.

1. Navigate to the **Admin** > **Settings** > **Data History** tray and adjust the *Alarm Retention Days*, *Alarm Retention Size*, and *Auto-Acknowledge Alarms* values as needed.
2. In the **Admin** > **Settings** > **Alarm Notifications** tray, verify that the alarm notifications options are correctly configured.
3. Go to the Admin > Alarm Monitor > Notification Profiles page and create notification profiles to enable additional notification channels.
4. Go to the Admin > Alarm Monitor > Alarm Policies page and:
   - Set the status of any alarm policies that are unnecessary or irrelevant to the environment to *Inactive* (must be done as a bulk action after selecting at least one policy).
   - Set the status of alarm policies whose events should be monitored but not reported in the Alarm Monitor views to *Store* (must be done as bulk action after selecting at least one policy).
   - Assign the appropriate notification profiles to any alarm policies that require them.

   > **Note**
   >
   > The *Timeout* and *Weight* values of an alarm policy can be adjusted at a later time, after evaluating reporting behavior for events under it.

5. Follow the *Flow Analytics configuration guide* to correctly set up FA-based functions and features.
6. Follow the *Plixer ML Engine configuration guide* to correctly set up machine-learning-based functions and features.

After the initial setup has been completed, it is highly recommended to continue to evaluate alarm and event reporting behavior and make further adjustments to the various elements' configurations as necessary.

4.2.4.2 Configuration checklist

The following checklist outlines the recommended order of configuration steps to fully set up a Scrutinizer deployment:

> **Note**
>
> Click on a checklist item for additional information and detailed instructions related to that configuration step.

Configuration step	Function/Benefit
Deploy Instance	Deploy the Scrutinizer hardware/physical or virtual appliance in your environment.
Appliance Setup Wizard	From the appliance terminal, run the setup questionnaire to configure an IP address, DNS hostname, NTP server, and HTTPS certificate.
Send Flows	Configure exporters/network devices to send flows to Scrutinizer (or a Replicator, if applicable).
Cloud Flows	Enable cloud flow collection from AWS, Azure, GCP, OCI and Zscaler.
SMTP Server	Configure an SMTP server to enable email notifications for alarms and on-demand/scheduled email reports.
SNMP Credentials	Configure SNMP credentials to enable importing of exporter names, interface names, and interface speeds.
Users	Create additional accounts/logins to customize settings and preferences for individual users.
User Groups	Create user groups for managing access levels and permissions for Scrutinizer users.
Defined Applications	Define rules for applications that are unique to your network to enhance reporting, filtering, and other functions.
IP Groups	Assign resources specific to your organization to IP groups for reporting, filtering, and inclusion/exclusion management.
External Authentication	Improve your security posture and simplify user management by leveraging AD, LDAP, SSO, Radius, and/or TACACS for user authentication.
Data History	Modify historical data retention settings to support your organization’s forensics and archiving needs.
Security Groups	Populate the default security groups (Firewalls, Core Exporters, Edge Exporters, Defender Probes) to automatically enable flow analytics algorithms and other features for similar devices.
Exclusion IP Groups	Verify that the DNS servers, Public WiFi, Network Scanners, DNS Servers, DHCP Servers, and SNMP Pollers IP groups are correctly populated to automatically define recommended exclusions for flow analytics algorithms.
Flow Analytics Inclusions	Define additional inclusions and exclusions (including custom security groups and IP groups) necessary for specific flow analytics algorithms.
Device/Mapping Groups	Organize exporters into groups to quickly find flow data sources, enable group report filters, and generate customizable network maps.
Dashboards	Create/customize one or more dashboards to consolidate frequently accessed information and drive workflows through the Plixer One platform.
Saved Reports	Create saved reports to quickly re-run the same report configuration with a single click.
Saved Report Thresholds	Add thresholds to saved reports to proactively watch for specific traffic/behaviors and trigger alarms (and notification profiles/actions) when the specified conditions are met.
Schedule Emailed Reports	Set up scheduled email reports to automatically re-run and send important reports as emails to any inbox.
Notifications	Create notification profiles that can be assigned to alarm policies to automatically send emails, forward details to your SIEM, or run custom scripts to do absolutely anything.
Deploy the ML Engine	Deploy and set up the ML Engine to enable advanced features, including network behavior modeling and anomaly detection in Plixer One Enterprise deployments.
ML Rules	Define inclusion and exclusion rules for sources to be monitored by the ML Engine.
ML Custom Dimensions	Define custom ML dimensions/applications to be monitored by the ML Engine.
Host Indexing	Enable host indexing to allow for faster and more efficient lookups of any hosts that have passed traffic on your network.
Resources	Allocate sufficient CPU and memory to support expected flow rates and enabled features.
Total Disk Space	Expand allocated disk space to support expected flow rates and configured data retention settings.

4.2.4.3 Custom firewall rules

Scrutinizer firewall rules are managed using individual `.nft` files in `/etc/nftables-plixer/` and enforced by the nftables framework.

The `00-base.nft` file defines the base nftables sets and rules provided by Plixer. This file can be used as a reference for the required format/structure of rule definitions. However, directly modifying this file is not recommended.

New custom rules can be added as follows:

1. SSH to the Scrutinizer server as the `plixer` user:

   ```
   ssh plixer@SCRUTINIZER_IP
   ```

2. Create a copy of the example file:

   ```
   cp /etc/nftables-plixer/20-local-rules.nft.example /
   ↪etc/nftables-plixer/20-local-rules.nft
   ```

   Alternatively, a blank file named `20-local-rules.nft` can be manually created in the same directory if one does not exist yet.

3. Edit `20-local-rules.nft` to add rule definitions (`20-local-rules.nft.example` has sample definitions commented out using `\#`).

4. Save the changes made to the `20-local-rules.nft`.

5. Restart the *nftables* service:

   ```
   sudo systemctl reload nftables
   ```

> **Note**
>
> The `00-base-rules.nft`, `10-plxr-*`, and `10-replicator-*` files may be modified or overwritten by Plixer processes or upgrades. `20-local-rules.nft` will not be affected by upgrades.

Example rule

The rule definition for the Zabbix agent would be added to the file as follows:

```
add element inet filter tcp_service_ports { 10050 }
```

Technical details

- Service port sets can be updated using either `tcp_service_ports` or `udp_service_ports`.
- The following user-defined chains can be used to define additional custom rules as needed:
  - `local_input_rules`
  - `local_output_rules`
  - `local_forward_rules`
- To view all rules run:

  ```
  sudo nft list ruleset
  ```

4.2.4.4 Device groups

Scrutinizer supports multiple user-defined entity grouping schemes, which can further enhance the way teams monitor, visualize, and derive insights from network data.

IP groups

IP groups can be used to categorize similar (e.g., device type, ownership/department, geolocation, etc.) flow-exporting devices for use in reports, filters, and *FA algorithm exclusion rules*. The Scrutinizer factory configuration includes default IP groups that should be populated as part of tailoring the system to the environment.

IP group definitions can be created/managed from the Admin > Definitions > IP Groups page.

Mapping groups

Mapping groups consist of devices that have been grouped together for the purpose of network mapping. Network maps will show network topology up to the interface level (i.e., not including endpoints) and can be tailored to a wide range of use cases using customizable elements.

The Monitor > Network Maps page is the primary interface for customizing and viewing network maps, while additional management options for mapping groups and map objects can be accessed via their respective pages under Admin > Settings.

Security groups

Security groups are device groups that can be used to enable one or more FA algorithms for *exporters of the same type*. The Scrutinizer factory configuration includes predefined security groups, which can be populated to automatically enable the recommended algorithms for the indicated device type.

Security groups can be created/managed from the Admin > Alarm Monitor > Security Groups page.

4.2.4.5 Distributed environments

Multiple Scrutinizer appliances/servers can be configured as a distributed cluster with a central, primary reporter and one or more remote collectors.

Distributed environments are capable of ingesting significantly higher flow volumes from a greater number of exporters. All admin, management, and reporting functions are handled from the primary reporter.

Distributed cluster setup

Distributed clusters can include any combination of hardware and/or virtual appliances, regardless of physical location.

To set up a distributed cluster, follow these steps:

1. Deploy the required number of Scrutinizer hardware or virtual appliances following the appropriate *deployment guides* and complete the *initial appliance setup* process.

2. Start an SSH session as the `plixer` user with the appliance that will be used as the primary reporter for the cluster.

3. Launch the *scrut_util* interactive CLI by running:

   ```
   /home/plixer/scrutinizer/bin/scrut_util
   ```

4. At the `SCRUTINIZER>` prompt, register each additional appliance as a remote collector:

   ```
   SCRUTINIZER> set registercollector APPLIANCE_IP
   ```

5. After registering all remote collectors, use the `exit` command to exit the **scrut_util** interactive CLI.

Once the Scrutinizer distributed cluster has been set up, exporters can be configured to send flows to any of the remote collectors. The web interface for the cluster can be accessed using the IP address or hostname of the primary reporter.

Note

- When registering remote collectors, it is highly recommended that one appliance/collector should also be assigned the *secondary reporter role*.

  ```
  set registercollector APPLIANCE_IP secondary
  ```

  This appliance can later be promoted to function as the primary reporter (using the `set selfreporter` *scrut_util* command) if the cluster's original primary reporter becomes unavailable.

- To avoid potential bottlenecks in distributed configurations that include hardware appliances, 10 Gb networking is strongly recommended. If the appliances are geographically dispersed, the WAN link should also support 10G.

Ports used

If appliances in a distributed cluster are unable to communicate with each other, it may be necessary to whitelist the connections between the remote collectors and the primary reporter.

The following network ports are used in communications between appliances in a distributed environment:

Collector(s) -> Reporter (UDP)	Collector(s) <-> Reporter (TCP)
514	22 80 (or 443) 6432 and 5432

Note

To learn more about licensing options for distributed environments or for additional assistance, contact *Plixer Technical Support*.

Certificate management

Run *these scripts* to generate certificate signing requests (CSRs) and install the signed certificates to remote nodes in a distributed cluster.

High availability

Scrutinizer distributed clusters support high availability (HA) configurations that include secondary reporters and/or backup collectors for redundancy.

Note

Contact *Plixer Technical Support* to learn more about HA licensing options.

Secondary reporters

In distributed deployments, a remote collector can be registered as a secondary reporter, which can be used to access the system if the primary reporter becomes unavailable.

To register a remote collector as a secondary reporter, enter the following *scrut_util* command from the primary reporter.

```
SCRUTINIZER> set registercollector COLLECTOR_IP secondary
```

After a collector has been registered as a secondary reporter, its IP address can be used to access a read-only version of the Scrutinizer web interface at any time. An updated backup of the primary reporter's configuration metadata will also be maintained on that collector.

If the primary reporter has become permanently unavailable, the secondary reporter should be promoted using the `set selfreporter` **scrut_util** command, as outlined in the *distributed environment setup guide*. This will lift the read-only status and restore full web interface functionality.

Note

- A new license key is not required when promoting a secondary reporter to primary status. The promoted reporter will operate normally with the old license until it expires. However, it cannot register new appliances as collectors and secondary reporters.
- If the original primary Scrutinizer reporter in a high-availability configuration becomes permanently unavailable, follow these steps to point the FlowPro probe to any new primary reporter.

Backup collectors

Distributed clusters can be configured to use backup collectors to enable high availability for flow collection functions.

To use a remote collector **Y** as a backup for remote collector **A**, do the following:

1. Configure all exporters sending flows to A to also send flows to Y.
2. In the web interface, navigate to **Admin** > **Resources** > **Exporters**, and then verify that the selected exporters are correctly sending flows to both collectors.
3. From the **Exporters** view, set the status of the duplicated exporters sending flows to Y to *Backup*.

If remote collector A becomes unavailable, the exporters that were previously set to *Backup* on remote collector Y must be set to *Enabled* to allow for continuous flow collection and reporting. Once A is online again, the status of the exporters should be reverted to *Backup*.

And if remote collector A is removed from the cluster configuration, it cannot be added back.

Hint

When managing a large number of exporters, filter the list to view only relevant exporters and use the checkboxes to set them to *Backup* or *Enabled* as a bulk action.

HA with Replicator

Replicator can simplify the process of setting up backup collectors by replicating flow data and forwarding it to multiple destination collectors.

View the Replicator online documentation or contact *Plixer Technical Support* to learn more.

4.2.4.6 Environment sizing

To ensure consistently optimal performance and continuous availability, Scrutinizer must be provisioned based on the functions and/or features required by its users.

This section outlines the recommended procedures for calculating the appropriate resource allotments for Scrutinizer deployments.

Note

Certain steps in these guides require access to the Scrutinizer web interface. For more accurate results, complete the *initial setup wizard* beforehand.

On this page:

CPU/RAM *CPU/RAM* Storage *Storage* Distributed clusters *Distributed clusters* Replicator *Local Replicator instance* Plixer ML Engine *ML Engine*

CPU/RAM

Follow the steps described in this section to determine the total number of CPU cores and amount of RAM that will be required by a Scrutinizer deployment.

Note

For additional guidelines related to distributed clusters, see *this section*.

1. Determine CPU and RAM requirements for flow collection, reporting, and core *alarm policies* based on expected flow rate and exporter count:

CPU cores and RAM based on flow rate and exporter count

	Exporters							
F/s	**5**	**25**	**50**	**100**	**200**	**300**	**400**	**500**
5k	8 CPU 16 GB	8 CPU 16 GB	10 CPU 20 GB	14 CPU 28 GB	20 CPU 39 GB	26 CPU 52 GB	32 CPU 67 GB	38 CPU 82 GB
10k	8 CPU 16 GB	8 CPU 16 GB	12 CPU 24 GB	18 CPU 36 GB	25 CPU 50 GB	32 CPU 65 GB	38 CPU 81 GB	43 CPU 97 GB
20k	16 CPU 32 GB	16 CPU 32 GB	16 CPU 32 GB	24 CPU 48 GB	32 CPU 64 GB	38 CPU 80 GB	43 CPU 96 GB	48 CPU 112 GB
50k	32 CPU 64 GB	32 CPU 64 GB	32 CPU 64 GB	32 CPU 64 GB	39 CPU 80 GB	44 CPU 96 GB	48 CPU 112 GB	52 CPU 128 GB
75k	46 CPU 96 GB	46 CPU 96 GB	46 CPU 96 GB	46 CPU 96 GB	46 CPU 96 GB	49 CPU 112 GB	52 CPU 128 GB	55 CPU 144 GB
100k	52 CPU 128 GB	52 CPU 128 GB	52 CPU 128 GB	52 CPU 128 GB	52 CPU 128 GB	52 CPU 128 GB	55 CPU 144 GB	58 CPU 160 GB
125k	58 CPU 160 GB	58 CPU 160 GB	58 CPU 160 GB	58 CPU 160 GB	58 CPU 160 GB	58 CPU 160 GB	58 CPU 160 GB	61 CPU 176 GB
150k	64 CPU 192 GB	64 CPU 192 GB	64 CPU 192 GB	64 CPU 192 GB	64 CPU 192 GB	64 CPU 192 GB	64 CPU 192 GB	64 CPU 192 GB

2. Determine additional CPU and RAM requirements to support the feature sets that will be enabled:

Note

- Each FA algorithm reports detections using one or more alarm policies, which are also enabled/disabled as part of the feature set. Policy-to-algorithm associations can be viewed in the Admin > Alarm Monitor > Alarm Policies view.
- The CPU and RAM allocations per feature are recommended for deployments with up to 500 exporters and a total flow rate of 150,000 flows/s.

Feature resource requirements and FA algorithms

Feature	CPU	RAM	FA Algorithms
Streaming (to a Plixer ML Engine or external data lake)	1 cores	0.4 GB	N/A
Basic Tuple Analysis	5.85 cores	3.3 GB	• *DNS Hits* • *FIN Scan* • *Host Reputation* • *ICMP Destination Unreachable* • *ICMP Port Unreachable* • *Large Ping* • *Odd TCP Flags Scan* • *P2P Detection* • *Packet Flood* • *Ping Flood* • *Ping Scan* • *Reverse SSH Shell* • *RST/ACK Detection* • *SYN Scan* • *TCP Scan* • *Network Transports* • *UDP Scan* • *XMAS Scan*
Application Analysis	0.25 cores	0.1 GB	• *Protocol Misdirection*
Worm Analysis	0.5 cores	0.2 GB	• *Lateral Movement Attempt* • *Lateral Movement*
FlowPro DNS Exfiltration Analysis	0.5 cores	0.2 GB	• *DNS Command and Control Detection* • *DNS Data Leak Detection*
FlowPro DNS Basic Analysis	0.25	0.1 GB	• *BotNet Detection*
JA3 Analysis	0.25	0.1 GB	• *JA3 Fingerprinting*
FlowPro DNS Server Analysis	0.25 cores	0.1 GB	• *DNS Server Detection*
FlowPro Domain Reputation Analysis	0.25 cores	0.1 GB	• *Domain Reputation*
Firewall Event Analysis	0.25 cores	0.1 GB	• *Denied Flows Firewall*
Scan Analysis	1.0 cores	0.4 GB	• *Bogon Traffic* • *Breach Attempt Detection* • *NULL Scan* • *Source Equals Destination*
Jitter Analysis	0.25 cores	0.1 GB	• *Medianet Jitter Violations*
DNS Lookup Analysis	0.25 cores	0.1 GB	• *NetFlow Domain Reputation*
DoS Analysis	0.5 cores	0.2 GB	• *DDoS Detection* • *DRDoS Detection*
Host Index Analysis	2.4 cores	2.4 GB	• *Host Watchlist* • *Incident Correlation* • *IP Address Violations*

3. Provision the Scrutinizer appliance with the CPU and RAM totals obtained from steps 1 and 2.

4. In the web interface, navigate to Admin > Resources > System Performance and verify that the correct CPU core count and RAM amount are displayed for the collector.

5. After confirming that CPU and RAM allocations have been correctly applied, go to Admin > Resources > System

Performance > Feature Resources and enable/disable features according to the selections made for step 2.

Once Scrutinizer is *fully configured and running*, CPU and RAM utilization can be monitored from the **Admin** > **Resources** > **System Performance** page using the *CPU Utilization* and *Available Memory* graphs. These graphs should be reviewed regularly (in addition to after resources are initially allocated), so that any necessary adjustments can be made.

Important

After making any adjustments to Scrutinizer's resource allocations, *launch scrut_util* as the `root` user and run the *set tuning* command to re-tune the appliance.

Note

- Events related to resource utilization (e.g. collection paused/resumed, feature set paused/resumed, etc.) are reported under the *System* category of alarm policies.
- Setting up large numbers of notification profiles, report thresholds, and/or scheduled email reports may also impact performance.

Storage

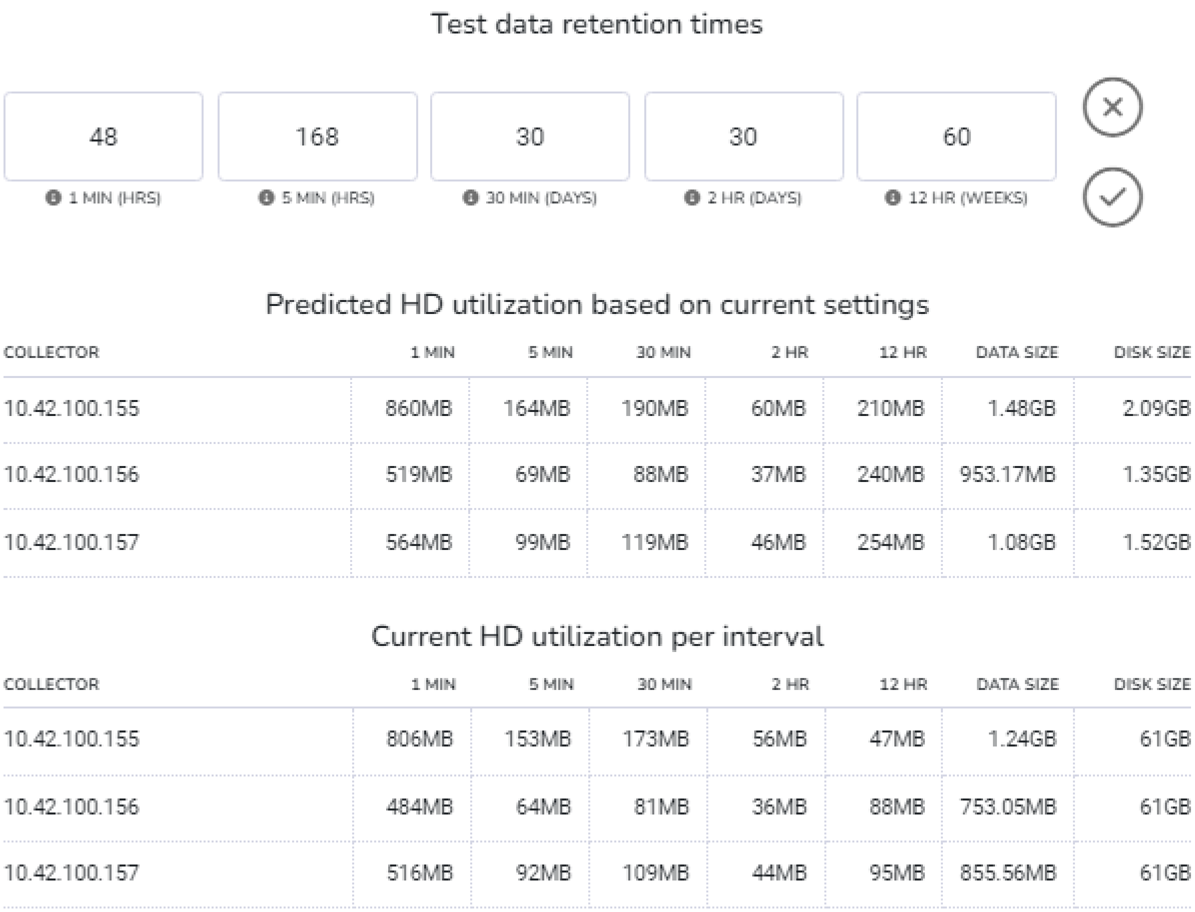
The Admin > Resources > System Performance page of the web interface summarizes disk utilization for individual collectors in a Scrutinizer environment. A more detailed view that shows actual and expected storage use for historical flow data can also be accessed by drilling into a specific collector.

Described below are the main factors that influence a Scrutinizer collector's disk utilization and recommendations for anticipating additional storage needs.

Data retention

Scrutinizer's data history settings can be used to adjust how long Scrutinizer stores *aggregated flow data*, alarm/event details, and other data. With the default settings, a collector provisioned with the minimum 100 GB of storage can store up to 30 days of NetFlow V5 data for a maximum of 25 flow-exporting devices with a combined flow rate of 1,500 flows/s.

For more accurate and detailed projections of disk space requirements based on specific data retention settings, the following database size calculator can be accessed from the data history settings tray:



Test data retention times

1 MIN (HRS)	5 MIN (HRS)	30 MIN (DAYS)	2 HR (DAYS)	12 HR (WEEKS)
48	168	30	30	60

Predicted HD utilization based on current settings

COLLECTOR	1 MIN	5 MIN	30 MIN	2 HR	12 HR	DATA SIZE	DISK SIZE
10.42.100.155	860MB	164MB	190MB	60MB	210MB	1.48GB	2.09GB
10.42.100.156	519MB	69MB	88MB	37MB	240MB	953.17MB	1.35GB
10.42.100.157	564MB	99MB	119MB	46MB	254MB	1.08GB	1.52GB

Current HD utilization per interval

COLLECTOR	1 MIN	5 MIN	30 MIN	2 HR	12 HR	DATA SIZE	DISK SIZE
10.42.100.155	806MB	153MB	173MB	56MB	47MB	1.24GB	61GB
10.42.100.156	484MB	64MB	81MB	36MB	88MB	753.05MB	61GB
10.42.100.157	516MB	92MB	109MB	44MB	95MB	855.56MB	61GB

The calculator shows both current and predicted disk usage for each historical flow data interval based on the retention times entered. Details are shown by collector, with total predicted usage and total storage currently available also included.

ℹ **Note**

- More detailed storage utilization information can be accessed by drilling into a collector from the Admin > Resources > System Performance page.
- Scrutinizer's functions are highly I/O intensive, and there are many factors that can impact the system's disk-based performance, such as the size/complexity of flows being received and flow cardinality. To ensure optimal performance, 15k HDDs or SSDs in a RAID 10 are recommended.

Auto-trimming

Scrutinizer automatically trims older historical flow data when available disk space falls below the *Minimum Percent Free Disk Space Before Trimming* value configured in the data history settings.

Auto-trimming can be disabled by unticking the *Auto History Trimming* checkbox, but flow collection and other functions may be paused when available storage runs low. The amount of storage for the collector can also be increased to retain older records.

Host indexing

When host indexing is enabled, it may become necessary to allocate additional CPU cores, RAM, and disk space to Scrutinizer collectors.

Host to host indexing can have a significant impact on disk utilization due to the two types of records stored:

- Continuously active pairs, for which records will not expire
- Ephemeral unique pairs, for which records will expire but are also replaced at approximately the same rate

Storage requirements

To approximate the amount of additional disk space that will be used by the host to host index:

1. Create/run a new Host to Host pair report and add all exporters that were defined as inclusions for the *Host Indexing* FA algorithm.
2. Set the time window to cover a period of at least 24 hours.
3. When the output of the report is displayed, click the gear button to open the Options tray and select *Global*.
4. In the secondary tray, select the *5m* option from the **Data Source** dropdown and click *Apply* before returning to the main view.
5. Note the total result count, which will be roughly equivalent to the number of active pairs.
6. Return to the **Options** > **Global** tray and switch to the *1m* data source option.
7. Subtract the previous result count from the updated total result count to determine the number of ephemeral pairs.

After obtaining the active pair and ephemeral pair counts, the following formula can be used to calculate additional disk space requirements for host to host indexing:

```
(Active pair count + Ephemeral pair count) * Exporter count * 200 B
```

where `Exporter count` corresponds to the total number of exporters/inclusions defined for the *Host Indexing* algorithm.

Utilization alerts

If the combined disk space used by the host and host pair databases reaches 100% of the *Host Index Max Disk Space* setting of the *Host Indexing* algorithm, host and host to host indexing will be suspended until storage becomes available again.

The following alarm policies are used to alert users to high disk utilization by host indexing:

Policy	Description
Host Index Disk Space Warning	Triggered when the disk space used by host indexing functions reaches/exceeds 75% of the specified *Host Index Max Disk Space*
Host Index Disk Space Error	Triggered when host indexing functions are suspended because the *Host Index Max Disk Space* has been reached
Host Index Disk Availability Error	Triggered when host indexing functions are suspended because disk utilization for the volume the host and host pair databases are stored on has reached/exceeded 90%

Host indexing functions will automatically restart once sufficient storage is available, either due to record expiry or because disk space has been added.

Distributed clusters

Distributed clusters consisting of one primary reporting server and multiple remote collectors allow Scrutinizer to scale beyond the single-appliance ceiling of 500 exporters with a total flow rate of 150,000 flows/s.

See below for sizing guidelines and recommendations for individual appliances in a distributed cluster.

Remote collectors

Resource allocation for each remote collector in a distributed cluster should follow the same guidelines/recommendations as that of a single Scrutinizer appliance:

1. Use the expected flow rate and exporter count for the collector to determine recommended *CPU and RAM allocations for core functions*.
2. Calculate the total additional CPU cores and RAM required to support the *features* that will be enabled for the collector and exporters associated with it.
3. Provision the collector with the minimum 100 GB of disk space and the total CPU and RAM obtained from the first two steps.

After the collector has been *registered as part of the cluster* and is receiving flows, continue to monitor resource utilization via the Admin > Resources > System Performance page and make adjustments when necessary.

Primary reporter

The primary reporter in a distributed environment requires the following **additional** resources (on top of the base resource requirements detailed above) based on the number of remote collectors in the cluster:

Resource	Minimum	Recommended
CPU cores	2x the number of remote collectors	4x the number of remote collectors
RAM	2 GB for every remote collector	4 GB for every remote collector

Note

Primary reporters that are receiving extremely high volumes of alarm/event data from collectors may require specialized resourcing. Contact *Plixer Technical Support* for assistance.

Local Replicator instance

Added in version 19.7.0: From version 19.7.0 onwards, Scrutinizer servers include a *local Replicator instance* that can be activated to enable automated replication and forwarding of incoming flow/packet streams.

The primary reporter in Scrutinizer deployments will require the following additional resources when the local Replicator instance is activated:

Resource	Minimum
CPU cores	2x the number of remote collectors
RAM	1 GB for every remote collector

Note

A single Replicator instance can replicate packets at rates close to line speed with sufficient CPU provisioning. However, the interface configuration and number of destination collectors must also be taken into consideration. If all packets are being replicated to two destinations, the outbound bandwidth utilization will be twice the inbound volume.

ML Engine

The Plixer ML Engine is a supplementary appliance that provides advanced anomaly and threat detection through Scrutinizer in Plixer One Enterprise deployments.

See below for sizing guidelines and recommendations for local and cloud-based ML Engine deployments:

Note

Sizing recommendations for the ML Engine are based on flow rates and asset counts. An “asset” is either an exporter interface or a host.

Local deployments

The following table shows the recommended resource allocations for a local Plixer ML Engine install:

CPU cores, RAM, and disk size based on flow rate and asset count

	Number of assets									
F/s	**150**	**300**	**450**	**600**	**750**	**900**	**1050**	**1200**	**1450**	**1700**
10k	8 CPU 40 GB 0.2 TB	12 CPU 80 GB 0.4 TB	16 CPU 112 GB 0.6 TB	20 CPU 136 GB 0.8 TB	24 CPU 160 GB 1.0 TB	28 CPU 184 GB 1.2 TB	32 CPU 208 GB 1.4 TB	36 CPU 232 GB 1.6 TB	40 CPU 256 GB 1.8 TB	44 CPU 256 GB 2.0 TB
20k	12 CPU 80 GB 0.4 TB	14 CPU 112 GB 0.6 TB	18 CPU 136 GB 0.8 TB	22 CPU 160 GB 1.0 TB	26 CPU 184 GB 1.2 TB	30 CPU 208 GB 1.4 TB	34 CPU 232 GB 1.6 TB	38 CPU 244 GB 1.8 TB	42 CPU 256 GB 2.0 TB	46 CPU 288 GB 2.2 TB
30k	16 CPU 112 GB 0.6 TB	18 CPU 136 GB 0.8 TB	20 CPU 160 GB 1.0 TB	24 CPU 184 GB 1.2 TB	28 CPU 208 GB 1.4 TB	32 CPU 232 GB 1.6 TB	36 CPU 244 GB 1.8 TB	40 CPU 256 GB 2.0 TB	44 CPU 288 GB 2.2 TB	48 CPU 320 GB 2.4 TB
40k	20 CPU 136 GB 0.8 TB	22 CPU 160 GB 1.0 TB	24 CPU 184 GB 1.2 TB	26 CPU 208 GB 1.4 TB	30 CPU 232 GB 1.6 TB	34 CPU 244 GB 1.8 TB	38 CPU 256 GB 2.0 TB	42 CPU 288 GB 2.2 TB	46 CPU 320 GB 2.4 TB	50 CPU 352 GB 2.6 TB
50k	24 CPU 160 GB 1.0 TB	26 CPU 184 GB 1.2 TB	28 CPU 208 GB 1.4 TB	30 CPU 232 GB 1.6 TB	32 CPU 244 GB 1.8 TB	36 CPU 256 GB 2.0 TB	40 CPU 288 GB 2.2 TB	44 CPU 320 GB 2.4 TB	48 CPU 352 GB 2.6 TB	52 CPU 384 GB 2.8 TB
60k	28 CPU 184 GB 1.2 TB	30 CPU 208 GB 1.4 TB	32 CPU 232 GB 1.6 TB	34 CPU 244 GB 1.8 TB	36 CPU 256 GB 2.0 TB	38 CPU 288 GB 2.2 TB	42 CPU 320 GB 2.4 TB	46 CPU 352 GB 2.6 TB	50 CPU 384 GB 2.8 TB	54 CPU 416 GB 3.0 TB
70k	32 CPU 208 GB 1.4 TB	34 CPU 232 GB 1.6 TB	36 CPU 244 GB 1.8 TB	38 CPU 256 GB 2.0 TB	40 CPU 288 GB 2.2 TB	42 CPU 352 GB 2.4 TB	46 CPU 352 GB 2.6 TB	50 CPU 384 GB 2.8 TB	54 CPU 448 GB 3.0 TB	56 CPU 448 GB 3.2 TB
80k	36 CPU 232 GB 1.6 TB	38 CPU 256 GB 1.8 TB	40 CPU 288 GB 2.0 TB	42 CPU 320 GB 2.2 TB	44 CPU 352 GB 2.4 TB	46 CPU 384 GB 2.6 TB	50 CPU 416 GB 2.8 TB	54 CPU 448 GB 3.0 TB	56 CPU 480 GB 3.2 TB	56 CPU 480 GB 3.4 TB
90k	40 CPU 256 GB 1.8 TB	42 CPU 288 GB 2.0 TB	44 CPU 320 GB 2.2 TB	46 CPU 352 GB 2.4 TB	48 CPU 384 GB 2.6 TB	52 CPU 416 GB 2.8 TB	54 CPU 448 GB 3.0 TB	56 CPU 480 GB 3.2 TB	56 CPU 512 GB 3.4 TB	56 CPU 512 GB 3.6 TB
100k	44 CPU 256 GB 2.0 TB	46 CPU 288 GB 2.2 TB	48 CPU 320 GB 2.4 TB	50 CPU 352 GB 2.6 TB	52 CPU 384 GB 2.8 TB	54 CPU 416 GB 3.0 TB	56 CPU 448 GB 3.2 TB	56 CPU 480 GB 3.4 TB	56 CPU 512 GB 3.6 TB	56 CPU 512 GB 3.6 TB

AWS deployments

When deploying the Plixer ML Engine as an AWS AMI, use the following table to determine the appropriate instance type and amount of storage:

Instance type (`xxxxxxx`*xlarge*) and Elastic Block Storage size based on flow rate and asset count

	Number of assets									
F/s	**150**	**300**	**450**	**600**	**750**	**900**	**1050**	**1200**	**1450**	**1700**
10k	r5a.2 0.2 TB	r5a.4 0.4 TB	r5a.4 0.6 TB	r5a.8 0.8 TB	r5a.8 1.0 TB	r5a.8 1.2 TB	r5a.8 1.4 TB	r5a.12 1.6 TB	r5a.12 1.8 TB	r5a.12 2.0 TB
20k	r5a.4 0.4 TB	r5a.4 0.6 TB	r5a.8 0.8 TB	r5a.8 1.0 TB	r5a.8 1.2 TB	r5a.8 1.4 TB	r5a.12 1.6 TB	r5a.12 1.8 TB	r5a.12 2.0 TB	r5a.12 2.2 TB
30k	r5a.4 0.6 TB	r5a.8 0.8 TB	r5a.8 1.0 TB	r5a.8 1.2 TB	r5a.8 1.4 TB	r5a.8 1.6 TB	r5a.12 1.8 TB	r5a.12 2.0 TB	r5a.12 2.2 TB	r5a.12 2.4 TB
40k	r5a.8 0.8 TB	r5a.8 1.0 TB	r5a.8 1.2 TB	r5a.8 1.4 TB	r5a.8 1.6 TB	r5a.12 1.8 TB	r5a.12 2.0 TB	r5a.12 2.2 TB	r5a.12 2.4 TB	r5a.16 2.6 TB
50k	r5a.8 1.0 TB	r5a.8 1.2 TB	r5a.8 1.4 TB	r5a.8 1.6 TB	r5a.12 1.8 TB	r5a.12 2.0 TB	r5a.12 2.2 TB	r5a.12 2.4 TB	r5a.12 2.6 TB	r5a.16 2.8 TB
60k	r5a.8 1.2 TB	r5a.8 1.4 TB	r5a.8 1.6 TB	r5a.12 1.8 TB	r5a.12 2.0 TB	r5a.12 2.2 TB	r5a.12 2.4 TB	r5a.12 2.6 TB	r5a.16 2.8 TB	r5a.16 3.0 TB
70k	r5a.8 1.4 TB	r5a.12 1.6 TB	r5a.12 1.8 TB	r5a.12 2.0 TB	r5a.12 2.2 TB	r5a.12 2.4 TB	r5a.12 2.6 TB	r5a.16 2.8 TB	r5a.16 3.0 TB	r5a.16 3.2 TB
80k	r5a.12 1.6 TB	r5a.12 1.8 TB	r5a.12 2.0 TB	r5a.12 2.2 TB	r5a.12 2.4 TB	r5a.12 2.6 TB	r5a.16 2.8 TB	r5a.16 3.0 TB	r5a.16 3.2 TB	r5a.16 3.4 TB
90k	r5a.12 1.8 TB	r5a.12 2.0 TB	r5a.12 2.2 TB	r5a.12 2.4 TB	r5a.12 2.6 TB	r5a.16 2.8 TB	r5a.16 3.0 TB	r5a.16 3.2 TB	r5a.16 3.4 TB	r5a.16 3.6 TB
100k	r5a.12 2.0 TB	r5a.12 2.2 TB	r5a.12 2.4 TB	r5a.16 2.6 TB	r5a.16 2.8 TB	r5a.16 3.0 TB	r5a.16 3.2 TB	r5a.16 3.4 TB	r5a.16 3.6 TB	r5a.16 3.6 TB

Azure deployments

When deploying the Plixer ML Engine as an Azure VM image, use the following table to determine the appropriate VM size and amount of storage:

VM size (*Standard_*`xxxxxx`) and Azure Disk Storage size based on flow rate and asset count

	Number of assets									
F/s	150	300	450	600	750	900	1050	1200	1450	1700
10k	D13_v2 0.2 TB	D14_v2 0.4 TB	D14_v2 0.6 TB	E20_v5 0.8 TB	E20_v5 1.0 TB	E32_v5 1.2 TB	E32_v5 1.4 TB	E32_v5 1.6 TB	E48_v5 1.8 TB	E48_v5 2.0 TB
20k	D14_v2 0.4 TB	D14_v2 0.6 TB	E20_v5 0.8 TB	E20_v5 1.0 TB	E32_v5 1.2 TB	E32_v5 1.4 TB	E32_v5 1.6 TB	E48_v5 1.8 TB	E48_v5 2.0 TB	E48_v5 2.2 TB
30k	D14_v2 0.6 TB	E20_v5 0.8 TB	E20_v5 1.0 TB	E32_v5 1.2 TB	E32_v5 1.4 TB	E32_v5 1.6 TB	E48_v5 1.8 TB	E48_v5 2.0 TB	E48_v5 2.2 TB	E48_v5 2.4 TB
40k	E20_v5 0.8 TB	E20_v5 1.0 TB	E32_v5 1.2 TB	E32_v5 1.4 TB	E32_v5 1.6 TB	E48_v5 1.8 TB	E48_v5 2.0 TB	E48_v5 2.2 TB	E48_v5 2.4 TB	E64_v5 2.6 TB
50k	E20_v5 1.0 TB	E32_v5 1.2 TB	E32_v5 1.4 TB	E32_v5 1.6 TB	E48_v5 1.8 TB	E48_v5 2.0 TB	E48_v5 2.2 TB	E48_v5 2.4 TB	E48_v5 2.6 TB	E64_v5 2.8 TB
60k	E32_v5 1.2 TB	E32_v5 1.4 TB	E32_v5 1.6 TB	E48_v5 1.8 TB	E48_v5 2.0 TB	E48_v5 2.2 TB	E48_v5 2.4 TB	E48_v5 2.6 TB	E64_v5 2.8 TB	E64_v5 3.0 TB
70k	E32_v5 1.4 TB	E32_v5 1.6 TB	E48_v5 1.8 TB	E48_v5 2.0 TB	E48_v5 2.2 TB	E48_v5 2.4 TB	E48_v5 2.6 TB	E64_v5 2.8 TB	E64_v5 3.0 TB	E64_v5 3.2 TB
80k	E32_v5 1.6 TB	E48_v5 1.8 TB	E48_v5 2.0 TB	E48_v5 2.2 TB	E48_v5 2.4 TB	E48_v5 2.6 TB	E64_v5 2.8 TB	E64_v5 3.0 TB	E64_v5 3.2 TB	E64_v5 3.4 TB
90k	E48_v5 1.8 TB	E48_v5 2.0 TB	E48_v5 2.2 TB	E48_v5 2.4 TB	E48_v5 2.6 TB	E64_v5 2.8 TB	E64_v5 3.0 TB	E64_v5 3.2 TB	E64_v5 3.4 TB	E64_v5 3.6 TB
100k	E48_v5 2.0 TB	E48_v5 2.2 TB	E48_v5 2.4 TB	E64_v5 2.6 TB	E64_v5 2.8 TB	E64_v5 3.0 TB	E64_v5 3.2 TB	E64_v5 3.4 TB	E64_v5 3.6 TB	E64_v5 3.6 TB

> **Note**
>
> To learn more about ML Engine licensing options and deployment procedures, contact *Plixer Technical Support*.

4.2.4.7 Flow Analytics

Scrutinizer includes a library of flow analytics (FA) algorithms, which are applied to all incoming flow data. This allows the system to provide additional traffic-based insights and report activity typically associated with threats to a network.

Note

To learn more about individual algorithms, see *this appendix section*.

Configuring Flow Analytics

To enable FA-based functions, several configuration steps must be completed after Scrutinizer has been deployed and set up.

This process helps ensure that Scrutinizer is fully adapted to an organization's NDR requirements.

Enabling/disabling algorithms

Because Scrutinizer is designed to support the full spectrum of enterprise applications, it may include FA algorithms that may not apply to certain network configurations. This will be based on the devices and elements present on the network, the types of flow data available, and/or organizational IT policies.

As part of optimizing the system's monitoring and reporting functions, all unnecessary FA algorithms should be disabled. This includes algorithms that:

- Only benefit devices or elements that are not present on the network
- Require flow data that is not being sent by devices on the network
- Target traffic or patterns that are made irrelevant by the organization's IT policies

The Admin > Alarm Monitor > Flow Analytics Algorithms page lists the current state of all FA algorithms (default: enabled).

Note

Most FA algorithms can also be tuned through *additional settings*, allowing them to be adapted to specific monitoring and detection requirements.

Disabling FA algorithms

To disable an algorithm, click on it to open the configuration tray and use the toggle. The algorithm can also be re-enabled this way at any time.

Multiple algorithms can also be disabled or enabled as a bulk action when one or more algorithms are selected.

Adding exporters

Scrutinizer selectively applies Flow Analytics to incoming flow data, based on the exporters defined for each algorithm.

To activate the system's FA-based functions, exporters must first be added to the enabled algorithms.

Security groups

Scrutinizer security groups are user-defined groups of exporters to which the same set of FA algorithms are applied. Security groups allow the exporter lists for all FA algorithms to be fully populated without the need to manually configure individual algorithms. Exporters can be added to security groups via the Admin > Alarm Monitor > Security Groups page.

If Flow Analytics is being configured for the first time, exporters should be added to the *Core Exporters* and *Edge Exporters* a few at a time. This will limit the volume of alarms that may need to be checked when *testing Flow Analytics settings* via the Alarm Monitor page.

The **Security Groups** view also allows new groups to be added and the settings for existing groups to be modified.

> **Hint**
>
> The default *Firewalls*, *Core Exporters*, *Edge Exporters*, and *Defender Probes* security groups are configured with FA algorithms based on the recommended exporter assignments.

Adding exporters individually

For more granular control over exporter-to-algorithm assignment, exporters can also be added to FA algorithms via the configuration tray of the Admin > Alarm Monitor> Flow Analytics Algorithms page.

Because alarm-triggering algorithms will only be triggered when the target is an internal address, public IP addresses must be defined as part of an IP group for them to be considered part of the protected network. For internal-to-internal and internal-to-external monitoring, core routers should be added to the relevant algorithms. For monitoring public assets, the edge routers of the relevant IP groups should be added to the algorithms.

Defining exclusions

To avoid unnecessary alarms and excessive processing load on the system, certain devices or traffic should be excluded from monitoring by specific FA algorithms.

Scrutinizer's factory configuration includes four IP groups that are defined as exclusions under the appropriate algorithms:

- DNS servers
- Public WiFi
- Network Scanners
- SNMP Pollers

These IP groups should be populated with the correct exporters to optimize Flow Analytics monitoring and reporting.

Adding exclusions to an FA algorithm

FA algorithms can also be configured with additional exclusions beyond those defined under the above-mentioned IP groups. This is done via the algorithm's configuration tray from the Admin > Alarm Monitor> Flow Analytics Algorithms page.

Exclusions can be defined by IP address, IP range, subnet, domain (via reverse DNS), or IP group.

> **Hint**
>
> The default IP group exclusions for an algorithm are also displayed under the *Exclusions* section of the configuration tray.

Additional settings

Scrutinizer's flow analytics functions can be further adapted to more unique network and/or security requirements through the configuration options below.

Global settings

The following global settings (Admin > Settings > Flow Analytics Settings) can be used to enable or configure additional FA-based features:

Setting	Description
Auto-Enable Defender	When checked, FlowPro Defender is automatically enabled for algorithms that support it.
Jitter by Interface	Sets the variation in packet delay due to queueing, contention, and/or serialization (Default: 80 ms); Also used for record highlighting in *Status* reports
Latency	Sets the latency value used for record highlighting in *Status* reports (Default: 75 ms)
Share Violations	When checked, allows the system to share details of cyber attacks coming from Internet IP addresses with the Plixer Security Team (May require firewall permissions); This information is used to further improve the global host reputation list. No internal addresses will be shared.
Top Algorithm Devices	Controls whether *Top X* FA algorithms are applied to all exporters or need to be configured individually

Algorithm settings

In addition to inclusions and exclusions, most FA algorithms have additional settings that control how they are applied to collected flow data. These settings include thresholds for adjusting detection sensitivity and traffic directionality inclusion/exclusion options.

For a full list of algorithm settings, see *this table*.

Custom reputation lists

The *Host Reputation* FA algorithm is capable of using custom lists in conjunction with Scrutinizer's default host reputation lists. When a host in any reputation list becomes the target of traffic, its address is reported in event is reported in event artifacts under the *Host Reputation* alarm policy.

To import a list of IP addresses as a custom host reputation list, follow these steps:

1. Add the hosts to a file, using one line for each IP address.

   Example:

   ```
   10.1.1.1
   10.1.1.2
   10.1.1.3
   ```

2. Save the file with a `.import` extension. (e.g., custom_threats.import)
3. Move the file to the `\scrutinizer\files\threats\` directory.

The file is imported hourly, at the same time that threat lists are updated.

Hint

To manually run the file import operation, run the following:

```
scrut_util --downloadhostreputationlists
```

Reporting options

Each alarm-triggering FA algorithm is associated with one or more alarm policies, under which anomalies and other insights are reported via the Scrutinizer alarm monitor. The *settings for these alarm policies* can also be modified to change the reporting behavior for the individual algorithms.

To learn more about alarm policies and the Scrutinizer alarm monitor, see the *alarms and events* section of this manual.

Notification profiles

To forward the details of alarms and events reported by an FA algorithm to one or more users or external systems, at least one notification profile must be created and assigned to the corresponding alarm policy.

To learn more about notification profiles, see the *alarm notifications* section.

FA dashboard gadgets

Certain gadgets that can be added to *Scrutinizer dashboards* rely on one or more FA algorithms for the data they report.

These gadgets require no further configuration and can be added to any dashboard as long as the corresponding algorithms have been enabled and correctly configured.

Hint

The **Flow Analytics Summary** gadget can be used to troubleshoot algorithm configurations. If there are algorithms that are taking longer than 5 minutes to run, check that the correct exporters have been added.

To learn more about dashboards and gadgets, see the *dashboards* topic of this documentation.

Testing and tuning

To ensure that flow analytics is properly configured, testing the various definitions, settings, and enabled features is strongly recommended. This can be accomplished by checking what alarms and events are being reported in the *Alarm Monitor views*.

When setting up flow analytics for the first time, the following process is recommended:

1. Navigate to **Admin** > **Definitions** > **IP Groups** and populate the *DNS Servers*, *Public WiFi*, *Network Scanners*, and *SNMP Pollers* groups to define basic exclusions for FA algorithms.
2. Review the *list of FA algorithms* in the **Admin** > **Alarm Monitor** > **Flow Analytics Configuration** and disable any algorithms that are irrelevant.
3. Define additional exclusions for individual algorithms in their configuration trays as needed.
4. Navigate to **Admin** > **Alarm Monitor** > **Security Groups** and add several exporters each to the *Core exporters* and *Edge exporters* security groups.

Once the first batch of exporters has been added, review the Alarm Monitor views to verify that alarms and events are being reported correctly. Afterwards, repeat Step 4 of the process and continue checking alarms and events until all exporters have been added to security groups.

Note

- If there are continuous or unnecessary alarms or events being reported, it may also be necessary to define additional exclusions for certain algorithms.

- To enhance response/resolution workflows, *create one or more notification profiles* and associate them with the appropriate alarm policies.

Further tuning

After the initial setup and testing have been completed, flow analytics functions can be further adapted to an environments monitoring and detection requirements through *global* and *individual algorithm* settings.

4.2.4.8 Importing data

Scrutinizer leverages a variety of user-customizable entity/resource labels, definitions, and groupings as part of its data aggregation and reporting functions. These details can be manually configured via the respective admin views or imported as a batch operation using the `import` utility.

This section covers the syntax, requirements, and other relevant information for each type of import operation.

> **Note**
>
> The `import` utility can be accessed via the *SCRUTINIZER> interactive prompt* or directly from the shell. The direct shell syntax can also be included in scripts to automatically update Scrutinizer's databases.

On this page:

ACL information

View details

To import custom ACL information from a file, execute the following from the *scrut_util interactive shell* (`SCRUTINIZER>` prompt):

```
import aclfile
```

Direct shell/script syntax

```
scrut_util --import aclfile
```

File requirements

- The file must contain the exact output when the command `show access-list` is run on the exporter.
- The file should be named `acl_file.txt` and saved to the `/home/plixer/scrutinizer/files/` directory.

Applications

View details

To import a list of application definition rules, execute the following from the *scrut_util interactive shell* (`SCRUTINIZER>` prompt):

```
import applications <PATH/FILE> [reset]
```

Direct shell/script syntax

```
scrut_util --import applications --file <PATH/FILE> [--reset]
```

File requirements

The file to be imported must be a CSV file.

Using the file `/home/plixer/scrutinizer/files/ipgroup_import.csv` for application rule definitions is recommended.

Definition format

Each application-rule pairing should be in a single line, following the format:

```
'APPLICATION NAME',RULE
```

Additional notes

- Rules can defined as any of the following:
  - Subnets
  - Single IP address
  - IP address ranges
  - Wildcard masks
  - Child rules (must be defined first)
  - Port and protocol
- For an application definition to be valid, it must include **at least** one port rule **and** one rule of any other type. The import file may include applications that do not meet this requirement, but they will not be considered a *defined application* by Scrutinizer.
- Passing the `reset` option will delete all existing application definitions/rules before the import operation.
- If the `reset` option is not used, imported rules will be added to the specified application if it already exists.
- Each import operation supports up to 100,000 application rule definitions.

Definition examples

Rule types:

```
'Application subnet rule',10.0.0.0/8
'Application single IP rule',10.1.1.1
'Application IP range rule',10.0.0.1-10.0.0.42
'Application wildcard mask rule',10.0.0.1/0.255.255.0
'Parent application with a child rule', 'My Child Application Rule'
'Application port and protocol rule',0-65535/256
```

ASN definitions

View details

To import a list of custom ASN definitions, execute the following from the *scrut_util interactive shell* (`SCRUTINIZER>` prompt):

```
import asns <PATH/FILE> [DELIMITER]
```

Direct shell/script syntax

```
scrut_util --import asns --file <PATH/FILE> [--delimiter <DELIMITER>]
```

File requirements

The file to be imported must be a CSV file, and the path provided must be relative to the `home/plixer/scrutinizer/` directory. The file's name should only include lower-case letters.

Definition format

Each ASN definition should be in a single line, following the format:

```
'AS_NUMBER',AS NAME,AS Description,IP_NETWORK(S)
```

Additional notes

- The optional `DELIMITER` parameter can be used to replace `` `` (space) for separating individual IP networks if the contents of the import file are formatted differently.
- `,` (comma) cannot be used as a custom delimiter, as it is reserved for separating elements in the definition.

Definition examples

```
213,My ASN,what a great autonomous system,10.0.0.0/8 192.168.0.0/16
214,Your List,this system is only meh,11.0.0.0/8
```

Custom hostnames

View details

To import a list of custom hostname assignments, execute the following from the *scrut_util interactive shell* (`SCRUTINIZER>` prompt):

```
import hostfile
```

Direct shell/script syntax

```
scrut_util --import hostfile
```

File requirements

The file should be named `hosts.txt` and saved to the `/home/plixer/scrutinizer/files/` directory.

Definition format

Each definition should be in a single line, following the format:

```
IPv4orIPv6ADDRESS   HOSTNAME   DESCRIPTION
```

Additional notes

- This command will alter the Scrutinizer database tables and should be used with caution.
- The description element in the definition is optional.

Definition example

```
10.1.1.4   my.scrutinizer.rocks   The best software in my company
```

Device GPS details

View details

To import a list of device/object latitude and longitude details for a specified geographical network map, execute the following from the *scrut_util interactive shell* (`SCRUTINIZER>` prompt):

```
import csv_to_gps <PATH/FILE> <GROUP NAME|GROUP_ID> [create_new] [FORMAT]
```

Direct shell/script syntax

```
scrut_util --import csv_to_gps --file <PATH/FILE> --group <GROUP_NAME|GROUP_ID>␣
↪[--create_new] [--file_format <FORMAT>]
```

File requirements

The file to be imported must be a CSV file, and the path provided must be relative to the `home/plixer/scrutinizer/` directory.

Definition format

Each set of details should be in a single line, following the format:

```
IP_ADDRESS,LATITUDE,LONGITUDE
```

Additional notes

- The imported GPS details are only assigned to objects for the specified device/mapping group. If the devices are assigned to other groups, they will retain the GPS details configured for those groups.
- The optional `FORMAT` parameter can be used to override the default `ip,lat,lng` element formatting in case the contents of the import file are formatted differently (e.g., `ip,lng,lat`).
- If the `create_new` option is used, objects will be created for devices in the import file that are not currently assigned to the specified device group.

Definition examples

```
10.169.1.3,37.7749,122.4194
192.168.6.1,40.7128,74.0059
```

Mapping groups

View details

To import a list of device/mapping group assignments, execute the following from the *scrut_util interactive shell* (SCRUTINIZER> prompt):

```
import csv_to_membership <PATH/FILE> <TYPE> [FORMAT]
```

Direct shell/script syntax

```
scrut_util --import csv_to_membership --file <PATH/FILE> --grouptype <TYPE>␣
↪[--file_format <FORMAT>]
```

File requirements

The file to be imported must be a CSV file, and the path provided must be relative to the home/plixer/scrutinizer/ directory.

Definition format

Each assignment should be in a single line, following the format:

```
IP_ADDRESS,GROUP_NAME
```

Additional notes

- The TYPE parameter specifies the device/mapping group type for any groups that will be created as part of the import operation. Valid values are plixer (for spatial maps) and google for geographical maps.
- The optional FORMAT parameter can be used to override the default ipaddr,group element formatting in case the contents of the import file are formatted differently (e.g., group,ipaddr).

Definition examples

```
10.169.1.3,Routers
192.168.6.1,Firewalls
```

Interface details

View details

To import a list of custom interface details to use for displaying utilization, threshold alerts, and other Scrutinizer functions, execute the following from the *scrut_util interactive shell* (SCRUTINIZER> prompt):

```
import ifinfo <PATH/FILE> [DELIMITER]
```

Direct shell/script syntax

```
scrut_util --import ifinfo --file <PATH/FILE> [--delimiter '<DELIMITER>']
```

Definition format

Each set of details should be in a single line, following the format:

```
INBOUND_SPEED,OUTBOUND_SPEED,NAME,HOST_IP,INDEX_NUMBER
```

Additional notes

- This command will alter the Scrutinizer database tables and should be used with caution.
- The optional `DELIMITER` parameter can be used to replace `,` (comma) for separating elements in each set of details if the contents of the import file are formatted differently.

Definition examples

```
10000000,10000000,WAN_Interface_1,192.168.1.2,2
20000000,20000000,WAN_Interface_1,192.168.1.4,11
```

IP groups

View details

To import a list of IP group inclusion definitions, execute the following from the *scrut_util interactive shell* (`SCRUTINIZER>` prompt):

```
import ipgroups <PATH/FILE> [reset]
```

Direct shell/script syntax

```
scrut_util --import ipgroups --file <PATH/FILE> [--reset]
```

File requirements

The file to be imported must be a UTF8-encoded CSV file.

Definition format

Each inclusion definition should be in a single line, following the format:

```
IP GROUP NAME,INCLUSION_RULE
```

Additional notes

- Rules can defined as any of the following:
  - Subnets (e.g., `Subnet Group,10.0.0.0/8`)
  - Single IP address (e.g., `Single IP Group,10.1.1.1`)
  - IP address ranges (e.g., `IP Range Group,10.0.0.1-10.0.0.42`)
  - Wildcard masks (e.g., `Wildcard Mask Group,10.0.0.1/0.255.255.0`)

  - Child groups (must be defined first; e.g., `Parent/Child Group, Subnet Group`)

- Passing the `reset` option will delete all existing IP group definitions before the import operation.
- If the `reset` option is not used, IP addresses covered by an imported inclusion rule will be added to the specified IP group if it already exists.
- Because each line can contain only one rule, an IP group containing multiple single IP addresses will need to be defined using a separate definition/line for each address. Multiple rules in separate lines for the same IP group are also supported (see examples below).
- Each import operation supports up to 100,000 IP group inclusion definitions.

Definition examples

Multiple single addresses:

```
HR Group,10.1.1.1
HR Group,192.168.3.4
HR Group,10.3.1.2
```

Multiple rules/types:

```
Subnet Group,10.2.0.0/16
New IP Group,10.0.0.1-10.0.0.42
New IP Group,10.0.0.1/0.255.255.0
New IP Group,Subnet Group
```

Sample workflow

CSV file (`ipgroups-import.csv`) contents:

```
HR Group,10.1.0.0/16
Sales Group,10.2.0.0/16
Sales VLAN1,10.70.1.0/24
Sales VLAN2,10.70.2.0/24
Sales VLAN3,10.70.3.0/24
```

To import the file:

```
scrut_util --import ipgroups --file /home/plixer/scrutinizer/files/ipgroups-import.csv
```

```
[plixer@scrutinizer]$ scrut_util --import ipgroups --file /
↪home/plixer/scrutinizer/files/ipgroups-import.csv
Found new ip_group: HR Group
Found new ip_group: Sales Group
Found new ip_group: Sales VLAN1
Found new ip_group: Sales VLAN2
Found new ip_group: Sales VLAN3
Added: 5 ip_group(s) with: 5 rules
```

To truncate existing rules and import a new batch:

```
scrut_util --import ipgroups --file /
↪home/plixer/scrutinizer/files/ipgroups-import.csv --reset
```

```
[plixer@scrutinizer]$ scrut_util --import ipgroups --file /
↪home/plixer/scrutinizer/files/ipgroups-import.csv --reset
Removing all ip_group rules
Found new ip_group: HR Group
Found new ip_group: Sales Group
Found new ip_group: Sales VLAN1
Found new ip_group: Sales VLAN2
Found new ip_group: Sales VLAN3
Added: 5 ip_group(s) with: 5 rules
```

4.2.4.9 ML Engine

When deployed as part of a Plixer One Enterprise environment, the Plixer ML Engine applies anomaly and threat detection techniques to the network data collected by Scrutinizer.

Contact *Plixer Technical Support* to learn more about Plixer One Enterprise licensing options.

On this page:

Overview *Overview* Inclusion & exclusion rules *Managing inclusion and exclusion rules* Managing dimensions *Managing dimensions* Global ML settings *Global ML settings* ML cluster settings *ML cluster settings*

> **Note**
>
> See *this section* for sizing recommendations for the Plixer ML Engine.

Overview

Once *deployed* and configured, the engine is able to ingest flow data through Scrutinizer and *apply multiple machine learning techniques* to identify potentially problematic activity on the network.

The Plixer ML Engine has several key functions that enable intelligent, multi-layered anomaly and threat detection in a Plixer One Enterprise deployment:

- **Comprehensive network behavior modeling**: Leveraging the large volumes of flow data collected by Scrutinizer, the engine is capable of building behavioral models encompassing network activity at any scale. It can then learn to recognize deviations and suspicious activity, such as data accumulation/exfiltration, tunneling, and lateral movement, that may indicate an attack on the network.
- **Accessible behavioral insights for network assets**: After being alerted to anomalous behavior, network and security teams can drill down into the associated hosts, IP address groups, and/or exporter interfaces to better understand the details of their involvement in the reported detection.
- **Highly configurable ML modeling**: The ML Engine monitors network activity based on user-customizable *dimensions* and *inclusion/exclusion rules*. Consistently repeated traffic patterns, asset/group importance, and data seasonality are all taken into consideration as well, resulting in models that are uniquely tailored to each environment.
- **ML-based malware detection**: Using pre-trained classification models, the engine is able to recognize generic activity patterns that are associated with common classes of malware, including command and control, remote access trojans, and exploit kits. This adds another layer of protection to further reduce risk and mean time to resolution (MTTR) when threats are detected.
- **Continuous observation and learning**: As it ingests additional flow data, the ML Engine updates its behavior models based on a schedule that defines weekdays, weeknights, and weekends to account for changes in legitimate activity patterns and improve recognition of advanced threats that attempt to disguise their behavior.

Managing inclusion and exclusion rules

To ensure that its behavior models represent only relevant network activity, the Plixer ML Engine can be uniquely tailored to its environment using custom rules defining inclusions and exclusions for its functions. These rules can be managed from the Admin > Alarm Monitor > ML Rules view of the Scrutinizer web interface.

Inclusion rules

An inclusion rule defines either a network address (hosts/subnets) or exporter interface as a network data source for the ML Engine. Each rule also includes a sensitivity setting (see below) that is applied to the asset specified.

Malware detection, which uses pre-trained classification models to recognize generic malware behaviors, can also be enabled for individual inclusions.

Inclusion sensitivity

An inclusion's sensitivity setting can be used to tune the engine's tolerance for behavioral deviations for the host/subnet or exporter interface. Lowering the sensitivity setting for an asset will cause even minor deviations to be reported as detections, resulting in a higher volume of alarms. Conversely, increasing the sensitivity will allow for greater deviation, which translates to fewer detections reported.

When defining inclusions, the sensitivity setting should be left at its default value. After a period of 7 days (recommended), if too many unwarranted detection alarms are triggered, the sensitivity can be increased to the next level.

Exclusion rules

Exclusion rules can be used to ignore one or more ML-driven detections for traffic originating from a specified source and/or bound for a specified destination.

If expected traffic/activity triggers alarms, one or more exclusion rules should be created to exempt the sources and/or destination addresses from the detections being reported.

Recommendations

Inclusion/exclusion rule recommendations

As part of the ML Engine's initialization, inclusion rules are automatically created for the twenty most suitable network assets (hosts and exporters/interfaces) based on its *default dimension definitions*. If necessary, additional rules should be created to cover all assets associated with critical/sensitive network activity ("crown jewel" assets) and hard-to-monitor traffic (e.g., IoT devices, operational technology, etc.).

The following resources are examples of network assets that are highly recommended for inclusion:

- AD servers
- DB servers
- DBS servers
- DHCP servers
- Web servers
- Source code repositories
- Object repositories
- FTP servers

If there are assets whose typical behavior is being reported as anomalous/suspicious, exclusion rules should be defined to exempt the traffic from superfluous detections.

Managing dimensions

The Plixer ML Engine's feature dimension list defines the protocols and ports to be observed on the network assets defined by its *inclusion/exclusion rules*. These dimensions are used by the engine to build its behavior models, which are used to report asset behavior insights, as well as deliver anomaly and threat alerts via the Scrutinizer alarm monitor.

The default configuration for the ML Engine includes recommended dimension definitions, which are used to automatically select suitable data sources as inclusions. After the engine is deployed and set up, dimensions can be managed from the Admin > Alarm Monitor > ML Dimensions view of the Scrutinizer web interface.

Dimension configuration

An ML dimension is defined by the following parameters:

- Inclusion/asset type the dimension applies to (host/subnet or exporter interface)
- Template field to use for grouping (`sourceipaddress` or `destinationipaddress`, host/subnet dimensions only)
- Aggregation method to use (`octetdeltacount` or `packetdeltacount`)
- Traffic port used

Note

A feature dimension is only observed for traffic associated with the type of *inclusion* (host/subnet or exporter interface) it was defined for.

Dimensions can be configured to apply to all or only internal traffic matching the definition. They can also be disabled and re-enabled as necessary.

Recommendations

Dimension recommendations

Once deployed, the ML Engine defaults to Plixer's recommended dimension definitions, which are based on the traffic in typical enterprise environments.

These default definitions should be reviewed and, if necessary, additional dimensions should be defined to monitor critical network services that are most often the target of attacks, such as:

- Authentication - Kerberos, NTLM
- Domain services - LDAP, DNS, DHCP
- File sharing services - SMB, NFS, CIFS
- Remote connectivity - SSH, Telnet, RDP, VNC, FTP
- Email protocols - SMTP, POP3
- Inter-process communication - ICMP
- Application protocols - HTTP, HTTPS
- Others - DB services, third-party APIs (especially those that connect to the Internet)

Global ML settings

The global ML settings under Admin > Settings can be used to configure parameters for certain ML functions and behaviors across all engines in an environment.

The default values for the above settings/options are recommended for new ML engine deployments but may be adjusted later as described *here*.

AD Users

The Plixer ML Engine is also able to ingest user activity data and access logs and alert users to anomalous behavior through user and entity behavior analytics (UEBA) detections.

UEBA alerts for Active Directory users can be enabled by adding the credentials for a Microsoft Azure account that is configured to store AD user sign-in logs under Admin > Settings > ML AD Users.

Alerts

There are three categories of alert settings that can be adjusted under Admin > Settings > ML Alerts:

Microsoft Office 365 alerts

These sensitivity values adjust the magnitude of deviation from typical behavior that will trigger the corresponding alerts. A higher value allows for greater deviation, resulting in fewer alerts for the corresponding activity.

- *Logon Sensitivity*: Unusual volumes of Office 365 login events
- *Unique Source Sensitivity*: Traffic coming from unusual numbers of unique hosts
- *Unique Location Sensitivity*: Traffic coming from unusual numbers of unique locations

Like *inclusion sensitivities*, these values should only be adjusted after assessing the accuracy of alarms/detections.

System vitals alerts

These thresholds control alerts and other actions related to high utilization of the ML Engine's resources.

- *CPU/RAM/Disk Alert Threshold*: Percentages at which a high utilization alert for the corresponding resource is triggered
- *Disk Reclaim Threshold*: Disk utilization percentage at which the ML Engine will attempt to delete old indexes from Elasticsearch

Initially, these thresholds should be left at their default values. If *alarms* are triggered, run an *ML Engine CPU*, *ML Engine Memory*, and/or *ML Engine Storage* report to assess whether threshold(s) need to be increased (for temporary spikes) or additional resources should be allocated to the engine (for sustained high utilization).

Kafka lag thresholds

These thresholds manage the amount of latency tolerated by the Kafka engine before the corresponding lag alert is triggered.

- *Kafka Netflow Lag Threshold*: Alerts for flow ingestion latency
- *Kafka K-means Lag Threshold*: Alerts for prediction latency
- *Kafka Alerts Lag Threshold*: Alerts triggered by automated process reconnaissance
- *Kafka Training Data Lag Threshold*: Alerts for behavior modeling latency
- *Kafka UEBA Lag Threshold*: Alerts for user and entity behavior analytics (UEBA) data latency

If *alarms* are triggered, run an *ML Engine Kafka Lag* report to determine whether there is a need to scale up the engine's resources.

Data limits

The ML Engine's data limit settings manage the maximum numbers of behavior models and hosts used for network/user activity patterns and prediction. The initial values set are based on the engine's default resource configuration, but they can be adjusted under Admin > Settings > ML Data Limits.

If there are *alarms* associated with these limits, the engine may need to be provisioned with additional resources to sustain the current volume of inclusions.

> **Note**
>
> To check the utilization for the current model limit, run an *ML Engine Model Count* report.

Training schedule

The settings under Admin > Settings > ML Training Schedule determine the seasonality applied when the ML Engine ingests traffic data, allowing it to distinguish between network activity during and outside of an organization's hours of operation.

The engine defaults to business hours of 8 am to 6 pm, from Monday to Friday. These settings can be changed after deployment if necessary.

ML cluster settings

The ML engine is built on Kubernetes, which deploys scalable pods to handle various tasks. Most services within the ML engine consume data from Kafka, which acts as the system's backbone for message passing both from Scrutinizer and between internal components.

Kafka allows the use of consumer groups which allow multiple pods to share workloads efficiently, making it easy to scale services horizontally by increasing the number of replicas. This supports high-throughput processing and flexible resource allocation across services like data ingestion and model training.

To ensure optimal performance based on the scale of the deployment and the volume of data processed, the engine management page can be used to *register* and manage ML engine deployments and configure various settings for individual engines.

Engine settings are accessed via the configuration tray, which is divided into the sections below.

Settings

The *Settings* secondary tray contains the following settings that can be adjusted to adjust resource allocations for specific engine tasks/services:

Ingestion Replica Count

This setting defines the number of data-ingestion pods running in the cluster. Adjusting this value helps scale the ingestion throughput and ensures SLAs are met.

These pods are responsible for consuming netflow data from Kafka topics, processing and aggregating flow records for both security (PSI) and network (PNI) monitoring, storing the processed data into Elasticsearch indices, and handling classification data for supervised ML models. The ingestion service runs continuously, processing data in one-minute intervals and logging heartbeats to ensure operational visibility.

Train Anomaly Detection Replica Count

This setting specifies how many pods are deployed to train ML models for the anomaly detection service.

This service uses techniques such as silhouette analysis and overfit deviation detection with configurable thresholds and limits. Once trained, the models are published to Kafka topics for use by downstream services. Scaling this service allows for faster and more resilient model training, particularly in environments with large or complex datasets.

Ingestion CPU & Memory (Min/Max)

These settings define resource limits allocated to the data ingestion service.

These resource limits ensure that each ingestion pod has the CPU and memory it needs to handle high-throughput data processing tasks. The ingestion service performs complex transformations, maintains multiple in-memory maps for real-time analytics, and conducts bulk insert operations into Elasticsearch. Typically, memory allocations in the range of 1 GB to 2 GB are required to support the various data structures used during processing.

Elasticsearch Memory and CPU (Min/Max)

These settings define resources allocated to the Elasticsearch cluster pods managed via the ECK (Elastic Cloud on Kubernetes) operator.

Since Elasticsearch is used to store and index all processed flow and classification data, and must support real-time search queries for machine learning and security operations, it is essential that sufficient resources are provided. The Java heap is configured with 8 GB (using -Xms8g -Xmx8g), and a total memory allocation of approximately 12 GB is recommended to provide additional headroom for OS and Elasticsearch operations. Similarly, minimum and maximum CPU allocations help maintain consistent indexing and query performance.

The Kibana UI can also be deployed alongside Elasticsearch by toggling on the *Enable Kibana* option.

Collectors

Collectors selected here will be used as data sources for ingestion by the current engine.

DGL IP Groups

IP groups added to the Deep Graph Learning inclusion list will be monitored by the engine to identify anomalous interactions between hosts.

4.3 Use Cases

See below for common use cases and sample workflows for *NetOps* and *SecOps* teams.

4.3.1 NetOps Use Cases

Custom reports

Aggregate flow data by any dimension to inspect any host or activity

Customizable observation points and reporting **Collaboration**

Streamline information sharing and enhance multi-role workflows

Team collaboration **Investigating congestion**

Monitor health and performance in real time to quickly identify root causes

Investigating network congestion **Scheduled email reports**

Proactively monitor specified network traffic from any email inbox

Scheduled email reporting **Network visualization**

Create and customize network maps to visualize what matters to your team

Network mapping and visualization **NOC dashboards**

Maintain multiple dashboards for diverse roles and workflows

NOC dashboards and forensics **NPM**

Continuously monitor network performance for deeper traffic insights

Network performance monitoring (NPM) **Capacity planning**

Monitor data circuit usage over time to plan future needs and optimize costs

Capacity planning **Cloud visibility and detection**

Bridge visibility between cloud and on-prem assets without probes

Cloud visibility and detection

4.3.2 SecOps Use Cases

Service behavior monitoring

Continuously monitor critical services for anomalous usage

Service behavior monitoring **General malware detection**

Monitor network activity to identify malware-infected hosts

General malware detection **Threat hunting**

Inspect device behavior details and pinpoint Indicators of Attack (IoAs)

Threat hunting **Lateral movement detection**

Monitor network activity to detect lateral movement behavior

Lateral movement detection **Incident response**

Enhance incident response with added visibility and UI-driven workflows

Incident response

4.3.2.1 Customizable observation points and reporting

With the Plixer One Platform (Core or Enterprise), users can use Scrutinizer to configure/run their own purpose-built reports. These reports are fully customizable and can be used to visualize network performance, identify problem points, and investigate root causes of network issues. Reports can also be continuously refined to filter, drill down, and/or pivot as part of monitoring or investigative activities.

Overview

Reports in Scrutinizer aggregate data from one or more devices/sources based on the dimensions defined in the base report type. To further adapt a report to more specific monitoring and investigative needs, there are a range of settings that can be modified.

Configuring reports

In addition to the base type and data sources, reports use the following settings when they are run:

- Time period covered (either *last X* or custom date/time ranges)

- Graph/visualization type
- Filters

Each report can have multiple filters in any combination of filter types (device/interface, domain, host addresses, etc.) defined as either inclusions or exclusions. Additionally, filters can be configured so that they include only source hosts, destination hosts, or both.

Report settings, including the base type and devices/sources, can be set/changed in the report creation wizard or when refining the output after the report is run.

Hint

In the report output view, table elements can be dragged to *Include* and *Exclude* drop zones to re-define the report's inclusions and exclusions. Additionally, clicking on a dimension element opens a tray that allows the user to pivot to any other report type available for that element.

Additional options

After a report is created/run, it can be saved and/or exported in several ways to enhance a wide range workflows.

With Plixer One Enterprise, saved reports can also be used to generate forecasts for *capacity planning* and to enable *more efficient collaboration* between team members.

4.3.2.2 Team collaboration

To support the growing scale and complexity of enterprise environments, the Plixer One Platform (Core or Enterprise) includes multiple functions that enable greater efficiency in collaborative processes and workflows:

- Save custom reports and allow other members to access/re-run them at any time
- Email reports directly to concerned parties or export them for use in external systems
- Compile alarm details and/or reports into collections for review/investigation by multiple team members
- Assign one or more notification actions (including email alerts) to alarm policies through customizable notification profiles.

Overview

Scrutinizer includes multiple features and functions that are designed to streamline the sharing of network and incident information between members and teams.

Saved reports

Scrutinizer reports function as a customizable network visibility interface, where you can continuously filter, drill down, and pivot to different report types to monitor specific network elements or *identify problem points*. Once a report configuration is saved, other users can be given access (through user groups) to re-run it or add it to their *dashboards*.

Hint

A saved report can also be used to set up a scheduled email report to automatically run and email the report to any number of users at regular intervals.

Report/notification emails

Once an email server has been configured, Scrutinizer can be set to send alerts and reports directly to user inboxes:

- On-demand email reports after any report is run
- Scheduled email reports at user-specified intervals
- Alarm/event email notifications, which are triggered via notification profiles assigned to alarm policies

Hint

Both email report types also include a link to run the report in the Scrutinizer web interface. PDF and/or CSV copies of the report may also be attached.

Collections

Collections are compilations of alarm/event data or reports that are assigned to specified users for review, analysis, or resolution. In addition, collections can be viewed by other users, who are able to add annotations directly to the collection item's details and/or engage in discussions via threaded notes/comments.

Hint

While reviewing a collection, a user can click on individual items to quickly jump to more detailed views.

Important

Collections are part of the **Plixer One Enterprise** solution. Contact *Plixer Technical Support* for more information.

Workflows

The following workflow(s) show how the Plixer One Platform can drive more efficient collaborative workflows through various functions:

Sharing information via collections

The network team discovers suspicious traffic and wants to share the information with an independently operating security team. Instead of exporting the information and sending it via email, they create a collection containing the relevant reports and/or alarm data that can be accessed by other Scrutinizer users at any time.

Workflow

1. From the *Alarm Monitor* view or *Report* page associated with the suspicious traffic, create a new collection from the Manage Collections submenu (star button) and set it as the active collection.

Tip

To use an existing collection instead, click the star button and select the collection from the menu.

2. To add an item after the active collection has been set, click the star button from any relevant alarm/event information view or report, and then click it a second time, after it has been replaced with a **+**.

3. Repeat the previous step to add additional items to the collection.

All Scrutinizer users can access existing collections via the Investigate > Collections page of the web interface. When inspecting a collection, users can add notes to the individual items or for the collection itself.

Hint

The default view of the Collections page displays all collections that have been assigned to the current user. To see other collections, switch to the *Other Collections* tab of the page.

Collections that are no longer relevant can be deleted by selecting them from the main *Collections* page and clicking the **Delete** button.

4.3.2.3 Investigating network congestion

In almost any modern enterprise environment, identifying the who, what, where, when, and why behind congestion issues requires tools that go beyond inundating network teams with large volumes of raw data.

Through Scrutinizer, the Plixer One Platform (Core or Enterprise) enables multiple approaches to dealing with network congestion issues:

- Drill down into network device/host activity to identify root causes for congestion by applying one or more filters and pivoting between different report types.
- Monitor network devices and/or interfaces for congestion in the Top Interfaces view.
- See real-time rates and utilization between devices and other objects in network maps by adding *connections* with custom color-coded thresholds.
- Get high utilization alerts via the Scrutinizer *Alarm Monitor* by adding user-defined thresholds to reports.

Overview

Teams can leverage the following Scrutinizer features/functions to proactively watch for network congestion, collect insights into the root cause(s), and respond efficiently.

Reports

Reports aggregate data from any number of user-specified devices and dimensions and can show sources of congestion and bandwidth consumption:

- Identify “Top Talkers” on the network using *Source* and *Destination* reports.
- View peak and 95th percentile in *Traffic Volume* reports.
- Check for latency and packet loss with FlowPro APM *Application Retransmission* reports.
- Apply any number of filters for subnets, applications, usernames and then pivot directly to another report type to narrow down your results.

Report Thresholds

Custom thresholds can be added to saved reports to monitor for congestion and trigger alarm monitor *alerts* when those thresholds are reached. With a report threshold configured, the report can be re-run to monitor for min/max bandwidth utilization and mitigate regression after congestion sources are identified.

Hint

If a notification profile is assigned to the *Report Threshold Violation* alarm policy, the threshold can be used to trigger notification actions, such as email alerts and CEF notifications for external tools.

Top Interfaces view

The Top Interfaces view (**Explore** > **Exporters** in the web interface) can be used to monitor all device interfaces, from the most saturated down to the least utilized. This allows network teams to identify which ones are most affected by congestion at a glance. The view can also be used to inspect highwater marks that indicate peak saturation over a period of time.

Hint

The **Explore** > **Exporters** page can be set to show either *By Interfaces* or *By Exporters* as the default in your user preferences menu.

Map Connections

After a network map is populated with devices and other objects, it can be further customized with connections representing activity between devices, objects, and/or interfaces. Connections can also be individually configured with utilization thresholds that change the color they're displayed in, giving teams a bird's eye view of potential congestion issues in real time.

Hint

Click on devices or interfaces in a network map to quickly jump to the Top Interfaces view filtered on the object.

Workflows

The following workflows show how multiple Plixer One Platform functions can help network teams mitigate, and/or investigate network congestion issues.

Monitoring for congestion issues

A user calls in reporting that everything on the network is taking an excessive amount of time to load, indicating network congestion.

Workflow

- Navigate to **Explore > Interfaces**
- Identify instantly if any interfaces are congested
- Open a "Conversations" Report to see the top source and destinations of bandwidth
- We may find that a host on the network is performing write intensive backups during the day and eating up all available bandwidth.

Tip

If Host Indexing is turned on, you can look up a user's IP and see all network devices that saw that address.

Note

Scrutinizer records *highwater marks* that represent the peak utilization for each interface.

Troubleshooting poor call quality

The sales teams reports that outbound calls have been of poor quality recently. Jitter happening sporadically on the call, making it difficult to conduct business efficiently.

Workflow

- Navigate to **Reports > Run Report > Select Report Types**
- Under the *Flowpro APM Reports* category, select a report like 'Host to Host Jitter All by SSRC'
- Open the report and note the report columns such as Source Jitter and Packet Loss
- We may find that we can measure the jitter and packet loss and see what the RTP payload type was. Perhaps the subnet traffic is not using class-based QOS and voice traffic isn't being prioritized.

Note

FlowPro is part of the Plixer One platform. To learn more, see the section on *FlowPro integration*.

4.3.2.4 Scheduled email reporting

With the Plixer One Platform (Core or Enterprise), NetOps teams can use Scrutinizer reports as a proactive monitoring tool for any type of network meta data by setting up scheduled email reports.

Overview

A scheduled email report is a saved report that has been set to run at specified intervals using the exact same configuration (graph, filters, etc.). Each time the report is run, its output is automatically emailed to one or more recipients.

Note

Scheduled email reports are different from on-demand report emails, which must be sent manually after a report is run.

All email reports contain a direct link to the primary report and may also include PDF/CSV copies of the report. One or more additional reports can also be run and sent in the email.

Setting up a scheduled email report

A scheduled email report can be set up after re-running a saved report (or after creating and saving a new report).

From there, click the *Export Report*/share button, select *Schedule Report* in the tray, and configure the following:

- A name for the scheduled email report configuration (used for configuration management and as the subject line of the email)
- One or more recipient addresses (comma-separated)
- Frequency and time (minute on the hour) to run and send the report
- (Optional) PDF and/or CSV format attachments (all included reports)
- (Optional) Additional reports to run and include in the email

Once set up, the report(s) will be run/sent at the specified intervals until the scheduled email report configuration is disabled or deleted.

Hint

To inspect, edit, or disable scheduled email report configurations, navigate to **Admin** > **Reports** > **Scheduled Email Reports**.

Workflows

The following workflow(s) show how the Plixer One Platform is able to continuously monitor specific network traffic through scheduled email reports:

Automating weekly reports

Important

To set up scheduled email reports, an email server must first be configured via the Admin > Integrations page.

Management wants to see summarized data concerning the network emailed on a weekly basis.

Workflow

First off, identify the details that are most critical to report on. Some examples are: top applications, top used ports, destination countries, etc. Regardless of the report types required, the same steps are used to add reports to your scheduled report.

1. Select **Reports > Run Report > Select Report Type** to start a report.
2. Choose **Destination Reports > Countries with AS**, and then select the appropriate network devices to include in the report.
3. Change the range of the report to **Last Seven Days** to show the entire weeks network data.
4. Save and give this report a name.
5. Export the report as a gadget from the Options tray.

Repeat the same steps for the other reports, making sure the time range is *Last Seven Days*

- Pair Reports > Conversations Apps
- Top > Protocols
- Top > Well Known Ports

Now that the reports that will be sent weekly have been created, they can now be assigned to a scheduled report.

Assign the frequency to *Weekly* and set time to the day of the week and time to see this email come through, "Friday 5:00pm". Options include adding PDF and CSV attachments along with the email.

Be sure to select the reports that were created for this scheduled email and add them to the include list. After a scheduled report configuration has been set up, it can be viewed or edited from **Admin** > **Reports** > **Scheduled Email Reports**.

4.3.2.5 Network mapping and visualization

With the Plixer One Platform (Core or Enterprise), network teams can leverage Scrutinizer's integrated network mapping functions to create and customize maps that are based on user-defined device groups. These maps are continuously updated in real time, allowing them to function as both a high-level view of network health and a starting point for investigating connectivity issues.

Overview

When creating a new map in Scrutinizer, users can select between *Spatial Maps* to fully customize the device layout or *Geographical Maps* for location-based arrangement.

After a network map is initially generated, it can be further customized/configured at any time. Existing network maps can be viewed from the **Monitor** > **Network Maps** page or as *dashboard gadgets*.

Spatial Maps

Using the following configuration options, spatial maps can be used to design fully customized topologies to meet different visualization requirements:

- Position map objects against custom backgrounds to recreate office layouts, wiring closet connections, and more.
- Add custom objects to represent non-exporters, such as external hosts
- Define connections between objects (devices, interfaces, and/or custom objects) to indicate static links, display interface utilization, or run a saved report using the connected objects
- Add custom utilization thresholds to connections to show overall network health and potential congestion issues
- Nest mapping groups within each other and create multi-layered maps to support network segment planning and monitoring
- Tailor maps to specific team role or workflow needs and manage access via *dashboards* and user groups.

> **Hint**
>
> Bulk management functions for mapping objects and groups can be accessed via the Mapping Objects and Mapping Groups pages under **Admin** > **Settings** in the web interface.

Geographical Maps

Object positions in geographical maps are determined by their longitudinal and latitudinal coordinates. Both manual coordinate entry and address lookups via Google Maps are supported.

Hint

Objects can be assigned unique coordinates/addresses for every map/group they are assigned to.

Geographical maps support similar configuration/customization options as spatial maps (except for object positioning and custom backgrounds) and can be used to enhance many of the same workflows. They are also ideal for monitoring the health and performance of geographically segmented networks.

Workflows

The following workflow(s) are examples of workflow enhancements enabled by Scrutinizer's live network maps in the Plixer One Platform:

Mapping your network

To streamline NOC workflows in their growing environment, the team decides that they need a visual representation of the network and critical applications.

Workflow

To set up the new map, navigate to **Monitor** > **Network Maps** and create a new spatial map:

1. Use a name that matches the coverage of the map (e.g., the entire network).
2. Assign all applicable devices (routers, firewalls, switches) as map objects.
3. Link devices as necessary by creating connections. Connections can be static lines, interface representations, or saved reports.

Hint

When a saved report is used as a connection, it will represent the traffic aggregated by the report. This can be anything from a layer 7 application (e.g., YouTube) to firewall events from a Cisco ASA. In the latter case, the connection will typically be grayed out (inactive), and can serve to quickly alert the network team when it becomes active.

If the network topography changes at a later time, the map can be updated to reflect the changes.

For larger networks, such as those that span multiple locations, it may be ideal to create smaller maps representing individual network segments and nest them under a larger map as objects. This will create a "global" map with a hub-and-spoke layout.

4.3.2.6 NOC dashboards and forensics

As ubiquitous as dashboards have become in network operations center (NOC) workflows, many tools remain limited by the lack of customization options for data sources, gadgets, and auxiliary features.

Scrutinizer dashboards–part of the Plixer One Platform (Core or Enterprise)–can be customized to support and enhance any number of unique user roles and/or workflows.

Overview

Scrutinizer users are able to create any number of uniquely configured dashboards to support and enhance their individual workflows.

Dashboard management

When *creating a new dashboard*, users can choose between starting with a copy of an existing dashboard or populating a "blank" dashboard with their own selection of gadgets.

Existing dashboards also have the following additional management/configuration options:

Set as default	Selects the dashboard as the default for the current user
Set as read-only	Locks dashboard settings and gadgets until toggled off
Modify user access	Shows or hides the dashboard for individual users
Modify user group access	Shows or hides the dashboard for user groups

Hint

To change the layout and gadgets for existing dashboards, switch to *edit mode* while dashboard is active.

Custom Gadgets

To complement the preconfigured gadgets bundled with Scrutinizer, network maps and reports can also be added to dashboards as gadgets. This allows users to view/access frequently used maps and reports directly from their preferred dashboard(s) instead of navigating to the corresponding sections of the web interface.

Existing network maps or reports (must be exported first) can be added when setting up a new dashboard or while in dashboard edit mode, provided the current user has access via their user group.

External gadgets

External gadgets are another type of custom gadget that embed data from third-party sources (via URL) in dashboards.

These gadgets can be used to expand visibility, acquire further insights, or provide convenient access to supplemental information.

Hint

External and report-based gadgets can be configured with custom refresh intervals to always display the data that is most relevant to users.

Workflows

The following workflow(s) show how the Plixer One Platform is able to enable and enhance UI-driven workflows with Scrutinizer Dashboards:

Multi-tenancy dashboards

As part of a multi-tenant environment, the operator wants to provide each customer with a dashboard for their network.

Workflow

Assuming two groups (A and B), each group should have exclusive logins so that only content relating to their group is accessible to their users.

This workflow assumes that each of these groups consists of a location with three network devices sending netflow data:

- Firewall
- Core Router
- Switch

The dashboard should contain a single top conversations report for the group's network and be accessible to all users under that group/location.

1. *Create a dashboard* for each group (e.g., Dashboard A and Dashboard B). This will allow you to export the appropriate reports to them after they have been created.
2. Create Group A's report:

   ### Report configuration

   1. Start by adding devices and select the IP addresses of Firewall A, Core Router A, and Switch A.
   2. Select *Conversations App* (under the *Recommended* category as the report type).
   3. Change the time window/range of the report to *Last 24 hours*.
   4. After running the report, save it under a name associated with Group A (e.g., Top Conversations A)
   5. Click the share button and select *Add to Dashboard*.
   6. In the secondary tray, select Dashboard A from the *Dashboard Tab* dropdown and choose what content to show in the gadget (graph, table, or both).

   > **Note**
   >
   > If a different name is entered in the *Report Name* field, a new, separate report will be saved. The new name will also be used as gadget label.

3. Repeat the previous steps using Firewall B, Core Router B, and Switch B, and export the report to Dashboard B.
4. Set up the report folders for each group:

   ### Report folder configuration

   1. Navigate to **Admin** > **Classic Admin** > **Reports** > **Report Folders**.
   2. Click the *New Folder* button and enter a name for Group A's folder (e.g., Report Folder A).
   3. Add the report that was created for Group A to the folder by selecting it and clicking the <- *Add* button.
   4. Repeat the steps to create the folder for Group B and add their report to it.

5. Create a map for each group's network:

   ### Network map configuration

   1. Navigate to **Monitor** > **Network Maps** and create a new spatial map for Group A (e.g. Map A).
   2. Assign Firewall A, Core Router A, and Switch A as map objects.
   3. Link the devices as necessary using connections.
   4. Repeat the steps to create the map for Group B.

> **Hint**
>
> The report previously created for Group A (or any other saved report) can be used to create a connection representing that traffic type between devices. These reports can also be added to dashboards for up-to-the minute display of the traffic covered.

6. Set up the user groups:

   ### User group configuration

   1. Navigate to **Admin** > **Users and Groups** > **User Groups** and click the **+** button to create a new group.
   2. In the tray, enter a name (e.g., Group A Users) and select *Guest* as the starting template from the dropdown.
   3. After the user group has been created, locate it in the main table and click the links under the columns to make the following changes:

   - Devices: Select only Firewall A, Core Router A, and Switch A.
   - Interfaces: Select only interfaces that should be visible to Group A (all interfaces associated with their devices, in most cases)
   - Reports: Select all reports and report folders created for Group A.
   - Dashboard Gadgets: Select only gadgets (based on saved report names) that were created for Group A.

   4. Repeat the steps to set up the usergroup for Group B.

7. Navigate to **Admin** > **Users and Groups** > **User Accounts** and click the **+** button to create login credentials for one or more users for each group. Use the dropdown in the tray to add each user to the appropriate user group.

> **Hint**
>
> Users obtained from LDAP or another identity provider can also be added to user groups.

After everything has been set up, users from each group will only have access to the devices/interfaces, reports, and dashboards/gadgets belonging to their group.

4.3.2.7 Network performance monitoring (NPM)

Without true visibility into traffic patterns and trends, additional provisioning may seem like the only way to keep up with a network's growth.

With Plixer One Enterprise, network teams can access detailed information related to application performance and performance costs, in addition to being able to examine end-to-end network conversation details through Scrutinizer's reporting and filtering functions. Users can also leverage the ML Engine to forecast any future network traffic/behavior.

Overview

Plixer One Enterprise includes multiple functions/components that can enhance a network team's ability to monitor and manage network performance down to the application level.

Reports

In Scrutinizer, reports can help network teams understand the root causes of traffic saturation on a network's top interfaces. When used in conjunction with alarms for interface threshold violations, they can get alerted to saturated circuits and will have the means to uncover what that traffic consists of.

APM

Plixer One Enterprise provides application performance monitoring functions that are designed to support teams in ensuring consistently optimal experiences for their users:

- Measure application round-trip time (RTT)
- Monitor latency for Layer 7 applications, clients, servers, and VoIP communication
- Diagnose issues using SSRC, ToS, jitter, retransmission rates, and other packet metrics

Forecasts

By combining the capabilities of Scrutinizer with the ML Engine, Plixer One Enterprise can provide users with forecasts of future network activity to support capacity planning initiatives. These forecasts can help network teams visualize trends of network growth and predict behavior based on the patterns exhibited by past activity.

Once a report has been configured with the correct settings and filters, it can be used to generate a forecast that predicts the state of the same traffic into the future.

Data history

Scrutinizer can be tuned to keep historical data for as long as needed through its data retention settings.

Because raw alarms come in off the wire and are stored each minute, the data stored for that interval offers the most granular historical information. To make more efficient use of disk space, however, Scrutinizer automatically aggregates that data and rolls it up into 5m averages for up to 2-hour intervals. This allows for historical data to be kept for a longer period of time.

To learn more about how Scrutinizer aggregates historical data, see *this section* of this documentation.

Important

APM-specific reports and forecasting are only available with Plixer One Enterprise. Contact *Plixer Technical Support* to learn more.

Workflows

The following workflow(s) show how the functions and features included in Plixer One Enterprise can help teams monitor network and application performance in their environment:

Monitoring for congestion issues

A user calls in reporting that everything on the network is taking an excessive amount of time to load, indicating network congestion.

Workflow

- Navigate to **Explore > Interfaces**
- Identify instantly if any interfaces are congested
- Open a "Conversations" Report to see the top source and destinations of bandwidth
- We may find that a host on the network is performing write intensive backups during the day and eating up all available bandwidth.

Tip

If Host Indexing is turned on, you can look up a user's IP and see all network devices that saw that address.

Note

Scrutinizer records *highwater marks* that represent the peak utilization for each interface.

4.3.2.8 Capacity planning

Through Scrutinizer, Plixer One Enterprise can leverage the capabilities of the ML Engine to generate forecasts of future network activity for capacity planning:

- Apply machine learning techniques to create dynamic baselines for network behavior
- Extend any Scrutinizer report into the future to forecast trends and predict changes to network activity
- Use AI-/ML-driven data analysis to predict VPN trends, proactively plan capacity, and align investments with business needs
- Gain visibility into encrypted VPN tunnels to detect threats
- Gain visibility into address pool utilization and trend its usage
- Associate users, devices, and applications with the consumption of bandwidth

Overview

Scrutinizer includes multiple tools and functions that can enhance a network team's capacity planning capabilities.

Traffic/behavior baselining

Using collected flow data, the *Plixer ML Engine* is able to create dynamic machine learning models of baseline network behavior.

Plixer One Enterprise can use these models to deliver additional capacity planning insights in two ways:

- Alarms for behavioral deviations that exceed a certain threshold (based on the configured sensitivity) using the *Plixer Network Intelligence Anomaly* policy
- Activity/deviation monitoring via **Behavior** tab when drilling into individual hosts from the Explore > Entities > Hosts view.

Reports

Scrutinizer's customizable reports are designed to help teams get to the bottom of any inquiry.

For capacity planning, they can be used to investigate traffic saturation on top interfaces and help determine whether additional provisioning will be required.

Forecasts

Forecasting is a Plixer One Enterprise feature that allows users to create forecasts of future network activity.

A forecast can be generated from any saved report and will comprise projections for the traffic included by the report configuration (e.g., devices, filters, etc.). This gives teams the ability to define the exact network activity to be forecasted as part of capacity planning.

HD utilization projections

On the Admin > Resources > System Performance page, clicking on a collector opens a view showing predicted HD utilization based on the current data retention settings. These projections can be used to ensure that sufficient disk space is always available to meet historical data storage needs.

Workflows

The following workflow(s) show how teams can leverage Plixer One Enterprise functions to enhance their capacity planning capabilities:

Forecasting and meeting business needs

The network team is asked to predict how long an organization's current infrastructure will continue to support their business needs. To visualize trends in network growth, they create report configurations for various aspects of the environment and use them to create ML-driven forecasts in Scrutinizer.

Workflow

Because Scrutinizer forecasts are based on reports, the environment's current capabilities should be split up into separate capacities, such as:

- WAN usage
- VPN traffic
- Subnet-to-subnet patterns
- BGP traffic
- Core router saturation
- Critical application latency

From there, one or more report configurations should be created and saved for each capacity. These reports can then be used to generate forecasts that will show emerging utilization trends. At the same time, any latency problems discovered may also indicate potential capacity issues that need to be addressed, depending on their frequency and degree of deviation from the baseline.

4.3.2.9 Cloud visibility and detection

The Plixer One Platform (Core or Enterprise) enables seamless visibility across on-prem and cloud-based resources in cloud or hybrid environments through cloud provider log ingestion in Scrutinizer.

Overview

After the corresponding cloud storage container is set up to receive log data from an AWS, Azure, or OCI virtual network, Scrutinizer can be configured to ingest the information via the container. Containers that have been set up as flow data sources in Scrutinizer are treated as exporters and support the same functions and configuration options as typical flow-exporting devices (e.g., flow analytics, ML rules, and reports).

Amazon VPC flow logs

To enable Amazon VPC flow log ingestion in Scrutinizer, the VPC must first be set to send log data to an Amazon S3 bucket with the *correct configuration*. Afterwards, the bucket should be added to Scrutinizer from the *Admin > Integrations > Flow Log Ingestion page* in the web interface.

The following *additional report types* can be run when one or more S3 buckets are selected as data sources for a report:

- Action
- Action with Interface
- Action with Interface and Dst
- Action with Interface and Src
- Availability Zones
- Dst Service
- Interface
- Pair Interface
- Pair Interface Action
- Src Service
- Src Service-Dst Service
- Traffic Path
- VPCs

Tip

To view only report types that apply to Amazon VPC flow logs, use the *Amazon AWS* category when selecting a report type.

Azure flow logs

Setting up Azure flow log ingestion in Scrutinizer requires an Azure Blob Storage container that is *correctly configured* and receiving log data from the virtual network. This container should be added to Scrutinizer from the *Admin > Integrations > Flow Log Ingestion page* in the web interface.

When one or more Azure blob containers are selected as data sources for a report, the following *additional report types* become available:

- Flow Decisions
- Flow Decisions Count
- Flow States
- Flow States Count

- All Details
- Resource IDs

Tip

To view only report types that apply to Azure flow logs, use the *Azure* category when selecting a report type.

4.3.2.10 Service behavior monitoring

Plixer One Enterprise addresses the limitations of traditional security technologies by applying AI and ML techniques to provide early, generic detections for activity associated with advanced persistent threats (APTs).

These detections rely on behaviors rather than signatures and give security teams an additional layer of defense against attempts to use common services to infiltrate, infect, and exploit network resources.

Overview

Plixer One Enterprise's approach to *anomaly detection* relies on the ML Engine to turn the flow data collected by Scrutinizer into behavioral models that represent typical host activity. All incoming flow data can then be compared against these baseline models to proactively scan for potentially malicious activity and alert security teams in real time.

Configuring anomaly detection

The ML Engine's anomaly detection functions can be adapted to any type of environment through its *configuration*:

Dimensions	Services/applications (protocol and port) whose behavior is modeled and monitored for anomaly detection
Inclusions	Hosts (by exporter or subnet) being monitored for anomalous behavior
Sensitivity	The tolerance for deviations from baseline service behavior for hosts associated with the inclusion

Defining dimensions and inclusions for the engine isolates traffic information to reduce the amount of "noise" and maximize the accuracy of detections. Organizations are also able to tune detections to their unique processes and workflows by adjusting the sensitivity for individual inclusions.

Hint

Low sensitivity is generally recommended for critical subnets (e.g., finance, HR, etc.) where all irregularities should be reported, while a *High* can be used for hosts whose security requirements are less strict.

Investigating anomaly detections

Once anomalous behavior is reported via an alarm, the appropriate response can be determined using a combination of Scrutinizer workflows, including:

- Drilling down into the alarm (e.g., *Plixer Security Intelligence*, *Lateral Movement Behavior*, etc.) and checking the timeline to determine whether the detection is an isolated observation or an ongoing event
- Inspecting event artifacts to see which hosts were involved and drilling into them to gain further insights from *Endpoint Analytics*
- Reviewing activity via the **Behavior** tab when drilling into hosts from the Explore > Entities > Hosts view.

- Running *Source* and *Destination* reports on the hosts to check for traffic between them and external IP addresses

Hint

After running an initial report, it can be refined directly from the output view to enable further investigation.

Workflows

The following workflow(s) show how alarms related to anomalous service behavior are used to investigate potential cyber attacks:

Detecting anomalies and deviations

Continuously monitor traffic anomalies or traffic deviations that exceed set thresholds using dynamic ML-modeled baselines.

Workflow

Machine learning allows Scrutinizer to alert users to anomalous traffic utilization patterns typically associated with security incidents.

Note

This workflow requires the Plixer ML Engine for predictive modeling. Contact *Plixer Technical Support* to learn more about licensing options.

All incoming flow data can be compared against these baseline models to proactively scan for potentially malicious activity and report discoveries in real time.

From there, the next steps should be to set up reports and use them to generate forecasts.

Identify which areas of the network (devices and interfaces) have the majority of traffic:

- What types of traffic would you expect to see – VoIP, HTTP, SQL?
- Business application traffic like Salesforce, AWS, Azure etc.
- DNS requests to dedicated DNS servers on the network

Now consider traffic that may be anomalous:

- Does Remote Desktop Protocol make sense on this network, is there a business use case for RDP?
- Should there be SSH traffic to critical hosts?

Based on the above considerations, create/run one or more reports to isolate traffic data for services, hosts, or device groups that are most likely to be involved in malicious activity. Once saved, these reports can then be used to forecast expected traffic patterns and highlight deviations (e.g., an anomalous ICMP data trend in outbound WAN usage for edge devices) that can be analyzed to identify threats.

Next steps would be to customize alerts for this behavior or other traffic deviations that exceed user-defined thresholds configured for the report(s).

Tip

Scrutinizer's alarm policies can be assigned custom notification profiles. To add one or more notification actions for all report thresholds, create a notification profile and assign it to the *Report Threshold Violation* policy.

4.3.2.11 General malware detection

Because all malicious activity leaves footprints in network traffic, the visibility provided by traffic data can be an invaluable asset against modern malware.

By ingesting large volumes of network information through Scrutinizer, Plixer One Enterprise can provide general malware detections and extract additional value from the same flow data.

Overview

The ML Engine uses *classification* - a machine learning technique that relies on models that have been trained on labeled data - to predict whether a host's behavior is indicative of common classes of malware, including command and control, banking trojans, exploit kits, etc. Each prediction is returned in the form of a percentage, which represents the degree to which the observed traffic patterns match those it has learned to be associated with malware. If that percentage exceeds a preset detection threshold, a high-severity event is generated under the corresponding alarm policy in the Scrutinizer alarm monitor.

Enabling malware classification

To optimize resource utilization, malware detection is configured at the ML inclusion level, enabling or disabling classification for all hosts associated with the inclusion. The *Malware Detections* setting can be accessed from the Manage ML Inclusions page, where it can be toggled on or off in the inclusion configuration tray.

Investigating malware detections

Once a detection is reported as an alarm, the appropriate response can be determined using a combination of Scrutinizer workflows, including:

> **Note**
>
> General ML-driven malware detections are reported under the *ML Engine malware alert* alarm policy. A separate *Malware Command and Conquer Activity Detected* policy is used for detections via Flow Analytics.

- Drilling down into the alarm and checking the timeline to determine whether the detection is an isolated observation or an active event
- Inspecting event artifacts to see which hosts were involved and drilling into them to gain further insights from *Endpoint Analytics*
- Running *Source* and *Destination* reports on the hosts to check for traffic between them and external IP addresses

Hint

After running an initial report, it can be refined directly from the output view to enable further investigation.

Workflows

The following workflow(s) are examples of Plixer One Enterprise's malware detections being used as starting points for investigating suspicious network activity:

Alerting on malware activity

Get alerted to any host demonstrating malware activity and send notification to security team.

Workflow

Becoming aware of suspicious activity

Scrutinizer and the Plixer ML Engine can be used together to help assess possible malware activity on your network.

The ML algorithms used for *malware classification* trigger alerts within Scrutinizer's alarm policies for traffic/activity that deviates from dynamic ML-modeled baselines.

Note

This workflow relies on the Plixer ML Engine to report classification-based detections. Additional host analysis and risk assessment functions are enabled through Endpoint Analytics.

Tip

Scrutinizer and FlowPro also use STIX/TAXII and other threat intelligence feeds to identify activity associated with common classes of malware and ransomware.

Responding to potential malware

Review the **Admin > Alarm Monitor > Alarm Policies** page and search for the *ML Engine malware alert* policy. Using a custom notification profile, this policy can be configured to trigger an email to one or more addresses. This can be used to alert security team members whenever there are malware detections that should be reviewed.

Hint

Other automated notification actions can also be defined under the same notification profile.

From the Alarm Monitor view within the UI, you could dive into the alarm policy and investigate the host with details on top applications and conversations.

Scrutinizer reporting can generate host-to-host reports to show the full extent of the host's communications with other IPs on the network. Any outbound traffic with remote hosts should be investigated by navigating to the **Reports** tab/section of the web interface and running destination reports.

Additionally, Endpoint Analytics may be able to provide MAC details for the host and report its own risk assessment based on internal algorithms, MS Defender, and Tenable.

4.3.2.12 Threat hunting

Plixer One Enterprise can enhance any team's threat-hunting capabilities by providing them with centralized access to rich, contextualized data accounting for every host and conversation in a network.

Through Scrutinizer, Plixer One Enterprise is also able to provide real-time alerts for generic malware and other anomalous traffic/activity, drive efficient workflows with its purpose-built UI, and integrate multiple threat intelligence functions. This gives teams the ideal starting point for their threat-hunting operations.

Overview

Scrutinizer plays two integral roles as part of a security team's threat-hunting program:

1. Collects traffic and host data for the entire environment (including *assets in the cloud*), storing hundreds of thousands of data points for investigations
2. Provides centralized access to all available data through various contextual views and reporting functions

This allows SecOps teams to efficiently search through and analyze device-level behavior and host conversations to search for suspicious activity and potential threats. Historical data can also readily be accessed to hunt for indicators of attack (IoA).

Visibility and workflow enhancements

Security teams using Plixer One Enterprise can leverage the following functions and features to hunt for threats:

Alarm monitor

The alarm monitor provides real-time alerts for anomalous behavior and other network activity violating Scrutinizer *alarm policies*. It functions as both a monitoring view for suspicious traffic and an interface for drilling into activity timelines and individual event artifacts, and more.

Customized reports

To further investigate alarms/events, users are able to run reports that can be tailored to their exact visibility requirements. These reports can also be used to drill deeper into specific data elements to identify infected hosts or malicious activity.

Configurable detection mechanisms

Configuration options for *Flow Analytics algorithms* and the *ML Engine* allow users to tailor Scrutinizer's monitoring and detection functions to their specific requirements. This ensures that detections are always relevant and can greatly reduce investigation/response times for security teams.

Note

Plixer One Enterprise includes additional detection techniques and mechanisms for security events.

Host indexing

With the *Host Indexing* FA algorithm enabled, a user is able to look up any IP address, find out whether or not the host has been seen on their network, and explore all activity associated with it. From the search results, the user can pivot directly to any applicable report and further investigate anomalous traffic originating from or targeting the host.

See also

For additional details on incident response workflows with Scrutinizer, see *this use case*.

Workflows

The following workflows are sample scenarios where the functions/features bundled with Scrutinizer are used in threat-hunting activities:

Using host index to identify malicious IPs

Host indexing allows users to quickly look up IP addresses seen on the network, making it ideal for monitoring hosts that have exhibited anomalous or suspicious behavior.

Workflow

To search the host index for malicious IP addresses:

1. Navigate to **Explore** > **Search** in the web interface.
2. In the *Host Index* subtab, use the dropdown to switch to *Multiple* search mode.
3. Paste in the comma-separated list of IoC (Indicators of Compromise) IP addresses into the field.
4. Review the traffic direction, byte counts, and first/last seen details for each host and, if necessary:

- Click on the hostname/IP to view additional traffic and alarm information associated with the host.
- Run a report filtered on the host by clicking the data source and selecting a report from the tray.

Hint

If further investigation is required, continue to refine the report configuration as needed.

See also

To learn more about configuring and refining reports, see *this use case*.

Reviewing Alarm Monitor for suspicious hosts

The Scrutinizer Alarm Monitor provides users with real-time alerts to both performance issues and security threats and allows them to drill into event details by policy violation or by host.

Workflow

To inspect activity for suspicious hosts using the Alarm Monitor:

1. Navigate to **Monitor** > **Alarm Monitor** in the web interface.
2. Switch to the **Hosts** subtab and add a filter to show only *Critical* severity violations.
3. Use the dropdown to switch to the *Event Connections* view to look for hosts involved in multiple events.
4. Drill into events or run reports filtered on potential threats as needed.

See also

To learn more about configuring and refining reports, see *this use case*.

Investigating off-hour network activity

Scrutinizer's monitoring and reporting functions can isolate traffic outside business hours and alert teams to potentially malicious activity taking place during an organization's off-hours.

Workflow

To proactively hunt for threats that remain dormant during business hours, security teams can leverage the following report filter options:

- Add a filter that excludes business hours. A report threshold can also be configured, so that any activity exceeding the specified value(s) can be tracked via the Alarm Monitor.
- Define the period of time outside business hours as the report's time window/range.
- Set the report's time window to *Last 24 hours* and compare traffic data during and outside business hours.

Hint

After Scrutinizer has been deployed, default business hours can be set in the **Admin** > **Settings** > **Reporting** tray. These hours can be changed when configuring a business hours report filter.

Important

The Plixer ML Engine uses separate baseline models for network behavior during and outside of business hours. The default 8 am to 5 pm setting can be changed in the **Admin** > **Settings** > **Reporting** tray.

Identifying exfiltration outside business hours

Scrutinizer is able to isolate network activity outside of business hours, allowing teams to quickly identify data exfiltration attempts and other malicious activity taking place outside business hours.

Workflow

Data exfiltration can be identified proactively within Scrutinizer by identifying and reviewing traffic leaving your network. The **Explore > Exporters > By Interface View** is a great place to start, as traffic is displayed as inbound/outbound columns.

By default this is sorted so that your most congested interface is displayed at the top. This may be worth reviewing as large amounts of traffic leaving the network may be exfiltration.

Even more likely, exfiltration happens in a "low and slow" attack approach where only small amounts of traffic leave the network periodically – avoiding causing spikes in traffic that may cause alarms.

Because inspecting individual interfaces one at a time is inefficient, Scrutinizer reports can be used to narrow down the scope of information to be reviewed. This allows for a more streamlined approach to proactively searching for unwanted/suspicious traffic.

The following example uses the *Destination Countries with AS* report type:

1. Select **Reports > Run Report > Select Report Type** to start an adhoc report.
2. Choose **Destination Reports > Countries with AS**, add the appropriate device(s), and run the report.

The report is likely to show multiple rows of autonomous systems and the corresponding country they are associated with.

Note

Class A, B, and C addresses are always classified as *Uncategorized* and will often include internal network addresses. In this scenario, these are likely associated with responses to internal destinations through outbound interfaces.

3. Help narrow your search by excluding traffic that you expect to see. What remains may be of use in identifying traffic leaving the network to a destination that is unintended.

When you have a subset of data that is more manageable, e.g., countries your organization does not do business with, you can begin to pivot to other report types. Changing the time frame or "zooming out" can also reveal possible threats in the form of suspicious traffic patterns.

4. Within your report, with same filters, set the timeframe to *Last Seven Days*.

Is there a ping every hour beaconing out? Same packet size of data leaving the network following a pattern?

At this point, your report likely has one or more country, AS, or host filters. Switching to another report type or using extended report options like host reputation or geo IP lookups can lead to additional insights.

Tip

Run a report against a core router that is likely to see a majority of your traffic. Alternatively, select **All Devices** to identify top network conversations across the entire network.

4.3.2.13 Lateral movement detection

Because indications of a cyber attack are not limited to traffic originating from external hosts, security teams require tools that can monitor internal network activity for potential threats, such as lateral movement.

Plixer One Enterprise employs multiple detection techniques to alert to behavior that may indicate lateral movement through their network by malicious actors.

Overview

Through Scrutinizer, Plixer One Enterprise combines deep network observability with multiple approaches to lateral movement detection to deliver meaningful alerts that enhance both proactive and reactive workflows.

As it continuously monitors and collects flow data from its environment, Scrutinizer uses the Alarm Monitor view to alert users to activity that matches potentially problematic or malicious patterns, including those associated with lateral movement techniques. The Alarm Monitor, *Network Maps* and *Dashboards* views allow users to pivot to *reports* and launch deeper investigations into typical indicators of lateral movement.

Hint

The Monitor > Alarm Monitor > ATT&CK tab classifies alarms using the MITRE ATT&CK framework and can be used to quickly filter for alerts related to lateral movement.

The following alarm policies are used to provide alerts specifically for potential lateral movement and based on different detection approaches/criteria:

Lateral Movement

Lateral Movement alarms are flow analytics detections that are triggered by traffic/activity that is indicative of techniques used to exploit remote services. Events under this alarm policy report the following details for the detection:

- Exporters/devices
- Violating hosts
- Target hosts

Lateral Movement Attempt

Lateral Movement Attempt alarms are flow analytics detections that are triggered by traffic/activity that is indicative of a worm attack on a specific port on a target host. Events under this alarm policy report the following details for the detection:

- Type of worm
- Destination/target port
- Violating hosts
- Target hosts

Lateral Movement Behavior

Lateral Movement Behavior alarms are machine learning detections that are triggered when the behavior of a *monitored host* deviates from baseline activity patterns in a way that is indicative of lateral movement. Events under this alarm policy report hosts that are communicating with an unusually large number of machines (based on behavior learned by the ML Engine) as violators.

Note

- The threshold at which irregular traffic/behavior associated with a host is reported as a detection can be adjusted by changing the sensitivity for the ML inclusion/source it belongs to.
- Because the *Lateral Movement* FA algorithm references existing lateral movement attempts for its detections, its scope can be customized by specifying traffic coverage (*external to internal*, *internal to external*, or *internal to internal*) for the *Lateral Movement Attempt* algorithm. E.g., if internal-to-internal traffic is disabled for the *Lateral Movement Attempt* algorithm, there will be no detections for internal-to-internal traffic under the *Lateral Movement* algorithm.

Workflows

The following workflows show how lateral movement detections in Scrutinizer can be used to investigate and respond to potential threats:

Investigating lateral movement alerts

Scrutinizer uses multiple lateral movement detection techniques, each of which corresponds to a separate alarm policy. This provides security teams with additional context on which to base their response strategies.

Workflow

After receiving a lateral movement alert either in Scrutinizer itself or via external SIEM, investigate the event:

1. Navigate to **Monitor** > **Alarm Monitor** in the web interface and search for *Lateral Movement* (FA), *Lateral Movement Attempt* (FA), or *Lateral Movement Behavior* (ML) violations.
2. Click on an alarm policy to open the summary view and review the activity timeline and hosts involved.
3. Drill into an event artifact to view a summary of details for a violation associated with a specific host.
4. To further investigate the activity of the host, click on the icon next to its IP address or hostname, and select an automatically filtered report to run.

> **Hint**
>
> For additional context and/or details related to how and why the host was compromised, review all alarms leading up to the lateral movement violation.

Uncovering data exfiltration

While proactively reviewing outbound traffic, the security team discovers activity that indicates a potential attempt to exfiltrate data.

Workflow

After discovering unusually high outbound utilization in the **Explore** > **Exporters** > **By Interface** view, run a report to redefine the scope of traffic that needs to be reviewed (e.g., *Destination Countries with AS*):

1. Run a new report for the exporters/devices exhibiting suspicious behavior, and select *Countries with AS* (under the *Destination Reports* category) as the report type. This will output a list of autonomous systems, along with the countries each one is associated with.

> **Note**
>
> Class A, B, and C addresses are always classified as *Uncategorized* and will often include internal network addresses. In this scenario, these are likely associated with responses to internal destinations through outbound interfaces.

2. Limit the scope of the report by dragging rows associated with expected traffic to the *Exclude* drop zone to the left and clicking **Apply** in the *Filters* tray.
3. After the report has been re-run with the additional exclusions, review the list for traffic bound for unusual destinations.
4. Once a more manageable subset of data (e.g., countries your organization does not transact with) has been achieved, refine the report to gain more insight:

- "Zoom out" to look for activity patterns by changing the time frame covered by the report.
- Inspect activity associated with the host, country, or autonomous system by clicking on it and pivoting to a different report type from the tray.
- Leverage additional tools (under the *Other Options* category in the tray) to obtain additional information.

For further investigation, continue to modify the settings of the report to gain visibility into hosts, traffic, etc. that remain suspicious.

4.3.2.14 Incident response

Plixer One Enterprise combines Scrutinizer's deep, environment-wide visibility and intuitive UI-driven workflows with advanced detection techniques for security events to enhance a team's ability to respond to threats.

Overview

Scrutinizer's "single-pane-of-glass" feature set is designed around providing maximum network observability via synergistic web interface functions and views that streamline monitoring and investigative activities.

Full visibility supporting incident response and other security processes

As part of an incident response plan, Scrutinizer ensures that SecOps teams have access to all the traffic and device information they need for investigation and remediation:

- Get comprehensive, contextualized details for intrusion detection system (IDS) and intrustion prevention system (IPS) events
- Access full network traffic forensics to watch for and investigate security information management (SIM) events
- View full IP to MAC address mapping history for all connected devices and endpoints
- See real-time and historical endpoint context and location
- Assess endpoint risk through layer 2 historical location tracing
- Glean additional insights from detection details via MITRE ATT&CK, *STIX/TAXII*, and *other integrations*

Web interface functions that promote more efficient response strategies and procedures

Scrutinizer enables more efficient general security and incident response workflows through multiple functions/features, including:

- Highly configurable UI views (alarm monitor, *dashboards*, network maps, etc.)
- Customizable data aggregation from any observation point(s) on the network
- Detections and alerts driven by by *AI/ML* and *Flow Analytics*
- Customizable notification options for alarm/event details
- Deep visibility for both on-prem devices and *assets in the cloud*
- *Collaborative features* that promote sharing investigation results/insights between members and/or teams

Workflows

The following workflows show how the additional visibility and workflow enhancements enabled by Scrutinizer can be leveraged by SecOps teams for monitoring and incident response:

Responding to Alarm Monitor security alerts

Scrutinizer leverages a range of technologies to alert users to anomalous and potentially malicious network activity through its library of alarm policies. Once policy violations are reported via the Alarm Monitor views, security teams can drill into individual event details to evaluate whether further investigation is necessary.

Workflow

To investigate an alarm policy (e.g., *Data Exfiltration*, *Data Accumulation*, etc.) violation reported in the Alarm Monitor:

1. Click on the alarm policy to open the summary view.
2. Review the activity timeline and hosts involved.
3. If further investigation is warranted, drill into individual event artifacts for more details.
4. Click the icon next to an IP address or hostname to run an automatically filtered report and examine additional activity/hosts associated with the event.

> **Hint**
>
> For additional context and/or details related to how and why the host was compromised, review all alarms leading up to the policy violation.

Scrutinizing an infected host

After a user is infected with a virus, the security team must identify what other hosts on the network may have communicated with the infected host.

Workflow

After the infected host is discovered/reported, the following steps can be used to identify other hosts it has interacted with:

> **Note**
>
> This workflow relies on usernames acquired from a network device (router, firewall, etc.) or through enabled integrations (e.g., Active Directory LDAP). If usernames are not available, host IP addresses can be used as identifiers instead.

1. Under **Explore** > **Exporters** > **Entities** > **Usernames**, search for the infected host/username and click on it. A new view will open.
2. Review the alarms/events associated with the host, which may include the following violations:

- *P2P* and *Lateral Movement* (infected host may be attempting to extend access further into the network)
- *TCP*, *UCP*, *XMAS Port Scan* (infected host may be pinging the network for reconnaissance)

3. Create/run a report with the username applied as a filter to identify all activity where the infected host was either the source or the destination of traffic. Ensure that the time range includes a period before the infection was reported or discovered.

> **Hint**

When viewing information associated with a username, click the graph icon to run a report with the username applied as a filter. The filter will be retained even when pivoting to other report types.

4. Review the output or pivot to different report types for insight related to who, what, when, where, why, and how the infected host communicated on the network:

- Protocols the host was seen using
- Countries the host communicated with
- Firewall events (through vendor-specific report types, e.g., ACL rules, NAT translations, etc.)
- Destination FQDN reports
- Activity associated with the host before and after the infection (for additional insight into the techniques used in the initial attack)

5. If the Host Indexing FA algorithm is enabled, navigate to **Explore** > **Search** to look up historical data associated with the IP address of the infected host. This information may provide additional insight based on typical communication patterns and reduce mean time to know (MTTK) during the investigation.

Note

If the *Use Host Index* option under **Admin** > **Settings** > **Reporting** is enabled, *Group* and *All Device* reports will use the host index to limit the scope of exporters checked when a host filter is applied.

4.4 Features and Functionality

4.4.1 Scrutinizer

Monitor

Monitor and investigate alarm policy violations and network activity

Monitor **Explore**

Look up any host and inspect exporters and entities

ui-explore **Investigate**

Collaborate in investigations and generate forecasts from reports

ui-investigate **Reports**

Monitor and investigate network activity using custom reports

ui-reports **Plixer ML Engine**

Enable ML-driven monitoring and detection

Machine learning **Admin**

Manage system settings, users, integrations and more

ui-admin

4.4.2 Plixer One

Endpoint Analytics

Enable enhanced endpoint monitoring

Endpoint Analytics **FlowPro**

Expand visibility and enable advanced analytics

FlowPro **Replicator**

Replicate and distribute packet streams

Replicator

4.4.2.1 Scrutinizer

Monitor

Monitor and investigate alarm policy violations and network activity

Monitor **Explore**

Look up any host and inspect exporters and entities

ui-explore **Investigate**

Collaborate in investigations and generate forecasts from reports

ui-investigate **Reports**

Monitor and investigate network activity using custom reports

ui-reports **Admin**

Manage system settings, users, integrations and more

ui-admin The Scrutinizer web interface is accessed by pointing any supported browser to `https://SCRUTINIZER_ADDRESS/ui/`, after the *server has been deployed and set up*. To use the Classic UI, use the URL `https://SCRUTINIZER_ADDRESS/oldui/` instead.

The preferred UI can also be set from within the web interface under the user menu.

UI overview

Show content

The Scrutinizer web interface enhances NetOps and SecOps workflows through a comprehensive feature set that transforms raw flow data into fully contextualized network intelligence.

The web interface pages/views are divided into four general categories that correspond to the most essential NetOps and SecOps workflows.

Hint

- Scrutinizer users can toggle between the persistent header tabs and the collapsible sidebar for navigation using the *Slim Navigation* option in the **Admin** > **Users & Groups** > **User Accounts** > **Preferences** tray.
- Click the **Help** (?) button in the header of any page to access the Scrutinizer online documentation at any time.

Tab/s	Description
Monitor	- Use customizable alarm policies to receive alerts when problematic or dangerous behavior is discovered on the network- Create custom dashboards using ready-to-use gadgets that display vital activity summaries and visualizations- Visualize and monitor activity between connected devices with user-defined network maps
Explore	- Drill down into flow-generating devices to examine activity, resource usage, and events generated- Inspect behavior, interactions, and events generated by individual entities- Look up specific host and host pairs in the system's host index to inspect details or verify if the host(s) has been seen on the network and investigate activity linked to it
Investigate	- Define collections of one or more alarms, events, and/or reports and assign them to analysts for investigation- View available forecasts to identify resource usage trends and identify future needs
Reports	- Create/run custom or preconfigured network activity reports that can be saved and used to generate ML-based forecasts- View/re-run and manage saved reports

If the *local Replicator instance* is enabled, an additional *Replicator tab/page* can also be accessed to monitor and manage flow replication parameters.

The functions and workflows under each UI tab are explained in further detail in the succeeding sections of this documentation.

Data aggregation method

Show content

Scrutinizer's *SAF* (Summary and Forensic) data aggregation method is an optimized system of storing flow data that makes use of summary tables to condense collected information without compromising transparency or accuracy.

How SAF works

With SAF, any incoming flow template with the required data elements is aggregated into a new template definition based on a tuple that includes `commonPort`. The resulting "summarized" template will omit all data elements that prevent aggregation (e.g., source and destination transport ports) but still contain all information required for the vast majority of reporting needs.

Hint

The aggregation logic used to create summary tables can be modified to suit different scenarios. Contact *Plixer Technical Support* for assistance.

The data elements retained in the summary tables are but not limited to:

- `intervalTime`
- `commonPort`
- `ingressInterface`
- `egressInterface`
- `sourceIpAddress`
- `destinationIpAddress`
- `octetDeltaCount`

- `octetDeltaCount_rev`
- `packetDeltaCount`
- `packetDeltaCount_rev`
- `flowDirection`
- `applicationId`
- `protocolIdentifier`

Once five 1m summary tables are available, the data averages for the top 1000 (default) conversations are rolled up into 5m tables, and the system continues the rollups to create 30m, 2h, and 12h tables.

ℹ **Note**

- If a collector's disk capacity will support it, the Flow Maximum Conversations value under **Admin** > **Settings** > **Data History** can be increased, which may improve reporting accuracy. Since this results in larger tables and certain report types taking longer to run, the value should be increased in increments until an ideal balance is achieved.
- When *Auto History Trimming* (under **Data History** settings) is enabled, 1m and 5m historical tables are trimmed to maintain the configured *Minimum Percent Free Disk Space before Trimming* value. Automatic trimming is also used to retain a similar level of historical data for all configured exporters.

Benefits of SAF aggregation

Because the summary tables created under SAF aggregation are drastically smaller in size than regular full-template tables, they benefit the Scrutinizer system in the following ways:

- Reduced disk utilization per table
- Increased historical data capacity
- Improved report render times
- Faster lookups before drilling into forensic data

While only summary data is rolled up into higher interval tables, Scrutinizer still retains the original forensic data, which is used by a handful of reports that require data elements not included in the summary tables. At the same time, the system also maintains a separate totals table for in/out byte counts per interface to allow for accurate utilization reporting without relying on SNMP.

ℹ **Note**

Systems that have been upgraded from versions prior to 18.x may still use the legacy data aggregation method that was the default in their original installs. To check, navigate to **Admin** > **Settings** > **Data History** and if the *Rollup Type* is not set to **Summary and Forensic**, contact *Plixer Technical Support* for assistance with switching.

Notes on collecting sFlow

When collecting sFlow, packet samples and interface counters should both be forwarded to the collector. Packet samples will be saved to the raw tables, and interface counters will be saved to the totals tables at one-minute intervals.

Important

Having an sFlow exporter (e.g., switch) that sends multiple templates for different flows may result in overreporting, if the flows contain the same or very similar information. Scrutinizer's frontend will run reports using data from all templates that match the information. To avoid this, use filters to specify a single template.

Monitor

The **Monitor** views of the Plixer Scrutinizer web interface provide comprehensive network visibility and real-time monitoring capabilities.

This section enables users to visualize network activity, track performance metrics, identify security threats, and investigate traffic patterns through customizable dashboards, interactive maps, and alarm management tools.

Alarms

Get alerted to alarm policy violations and drill into details by policy, host, or MITRE ATT&CK category

monitor-alarms **Dashboards**

Use custom dashboards to enhance workflows and support diverse roles

Dashboards **Network maps**

Visualize network topology and activity in customizable maps

monitor-maps **Topology**

View discovered topology and run reports for selected hosts or interfaces

Topology (BETA)

Alarm Monitor

The Scrutinizer **Alarm Monitor** subsection/page is Scrutinizer's main interface for monitoring and investigating active alarm policy violations. The page is divided into three subtabs, which allow for different starting points when investigating events.

Policies alarms-policies Hosts alarms-hosts ATT&CK alarms-attack For additional background and recommended configuration steps related to Alarm Monitor functions, see the *configuration guide for alarms and events*.

Policies

The **Monitor** > **Policies** tab/view is the default Alarm Monitor view and can be used to investigate alarms within the specified time period based on the alarm policy violated.

The overview table can be set to include any of the following columns via the **Available Columns** button:

- **Severity**: Distribution of individual events under the policy based on severity
- **Risk**: Aggregated risk level
- **Events**: Total number of violating events under the policy
- **Violators**: Total number of hosts observed as violators under the policy
- **Targets**: Total number of hosts observed as targets under the policy
- **First Observed**: Timestamp of the first violating event within the specified time period
- **Last Observed**: Timestamp of the most recent violating event within the specified time period

- **Category**: Policy category
- **Technology**: Plixer One component where the alarm originated

The host counts in the **Violators** and **Targets** columns also function as shortcuts to pivot to the Hosts view with a filter for the policy applied.

Note

- Risk information requires *Endpoint Analytics*.
- For a full list of alarm policy categories and violation descriptions, see *this table*.

Editing policy settings

To edit the *settings* of the policy for an active alarm, select **Edit Policy** from the three-dot menu in the list/table.

This will open the settings tray in the alarm policy management view, where the policy's weight, timeout, and state can be modified. *Notification profiles* can also be created and assigned to the policy from this tray.

Inspecting hosts

Clicking the ⍰] icon in the *Violators* or *Targets* column of the table opens a tray listing violating and targeted hosts involved in the alarm. This tray can be used to select one or more hosts to apply as filters or view alarm details for any of the hosts involved.

Alternatively, clicking on the host count in the *Violators* or *Targets* column opens the Alarm Monitor Hosts tab with a filter for the policy applied.

The tray also includes toggles to hide/show system policy violations and acknowledged events in the active alarm list.

Managing exclusions

To add or remove exclusions for an active alarm policy, select **Manage Exclusions** from the three-dot menu in the list/table.

For *Scrutinizer* alarm policies (indicated in the *Technology* column), this will open the FA algorithm management view, from where exclusions can be added to or removed from the algorithm driving the policy. For *Plixer Machine Learning* policies, the option will open the FA algorithm management view instead.

Individual hosts can also be added to FA algorithm or ML detection exclusion lists by opening the violators/targets tray and clicking the ⍰ icon for one or more hosts.

Alarm summary

Clicking on a policy in the main list opens the summary/details view for the alarm, which includes a chart/timeline summarizing observation details and a list of artifacts for separate events/violations under the same policy.

The following visualizations can be selected from the *View* dropdown:

- **Events Scatter Plot** - Shows distribution of the events and observations
- **Events Timeline** (default) - Shows the individual events and their durations in a timeline for the specified time period
- **Entities** - Shows observation distribution among top violators, IP groups, and targets

Note

Scrutinizer aggregates continuous or consecutive observations within the policy's *Timeout* setting as a single event. See *this page* on the alarm/event life cycle for further details.

Event list

The event list of the alarm summary view can be used to drill into the artifacts for discrete events/violations within the specified time period. The summary table lists total number of observations aggregated as well as the basic details (severity, hosts, etc.) for each event.

Hint

Mouse over the graph icon in the event list for additional shortcuts/options (varies by policy).

Click on an artifact to open a tray containing the *full details* for the event:

- Severity
- Start/end timestamps
- Most recent event message generated
- All hosts observed as targets
- All hosts observed as violators
- All events with matching violating criteria

In the tray, clicking on the link icon for target or violator opens the host details view, where the details for all alarms associated with the host can be investigated. Details for other events with the same violating criteria (based on the alarm policy) can also be viewed in a secondary tray by clicking the view (eye) icon.

Auto-Investigate policy

The *Auto-Investigate* alarm policy reports sequential incident/event chains wherein each targeted host becomes the next violator in the sequence. Each chain includes all discrete events starting from the initial incident and ends when the target cannot be confirmed as the next violator.

When an Auto-Investigate alarm is active, its summary view will list all incident chains (aggregated by the initial violating host) instead of individual events.

Investigation details

Clicking the microscope icon in the list/table opens the investigation subview for the selected initial violator, which can be used to inspect the following information for all incident chains linking back to it:

- All incident chains with the same initial violator, including violators, targets, and exact timelines
- Visualized links between violators, policies, and targets
- Event distribution over time
- Event, target, and violator counts for all policies violated
- Number of policy violations, linked event violator counts (including itself), and roles for all hosts

The policy and host lists also link back to their respective Alarm Monitor views for further investigation and cross-referencing.

Hosts

The **Monitor > Policies** tab can be used to investigate alarms within the specified time period based on a target or violating host.

The overview table can be set to include any of the following columns via the **Available Columns** button:

- **Severity**: Distribution of individual events under the policy based on severity
- **Behavior**: Host behavior information (Click the icon to view behavior summary or drill into the host behavior subview.)
- **Risk**: Endpoint risk level (Click the icon to view endpoint details.)
- **Country/Group**: IP group or country associated with the host
- **As Target**: Total number of events with the host as a target
- **As Violator**: Total number of events with the host as a violator
- **Policies**: Total number of policy violations involving the host as a target or violator
- **First Observed**: Timestamp of the first violating event involving the host within the specified time period
- **Last Observed**: Timestamp of the most recent violating event involving the host within the specified time period

The three-dot icon/menu can be used to access the host information summary tray or pivot to any report supported by the host.

Note

- Behavior information requires a Plixer One Enterprise license.
- Risk information requires *Endpoint Analytics* integration to be enabled.
- The **Country/Group** column will display IP groups for internal hosts and countries for external addresses. Addresses can be designated as internal or external as part of IP group definitions.

Host details

Clicking on a hostname/address in the main list opens the host details page, which includes an overview pane and three (four if the host is an exporter) subviews with detailed insights related to the host's activity.

Note

If *Endpoint Analytics* integration is enabled, the overview pane will include a section with additional endpoint information and a link to the corresponding Endpoint Analytics view.

Traffic

The host traffic subview can be used to inspect a host's activity based on its communications with other hosts and/or IP groups.

This subview visualizes activity data for the host using the following charts:

- An activity timeline showing the inbound (green) and outbound (blue) rates over the specified time period in an activity timeline
- A traffic distribution chart of source IP groups where this host is the destination

- A traffic distribution chart representing the host's activity by defined application used
- A traffic distribution chart of destination IP groups where this host is the source

Each chart also includes a shortcut button to run a filtered report to break down the host's activity in greater detail.

Behavior

The host behavior subview can be used to investigate a host that has been observed by the Plixer ML Engine to be exhibiting anomalous behavior.

Host behavior insights for the selected ML dimension are summarized in the following:

- A timeline showing the deviation criteria (e.g., bytes, IP address count, etc.), magnitude (based on the host's typical activity patterns), and threshold for the selected dimension
- A table/list of timestamps and details for individual behavior deviations

To see behavior information for a different feature dimension, use the dropdown and select another dimension with an anomalous behavior count.

Further investigation is recommended for hosts with deviation magnitudes exceeding the indicated threshold.

Note

- Behavior data will only be available for hosts that are covered by the ML Engine's inclusion rules and have exhibited anomalous behavior.
- Behavior modeling and other ML Engine functions require a Plixer One Enterprise license. Contact *Plixer Technical Support* to learn more.

Alarms

The host alarms subview can be used to investigate alarms in which the host was involved as a target and/or violator.

This subview includes two overviews of all unacknowledged alarms associated with the host:

- A timeline showing individual *events* by alarm policy violated
- A summary table (similar to the main Alarm Monitor policies view) with details for all policies with violations involving the host

Drilling in from the summary table opens the alarm details view for the policy, where event artifacts can be inspected individually.

Interfaces

The host interfaces subview consists of a table listing all interfaces on a flow-exporting device along with their inbound and outbound activity details.

Note

Inbound and outbound activity details use rates by default. If custom interface speed has been assigned to an interface, utilization will be used instead.

To show highwater activity (inbound or outbound) details for an interface, hover over the corresponding information (**i**) icon in the table. Shortcuts to run reports or drill into interface traffic/behavior can be accessed from the three-dot menu.

Additional options

To support workflow efficiency, the host details page header includes buttons to access the following functions:

- Changing the time period/range covered
- Pivoting to any supported report type filtered on the current host
- Viewing additional details and information from integrated sources (**Learn more** button)
- Applying filters (alarms and interfaces subviews only)

ATT&CK[1]

The **Monitor** > **ATT&CK** tab can be used to investigate events based on the tactic, technique, and sub-technique assigned by the MITRE ATT&CK framework.

Events are plotted in a timeline, where the user is able to drill into them individually to open a tray containing the following:

- MITRE ATT&CK tactic and technique information, with links to the relevant MITRE ATT&CK knowledge base articles
- Shortcuts to the **Policies** or **Hosts** Alarm Monitor tab with filters for the event's details applied
- Basic event information

The page also includes the MITRE ATT&CK Enterprise Matrix, with technique classifications highlighted to match the corresponding events in the timeline.

Hint

Click on a technique cell in the matrix to view the policies violated in the **Policies** tab.

Applying filters

To further facilitate monitoring and investigation, the Scrutinizer Alarm Monitor views support multiple approaches to applying filters to the Alarm Monitor views.

Time range filter

The Alarm Monitor views can be set to show alarm/event information for either a custom date and time range or a specified *Last X* period (last 15 minutes, last 24 hours, last week, etc.).

To view data for a different period, click the **Time Range** (calendar) button and configure the range to apply.

Hint

When a custom range is specified, click the up/down arrows to automatically adjust the dates to cover the same period of time.

Card/chart filters

By default, the **Policies** and **Hosts** tabs use sparkline cards to summarize severity distribution across policies or hosts. These cards can be clicked to apply a filter for policy violations or hosts matching the selected severity.

Other visualization types (timelines and connection diagrams) showing different event details (events, alarm policy category, etc.), can be selected from the **View** dropdown and used to quickly apply the corresponding filter.

[1] © 2022 The MITRE Corporation. This work is reproduced and distributed with the permission of The MITRE Corporation.

Advanced filters

Clicking the **Filters** button opens a tray where one or more filters can be manually configured.

The following filtering options are available:

- Policy
- Severity
- Risk
- Hosts
- Violators
- Targets
- Category (of alarm policy)

To apply a filter, expand the filter option/section, and select the criteria to use. Multiple options and criteria can be applied at the same time.

> **Note**
>
> - The *Risk* filter is only available when the Endpoint Analytics integration is enabled. To learn more about Endpoint Analytics integration in Scrutinizer, see *this section* of this documentation.
> - The filter options tray also includes an option to show policies and hosts associated with events that have already been acknowledged.
> - When exporting alarm/event data (via the **Options** button/tray), use the *Export CSV (All)* option to ignore any filters currently applied.

Acknowledging events

Once an event has been investigated and/or resolved, it should be acknowledged to clear it from all Alarm Monitor views. This reduces the volume of active alarms and/or events at any given time and can further streamline investigative processes.

Acknowledging events is part of Scrutinizer's recommended investigation and resolution workflow.

> **Hint**
>
> To show/hide acknowledged events in the Alarm Monitor views, open the filter options tray and toggle the **Show Acknowledged Events** option on/off.

Acknowledging can be done by alarm policy or by event.

Acknowledging by policy

From the main view of the **Policies** tab, acknowledging an alarm policy automatically flags all events generated under the policy as acknowledged.

To acknowledge by alarm policy:

1. While on the **Policies** tab of the Alarm Monitor view, select the policy by ticking its checkbox.
2. If acknowledging more than one policy, verify that the correct policies have been selected.
3. Click **Acknowledge Selected Events**.

> **Note**
>
> The **Acknowledge Selected Events** button is only available when at least one policy checkbox is ticked.

Once acknowledged, the alarm policy and all events associated with it will be hidden from all Alarm Monitor views.

Acknowledging by event

Acknowledging can also be used to clear only events that match the same criteria. This allows other events under the same policy (as well as the alarm policy itself) to be retained in Alarm Monitor views.

Events are acknowledged from the summary view of the **Policies** tab as follows:

1. Scroll down to the **Event List** section of the page.
2. Select the artifact linked to the criteria/events to be acknowledged by ticking its checkbox.
3. If selecting more than one artifact, verify that the correct checkboxes have been ticked.
4. Click **Acknowledge Selected Events**.

> **Note**
>
> The **Acknowledge Selected Events** button is only available when at least one policy checkbox is ticked.

Once acknowledged, the event(s) will be hidden from all Alarm Monitor views.

Dashboards

Scrutinizer further enhances diverse network and security workflows through user-configurable dashboards, which can be configured and accessed via the **Monitor** > **Dashboards** page of the web interface.

Teams can set up and save any number of fully customized dashboards, allowing for the use of purpose-built views to address even the most unique monitoring or investigative requirements.

This section discusses the features and functions accessed via the **Monitor** > **Dashboards** tab/section of the web interface, and includes detailed guides for the creation, customization, and management of dashboards.

On this page:

Creating dashboards *Creating a new dashboard* Viewing dashboards *Viewing dashboards* Editing dashboards *Editing/customizing dashboards* Dashboard management *Dashboard management* Dashboard gadgets *Dashboard gadgets*

Creating a new dashboard

Creating a dashboard in the **Monitor** > **Dashboards** page allows users to enhance various network and security workflows by enabling tailored views to meet even the most specific monitoring or investigative needs. It also allows users to switch between different unique views to organize monitoring requirements and workflows. Creating at least one dashboard is highly recommended when *setting up a new Scrutinizer environment*.

To create a new dashboard, follow these steps:

View instructions

1. Click the **Dashboard Options** (gear) icon, and then click **Add New Dashboard**.
2. Enter a unique name for the dashboard.
3. [Optional] Tick the *Default Dashboard* checkbox to set the dashboard as the default for the current user.
4. [Optional] Tick the *Read Only* checkbox to lock the dashboad configuration.
5. [Optional] Select one or more *gadgets* to add to the new dashboard in the *Gadget Selection* list.
6. When done, click **Save** to create the new dashboard with the selected gadgets, if any.

Once the dashboard has been created, it will replace the current view and can be accessed via **Dashboard Options > All Dashboards** at any time.

Hint

- Gadgets can be *added to or removed from a dashboard* at any time.
- An existing dashboard can be used as a base for a new dashboard by *creating and editing a copy*.

Welcome dashboard

New Plixer One or Scrutinizer deployments will have a default welcome dashboard to assist with the setup/configuration process.

View details

The welcome dashboard will be populated with the following *gadgets*:

- *Configuration Checklist* - Displays the completion percentage of the configuration checklist for new deployments
- *Quick Start* - Contains links to documentation for dashboard functions
- *Enabled Exporters* - Shows the current utilization of the active license's maximum exporter count
- *Alarm Monitor* - Displays a graphic summary of alarms generated within the last 24 hours, categorized by severity
- *Contact Us* - Provides Plixer's contact information for additional support.

Viewing dashboards

The **Monitor > Dashboards** view will always default to the current user's default dashboard. Clicking the **Dashboard List** (gauge) button opens a tray where other existing dashboards can be accessed.

Hint

- A user's default dashboard can be set in the *Edit Dashboard* tray or as a user preference via the Admin > Users & Groups > User Accounts view.
- Other dashboards (all, current user, or user group) can also be accessed from the **Dashboard Options** tray.

Full-screen mode

The **Full Screen** option expands the dashboard to occupy the entire screen, maximizing the space available for viewing data visualizations and insights. This feature can be ideal for large displays in NOCs.

While in full-screen mode, the bottom toolbar includes additional options to refresh all gadgets at once or edit the gadget layout.

Editing/customizing dashboards

Existing dashboards can be modified or further customized by clicking the **Edit Dashboard** icon (pencil icon). A dashboard cannot be modified if it has been set to read-only. To edit an existing dashboard but retain a copy of its current state, *create a copy* of the dashboard before making changes.

Adding/removing gadgets

To add a predefined or existing *custom gadget* to a dashboard, follow these steps:

View instructions

1. Click the **Edit Dashboard** (pencil) icon.
2. Under *Gadgets*, click the **Add Existing Gadget** (**+**) button.
3. Select a *gadget category* (or use the search field):
4. Select the gadgets to add using the checkboxes.

Gadgets are added to the dashboard as they are selected. To remove gadgets, click on the *Delete* (trash icon) in the gadget list in the **Edit Dashboard** tray.

> **Hint**
>
> Click *Modify/Rename* in the **Edit Dashboard** tray to edit the current dashboard's name. Read-only mode (locks the dashboard's current configuration) can also be toggled on or off from the same tray.

Gadget layout

After gadgets have been added to a dashboard, clicking the **Edit Gadget Layout** button enables a layout-edit mode where gadgets can be resized and repositioned as desired.

To save the new layout and toggle the edit mode off, click the button a second time.

Dashboard management

The **Dashboard Options** (gear button) tray provides access to the following dashboard management functions:

Deleting dashboards

To delete an existing dashboard, select *All Dashboards* in the tray, and then click the corresponding delete icon in the secondary tray. Dashboards can also be deleted from the *Dashboard Edit* tray by selecting *Remove This Dashboard* in the *Options* section.

Users that do not have the Dashboard Admin permission are unable to delete dashboards created by other users. Dashboards that are set as default or read-only cannot be deleted.

Copying dashboards

Selecting *Copy This Dashboard* creates a copy of the current dashboard with a specified name. This option can be used to create a modified dashboard configuration (add/remove gadgets, change layout, etc.) without overwriting the source dashboard's current state.

Copies can also be created from *Dashboard Edit* tray.

Deleting gadgets

To delete an existing gadget, select *All Gadgets*, and then click the corresponding delete icon in the secondary tray.

The search field and gadget category dropdown can also be used to filter the list.

Managing user and user group access

Users with the Dashboard Admin permission can grant or restrict access to dashboards by user or by user group.

To grant a user or user group access to one or more existing dashboards:

1. Click *User Dashboards*/*User Group Dashboards*.
2. In the secondary tray, select the user/user group from the dropdown.
3. Use the checkboxes to select the dashboards the user or group can access.
4. Click the **Save** button to save any changes made.

Once granted access to a dashboard, the user/group member will be able to view and copy any dashboard. However, they will not be able to edit or delete dashboards created by other users.

Note

- To fully access a dashboard, a user must also be granted access to the dashboard's gadgets through their user group.
- A user can only manage dashboard access for other members of their user group(s). Additionally, they are only able to grant access to dashboards that have been created by members of their user group(s).

Dashboard gadgets

Scrutinizer dashboards can be tailored to any workflow or role using any combination of gadget types and configurations.

Once a gadget has been added to a dashboard, it will automatically refresh and can be clicked to access more detailed views.

Hint

A refresh progress bar and countdown is displayed in the header of each gadget. Clicking the refresh button will force an update and reset the refresh timer.

A gadget's default refresh time (5 minutes) can be changed by *editing the gadget's properties*.

Gadget categories

When *adding gadgets to a dashboard*, available gadgets are divided into the following categories:

- *Custom* - User-defined *iFrame or interface gadgets*
- *Flow Analytics* - Gadgets for monitoring alarm information
- *Flow Reports* - Reports that have been saved and exported for use as gadgets
- *Maps* - Gadget versions of existing network maps
- *Plixer* - Auxiliary gadgets for monitoring system functions/features

- *Top n* - Gadgets for monitoring top entities associated with specified metrics
- *Vendor Reports* - Gadgets that report data from other Plixer One platform components or third-party integrations
- *Vitals* - Gadgets for monitoring system vitals

Flow report gadgets

After a report has been saved, it can be exported as a gadget and used to monitor the configured network performance or behavior from a dashboard.

Report gadgets can be configured to show only the graph, only the table, or both. The type of visualization used can also be specified as part of the *gadget configuration*.

See also

See this page to learn more about configuring and managing reports.

Map gadgets

Network maps automatically become available as gadgets upon creation, allowing users to monitor one or more mapped topographies from the same dashboard.

If the source map configuration is updated at a later time, the changes will be reflected in the gadget when it is next refreshed.

See also

See this page to learn more about creating and customizing network maps.

Custom gadgets

Scrutinizer dashboards support two types of custom gadgets:

Interface gadgets

Interface gadgets visualize activity details for one or more specified exporters and/or interfaces.

To create a new interface gadget:

1. Enter dashboard edit mode, and then click the **Create New Gadget** button in the tray.
2. Select *Interfaces* as the type in the secondary tray.
3. Configure the display options (auto-refresh interval, activity metric, etc.) for the gadget.
4. Select the exporters and/or interfaces to monitor in the gadget.
5. Click **Save To Dashboard**.

Once saved, the gadget will be added to the current dashboard. It will also be made available when adding gadgets to other dashboards.

iFrame gadgets

iFrame gadgets can be used to display another webpage (including external pages) in a dashboard.

To create a new iFrame gadget:

1. Enable external links(see below) in Scrutinizer if not already enabled (required).
2. Enter dashboard edit mode, and then click the **Create New Gadget** button in the tray.
3. Select *iFrame* as the type in the secondary tray.
4. Enter the URL for the page to display (must include the `http(s)://` prefix).
5. Click **Save To Dashboard**.

Once saved, the gadget will be added to the current dashboard. It will also be made available when adding gadgets to other dashboards.

Note

Certain HTTP content may not load if Scrutinizer is configured to use HTTPS.

Enabling external links

To be able to create and use custom iFrame gadgets, external URLs must be allowed on the primary Scrutinizer reporter.

Note

The NOAA Weather gadget (included in the *Plixer* gadget category) also requires external URLs to be enabled.

External URLs are disabled by default and can be enabled as follows:

View instructions

1. On the primary reporter, comment out the following line in the `location /ui` and `location /oldui` blocks in `/etc/nginx/webapp.d/conf/plixer-nginx-server.conf`:

   ```
   add_header Content-Security-Policy "frame-src 'self';"
   ```

   Important

   The line **must** be commented out in both blocks.

2. Restart the relevant services:

   ```
   sudo service nginx restart
   sudo service memcached restart
   ```

Once the services have been restarted, the option to create custom iFrame gadgets will be available in the dashboard settings tray.

Editing a gadget

After a gadget has been added to a dashboard, its properties can be updated by entering the dashboard edit mode and clicking the edit/pencil icon in the *Gadgets* section of the tray.

Once the desired changes have been made, click **Save** to save the new gadget configuration.

Managing access to gadgets

Access to gadgets can be managed as part of user group permissions as follows:

View instructions

1. Navigate to **Admin** > **Users & Groups** > **User Groups**.
2. Click on the name of the user group to modify.
3. In the tray, click the edit/pencil icon for *Dashboard Gadgets*.
4. Use the checkboxes to select the gadgets to be made available to members of the user group (or tick the *All Gadgets* checkbox to grant access to all gadgets).

Changes are automatically saved as gadgets are selected.

Network maps

The **Monitor** > **Network Maps** page is the interface for viewing/monitoring, creating, and customizing network maps. Management views for mapping groups and objects can also be accessed from this page.

This section contains guides and information on the **Network Maps** tab/section of the web interface, as well as further details related to Scrutinizer's network mapping functions.

On this page:

Overview maps-basics Creating maps maps-create Viewing maps maps-view Map customization maps-customize Managing mapping groups maps-manage-groups Managing mapping object maps-manage-objects

Network mapping overview

Scrutinizer's integrated network mapping functions allow users to create dynamic, highly-customizable topology visualizations that can greatly enhance network monitoring and management workflows.

Network maps can be created as one of two types:

- **Spatial maps** allow map objects to be manually positioned in any layout. Custom connections, objects, and backgrounds (e.g., wiring cabinet, office floor plan, etc.) can also be used to increase the level of detail.
- **Geographical maps** use the longitudinal and latitudinal coordinates associated with map objects to automatically position them on a global map. Geomaps can help identify devices with issues, even when there are multiple topologies dispersed across different physical locations. Coordinates can be entered via the mapping object management view.

Existing maps can be viewed from the main **Network Maps** page and/or added to *dashboards*.

Important

- Geographical maps require a Google Maps browser API key, which can be entered in the options tray of the mapping group management view.

- Access to the Internet is required for Google Maps geolocation requests. If Scrutinizer is unable to reach the Internet normally, a Google Maps proxy server can be configured under Admin > Settings > Google Maps Proxy Server.

Mapping/map groups

Network maps are populated by assigning objects to mapping groups. Maps/groups can be created from either the main **Network Maps** page or the mapping group management view.

Groups are populated as part of creating a new map, but they can also be manually defined from the mapping group management view.

> **Note**
>
> Mapping groups are a separate *grouping scheme* from IP groups.

Object membership for existing groups can be modified at any time, and the map will automatically be updated the next time it refreshes.

For further details on mapping groups, see the page on mapping group management.

Mapping/map objects

Each mapping group can contain any number of objects of the following types:

- Devices/exporters
- Other mapping groups
- Custom map objects (spatial maps only)

Map objects can be added or reconfigured while in map edit mode (objects assigned to current map/group only) or from the mapping object management view.

For further details on mapping objects, see the page on mapping object management.

Connections

Connections are used to add links between objects in network maps and can serve the following functions:

- Show basic association between objects
- Display status/activity between interfaces
- Run a saved report for the connected objects

Each map can be configured with any number of connections, allowing users to tailor maps to their monitoring needs.

To learn more about adding and configuring connections, see this section.

Creating a new map

New network maps/mapping groups can be created from the main **Monitor** > **Network Maps** page or the mapping group management view.

Note

After a spatial map/group is first populated, map objects will be stacked on top of each other until they are manually repositioned in map edit mode. Objects in new geographical maps will be similarly clustered, unless they have had an address or GPS coordinates associated beforehand.

To add/create a new network map, follow these steps:

1. Navigate to **Monitor** > **Network Maps**, and then click the **Add** button.
2. In the **New Group** tray, select whether to create a spatial map or geomap.

Important

Geographical maps require a valid Google Maps browser API key to be displayed correctly. If no key has been added, it can be entered in the field provided when creating a new geomap. An API key can also be added via the mapping group management view, in the options (gear) tray.

3. Enter a name (required) and description (optional) for the map.
4. Click the **Apply** button to automatically open the map configuration tray.
5. [Optional] In the **Add Object** secondary tray, use the checkboxes to select the devices and/or mapping groups to add as objects to the new map.

   Note

   Objects are added to the map in real time as they are selected.

6. [Optional] Add one or more custom objects to the new map by clicking the **+** button under *Objects* in the primary tray.
7. [Optional] Add one or more connections between objects in the new map by clicking the **+** button under *Connections* in the primary tray.
8. [Optional] Add a background image for the map by clicking *Background* in the primary tray and either selecting one of the provided images or uploading a custom background.
9. [Optional] Apply optional map settings to the new map under *Settings* in the primary tray.
10. Close the tray to return to the previous view.

Once a map has been created, it can be reconfigured or further customized at any time, through the optional steps described above. The map configuration tray can be accessed while in map edit mode or by selecting *Settings* in the three-button menu in the mapping group management view.

Viewing network maps

When navigating to the main **Monitor** > **Network Maps** page, the default map for the current user is displayed. To bring up a different map, click the All Maps button and select it from the list. Network maps can also be accessed by clicking on the map/group name in the mapping group management view.

The main map view includes the shortcuts/buttons to the following views and functions:

Mapping Groups	Opens the mapping group management view
Mapping Objects	Opens the mapping object management view
Add	Opens the new map/group tray
Refresh	Updates the map to reflect most recent collected data
Report Menu	Opens a tray from which any report applicable to the group can be run
Edit Map	Switch to map edit mode
Options	Configure the following global network map settings: - Refresh interval (in minutes) - Set connections to show rate or utilization - Use resolved hostnames instead of IP addresses as object labels

Exporter/device map objects can be clicked to inspect rate or utilization by interface (via the Explore > Exporters view). Child groups can also be clicked to drill into their maps.

To run a report, modify object properties (including location information), or create a connection from the object, right-click an exporter or group object at any time.

> **Note**
>
> - Full screen mode can be used to monitor network maps on larger displays. The zoom level can also be adjusted as needed using the corresponding buttons.
> - To access data for all objects assigned to a network map, a user must also be granted access to all included devices and/or interfaces.

Map selection

The **All Maps** tray can be used to quickly switch between existing maps.

Clicking the reports (graph) icon opens the **Available Reports** tray, which allows the user to quickly pivot to any report applicable to the group. Hierarchical display for the map/group list can also be toggled on or off as needed.

> **Hint**
>
> If *Map Hierarchy* is enabled, expand a parent group to see all maps/groups added to it.

The *View All Mapping Groups* and *View All Mapping Objects* links can be used to navigate directly to the mapping group and mapping objects management views.

Adding maps to dashboards

Network maps can be added to dashboards, where they can be viewed alongside other *dashboard gadgets*.

To learn more about dashboards and gadgets, see *this section* of the manual.

Map customization

Network maps in Scrutinizer can be uniquely tailored to any type of environment/topology using several customizable elements.

Map edit mode

Once a map/group has been created, it can be further re-configured/customized by switching to map edit mode in the main network map view. This allows map objects in spatial maps to be freely repositioned.

The map editing toolbar provides access to the following functions:

- Configure map settings
- Mapping objects
- Select and drag objects
- Inspect/edit object membership
- Auto-align objects
- Edit object layering (*Bring to front*, *Send to back*, etc.)

After making changes, click the save button to update the map/group.

Additional shortcuts can also be accessed by right-clicking any object. Clicking the **Edit Map** button a second time exits map edit mode.

Note

- To assign location information to a geographical map object, left-click on the object to enable editing and select *GPS Location* in the secondary tray.
- Location information for an object is unique for each geographical map/group it is a member of.

Custom objects

Custom objects are non-device/non-group objects that can be displayed as icons (similar to regular map objects) or text boxes in a network map.

To add a new custom object, follow these steps:

1. Navigate to the mapping object management view and click the **+** button.
2. Use the dropdown to select the object type to create (*Icon* or *Text Box*), and then configure the following properties for the object:

Icon object properties	
Icon	Icon graphic to use for the object in the network map
Color	Color to apply to the icon
Size	Size of the icon
Weight	Icon variant to use
Label	Label to display for the object in the network map
Link	Complete URL of page to open when object is clicked (for example, *http://www.plixer.com*)
Description	Custom description to display in the tray object list and the mapping object management view

Text box object properties	
Label	Label to display for the object in the network map
Type	Determines whether the selected color is applied to the text or the background
Link	Complete URL of page to open when object is clicked (for example, *http://www.plixer.com*)
Description	Custom description to display in the tray object list and the mapping object management view
Shape	Shape to use as the background for the text
Dimensions	Dimensions to apply to the selected shape (length, width, radius, etc.)
Color	Color to apply to the selected shape

3. Verify that the correct details have been entered, and then click the **Apply** button to save the new custom object.

Once a custom object has been created, it can be added to any map/group at any time. It can also be repositioned while in map edit mode and used as an endpoint in map connections (line and saved report connections only).

> **Note**
>
> - After adding a custom object to a geographical map, switch to map edit mode and left-click on the object to assign location information to it.
> - Custom objects can also be defined while in map edit mode or from the mapping group management view, under *Objects* in the map settings/configuration tray.

Connections

To add links showing relationships or network activity between objects in a network map, define one or more connections for the group.

To define one or more connections for a map/group, follow these steps:

1. Navigate to the mapping group management view and select *Settings* from the three-dot menu for the map to add connections to.
2. In the tray, click the **+** button under *Connections*.
3. In the secondary tray, use the two *Endpoint* dropdowns to select the objects to link with the connection.
4. Select a connection type from the *Type* dropdown, and then configure the required properties:

Connection Type	Function	Properties
Interface	Displays activity/utilization between the two endpoints on the specified interface	Interface/instance
Line	Static line linking the two objects	• Color • Label (optional)
Saved Report	Runs a specified saved report when clicked and changes color if thresholds are configured	• Saved report to run • Yellow, orange, and red thresholds (optional)

5. Verify that the correct details have been entered and click the **Apply** button.

After a connection has been added to a map, mouse over the connection to view additional details.

> **Note**

- If a map/group has existing connections, they can be edited or deleted by clicking the corresponding icons in the list.
- Connections can also be added to a map by opening the tray while in edit mode.

Interface connections

Interface connections are dynamic links whose visual properties indicate real-time status and traffic/activity between objects:

- The arrow **orientation** indicates the directionality of the highest traffic volume (as indicated in the connection label).
- The object **closer to the arrow** is the device/interface on which activity is metered (i.e., the object that the activity displayed is *inbound to* or *outbound from*).
- The connection's **color** indicates one of the following:
  - **Green**: Active
  - **Yellow/orange/red**: Utilization reaching global thresholds configured under Admin > Settings > Thresholds
  - **Blue**: No bandwidth statement available
  - **Grey**: No traffic
  - **Dashed grey**: No flow data received in the last 5 minutes

Note

Utilization percentages can only be displayed for interfaces whose speeds are known (via SNMP or a custom setting).

Saved report connections

Saved report connections can be configured to run any saved report that applies to the two endpoints defined in the connection.

The yellow, orange, and red thresholds configured for the connection apply to the total of the rightmost/calculated column of the report. These thresholds are independent of report thresholds, which can be added to any saved report to trigger alarms.

Backgrounds

Spatial maps support the use of custom backgrounds (e.g., wiring cabinets, office floor plans, etc.), which can be uploaded under *Background*, in the map settings/configuration tray.

This can be done while in map edit mode or from the mapping group management view.

Managing mapping groups

The **Network Maps** > **Mapping Groups** view can be used to create, configure, and manage mapping groups.

It lists all existing maps/groups, alongside the following details for each one:

- Status
- Type (spatial or geographic)
- Description (if set)

- Timestamp when the group was last modified
- User that created the map

Clicking on a group name displays the map in the main **Network Maps** page. The map settings/configuration tray and other shortcuts can be accessed from the three-dot menu.

Global mapping group/page settings

The **Options** tray (gear button) contains global network map settings, as well as options for the mapping group management view, including:

- Default map for the current user
- Google Maps browser API key and TLD
- Map refresh interval in minutes
- Enable/disable hierarchy view in the mapping group management view (independent of the map selection tray toggle in the main **Network Maps** view)

Bulk actions

When one or more maps/groups are selected, the following batch operations can be performed via the **Bulk Actions** tray:

- Add or remove objects for all selected groups (shows objects common to all selected groups)
- Run a report filtered on all devices/objects included in the selected groups
- Delete the selected groups
- Clear selection

Creating a new map/group

New maps can be created from the mapping group management view by clicking the **+** button and entering the map type and name for the new group. After the group has been created, it can be further customized (membership, connections, etc.) via the configuration tray.

Clicking a group name will display the map in the main **Network Maps** view, where it can be further configured in edit mode.

Configuration tray

Selecting **Settings** from the three-dot menu opens the configuration tray for that map/group, from where the map/group can be reconfigured/customized at any time.

The tray is divided into the following main sections (also accessible in map edit mode in the main **Network Maps** view):

Settings	Inspect/edit general settings
Objects	Manage object membership or add custom objects to the group
Connections	Define or manage connections for the group
Background	Upload/select a background for the network map

In addition, the configuration tray includes shortcuts for the following actions/functions:

- Run a specified report filtered on the devices/objects included in the group
- Create a duplicate of the selected map
- View the map in the main **Network Maps** view

- Set the selected map as the default map for the current user

Settings

The *Settings* section of the configuration tray contains the following general map settings/options:

Name	Name identifying the map/group
Auto-add devices	Automatically add devices with resolved hostnames matching the specified regular expressions (RegEx)
Truncate labels on	Shortens map object labels by omitting the entered string
Description	Optional description to add to the map/group (can also be viewed by clicking the **i** icon in member object lists)
Pass status	When enabled, the current group's status will be reflected in its map icon in parent maps.

Objects

When expanded, the *Objects* section of the tray shows all map objects currently assigned to the group. Objects can also be reconfigured or deleted by clicking the corresponding icon in the list.

To manage object membership for the current group, click the edit (pencil) button and select/deselect devices, groups, or custom objects in the secondary tray.

Connections

To learn more about map connections, see this section of the map customization guide.

Backgrounds

To learn more about adding custom backgrounds to spatial maps, see this section of the map customization guide.

Managing mapping objects

The **Network Maps > Mapping objects** view can be used to manage mapping object properties and group membership. New custom objects can also be defined from this view.

The main view lists all map objects currently assigned to at least one network map/group, alongside the following details:

- Icon assigned (reflects device or group availability)
- Type (exporter, user-created/custom, or map/group)
- Status
- Link that will be opened when the object is clicked (custom objects only)
- Number of maps/groups the object has been assigned to
- Timestamp when the group was last modified

Clicking on a group name opens the object properties/configuration tray, from where the object can be edited or assigned to maps/groups.

Bulk actions

When one or more maps/groups are selected, the following batch operations can be performed via the **Bulk Actions** tray:

- Add selected objects to one or more maps/groups
- Remove selected objects from one or more groups they share

- Add GPS location details to all selected objects

Object properties

Exporter and group object icons can be customized through the following properties:

- Icon
- Color
- Size
- Weight (icon variant)
- Label
- Link
- Description

These properties are applied to the object icon across all maps/groups an object has been assigned to.

Group membership

To manage group membership for the current object, expand the **Current Groups** section of the object properties tray.

Clicking the **+** button opens a secondary tray, where the object can be assigned to or removed from one or more maps/groups.

GPS location

Objects assigned to geographical maps are automatically positioned based on their location.

To enter GPS coordinates for an object, click **GPS Location** in the object properties tray. An address can also be entered instead, for which coordinates will automatically be obtained via GPS lookup.

Custom objects

To learn more about adding/defining custom objects, see this section of the map customization guide.

Topology (BETA)

The **Monitor** > **Topology** page shows a topological visualization of network equipment and their connecting interfaces by leveraging the Cisco Discovery Protocol and Link Layer Discovery Protocol.

Devices

Drilling into a device in the map opens a tray with additional details (device name/type, location, etc.) and the following options:

- **View** - Opens the host details page to view traffic summaries, interface activity, and any alarms/events or anomalous behavior associated with the host
- **Reports** - Run any report type supported by the device

Interfaces

Clicking a connection in the map opens a tray showing connected devices and interfaces used.

To view further details or run a supported report type, drill into a host or interface from the tray.

Added in version 19.7.0: The **Topology** view/feature is currently in beta and will have expanded functionality in future releases.

Explore

The **Explore** views of the web interface can be used to quickly look up information on exporters, hosts, and other entities (users, applications, etc.) in the **Scrutinizer** environment.

This section covers the different functions and types of information that can be accessed via **Explore** views of the web interface.

On this page:

Exporters explore-exporters Entities explore-entities Search explore-search

Exporters

The **Explore** > **Exporters** tab can be used to look up information for all devices sending flows to **Scrutinizer** collectors.

The main view lists device status, traffic information, and other details either by interface (default) or by exporter and provides access to a summary tray for drilling into the corresponding alarm and host views. The left-hand mapping/device group pane can be used to apply filters and manage mapping group settings, membership and connections.

Interfaces view

The *By Interface* view lists the associated exporter as well as inbound and outbound activity details for each interface. A status icon indicates whether the exporter is available (green) or offline (red).

The following options can be accessed by clicking the exporter address/hostname, interface name, or three-dot menu in the table:

- **Reports**: Run any report supported by the exporter
- **Information**: Shows general interface information and links to the Admin > Interfaces management view filtered on the interface
- **Exporter**: Opens the **Alarms** subtab of the host details view for the exporter
- **View Interface**: Opens to the host details view for the interface
- **View Exporter Alarms**: Opens the Alarm Monitor > Hosts view filtered on the exporter
- **Reset Highwater Inbound**: Resets highwater mark data for inbound traffic
- **Reset Highwater Outbound**: Resets highwater mark data for outbound traffic
- **Reset Highwater Both**: Resets highwater mark data for both inbound and outbound traffic

> **Note**
>
> - The units buttons can be used to toggle between bits, bytes, and utilization percentage in the *Inbound* and *Outbound* columns. Utilization will only be available for interfaces whose speeds are known (via SNMP or custom settings).
> - The bulk actions tray, which contains options to run applicable reports and reset highwater values, can be accessed after one or more exporters or interfaces are selected using the checkboxes.

Exporters view

The *By Exporter* view lists exporter hostnames/addresses alongside the following details:

- Current status of the exporter (green: available, red: offline)
- Number of mapping groups the exporter is assigned to
- Number of interfaces associated with the exporter
- Average packets per second over the last 12 hours
- Average flows per second over the last 12 hours
- Timestamp of the most recent flow received from the exporter

In this view, the following options can be accessed by clicking the exporter address/hostname or three-dot menu in the table:

- **Reports**: Run any report supported by the exporter
- **Information**: Shows general exporter information and links to the Admin > Exporters management view filtered on the exporter
- **Exporter**: Opens the **Alarms** subtab of the host details view for the exporter
- **Interfaces**: Switches to the **By Interface** view filtered on the exporter
- **Tags**: View/manage custom tags for the device
- **Mapping**: Edit object icon properties, mapping group membership, or location details for the exporter
- **Admin**: Opens the Admin > Exporters management view (no filters applied)
- **View Exporter Alarms**: Opens the Alarm Monitor > Hosts view filtered on the exporter

Note

- Click the details in the *Groups* and *Interfaces* columns of the table to quickly access the corresponding options in the tray.
- In the *By Exporter* view, the bulk actions tray contains options to run reports, add custom tags, and edit mapping details for all selected exporters.

Mapping group pane

The mapping group pane lists all current mapping/device groups and provides quick access to the following functions:

- Run any report supported by the group’s devices/exporters
- View the network map for the group
- Apply a filter for the group’s exporters or interfaces to the main list/table (click the filters button for additional options)
- Create a duplicate of the selected network map

In addition, the *Modify* option opens a tray where the settings, membership, connections or settings for the network map can be modified.

Entities

The **Explore** > **Entities** tab can be used to look up and inspect the individual data entities–both user-defined and discovered–monitored by **Scrutinizer** as part of network activity.

The page is divided into separate subtabs displaying the following details for each entity type:

Usernames

- Host associated with the observation
- Data source
- Machine name (if available)
- Timestamp when the username was first seen on the host
- Timestamp when the username was last seen on the host

Applications Defined

- Number of exporters the application was observed on
- Total number of flows with data associated with the application
- Average packet rate for activity involving the application
- Average data transfer rate for activity involving the application

Hosts - Sources/Destinations/Pairs

- Source and/or destination IP address(es)/hostname(s)
- Number of exporters the source, destination, or pair was observed on
- Total number of flows with data associated with the host(s)
- Average packet rate for activity involving the host(s)
- Average data transfer rate for activity involving the host(s)

Autonomous Systems - Sources/Destinations/Pairs

- Source and/or destination autonomous system(s)
- Number of exporters the source, destination, or pair was observed on
- Total number of flows with data associated with the autonomous system(s)
- Average packet rate for activity involving the autonomous system(s)
- Average data transfer rate for activity involving the autonomous system(s)

IP Groups - Sources/Destinations/Pairs

- Source and/or destination IP group(s)
- Number of exporters the source, destination, or pair was observed on
- Total number of flows with data associated with IP group(s)
- Average packet rate for activity involving the IP group(s)
- Average data transfer rate for activity involving the IP group(s)

Countries - Sources/Destinations/Pairs

- Source and/or destination country/countries
- Number of exporters the source, destination, or pair was observed on
- Total number of flows with data associated with the country/countries
- Average packet rate for activity involving the country/countries
- Average data transfer rate for activity involving the country/countries

Protocols

- Number of exporters the protocol was observed on
- Total number of flows with data associated with the protocol
- Average packet rate for activity involving the protocol
- Average data transfer rate for activity involving the protocol

Clicking on an entity in any subtab opens a summary page (similar to the host traffic subview) that contains visualizations of the entity's activity as well as report shortcuts for deeper investigations.

Note

Shortcut links to manage application definitions, protocol exclusions, and *FA algorithm exclusion rules* are included in the corresponding subtabs.

Search

The **Explore** > **Search** tab allows users to search the **Scrutinizer** host index to quickly verify whether or not a host has been seen on the network. Searches can be performed for either individual hosts or pairs (host to host). Simultaneous lookups for multiple hosts or pairs are also supported.

Important

To be able to search for hosts and host pairs, the corresponding indexing feature must be enabled.

The following are the available details displayed in the search results:

- Host
- Traffic direction (inbound, outbound, A > B, B > A, bidirectional)
- First and last seen timestamps
- Exporter/source of collected data
- Bytes in and out
- Packets in and out
- Flows in and out

To show fewer details in search results, click the table button and untick the checkboxes for the columns to be hidden.

In the search results, drilling into a host will display a summary of its activity on the network. Clicking on a data source opens a tray that allows the user to quickly pivot to any supported report type.

Enabling host indexing

When host indexing is enabled, **Scrutinizer** will store records for all hosts that pass traffic on the network. Records for host pairs can also be stored (and searched through) by enabling host to host indexing as indicated below.

To enable host indexing:

1. Navigate to Admin > Alarm Monitor > Flow Analytics Algorithms.
2. Open the configuration tray for the *Host Indexing* algorithm.
3. Add sources/inclusions for the algorithm either indvidually or using security groups.

> **Hint**
>
> Recommended inclusions for host indexing are internal/core routers, edge routers, and public IP addresses that have been assigned to IP groups.

4. If there are sources (IP addresses/ranges, domains (by reverse DNS), IP groups, etc.) that should not be indexed, add them as exclusions.
5. Expand the **Settings** secondary tray to configure the following:
   - Days of Host Index Data Retention
   - Host Index Database
   - Host Indexing Domain Socket
   - Host Index Max Disk Space
   - Host Index Sync Interval Minutes
   - Host to Host Database
   - Window Limit
6. (Optional) Enter a database path in the *Host to Host Database* field to enable host pair indexing. To disable the feature, leave it blank.
7. Use the toggle to enable the algorithm and close the tray.

Once the algorithm has been configured and enabled, users can use the **Explore** > **Search** view to search the host or host pair (if enabled) index.

> **Hint**
>
> If the *Use Host Index* option (Admin > Settings > Reporting) is enabled, only exporters that a host has been seen on will be searched when data is aggregated for a report. This can significantly reduce the time it takes to run reports.

Resource requirements

When host indexing is enabled, additional resources may need to be allocated to the **Scrutinizer** collectors as described *here*.

Host index population from historical data

If host indexing is not immediately enabled after **Scrutinizer** is deployed, the database can be backfilled at a later date using historical data.

To populate the host index database from historical tables, follow these steps:

1. SSH to the **Scrutinizer** server as the `plixer` user.
2. Stop the host index service:

   ```
   sudo systemctl stop scrutinizer-host-index
   ```

3. Run the following to populate the database using the specified historical data tables and time range/window:

   ```
   host_index --db_config --verbose --populate_from_history --
   ↪table_interval=INTERVAL_TABLE --date_start="<START_DATE_TIME>" --date_end="
   ↪<END_DATE_TIME>"
   ```

   where:

   - `START_DATE_TIME` and `END_DATE_TIME` must be formatted as `YYYY-MM-DD HH:MM`, with the time in 24-hour format (leading zeroes should be omitted).
   - `INTERVAL_TABLE` is an integer that specifies the *aggregation interval tables* and should be set to `1`, `5`, or `30`.

Note

- If the time element is omitted from `END_DATE_TIME`, data from the end date specified will be excluded from the operation.
- The utility can also be used to repopulate the host index database in case of data corruption. However, it is highly recommended to contact *Plixer Technical Support* for assistance with restoring data.

Investigate

The **Investigate** views of the web interface provide access to Scrutinizer's collaborative investigation and ML-powered forecasting functions (requires Plixer One Enterprise).

This section covers the **Collections** and **Forecasts** views and includes detailed guides for their associated functions.

On this page:

Collections investigate-collections   Forecasts investigate-forecasts

Collections

Collections are bundles of one or more alarms, events, and/or reports that are compiled and assigned to a specific user for further review and analysis.

Once created, a collection can be annotated and reassigned, allowing multiple users (e.g., NetOps and SecOps) to share workloads and collaborate in investigations.

Collections page

The **Collections** page of the **Investigate** section lists all existing collections and is split into two tabs: **Assigned to Me** (current user) and **Other Collections**.

Along with each collection's name, the table also shows the following details:

- An indicator that shows the current active collection (green checkmark)
- User who created the collection
- Date and time the collection was created
- Date and time the collection was assigned
- User to whom the collection is currently assigned
- Number of alarms, events, and/or reports that have been added to the collection

From the main **Collections** page, the following actions are available:

- **Viewing collections** - Click on a collection's name to open its summary page.
- **Deleting collections** - Select one or more collections, and then click the **Delete** button.
- **Reassigning collections** - Click the username under a collection's *Assigned User* column to assign it to a different user.
- **Setting the active collection** - Use the radio buttons to set/change the active collection. For additional information, see the subsection on managing collections.
- **Filtering options** - Click the filter button to view available filtering options for the list.
- **Options** - Click the gear icon to view the available options for the list.

Inspecting collections

A collection's summary page lists all alarms, events, and reports added to the collection as links that allow the user to drill down into each item. Annotation can be added to the summary page in threaded view using the **Notes** card.

In addition, the table also lists the following details for each item:

- Type of item
- Additional details, such as the number of individual events, hosts involved, or report type (click **+** to expand)
- Date item was added to the collection
- User who added the item
- Any notes related to the alarm, event, or report added by users

> **Hint**
>
> When adding notes to a report item in a collection, the text field will be pre-populated with basic information about the report.

To remove items from the collection, select one or more items using their checkboxes, and then click the **Delete** button.

Collection management

The collection management menu can be accessed from either of the following:

- **Alarm monitor view**

1. Navigate to either **Alarm Monitor > Policies** or **Alarm Monitor > Hosts** tab.
2. In the **Alarm Policy** or **Host** list, hover over the star icon, and then select **Manage Collections**.

- **Current report view**

1. Navigate to **Reports > Run Report**, and then create/run a new report.
2. After the report is run, hover over the star icon, and then select **Manage Collections**.

Creating a new collection

To create a new collection, click the **Add New Collection** (**+**) button, and then enter a unique name for the collection. Select a user to assign the collection to, and then click the **+** button to save the collection.

> **Note**
>
> The name and user fields must both be filled to create a new collection.

Once the collection has been successfully created, it will be added to the list in the management menu.

Setting the active collection

To set/change the current active collection, open the management menu, and then select the collection from the list. The green checkmark beside the collection name indicates that it is the current active collection. Only one collection can be set as active at a time.

The active collection can also be set from the main collections page.

Adding alarms, events, or reports to a collection

1. Click the star button to open **Manage Collections** menu.
2. Click the button a second time (after it turns into an add (**+**) button).

This automatically adds the alarm, event, or report to the active collection.

To remove the item from the active collection, click the star button, and then click the button a second time (after it turns into a minus (**-**) button).

Forecasts

When paired with the Plixer ML Engine, Scrutinizer is able to use the aggregated flow data of a specified report to generate forecasts of future network activity and/or resource utilization.

> **Important**
>
> Forecasts require an active Plixer One Enterprise license. To learn more about licensing options, contact *Plixer Technical Support*.

Generating forecasts

To generate a forecast, a report must first be run to define the scope of data for extrapolation.

The following data elements in a report will be used to generate the forecast:

- Hosts
- Data points
- Time period covered
- Filters applied

In the results/output, click the **Forecast** button, and then enter a name to save the new forecast under. The main Investigate > Forecasts page will automatically be displayed after the forecast is created.

Note

The amount of time it takes to generate a forecast varies, depending on the amount of data that needs to be processed.

Forecast horizon and seasonality customization

By default, Scrutinizer applies a recommended forecast horizon and seasonality based on the volume of data sampled in the report used.

To manually define the horizon and seasonality instead, the filename for the forecast should be formatted as follows:

```
[forecast_name] ? <horizon_integer> <time_unit> with [no|auto|null] season␣
↪[<season_integer> <time_unit>]
```

Natural language is also supported, so a forecast titled:

```
VPN Usage ? for 3 months with a season of 14 days
```

will generate a forecast with projected values for 3 months (after the end of the report time range/window) and a seasonality of 14 days.

Viewing Forecasts

All previously created/saved forecasts can be accessed from the **Investigate** > **Forecasts** page. Forecasts that are marked *Complete* under the **Status** column are ready to view.

Clicking on a forecast opens a detailed view with two sections:

Forecast timeline

The forecast timeline plots the data aggregated by the base report (solid lines) and shows the extrapolations (broken lines) up to the horizon of the forecast. Hovering over a line will show the upper and lower bounds of potential deviation (highlighted region), as well as additional details for the data element used to aggregate the data (hosts, applications, etc.).

The timeline can be viewed as either a line or step graph.

Inbound events

In addition to the timeline, the forecast details view includes a table listing the following information for each host, application, etc.:

- Rank (based on the forecast's calculated data)
- Date and time when the calculated data is expected to reach the expected maximum value
- Expected maximum value of the calculated data
- Upper bound for deviation in the calculated data's expected maximum

When applicable, the table links directly to the relevant **Explore** summary page for each element. The base report for the forecast can also be re-run at any time by clicking the **View Report** button.

Forecast management

The main **Investigate** > **Forecasts** page can be used to access forecasts after they are created and includes the following details for each forecast:

- ID number assigned to the forecast
- Forecast name/filename
- Name of the report used for the forecast (click to re-run)
- Forecast creator
- Current status of the forecast (*Initializing* -> *Starting* -> *Data Retrieval* -> *Processing* -> *Strategy Selection* -> *Learning* -> *Prediction* -> *Complete*)
- Timestamp when the forecast became ready to view

In some cases, it may take up to several minutes for the Forecasting task to progress from *Initializing* to *Complete*.

Updating forecasts

Clicking the refresh icon reinitializes the forecast using the most up-to-date dataset for the base report's time window/range settings.

Forecasts based on reports with a custom date and time range (i.e., not *Last X*) can also be refreshed but will result in the same projections. To obtain an updated forecast, re-run the report with adjusted date and time settings, and then generate a new forecast.

Deleting Forecasts

To delete one or more forecasts, select the forecasts using the checkboxes and then click the *Delete* button to permanently delete them.

Reports

The **Reports** views of the Scrutinizer web interface are used to create, run/view, and manage reports. Advanced features, such as defining custom report thresholds, setting up scheduled email reports, and creating forecasts (requires Plixer One Enterprise), can also be accessed from these views.

This section comprises detailed guides for leveraging the various functions related to reports in Scrutinizer.

On this page:

Creating reports reports-create Report output reports-view Refining report results reports-refine Report filters reports-filters Saved reports reports-manage Email reports reports-email Report functions reports-functions

Creating reports

Reports are fully configurable network data aggregations that enable customized transparency for any asset or activity on the network.

When a report is run, traffic data matching the specified filters (time window, sources/devices, etc.) is collated based on the selected *report type*. The results are then displayed in the output view.

Creating/running a new report

To create/run a new report, navigate to the **Reports** > **Run Report** page and follow these steps:

1. Select between the two options to start creating a report:
   - **Select Devices**: Select one or more devices to use as data sources for the report before specifying the report type.
   - **Select Report Type**: Select a *report type* to define the data aggregation criteria before specifying data sources.
2. After the devices and report type have been selected, configure the following settings/filters for the report:
   - **Time Window**: Select a *Last* **X** time window or specify a custom range to be covered by the report (default: *last 24 hours*).
   - **Display Type**: Select the graph or chart for result visualization in the output view.
   - **Additional Filters**: Define any additional filters to be applied to the report.
3. Click **Run Report**.

A progress bar is shown as the report is being run, after which the report results/output view will be displayed.

> **Note**
>
> - Settings and filters can be modified after a report is initially run to refine the results for further investigation.
> - When a *last* **X** time window is selected, clicking the up or down arrow will automatically shift the date/time period covered forward or backward.

Saving reports

After a report is created and run, the configuration can be saved by clicking the save (disk) button in the output view. See this page for further details on saving and managing reports.

Running reports via URL

A *Host to Host* pair report can be run against all available devices with a filter for a specified IP address (`FILTER_IP`) using the following URL format:

```
https://SCRUTINIZER_ADDRESS/ui/reports/run-report/search/el/FILTER_IP
```

Scrutinizer will also accept a `FILTER_IP` in hex format but only if the IP address belongs to an exporter.

Report type groups

When configuring a report, only report types and groups that are supported by the flow data currently available are displayed. The *Recommended*, *Recent* (last 16 report types run), and Designed Reports categories can also be used to quickly find frequently used report types.

Adding report types to/from the Recommended group

The *Recommended* report type group can also be customized to better support specific workflows:

1. **Identify the numerical ID of the report type to be added/removed:**

    1. Run the report type to be added/removed in the Scrutinizer web interface.
    2. Extract the report type langkey from the URL (`reportTypeLang%22:%22REPORT_TYPE_LANGKEY%22`) after the report completes running.
    3. SSH to the Scrutinizer server or primary reporter as the `plixer` user, and then access the database:

        ```
        ssh plixer@SCRUTINIZER_IP
        psql plixer
        ```

    4. Run the following query to obtain the numerical report type ID:

        ```
        SELECT
          rpt_lang,
          rt_id
        FROM plixer.report_types
        WHERE rpt_lang IN('REPORT_TYPE_LANGKEY');
        ```

        To obtain multiple report type IDs, separate the langkeys with commas within the parentheses (e.g., ('LANGKEY_1','LANGKEY_2',...))

2. Run the following command to add the report type ID(s) to the *Recommended* report type group:

    ```
    INSERT INTO plixer.report_types_recommended
    (rt_id,rtg_id)
    VALUES
    (REPORT_TYPE_ID,72)
    ON CONFLICT DO NOTHING;
    ```

    Multiple IDs can also be included in the above commands (in the same `(ID,72)` format; one per line and separated by commas)

Note

To remove report types from the *Recommended* report type group, obtain the numerical IDs as described above, and then run (also supports multiple IDs):

```
DELETE FROM plixer.report_types_recommended
WHERE rt_id IN (REPORT_TYPE_ID)
```

User-defined groupings/labels

Additional report type groups can also be created to further customize how reports are displayed within the *Recommended* group:

1. Create the new group:

    ```
    INSERT INTO plixer.report_types_menu_groups
    (rtg_id, order_id, group_lang)
    VALUES
    (NUM_GROUP_ID,4,'customGroupName')
    ON CONFLICT DO NOTHING;
    ```

    where:

- NUM_GROUP_ID: must be greater than 10000 (values below are reserved)
- customGroupName: any string with no spaces or special characters

2. Add a language entry for the new group:

```
INSERT INTO
languages.custom
(id,string)
VALUES
('customGroupName','GROUP NAME IN UI')
ON CONFLICT DO NOTHING;
```

3. Add report types to the *Recommended* group using the report type ID and the new custom group ID:

```
INSERT INTO plixer.report_types_recommended
(rt_id,rtg_id)
VALUES
(REPORT_TYPE_ID,NUM_GROUP_ID)
ON CONFLICT DO NOTHING;
```

4. Refresh the UI views:

```
REFRESH MATERIALIZED VIEW plixer.report_types_reports_in_group;
REFRESH MATERIALIZED VIEW plixer.report_types_template2id;
REFRESH MATERIALIZED VIEW plixer.report_types_all_report_groups;
```

After the UI refresh, report types in the *Recommended* group will be displayed with the new group name in the web interface.

Custom reports

To learn more about creating custom reports, see the Report Designer topic in the Classic UI section of this documentation.

Report output

The output of a report will mainly consist of two classes of data: the grouping criteria/entities(sources/destinations, IP groups, users, etc.) and their aggregated activity data.

After a report completes running, the results are displayed in both graph and table formats in the output view, where the reports original settings can continuously be refined to create the visibility required for the current task.

Graph details and functions

Each report type supports multiple interactive graph options to visualize the data for the top ten grouping entities based on their activity. An *Others* entity, which combines the aggregated activity data for all entities outside the top ten, is also included.

The **Graph** dropdown allows the user to quickly switch between the available visualizations directly from the output view. Additional details for any entity or activity can be viewed by hovering over the corresponding graph element.

Table/list details

The output view table functions as both a summary of the report results and a legend for the graph. The columns to the left (without the sorting arrows) list report type's grouping entities, while the right-hand columns are used for the aggregated activity details. Traffic values can be displayed as average rates or totals (for the entire time range) by selecting the corresponding global setting in the Options tray.

Clicking on an entity in any grouping criteria column (e.g., source, application, or destination in a *Conversations App* report) opens a tray from where any supported report type can be run.

Hint

- Timeline graphs (line, step, stacked bar, etc.) can be used to apply a new time range to the current report. To do this, click on the graph once, and then click and drag to highlight the new range to use.
- To hide the graph for the current report, click the **Hide** button in the header.
- Individual cells in the grouping criteria columns of the table can be dragged to the left into *inclusion* and *exclusion* dropzones to configure additional filters for the current report (click the **Apply** button in the tray when done).

Filters tray

Clicking the **Report Filters** button in the output view opens a tray where the filters for the current report can be redefined.

To add a new filter, do the following:

1. Click the **Filters** button to open the tray.
2. In the tray, click the **+** button.
3. Select filter type for the new filter.
4. Configure the required details for the filter (varies by filter type).
5. Click the **Add** button.
6. In the primary tray, click the **Apply** button to re-run the report with the new filter(s) applied.

Existing filters can be modified by clicking the edit (pencil) button or removed by clicking the delete (trash bin) button.

For a full list of supported report filters, see this page.

Flow Hopper view

The results table of *Connections by Bytes* pair reports includes the option to switch to the Flow Hopper view, which can be used to retrace the path taken by a flow traversing the network. The path shown in this view will remain accurate even if topology has changed since the time of the flow.

Note

Flow Hopper requires all devices in the flow path to be exporting NetFlow v5 or higher to the collector. Next-hop routing information and read-only SNMPv2 or v3 access to the router is also required.

If an asymmetric flow path is observed (i.e., a different return route), the connection will be drawn out accordingly. Hovering over each router or layer 3 switch in the view will display all details included in the flow template. Changes in element values (e.g., DSCP, TTL, octets, etc.) between ingress and egress metered flows are highlighted as well.

Additional options

Clicking the **Options** button in the header opens a tray containing the following option submenus:

Global	*Data*: Toggle between rates or totals in report results. *Data Source*: Specify an *aggregation/roll-up table* to use for reports. *Data Units*: Toggle between bits or bytes in report results. *Interfaces*: Enable/disable grouping report results by interface. *Data Mode*: Toggle between *summary and forensic* flow data to run reports. *Show Others*: Enable/disable including the *Others* grouping entity in report results. *Show Host Names*: Toggle between host IP addresses and hostnames in report results. *Rows*: Select the number of grouping categories to include in report results.
Table	*Peak*: Show/hide additional column for peak activity details. *95th*: Show/hide additional column for 95th percentile activity details. *Values*: Toggle between formatted/rounded and raw calculated activity data in the report table.
Threshold	Configure a custom threshold for the current report.
Details	*Collectors*: View expanded details for the collectors associated with the data sources of the current report. *Exporters*: View expanded details for the exporters/data sources used for the current report. *Report JSON*: View the report JSON (for *reporting API calls*)

Note

- Toggle on **Display Advanced Options** in the tray to access the *Data Mode* and *Values* settings.
- If the *Rows* setting is increased beyond 10, additional grouping criteria/entities will be displayed in gray in the graph.
- Use the **Copy to clipboard** button to quickly copy the report JSON to your clipboard.

Refining report results

After a report is run, the output view can be used to further investigate any entity or activity included in the report results.

Sample use cases and workflows for reports can be found in *this section* of this documentation.

Switching between graphs

After a report has been run, the **Graph** dropdown allows the user to freely switch between the different graph and chart types supported by the report type.

This allows teams to highlight different aspects of a report's results as needed for their resolution or investigation.

Modifying the time range

The current report can be re-run to cover a different time range of flow data, allowing teams to inspect activity for the same grouping criteria at different points in time.

The period of time covered by the current report configuration can be adjusted via the time range selector in the main output view or by highlighting (click and drag) an area in any timeline graph.

Editing filters

Once a report completes running, its initial filter configuration can be modified to highlight activity for specific grouping entities.

In the main output view, click the **Filters** button to add, modify, and/or remove filters. Additional filters can also be defined by dragging entities from the table's grouping criteria columns into the corresponding dropzones on the left side of the page. After the new filter configuration has been set up, click the **Apply** button in the tray to re-run the report.

Pivoting to other report types

The **Report Type** dropdown in the main output view can be used to run a different report type using the current data sources, filters, and other settings. This function can be used when additional context is required to further investigate a host or activity on the network.

Additionally, a different report type can be filtered for a specific entity in any of the table's grouping criteria columns. This is done by clicking on the entity and selecting the report to run in the **Available Reports** tray.

Report filters

Reports can be run using any combination of filters, including data sources (devices) and the time window covered.

The following table lists all additional filters that can be applied either before a report is first run or from the output view:

Type	Description	Parameter(s)	Option(s)
Applications	Filters results for a selected NBAR application	NBAR application	Restriction
Applications defined	Filters results for a selected defined application (based on definitions under Admin > Definitions > Applications)	Defined application	Restriction
Autonomous system by tag	Filters results for the selected autonomous system (AS) tags	Autonomous system (by AS number)	Direction, restriction
Business hours	Filters results for activity during specified business hours	Start hour, end hour, time zone, days	N/A
Calculated column filter	Filters results based on values in one of the report's calculated columns	Filter column, comparison operator and value	N/A
Country	Filters results for the selected country	Country	Direction, restriction
Device/interface	Filters results for activity associated with the specified devices, interfaces, or mapping groups	Device Interface (if a device is selected) Mapping group (if *Group* is selected)	N/A
Domain	Filters results for the specified domain	Domain	Direction, restriction
Flow template	Filters results for the selected template	Flow template	Restriction
Host list	Filters results for the specified hosts	Host IP address(es)	Direction, restriction
Host to host	Filters results for activity between the specified host pair	Host pair IP addresses	Restriction
IP Groups	Filters results for the selected IP group (defined under Admin > Definitions > IP Groups)	IP group name	Direction, restriction
IP host	Filters results for the specified host IP address	Host IP address	Direction, restriction
IP range	Filters results for the specified range of IP addresses	Starting and ending IP addresses	Direction, restriction
IP subnet	Filters results for the specified subnet	Subnet address and mask	Direction, restriction
Internal host	Filters results for activity associated with internal hosts	N/A	Direction, restriction
Port speed	Filters results for the specified inbound and outbound port speeds	Inbound and outbound port speeds	N/A
Protocol	Filters results for communications using the selected protocol	Protocol	Restriction
Sample multiplier	Used to correct the report's results for devices that use flow sampling	Multiplier value	N/A
Source/destination port	Filters results for the specified source or destination port(s)	Port number or range	Direction, restriction
Subnet to subnet	Filters results for activity between the specified subnet pair	Subnet pair addresses and masks	Restriction
TCP flags	Filters results for traffic with the selected TCP flag	TCP flag	Restriction
Type of Service	Filters results for traffic with the selected ToS	Type of service	Restriction
Well-known port	Filters results for the selected well-known port	Well-known port	Restriction
Wildcard mask	Filters results for the specified network and wildcard mask	Network address and mask	Direction, restriction

- *Direction* options: Source, destination, or both
- *Restriction* options: Include or exclude

Important

The additional filters that can be added to a report vary based on the selected devices/interfaces and report type. More filters may also become available when Scrutinizer has access to devices from certain vendors or is configured with additional integrations.

TCP flag filters

In the *Report Type* dropdown, you can run a TCP Flags report to retrieve information about the TCP flags set in TCP packets observed during a network analysis or packet capture.

To run the report, do the following:

1. Navigate to the **Reports** > **Run Report** page.
2. Select one of the two starting points to create a report.

Note

For more information, refer to the Creating/running reports section.

3. In the *Report Type* dropdown menu, select **Designed Reports**, and then select **TCP Flags**.
4. Configure the following settings:
   - Time Window
   - Display Type
5. In the *Additional Filters* field, select **Advanced Filters**.
6. In the *Select Element* field, select **tcpcontrolbits**.
7. Select **Equal** in the *Select Comparison* field, type in *SYN*, and then click **Add**.
8. Click **Run Report**.

Note

Setting this filter generates a TCP Flag report using the SYN (Synchronize) flag in TCP packets observed during a network analysis or packet capture.

Saved reports

After a report has been created and run, it can be saved and re-run at any time from the **Reports** > **Saved Reports** subtab. This page also functions as the management view for saved reports.

Saved reports can be re-run with either the same original configuration or modified settings. They can also be used to set up custom thresholds to trigger alarms and scheduled email reports.

Hint

Access to specific reports and/or report folders can be defined as part of *user group permissions* from the Admin > Users & Groups > User Groups page.

Saved report list

To re-run a saved report, click on the report name in the main view of the **Saved Reports** subtab. Filters, including report folders, can be applied to the list, and it can be displayed in a tabular list or as individual tiles.

Both viewing modes indicate whether the following functions have been enabled or configured for each saved report:

- Custom threshold
- Dashboard gadget
- Scheduled email
- Added to dashboard(s) as a gadget (count)

In addition, the list mode table also indicates the report type, the last-run timestamp, and the creator of each report.

Deleting saved reports

To delete one or more saved reports, select the report(s) using the checkboxes and select *Delete* in the bulk actions tray.

Report folders

After a report has been saved, it can be assigned to one or more user-created folders.

Report folders can be used to organize/filter reports in the **Saved Reports** view. They can also be used to simplify report access management through user group permissions.

Creating report folders

New folders can be created from the **Saved Reports** view as follows:

1. Click the report folders button.
2. In the **Report Folders** tray, click the add (**+**) button.
3. Enter a name for the new report folder in the secondary tray.
4. Click the **Save** button.

Once created, the report folder will be added to the list in the **Report Folders** tray.

Note

Existing report folders cannot be renamed. However, a new folder with the desired name can be created and populated with the same saved reports.

Adding saved reports to folders

There are three ways to assign saved reports to folders:

- When entering a name to save a report, use the dropdown to select a folder to assign it to (*Unfoldered* saves the report without adding it to any folders).

- In the **Report Folders** tray, click the edit (pencil) icon to make changes to the membership list of the selected folder.
- From the main **Saved Reports** view, select one or more saved reports using the checkboxes, and then use the *Move to folder* option in the **Bulk Actions** menu/tray.

Folder management

By default, the main **Saved Reports** view lists all saved reports accessible by the current user. To view only reports assigned to a specific folder instead, open the **Report Folders** tray and select the folder using the link icon.

The following functions can also be accessed via the folder list:

- Edit folder membership (edit/pencil icon)
- Delete folder (delete/bin icon)

Exporting reports

After a report is run, the results can be exported in PDF or CSV format from the **Export** (share button) tray in the output view.

> **Hint**
>
> PDF and or CSV copies of a report can also be attached to email reports.

Email reports

Once an email server has been configured, reports can be forwarded to any email address to provide external access to network data.

Email reports include a link to view the report in the Scrutinizer web interface. PDF and/or CSV copies of the report may also be attached.

On-demand reports

After any report is run, the results can be sent to one or more specified email addresses.

To send an email report, select *Email Report* in the export options tray (share button), and then enter the following details:

- Sender email address
- Recipient email address(es)
- Subject (optional)
- Message (optional)

Tick the appropriate checkbox(es) to attach PDF and/or CSV copies of the report results, if desired, and then click **Send**. A message confirming that the email report has been sent will be displayed.

Scheduled reports

Saved reports can be scheduled to run at specified intervals and sent to one or more recipients, enabling continuous network monitoring from any email inbox.

Hint

Configure a *last* **X** time range/window for a report to send/receive regular updates for any type of network metadata.

Creating a scheduled report

To set up a scheduled email report for a report:

1. Create, run, and save the report.

   Note

   Scheduled reports filtered on a specific date/time range will send either the same or no output when they are re-run.

2. In the output/results view, click the share button to open the export options tray.
3. Select *Schedule Report*.
4. In the secondary tray, enter/configure the following details:
   - A name for the scheduled report (used in the email subject line and for scheduled report management)
   - Recipient email address(es)
   - Frequency and exact minute on the hour that the email report should be re-run and sent
5. [Optional] Tick the appropriate checkbox(es) to attach PDF and/or CSV copies of the report results.
6. [Optional] Select additional reports to include in the scheduled email.
7. Click the **Save** button to save the scheduled email report configuration.

Once set up, a scheduled report will continue to be re-run and emailed at the scheduled intervals until it is disabled or deleted.

Note

New scheduled report configurations can be created from the management view, without having to run the saved report(s) beforehand. This can facilitate setting up multiple email configurations for reports that have been previously run/saved.

To create a new scheduled report from the management view, click the add (+) button and follow the steps above, starting from step 4.

Configurations can also be modified at any time by clicking the saved report name/subject to open the settings tray.

Scheduled report management

The **Reports > Scheduled Reports** subtab is the management view for scheduled report configurations. Scheduled reports can be created, reconfigured, and deleted from this page.

The table/list shows all current scheduled email reports and includes the following information for each configuration:

- Name/email subject
- Schedule details (frequency, time, day or date)
- Expected execution/run time

- Timestamp of the last run/email
- Configured recipient email addresses

One or more filters can also be applied to show only scheduled reports that match the defined criteria.

Deleting scheduled reports

To delete one or more scheduled reports that are no longer needed, use the checkboxes in the main view to select them, and then select *Delete* from the bulk actions tray.

Scheduled reports can also be temporarily disabled by ticking the **Disable** checkbox.

Report functions

Reports can be used to further enhance network monitoring and investigative workflows through the functions described below.

Report thresholds

Report thresholds allow you to monitor key metrics from saved reports and receive alarms when specific conditions are met. This feature helps you proactively detect unusual or critical traffic patterns without constantly checking the reports manually.

When you add a threshold to a saved report, the system automatically evaluates that report every 5 minutes, checking the last 5 minutes of data, regardless of the original timeframe set when the report was saved. This ensures thresholds always reflect the most recent network activity.

> **Note**
>
> Having a large number of active *Report Threshold Violation* alarms–particularly total reports (as opposed to rate)–may result in performance issues. The total number of concurrent report processes that can be run at a time for threshold checks can also be adjusted under Admin > Settings > Reporting.

Adding a threshold

1. Run and save a report, click the gear button to open the options tray, and then select **Threshold**.
2. In the *Threshold* settings tray, select whether the threshold should be applied **per row** or to the **total of the calculated column**.
3. Select the appropriate comparison operator (>= or <=) for the desired criteria.
4. Enter the numeric threshold value and choose the unit prefix (kilo-, mega-, or giga-).
5. From the dropdown menu, select the notification profile to trigger when the threshold exceeds or falls below the specified limits.
6. Click **Save**. Scrutinizer will now evaluate this threshold every 5 minutes for the latest 5 minutes of data.

Modifying or deleting a threshold

1. Re-run the saved report, and then click the **Filters** button in the output view.
2. In the **Filters** tray, locate the report threshold.
3. Click the pencil icon to modify the threshold or click the delete (**X**) icon to delete it.

Threshold evaluation

- **Per-Row**: Threshold checks are applied per row of the saved report. For example, if the report is saved with Top 50 results, the system can evaluate up to 50 rows. If a threshold condition is met on each row, up to 50 notifications could be generated.
- **Total**: Scrutinizer sums the values in your sorted column across all rows and compares the aggregate against the set threshold.
- **Sorting Column**: The threshold always applies to the column that the report is sorting or trending on. This ensures consistency between what you see in the report and what the threshold monitors.
- **Directional Reports**:
  - **Bidirectional Reports**: Threshold checks apply only to inbound values, even though the report shows both directions.
  - **Outbound-Only Reports**: If the report is saved as outbound, threshold evaluations target outbound values exclusively.

Threshold violations and notifications

When a threshold is violated, a *Saved Report Threshold Alarm* is generated under the *Report Threshold Violation* policy in Alarm Monitor. One or more notification profiles (email, SNMP, syslog, etc.) can be assigned to the alarm. For example, if the notification profile type is email, the full report will be emailed to you at the time of the violation. For more information, see the Notification profiles section.

Report gadgets

Reports can be added to *dashboards* as gadgets, enabling continuous active monitoring of any specified network traffic/activity.

To create/configure a dashboard gadget for a report, follow these steps:

1. Run the report (new or saved).
2. In the output/results view, open the export options tray and select **Add to Dashboard** (or **Edit Gadget**, if the gadget was previously configured).
3. Enter a name for the gadget. If the report has not been saved, it will be saved under the name entered.
4. Select a dashboard to add the gadget to from the *Dashboard Tab* dropdown. Select **Don’t send to dashboard** to manually add the report gadget to dashboards at a later time.
5. In the *Type* dropdown, select whether the gadget should show the report graph only, the table only, or both.
6. [*Graph* or *Graph & Table*] Select the gadget graph type and the report column to sort by.
7. [*Table* or *Graph & Table*] Use the checkboxes to select the columns to display in the gadget table.
8. [Optional] Expand the **Display Options** section of the tray to modify the default layout and behavior of the gadget.
9. Click the **Save** button to save the gadget configuration.

After a report gadget has been configured/saved, it will be included in the list of available gadgets when *creating* or *editing* a dashboard.

> **Note**
>
> - To view a report in a dashboard, the current user must be granted access to both the report and the dashboard(s) through their user group.

- If the default gadget name for a saved report is changed, a new saved report will automatically be created under that name. If the gadget is renamed multiple times, the saved reports are still created, but only the most recent name change is applied to the gadget.

Adding reports to collections

A collection can include one or more reports (in addition to alarms, events, and/or hosts) for review by the assignee(s).

To add a report to the current active collection:

1. Run the report.
2. In the results/output view, click the star button to open the collections menu.
3. Click the button a second time (after it turns into a **+** button).

If the report was previously added to the active collection, clicking a second time (**-** button) will remove it. To add the report to a different collection, select *Manage Collections* and then set that collection as active, before following the same steps. Reports can be included in multiple collections.

Reports in collections can be re-run directly from the collection summary page.

Creating forecasts

As part of the Plixer One Enterprise platform, Scrutinizer can further leverage the data aggregated by a report to generate a forecast of future traffic/activity.

A forecast can be generated after running any report by clicking the **Save Forecast** button. It can then be viewed via the main Investigate > Forecasts page.

To learn more about creating, viewing, and managing forecasts, see this section of this documentation.

To create a new forecast, click the **Save Forecast** button in the report output/results view.

Admin

The **Admin** views of the Scrutinizer web interface are used to access the system's administrative and configuration functions.

For ease of navigation, the different admin pages/views are organized into categories in the **Admin Menu** tray, which can be accessed from any admin page/view via the three-dot button.

Hint

The Classic UI Admin page can be accessed via either the icon next to the *Admin* text in the web interface header or the *Classic Admin* link in the tray.

Admin Dashboard

The **Admin Dashboard** provides a visual overview of the functions and performance of the Scrutinizer environment. It is the default view opened when clicking on the **Admin** text in the web interface header.

This page comprises the following interactive dashboard gadgets:

System: CPU	*Displays system performance metrics in timelines or charts* Click on a metric to switch views. Click on the *Vitals* icon to view server health.
Storage: Free Disk System	*Displays available storage per collector* Click on a storage element to switch views. Click on the *Vitals* icon to view OS health.
Services: Collector	*Displays the status of system services per collector* Hover over a chart element to view additional details. Click on the *Vitals* icon to view exporter health.
Configuration Status	*Shows the overall configuration progress for Scrutinizer and can be expanded to show the detailed configuration checklist* Click on a configuration item to view its current status and accept/decline the item. Click the *Launch* icon to open the relevant documentation page for an item, or hover over the *Dependencies* icon to see other related or required configuration items.
User Activity	Shows activity for individual users in a timeline

Note

- Click the **X** button to close the expanded tables for the vitals gadgets. To collapse the configuration checklist, click the progress bar a second time.
- A configuration status dashboard gadget is also included in the default **Welcome** *dashboard* for Scrutinizer installs.

Vitals LEDs

Three notification LEDs for system vitals are persistent across all admin pages/views and can be used to monitor the general health of the Scrutinizer environment.

These LEDs correspond to the following system components/functions, from left to right:

- Server
- Software
- Exporter

Hovering over an LED will display additional details related to the component's current statuse. Each LED also functions as a shortcut to return to the admin dashboard with the corresponding vitals gadget expanded.

Admin Menu tray

The **Admin Menu** tray is the main access point for administrative functions in Scrutinizer. The tray can be opened from any admin page/view by clicking on the three-dot button.

The admin tray search field supports lookahead searching and can be used to quickly find settings, configuration views, or help descriptions that match the entered string.

Note

Admin views marked with a **[->** are still only accessible via the Classic UI of the web interface.

Settings

The **Admin** > **Settings** page provides access to global settings for Scrutinizer's core functions and behavior, organized under the subcategories listed in the table below.

Click on a setting/subcategory below to learn more:

AI Settings	Configure AI settings including AI server URL, API Key, and which model to use
Alarm Notifications	Configure global alarm message options and *Flow Inactivity* and *Interface Threshold Violation* alarm settings
Collector	Configure global collector settings and low resource fallback options
DNS	Set DNS cache retention duration and resolution attempt timeout
Data History	Set alarm and flow data history retention durations
Flow Analytics Settings	Configure *global settings* and auto-enable FlowPro Defender for appropriate algorithms
Global Authentication Settings	Configure user session and login security options (See also: user and user group settings)
Google Maps Proxy Server	Configure proxy server settings for Google Maps requests
Login Banner	Add a custom message to the Scrutinizer login page
ML AD Users	Configure Azure account info for integrating AD Users with Machine Learning (for UEBA alerts)
ML Alerts	Manage alarm thresholds for Plixer ML Engine vitals and Office 365 detection sensitivities
ML Data Limits	Set model and host/subnet limits for user and network behavior learning
ML Training Schedule	Set business hours for network behavior observation and modeling
Mapping Groups	Define and manage device groups for network mapping
Mapping Objects	Define custom map objects and manage object/group object properties
Reporting	Customize Scrutinizer reporting engine functions
System Preferences	Configure general Scrutinizer environment preferences/settings
System/New User Default	Set up default preferences/settings for new users
Thresholds	Customize color thresholds for displaying utilization

Definitions

The **Admin** > **Definitions** category contains management views for the various user-defined elements and groupings used by the Scrutinizer system.

Hint

In views that include selection checkboxes, bulk actions become available after one or more items are selected.

Click on a setting/subcategory below to learn more:

Applications	Define custom applications using IP address and port rules
Autonomous Systems (AS)	View autonomous system number assignments and activity information
Host Names	Define custom hostname-to-IP mappings and static subnet labels for reporting
IP Groups	Define rule-based IP range/subnet groups for reporting
MAC Addresses	Add and manage custom MAC address labels
Protocol Exclusions	Define protocol exclusion rules for reporting
SNMP Credentials	Manage SNMP credential sets for polling exporters in the environment
Type of Service	Add custom labels for Type of Service (ToS) and Differentiated Services Code Point (DSCP) values in reports (*ToS Family* must first be set under Admin > Settings > Reporting)
Well-Known Ports	Add and manage well-known port definitions

Note

This category includes views/pages under the **Admin** > **Definitions** tab of the Scrutinizer Classic UI.

Users & Groups

The **Admin** > **Users & Groups** category provides access to settings, options, and functions related to user management and access control.

Hint

In views that include selection checkboxes, bulk actions become available after one or more items are selected.

Click on a setting/subcategory below to learn more:

Auditing Logs	View logs of Scrutinizer web interface user actions
Authentication Providers	Add and configure third-party authentication methods/servers
Authentication Settings	Configure global options for local and third-party authentication methods
Authentication Tokens	Add and manage user authentication tokens
User Accounts	Manage user accounts and preferences
User Groups	Set up local user groups and manage access to features and resources

Integrations

The **Admin** > **Integrations** category provides access to the configuration views for the various third-party integrations that can be enabled in Scrutinizer.

Click on an integration type below to learn more:

3rd Party Integration	Enable/disable and configure *third-party integrations* for Explore > Exporters view
ASA ACL Descriptions	Add/edit ASA firewall credentials for ACL description retrieval
Email Server	Configure SMTP server settings for email notifications and reports
Flow Log Ingestion	Configure and manage flow data ingestion for *cloud resources/services*
STIX-TAXII	Add and manage STIX-TAXII threat intelligence feeds
ServiceNow	Configure and manage ServiceNow instances for incident/ticket generation via notifications and collections
Viptela Settings	Enable/disable and configure Viptela integration for Cisco vManage devices

Flow log ingestion

Scrutinizer can be configured to ingest flow logs from cloud data sources, enabling seamless visibility between on-prem and cloud-based assets.

Data sources are added from the **Admin** > **Integrations** > **Flow Log Ingestion** page as follows:

1. Click the **+** button to open the configuration tray for a new data source:
2. Select the service/type of data source to be added.
3. Enter the *required details* in the secondary tray.
4. [Optional] Click **Test** to verify that Scrutinizer can access the data source.
5. Click **Save** to save the data source configuration.

Once flows originating from a cloud data source are being ingested, any exporters reported–either as part of flow contents or in attached metadata–will be added to Scrutinizer. These devices can then be used similarly to regular exporters in Scrutinizer's functions (e.g., reports, network maps, Security Groups, etc.).

> **Hint**
>
> To delete one or more data source configurations, select them using the checkboxes and use the *Delete Integrations* option in the **Bulk Actions** tray.

For further information and additional set-up steps for specific cloud providers, see the corresponding sections below:

- *Amazon Web Services VPC flow log ingestion*
- *Azure flow log ingestion*
- *Oracle Cloud Infrastructure Streaming flow log ingestion*
- *Google Cloud Platform VPC flow log ingestion*
- *Zscaler ZIA flow log ingestion*
- *Zscaler ZPA flow log ingestion*

Alarm Monitor

The **Admin** > **Alarm Monitor** category covers the configuration and management views for functions related to events/detections and alert delivery.

Click on a settings subcategory below to learn more:

Alarm Policies	Reconfigure, enable/disable, and assign notification profiles to alarm policies
Flow Analytics Algorithms	Reconfigure, enable/disable, and add inclusions/exclusions to FA algorithms
ML Dimensions	Define traffic for the Plixer ML Engine to monitor for behavior modeling
ML Rules	Define subnet, host, or interface inclusion/exclusion rules for ML Engine observation
Notification Profiles	Create and manage profiles to assign notification actions by alarm policy
Security Groups	Create and manage IP address security groups to define FA algorithm inclusions

Reports

The **Admin** > **Reports** category includes management views for report-related functions.

Flow Report Thresholds	Manage custom report thresholds to trigger alarms and/or notifications
Report Designer	Create/manage custom report configurations
Report Folders	Create and manage folders to organize saved reports
Scheduled Email Reports	Set up and manage scheduled email report configurations

Note

Report threshold, folder, and scheduled email report management options can also be accessed from the main Reports views of the web interface.

Plixer

The **Admin** > **Plixer** options are used to access licensing and management views for Scrutinizer and other Plixer One platform components/products:

Endpoint Analytics	Configure and enable/disable Endpoint Analytics integration
FlowPro Licensing	Register a new FlowPro license key or view details for the current license
Replicator Licensing	Register a new Replicator license key or view details for the current license
Scrutinizer Licensing	Register a new Scrutinizer license key or view details for the current license

Note

- This admin category includes pages/views from the **Admin** > **Settings** section of the Scrutinizer Classic UI.
- Additional licensing may be required to enable integration with certain Plixer components. Contact *Plixer Technical Support* to learn more.

Resources

The **Admin** > **Resources** category provides access to pages/views for monitoring and managing Scrutinizer features and elements in the environment.

Click on a settings subcategory below to learn more:

Collectors	Manage Scrutinizer collectors and Plixer ML Engines in the environment
Exporters	Manage and add protocol exclusions to flow-exporting devices in the environment
FlowPro Capture Rules	Define and manage packet capture rules for FlowPro probes
FlowPro Probes	Manage FlowPro probes sending data to Scrutinizer collectors
Interfaces	Manage Scrutinizer settings and SNMP credentials for individual interfaces
ML Engines	Manage host settings for Machine Learning Engine
Replicators	Manage host settings for Replicators
SNMP Credentials	Manage SNMP credential sets for polling exporters in the environment
System Performance	View current and predicted resource utilization for individual Scrutinizer collectors

4.4.2.2 Endpoint Analytics

Endpoint Analytics enables access to the following additional endpoint details in Scrutinizer (for example, via the **Monitor > Hosts** view):

- MAC address
- Endpoint Analytics profile
- OS
- Switch port location
- Risk profile, etc.

Note

Further details are available in the Endpoint Analytics online documentation.

Configuration Guide

After setting up an Endpoint Analytics account, configure integration in Scrutinizer as follows:

1. Navigate to **Admin** > **Plixer** > **Endpoint Analytics**, and then tick the **Enable** checkbox.
2. Enter the IP address or hostname to send API requests to.
3. Enter the password to send with API requests.
4. Enter the port to use for sending API requests.
5. Use the dropdown to select the communication protocol for API requests.
6. Enter the username to send with API requests.
7. Click **Save**.

Important

Scrutinizer retains date and time data reported by Endpoint Analytics, which is based on the time zone of the account used for integration.

Troubleshooting

If there are issues with the integration, try the following steps:

- Check Scrutinizer logs for errors.
- Verify that the correct credentials were entered during configuration.

For additional assistance, contact *Plixer Technical Support*.

4.4.2.3 FlowPro

Once *registered and deployed*, FlowPro probes enable the following additional functions/features in Scrutinizer.

Note

Further details are available in the FlowPro online documentation.

Reports

The following additional report types are enabled by FlowPro:

Application Latency	Application latency reporting measures the delay or time an application takes to send a request and receive a response. It is a critical metric for assessing the responsiveness of applications.
Application Latency (old)	This report refers to historical data on application latency, providing insights into how latency has changed. Analyzing historical data can help identify trends and potential issues.
Host Jitter	Jitter is the variation in the delay of received packets. Host Jitter measures the irregularity in the packet arrival timing at the destination host. It is crucial for understanding network stability and potential performance issues.
Host Jitter By SSRC (Dst)	This report breaks down the host jitter by Synchronization Source (SSRC) at the destination. SSRC is a unique identifier assigned to each synchronization source in a multimedia session.
Hosts Latency (Dst)	Measures the latency at the destination host; it provides insights into the delay experienced by packets as they reach their destination.
Hosts Latency (Src)	Like Hosts Latency (Dst), this report measures latency at the source host. It helps in understanding the delay introduced by the source system.
Host to Host Latency	Host to Host Latency measures the overall latency between two hosts, from source to destination. It considers the complete round trip time for data transfer between the specified hosts.
Re-transmission by Application	Indicates the number of times an application has to retransmit data due to packet loss or other network issues. High re-transmission rates may suggest network congestion or unreliable connections.
Re-transmission Host to Host	Like Re-transmission By Application, this metric focuses on retransmissions between two hosts.
Top Applications	This report provides information on the network's most used or resource-intensive applications. Monitoring top applications helps identify bandwidth consumption and potential performance bottlenecks.

Alarms

The following additional flow analytics algorithms and their corresponding alarm policies are enabled by FlowPro:

BotNet Detection	Alerts for large numbers of failed unique DNS lookups
DNS Command and Control Detection	Alerts for DNS TXT messages at the network perimeter whose volume or size exceed a specified threshold
DNS Data Leak Detection	Alerts for messages with suspicious DNS names whose volume or size exceed a specified threshold
DNS Server Detection	Alerts for new DNS servers based on packet exchanges between clients and servers
Domain Reputation	Alerts for traffic associated with suspicious domains (based on a Plixer-maintained reputation list)
JA3 Fingerprinting	Alerts for suspicious encrypted traffic based on TLS handshake data and known signatures

Selective packet capture

FlowPro also enables targeted traffic sampling in Scrutinizer through custom packet capture rules. These rules can be defined from the web interface or via *API request*.

Troubleshooting

If there are issues with any FlowPro feature, try the following steps:

- Check Scrutinizer logs for errors.
- Verify that the correct credentials were entered during configuration.

For additional assistance, contact *Plixer Technical Support*.

4.4.2.4 Machine learning

Through the Plixer ML Engine, Scrutinizer is able to leverage advanced AI, machine learning, and deep learning technologies to provide real-time anomaly detection and reporting.

> **Note**
>
> To learn more about ML Engine licensing options, contact *Plixer Technical Support*.

Once set up, the engine enables the following functions in Scrutinizer:

Anomaly recognition

As it ingests data through Scrutinizer, the Plixer ML Engine builds behavior models based on the current inclusion/exclusion rules and dimensions configured. These models encompass all network activity, including applications and communications to/from external hosts.

When a sufficient volume of data has been ingested, the ML Engine is able to use models that represent typical, legitimate activity patterns as a baseline and recognize deviations that may indicate threats and other anomalies. Deviations that exceed the specified thresholds are then reported as alarms and events via the Scrutinizer web interface.

The ML Engine's detection and reporting functions can be adapted to any type of enterprise network by defining the *inclusions, dimensions, and sensitivity/threshold values* that best suit an organization's environment.

Malware detection

Because irregular behavior by itself is only indicative of a possible threat and may or may not need remediation, the Plixer ML Engine utilizes additional pre-trained ML models to classify the anomalies it observes through Scrutinizer and report whether the anomaly actually constitutes malicious activity.

> **Note**
>
> The pre-trained models packaged with the ML Engine are IP-agnostic and allow Scrutinizer to alert users to potential threats without needing previously known domain or IP-based signatures.

This classification process is divided into four steps:

1. The engine ingests flow data containing anomalous traffic streamed from Scrutinizer.
2. The data is preprocessed by the ML Engine into feature vectors that can be used by the pre-trained ML models.
3. The resulting data is used as the input for the different pre-trained ML models.
4. Each ML model outputs a probability score, which represents the likelihood that the anomaly observed constitutes malicious behavior.

Once probability scores have been obtained, Scrutinizer compares them to a user-configurable threshold to determine whether or not an alarm should be generated for the host.

> **Note**
>
> The ML Engine regularly checks for updates that may include newer versions of the pre-trained ML models it uses.

Continuous learning

To combat the growing sophistication of modern threats, the Plixer ML Engine is also equipped with deep learning capabilities that take advantage of the large quantities of flow data collected by Scrutinizer to identify complex behavioral patterns and enable advanced features, such as link prediction.

The ML Engine's deep learning-based threat detection processes can be summarized in the following steps:

1. Flow data collected by Scrutinizer is forwarded to a datastore module for preprocessing.
2. Once preprocessed, the data is forwarded to the engine, which runs it through a multi-layered neural network designed to discover behavioral patterns in the data.
3. The neural network uses the patterns to learn how devices on the network typically interact with each other.
4. After an anomaly has been detected and classified, the system uses link detection to analyze the device's interactions with other devices on the network.
5. If the deviation from what the ML Engine has learned as typical behavior exceeds a set threshold, the device involved is added to an endpoint monitoring protocol.

Devices that have been flagged for further monitoring will trigger alarms under Scrutinizer's alarm monitor, allowing security teams to decide whether immediate action is necessary.

4.4.2.5 Replicator

After a Replicator license is registered, the **Replicator** web interface page becomes available and allows users to monitor activity, manage replication parameters (profiles, policies, collectors, etc.), and enable *Auto Replicate*.

> **Note**
>
> See the Replicator online documentation for further details about Replicator's core functions and other deployment options.

On this page:

Overview

Overview

The **Overview** view is a dashboard summarizing statistics related to Replicator's functions.

Real-Time Statistics

The **Real-time Statistics** gadgets contain visualizations for the following statistics:

- Total number of exporters in profiles
- Total number of collectors in profiles
- Total number of unique collector-exporter pairs across all profiles
- Current number of profiles configured
- Average packet rate (in and out)
- Average bit rate (in and out)
- Packet rates over time (in and out)
- Bit rates over time (in and out)
- Total bits received and sent
- Total number of packets received and sent

Hovering over the packet or bit rate timeline will display details for a specific point in time.

Topology

The **Topology** gadget shows the flow of packets from exporters to their destination collectors.

Filters

When a filter is defined from the **Filters** menu in any of the Replicator UI views, the filter is applied to all views, including the Overview.

The information that can be displayed in the Overview will reflect the type of filter applied:

- If a profile filter is applied, only outbound/replicated traffic for the selected profile will be displayed. This is because inbound traffic vitals are not associated with a profile (i.e., inbound traffic can apply to no profiles, a single profile, or multiple profiles).

- If a Replicator instance filter is applied, only inbound and outbound traffic details for that instance will be displayed.

Exporters

The **Exporters** view displays all exporters currently sending packets/flows to the Replicator instance.

The list/table includes the following details for each exporter:

- IP address
- Port number
- Current status of the exporter
- Replicator instance associated with the exporter
- Timestamp when the exporter was last confirmed as available
- Total number of profiles associated with the exporter
- Total number of collectors associated with the exporter
- Timestamp when the exporter was last modified

Clicking the IP address in the main list/table opens the summary/details view for the exporter. Managing profiles and collectors associated with the exporter can also be done here.

Additional exporter actions

In the main list/table, the three-dot menu also includes shortcuts to view names, profiles, and collectors associated with the exporter.

Clicking **Add To Profiles** from the three-dot menu opens a secondary tray where you can select one or more profiles to add the exporter to.

Advanced filters

Clicking the **Filters** button opens a tray where one or more filters can be manually configured.

The following filtering options are available:

- Replicator
- Profile Name
- Exporter
- Exporter Port
- Collector
- Collector Port

To apply a filter, expand the filter option/section, and select the criteria to use. Multiple filters and criteria can be applied at the same time to further refine results.

Profiles

The **Profiles** view can be used to create, edit, and manage replication profile configurations.

The list/table shows the following details for all profiles currently saved on the Replicator instance:

- Profile name
- Replicator instance the profile was created on

- Number of policies added to the profile
- Number of exporters included in the profile
- Number of collectors assigned to the profile
- Username of the user who created the profile
- Date and time the profile was last modified

Creating a new profile

View instructions

1. Click the **+** button.
2. In the *Add Replicator Profile* tray, enter a name for the profile.
3. Select a profile type from the *Type* dropdown:

Profile types

- *IPv4 HA Dual Exporters*: Rewrites the header of IPv4 packets from a *redundant exporter pair* to show a specified IP address and port as their origin
- *IPv4 Spoofing*: Rewrites the header of IPv4 packets to show the source exporter as their origin
- *IPv6 Spoofing*: Rewrites the header of IPv6 packets to show the source exporter as their origin
- *Plixer Exporter Spoofing*: Modifies the packet header to include the origin exporter for Plixer collectors (used only in cloud environments, where conventional spoofing is not possible)
- *Auto Replicate Seed*: (Plixer One/Scrutinizer deployments only) Used to enable *autoreplicate* across one or more remote collectors in *distributed clusters*
- *Auto Replicate Collector*: (Plixer One/Scrutinizer deployments only) Used to associate collectors with the seed profile when automatic load balancing is enabled

> **Note**
>
> IPv4 spoofing profiles replicate IPv6 datagrams, while IPv6 spoofing profiles do not replicate IPv4 datagrams.

4. Select the Replicator instance to associate the profile with.
5. Add a description for the profile, and then select whether to enable or disable the profile.
6. Click **Save**.

Once created, new profiles will be added to the main **Profiles** list/table and can be further configured at a later time.

Deleting profiles

Profiles can be deleted by ticking one or more profiles in the main list/table, and then clicking **Delete** via the *Bulk Actions* button.

Editing a profile

Clicking the profile name or selecting **Edit** from the three-dot menu in the main list/table opens the profile settings tray where the following can be modified:

- Name

- Type
- Replicator
- Description
- Enable/disable the profile
- Policies added to the profile
- Exporters added to the profile
- Collectors assigned to the profile

Adding Replicator policies to a profile

View instructions

1. Select **Add Replicator Policy** from the three-dot menu in the list/table.
2. Enter the subnet/CIDR for the exporters to be defined by the policy.
3. Select whether to include or exclude the specified subnet/CIDR for the policy.
4. Click **Save**.

Assigning collectors to a profile

Select **Add Collectors** from the three-dot menu in the list/table, and then in the *Add Collectors* tray, select the collector to assign to the profile.

Additional profile actions

In the main view, the three-dot menu for profiles also includes shortcuts to view policies and collectors assigned to a profile.

Filtering options

Clicking the **Filters** button opens a tray where one or more filters can be manually configured.

The following filtering options are available:

- Replicator
- Profile Name
- Exporter
- Exporter Port
- Collector
- Collector Port

To apply a filter, expand the filter option/section, and select the criteria to use. Multiple options and criteria can be applied at the same time.

Collectors

The **Collectors** view can be used to view all collectors that are currently assigned to at least one *profile*.

The list/table includes the following details for each collector:

- IP address

- Port number
- Current status of the collector
- Replicator instance associated with the collector
- Timestamp when the collector was last confirmed as available
- Total number of profiles associated with the collector
- Total number of exporters associated with the collector

Adding a Replicator collector

To add a Replicator collector, click the **+** button, and then enter the the following details in the **Add Replicator Collector** tray:

- Collector IP address
- Port to use on the collector
- Replicator instance to associate with the collector
- [Optional] Description for the collector

Once saved, the collector will be added to the main **Collectors** list/table. The collector configuration can be further configured or modified at a later time.

Changed in version 19.7.2: Replication to internal/Plixer collectors (including *autoreplication*) is now supported with any Plixer One license type.

External collectors (as well as additional Replicator instances) can be enabled with a Replicator license key. Contact *Plixer Technical Support* for more information.

Editing a collector

Clicking the collector IP address or selecting **Edit** from the three-dot menu in the main list/table opens the collector settings tray where the following can be modified:

- IP address
- Port number
- Description
- Profiles associated with the collector
- Exporters associated with the collector

Adding a collector to profile/s

Select **Add To Profiles** from the three-dot menu in the main list/table, and then in the *Add To Profiles* tray, select one or more profiles to add the collector to.

Additional collector actions

In the main view, the three-dot menu also includes shortcuts to view profiles and exporters associated with the collector.

Advanced filters

Clicking the **Filters** button opens a tray where one or more filters can be manually configured.

The following filtering options are available:

- Replicator
- Profile Name
- Collector
- Collector Port
- Exporter
- Exporter Port

To apply a filter, expand the filter option/section, and select the criteria to use. Multiple options and criteria can be applied at the same time.

Auto Replicate

Replicator **Auto Replicate** allows flow streams to all be sent to a single Replicator instance, which will then automatically distribute them across collectors in the cluster based on their available capacity.

Auto Replicate is enabled by creating a collector profile for each destination collector and associating them with a seed profile with the necessary exporter inclusion/exclusion policies. The profiles must be created on the local Replicator instance on a Scrutinizer primary reporter or a *headless deployment* registered with the primary reporter. Destination collectors for autoreplication must be registered with the same primary reporter, which will have access to all collector configurations and current loads.

> **Note**
>
> Multiple seed profiles can be created to enable autoreplication for separate collector groups. All seed profiles are automatically discovered and processed when rebalancing and assigning exporters.

Creating a collector profile

Auto Replicate Collector profiles define the destination collectors for a seed profile. A collector profile must be created for each collector and then assigned to the seed profile for its cluster.

To create a new collector profile:

View instructions

1. Navigate to **Replicator > Collectors**, and then click the **+** icon to *create a new collector profile*.
2. In the *Add Replicator Profile* tray, enter a name for the profile.
3. Select **Auto Replicate Collector** as the profile type, and then select the Replicator instance to create the profile on.
4. Enter the collector's IP address and port number to use.
5. Enter an exporter count limit and a flow rate limit for the collector.
6. [OPTIONAL] Add a description for the collector/profile.
7. Click **Save**.

Repeat the above steps to create a collector profile for each destination collector, and then proceed to create the seed profile.

Creating a seed profile

The Auto Replicate Seed profile contains *inclusion and exclusion policies* that define the exporters/streams that should be autoreplicated.

Flows from matching exporters are sent to one of the collectors in the cluster (must be defined by *collector profiles*). Each new exporter is always assigned to the collector with the most available capacity. If a collector becomes overloaded (based on exporter count or flow rate) as a result, the exporter will be reassigned to a collector with the required capacity available.

To create a new seed profile:

View instructions

1. Navigate to **Replicator > Profiles**, and then click the **+** icon to *create a new profile*.
2. In the *Add Replicator Profile* tray, enter a name for the profile.
3. Select **Auto Replicate Seed** as the profile type in the dropdown, and then select the Replicator instance to create the profile on.
4. [OPTIONAL] Add a description for the collector/profile.
5. Click **Save**, and then return to the main **Profiles** view.
6. Click on the newly created profile to open the configuration tray.
7. Create inclusion and exclusion policies to define source exporters for autoreplication.
8. Select the collector profiles of all destination collectors to associate with the seed profile (only collector profiles not currently associated with a seed profile can be selected).
9. Click **Save**.

Once the seed profile has been configured, enabling it will start autoreplication.

> **Note**
>
> By default, new exporters/streams are assigned to collectors once every hour, and collectors are checked to verify that they are not over capacity once a day. These times can be adjusted in `/home/plixer/scrutinizer/files/conf/rebalance.yaml`. Rebalancing can also be manually initiated via the seed profile in the Replicator UI. Exporter reassignment is kept to a minimum to improve system performance.

High-availability replication

A Replicator instance can be paired with a secondary instance to create a high-availability pair that ensures uninterrupted flow data replication. Configuration data is synced between the primary and secondary instances for seamless failover.

> **Note**
>
> Any unpaired headless deployment without saved profiles can be set as a secondary instance in an HA pair.
>
> In *distributed Scrutinizer clusters*, the secondary reporter or any remote collector can be used as a secondary Replicator instance by *enabling the Replicator service*.

Multi-network mode

When the primary and secondary Replicator instances are on different subnets (i.e., a virtual IP address cannot be used), the flow data to be replicated must be sent to both Replicator instances.

After a high availability pair is set up in this mode, the primary instance continuously sends UDP heartbeat packets to the secondary instance (1 packet per second). If the secondary fails to receive two consecutive heartbeat packets, it immediately starts replication. Once a heartbeat packet is received from the primary instance again, the secondary syncs any configuration updates, stops replication, and reverts to the standby state.

> **Note**
>
> Multi-network mode is the default high-availability configuration. If the primary and secondary Replicator instances are on the same network, enabling virtual IP/single-network mode is recommended.

Enabling multi-network HA on a Replicator

To create a multi-network HA pair, follow these steps:

View instructions

1. Go to Admin > Resources > Replicators, and then click on the name of the Replicator instance to use as the primary.
2. In the Replicator configuration tray, toggle the **High Availability** switch to *On* (will not be displayed if no secondary instances are available).
3. Select the Replicator instance to use as the secondary in the *Secondary IP* dropdown.
4. Click **Save** to create the HA pair.

After the multi-network HA pair has been saved, configure all exporters to send flow data to both the primary and secondary instances.

Virtual IP mode

When the primary and secondary Replicator instances are on the same network, they can receive flow data packets via a shared virtual IP address.

After a high-availability pair is set up in this mode, the availability of the primary and secondary instances is monitored using the Virtual Router Redundancy Protocol (VRRP). If the primary Replicator instance becomes unavailable, the specified virtual IP address is immediately reassigned to the secondary instance, which then starts replication (handover typically takes ~1 second). Once the primary becomes available again, it re-assumes responsibility for the IP address and resumes replication after a user-defined delay (see instructions below).

Enabling virtual IP HA on a Replicator

View instructions

1. Go to Admin > Resources > Replicators, and then click on the name of the Replicator instance to use as the primary.
2. In the Replicator configuration tray, toggle the **High Availability** switch to *On*.
3. Select the Replicator instance to use as the secondary in the *Secondary IP* dropdown.
4. Enable *Virtual IP*, and then enter the following details in the provided fields:
    - Virtual IP address: IP address to be shared between the primary and secondary instances

- Virtual router ID: Virtual router ID to assign to the HA pair (must be unique to the pair to avoid conflicts with other software devices using VRRP)
- Failover delay: Length of time that the primary instance must be online again before it takes over the virtual IP and replication (to avoid flapping)

> **Note**
>
> The failover delay is meant to allow all services on primary instance to fully restart after a reboot/outage. A delay of at least 2 minutes is recommended (default: 5 minutes).

5. Click **Save** to create the HA pair.

After the VIP HA pair has been saved, configure all exporters to send flow data to the virtual IP address specified.

Reverting HA pairings

To revert paired Replicator instances back to single appliances, toggle off *High Availability* for the primary instance.

This will unpair the instances and allow them to be used as single deployments again. The primary instance will retain all profiles, collectors, and other settings previously applied, and the secondary instance will be reverted to its unused, post-deployment state.

High-availability exporter pairs

Replicator can automatically manage flow data streams from a specified pair of redundant IPv4 exporters using the *IPv4 Dual HA Exporters* profile type.

After a profile of this type is *created*, it must be configured as follows:

View instructions

1. Select the Replicator instance to use.
2. [Optional] Add a description for the profile.
3. Enter the spoofed IP address to use for replicated streams.
4. [Optional] Enter the spoofed port to use for replicated streams.
5. Create exactly two */32* policies (one for each HA exporter).
6. Select the preferred/primary source.
7. Set the amount of time to wait for the preferred source.
8. Add collectors to the profile (or define new collectors, if necessary).

After the profile has been configured and enabled, flow data from the preferred source will be replicated and forwarded to the specified collector(s). If the preferred source becomes inactive for the specified wait time, replication will start for the stream from the other exporter/policy defined in the profile. The same spoofed IP address and port will be used regardless of the active source.

4.5 Advanced Services

4.5.1 Administration and management

APIs

Leverage Scrutinizer APIs for external integration

APIs **Backups**

Create/restore system or config-only backups

Backups **Certificate management**

Manage and configure security certificates

Certificate management **Data migration**

Migrate configuration and historical data between Scrutinizer servers

Data migration **Database expansion**

Expand appliance database capacity

Database expansion **Interactive CLI**

Manage Scrutinizer with the scrut_util interactive command line utility

Interactive CLI **Upgrades and updates**

Install Scrutinizer and Plixer ML Engine upgrades and security patches

Upgrades and updates

4.5.2 Integrations

4.5.2.1 Log ingestion

AWS VPC logs

Enable/configure AWS VPC log data ingestion

AWS VPC logs **Google Cloud VPC logs**

Enable/configure GCP VPC log data ingestion

Google Cloud VPC logs **Microsoft Azure logs**

Enable/configure Microsoft Azure log data ingestion

Microsoft Azure logs **Oracle Cloud VCN logs**

Enable/configure OCI VCN log data ingestion

Oracle Cloud VCN logs **Zscaler ZIA logs**

Enable/configure ZIA log data ingestion

Zscaler ZIA logs **Zscaler ZPA logs**

Enable/configure ZPA log data ingestion

Zscaler ZPA logs

4.5.2.2 Network management

Cisco FireSIGHT

Security management platform integration for enhanced threat visibility

Cisco FireSIGHT **Endace**

Packet capture and analysis integration for deep network investigation

Endace **Kubernetes Flow Exporter**

(BETA) Monitoring and visibility for Kubernetes clusters

third-kubernetes **SD-WAN log ingestion**

Log data ingestion and visibility for software-defined networks

SD-WAN log ingestion

4.5.2.3 Analytics & SIEM

Grafana

Enable/configure Grafana integration

Grafana **SolarWinds**

Enable/configure SolarWinds integration

SolarWinds **Splunk**

Enable/configure Splunk integration

Splunk **STIX-TAXII**

Enable/configure STIX-TAXII integration

STIX-TAXII

4.5.2.4 Enterprise systems

PRTG

Enable/configure PRTG integration

PRTG **ServiceNow**

Enable/configure ServiceNow bi-directional integration

ServiceNow **Username reporting**

Enable user correlation via Microsoft AD or Cisco ISE

Username reporting

4.5.3 Platform extension

MCP server (Beta)

Allow other LLMs to run Scrutinizer reports

MCP server **Localization**

Define Scrutinizer UI element translations

Localization **Reverse-path filtering**

Configure reverse-path filtering and virtual routing

Reverse-path filtering **Streaming to data lakes**

Stream Scrutinizer data to data lakes

Streaming to data lakes

4.5.3.1 APIs

Scrutinizer supports API access for the following function sets:

On this page:

Capture rule configuration *Capture rule configuration* IP group management *IP group management* Host index search *Host/host-to-host index search* Reporting *Reporting* User account management *User account management*

Capture rule configuration

Selective packet capture (requires *FlowPro*) rules can be added via API, which requires the following fields:

- `authToken` - Admin authentication token generated by Scrutinizer (required for API access)
- `rm` - `flowpro_capture_rules` (runmode corresponding to the function set being accessed)
- `name` - Name to assign to the new capture rule
- `server_ip` - Packet source/server IP address or CIDR
- `client_ip` - Packet destination/client IP address or CIDR
- `max_packets` - Maximum number of packets to capture
- `stops_on` - End date/time for capturing packets as UNIX epoch timestamp
- `well_known_port` - Well-known port to monitor for packets
- `retention_hours` - Duration to store captured packet data
- `enabled` - State to add the rule in (`1`: enabled; `0`: disabled)
- `action` - `add` (adds/creates a new capture rule as defined in the request)

Request example

New rule API call

```
curl --location 'https://<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=flowpro_capture_rules' \
--form 'name="LDAP Traffic 2"' \
--form 'server_ip="1.1.1.1/32"' \
--form 'client_ip="2.2.2.2/32"' \
--form 'max_packets="1000"' \
--form 'stops_on="1743048000"' \
--form 'well_known_port="393605"' \
--form 'retention_hours="168"' \
--form 'enabled="1"' \
--form 'action=add'
```

IP group management

The following fields are required for all IP group management API requests:

- `authToken` - Admin authentication token generated by Scrutinizer (required for API access)
- `rm` - `ipgroups` (runmode corresponding to the function set being accessed)
- `action` - One or more of the following *actions* to be initiated by the request:

- *saveRule* - Creates an IP group with the specified rule
- *update* - Modifies an existing IP group
- loadTreeRootFast - Loads a condensed list of all IP group names and IDs
- *search* - Searches for an IP group by name
- loadRules - Loads a list of all rule definitions for an IP group
- *deleteRule* - Removes a rule from an IP group
- *delete* - Deletes an IP group
- deleteAll - Deletes all IP group definitions from Scrutinizer

Rule definitions

Use the following JSON object formats to pass IP group inclusion rule definitions in requests:

Single IP address(es)

```
[
 {
  "type": "ip",
  "address": "<IP_ADDRESS_1>"
 },
 {
  "type": "ip",
  "address": "<IP_ADDRESS_2>"
 }
]
```

IP address range

```
[
 {
  "type": "range",
  "sip": "<START_IP>",
  "eip": "<END_IP>"
 }
]
```

Subnet

```
[
 {
  "type": "network",
  "address": "<ADDRESS>",
  "mask": "<SUBNET_MASK>"
 }
]
```

Wildcard mask

```
[
 {
  "type": "wildcard",
  "address": "<ADDRESS>",
  "mask": "<WILDCARD_MASK>"
 }
]
```

Child group

Child groups must be created before they can be added to parent groups.

```
[
 {
  "type": "child",
  "child_id": "<CHILD_IPGROUP_ID>"
 }
]
```

All IP addresses

```
[
 {
  "type": "ipall",
  "all": 1
 }
]
```

Request examples

Below are additional details and request examples for IP group management API call `action` field.

saveRule

Creating new IP groups using the `saveRule` action requires the following additional fields:

- `new_fc` - Specifies a name for the new IP group
- `added` - Specifies a JSON array of one or more *inclusion rule definitions* to add to the new IP group

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=ipgroups' \
--form 'action=saveRule' \
--form 'new_fc=UK Data Center' \
--form 'added=[
    {
        "type": "ip",
```

(continues on next page)

(continued from previous page)

```
        "address": "10.30.10.1"
    }
]'
```

Returned JSON object

```
{
  "removed": [],
  "updated": [],
  "added": [
      {
          "rule_id": 506588,
          "cid": null,
          "type": "ip",
          "address": "10.30.10.1"
      }
  ],
  "warnings": [],
  "fc_id": 16900006,
  "myrules": "IP Address:10.30.10.1",
  "fc_name": "UK Data Center",
  "rule_id": 506588,
  "total": 1
}
```

update

Modifying existing IP groups using the `update` action requires an `fc_id` field and accepts the following optional fields for adding, replacing, or removing rule definitions:

- `name` - Replaces the current name of the IP group if included
- `added` - Specifies a JSON array of one or more *inclusion rule definitions* to add to the IP group
- `updated` - Specifies a JSON array of rules (based on the included `rule_id` field) that will be overwritten with the new definitions provided
- `removed` - Specifies a JSON array of rule IDs to be deleted

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--header 'Content-Type: application/json' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=ipgroups' \
--form 'action=update' \
--form 'fc_id=16900006' \
--form 'name=Renamed Group' \
--form 'added=[
    {
        "type": "ip",
        "address": "10.1.4.66"
```

(continues on next page)

(continued from previous page)

```
    }
]' \
--form 'updated=[
    {
        "rule_id": "84",
        "type": "ip",
        "address": "192.1.0.0"
    }
]' \
--form 'removed=[114]'
```

search

Searching IP groups using the `search` action requires the following additional fields:

- `name` - IP group name or string to search for
- `fc_name_comp` - Specifies the comparison operator to search with (`like` or `notlike`)
- `page` - Specifies the number of pages of results to load (default: one page)
- `maxRows` - Specifies the maximum number of results per page in the response

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--header 'Content-Type: application/json' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=ipgroups' \
--form 'action=search' \
--form 'name=UK' \
--form 'fc_name_comp=like' \
--form 'page=1' \
--form 'maxRows=10'
```

deleteRule

Deleting rule definitions using the `deleteRule` action requires the following additional field:

- `rule_id` - The ID of the rule definition to delete from its IP group

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--header 'Content-Type: application/json' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=ipgroups' \
--form 'action=deleteRule' \
--form 'rule_id=506588'
```

Returned JSON object

```
{
  "fc_id": 16900006,
  "success": 1,
  "myrules": "",
  "rule_id": 506588,
  "total": 0
}
```

delete

Deleting IP groups using the `delete` action requires an additional `json` field containing an array of IP group objects with `id` fields specifying the group IDs to be deleted.

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--header 'Content-Type: application/json' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=ipgroups' \
--form 'action=delete' \
--form 'json=[
    {
        "id": "16900032"
    }
]'
```

Returned JSON object

```
{
  "processedCount": 1,
  "removed": [
      "16900032"
  ]
}
```

Host/host-to-host index search

The following fields are required for host/host-to-host index search API requests:

- `authToken` - Admin authentication token generated by Scrutinizer (required for API access)
- `rm` - `quick_search` (runmode corresponding to the function set being accessed)
- `action` - `check_hosts` for host index searches or `check_host2host` for host pair index searches
- `data_requested` - An array of IP addresses (for host index searches) or sub-arrays of IP address pairs (for host pair index searches)

Note

Each request can include multiple IP addresses or pairs.

Request examples

Host index search

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=quick_search' \
--form 'view=quick_search' \
--form 'action=check_hosts' \
--form 'data_requested={
    "hosts": [
        "10.1.5.2"
    ]

}'
```

Returned JSON object

```
{
  "check_hosts": {
    "found_hosts": {
      "10.1.5.2": {
        "exporters": [
          {
            "ip": "10.30.50.10",
            "first_seen": 1613785014.59242,
            "bytes": 7876890358,
            "packets": 5523590,
            "reverse_packets": 1691102,
            "flow_count": 78469,
            "hex": "0A1E320A",
            "reverse_bytes": 193222576,
            "direction": 3,
            "reverse_flow_count": 79056,
            "last_seen": 1613799043.12985
          },
          {
            "reverse_bytes": 193935041,
            "hex": "0A1E320C",
            "last_seen": 1613799043.12985,
            "reverse_flow_count": 80011,
            "direction": 3,
            "bytes": 7877702902,
            "reverse_packets": 1693619,
            "packets": 5526105,
            "flow_count": 79507,
            "first_seen": 1613785014.59243,
            "ip": "10.30.50.12"
          }
        ],
```

(continues on next page)

(continued from previous page)

```
        "last_seen": 1613799043.12985,
        "first_seen": 1613785014.59242
      }
    }
  }
}
```

Host-to-host index search

API request

```
curl --location --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=quick_search' \
--form 'view=quick_search' \
--form 'action=check_host2host' \
--form 'data_requested={
    "host2hosts": [
        ["10.1.5.1","10.30.1.114"]
    ]
}'
```

Returned JSON object

```
{
  "check_host2host": {
    "found_hosts": {
      "10.1.5.1-10.30.1.114": {
        "first_seen": 1613785134.63046,
        "exporters": [
          {
            "reverse_flow_count": 387,
            "reverse_bytes": 47788,
            "packets": 747,
            "direction": 3,
            "flow_count": 385,
            "ip": "10.30.58.10",
            "last_seen": 1613799843.15848,
            "reverse_packets": 749,
            "first_seen": 1613785134.63046,
            "hex": "0A1E320A",
            "bytes": 165439
          },
          {
            "ip": "10.30.58.12",
            "last_seen": 1613799843.15848,
            "reverse_flow_count": 390,
            "reverse_bytes": 48172,
            "packets": 747,
            "direction": 3,
```

(continues on next page)

(continued from previous page)

```
            "flow_count": 385,
            "bytes": 165439,
            "hex": "0A1E320C",
            "reverse_packets": 755,
            "first_seen": 1613785134.63046
          }
        ],
        "last_seen": 1613799843.15848
      }
    }
  }
}
```

Reporting

The following fields are required for all IP group management API requests:

- `authToken` - Admin authentication token generated by Scrutinizer (required for API access)
- `rm` - `report_api` (runmode corresponding to the function set being accessed)
- `action` - `get` (runs the report defined in the request)
- `rpt_json` - JSON object defining the *parameters* of the report to be run
- `data_requested` - Specifies the *elements of the report* to be included in the response

Report parameters

Each report API request must specify the parameters for the report using the following elements of the `rpt_json` object:

Parameter details

Object Element/Field	Report Parameter	Available Options	Example

`reportTypeLang`	*Report type*	• `conversations`: Conversations WKP
		• `host2host`: Host to host
		• `ipGroupGroup`: IP group to IP group
		• `applications`: Applications defined
		• `country2country`: Country to country
			`"reportTypeLang":`
			`↪"conversations"`

`filters`	Exporter and interface filter	• `in_<EXPORTER_IP_HEX>_` Includes all interfaces on the specified exporter
		• `in_<EXPORTER_IP_HEX>_` Includes interface index `N`
			`"filters": {`
			`    "sdfDips_0":`
			`↪"in_0A190101_ALL"`
			`}`

`reportDirections`	Traffic directionality (relative to interfaces included)	`inbound` or `outbound`	`"reportDirections": {`
			`    "selected":`
			`↪"inbound"`
			`}`

`times`	Report time range/window and time zone to display dates in (use `scrut_util --show tzlist` for a list of valid timezones)	• `LastFiveMinutes`
		• `LastTenMinutes`
		• `LastFifteenMinutes`
		• `LastTwentyMinutes`
		• `LastThirtyMinutes`
		• `LastFortyFiveMinutes`
		• `LastHour`
		• `LastFullHour`
		• `LastThreeDays`
		• `LastSevenDays`
		• `LastThirtyDays`
		• `Today`
		• `Yesterday`
		• `Last24Hours`
		• `ThisWeek`
		• `LastWeek`
		• `ThisMonth`
		• `LastMonth`
		• `ThisYear`
		• `LastYear`
		• `Custom` (requires additional `start` and `end` fields to specify)
			`"times": {`
			`    "dateRange":`
			`↪"LastFiveMinutes",`
			`    "clientTimezone":`
			`↪"America/New_York"`
			`}`

`dataMode`	*Aggregation method* to apply to collected data	• `saf` (default)
		• `traditional` (used for legacy support)
			`"dataMode": {`
			`    "selected": "saf"`
			`}`

`rateTotal`	Selects between rate (packets/s, bits/s, etc.) or total traffic in the report output	`rate` or `total`	`"rateTotal": {`
			`    "selected": "total"`
			`}`

`dataGranularity`	Source data granu-	• `auto` (API selects an	

Response data

The data_requested field specifies how to format the graph and table of the report output.

JSON object example

```
{
    "inbound": {
        "graph": "none",
        "table": {
            "query_limit": {
                "offset": 0,
                "max_num_rows": 10
            }
        }
    }
}
```

Note

The directionality specified in the data_requested object must match the reportDirections field.

Request example

The following API call runs a default report against all interfaces of the specified device for the last 5 minutes:

API request and response details

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--header 'Content-Type: application/json' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=report_api' \
--form 'action=get' \
--form 'rpt_json=
{
    "reportTypeLang": "conversations",
    "filters": {
        "sdfDips_0": "in_0A190101_ALL"
    },
    "reportDirections": {
        "selected": "inbound"
    },
    "times": {
        "dateRange": "LastFiveMinutes",
        "clientTimezone": "America/New_York"
    },
    "dataMode": {
        "selected": "saf"
    },
    "rateTotal": {
```

(continues on next page)

(continued from previous page)

```
        "selected": "total"
    },
    "dataGranularity": {
        "selected": "auto"
    },
    "bbp": {
        "selected": "bits"
    }
}' \
--form 'data_requested=
{
    "inbound": {
        "graph": "none",
        "table": {
            "query_limit": {
                "offset": 0,
                "max_num_rows": 10
            }
        }
    }
}'
```

Returned JSON object

The following condensed response shows the typical structure of the object returned for a report API request:

```
{
    "report": {
        "request_id": "0xed184820e4b611eab58f1fc02130f7f9",
        "table": {
            "inbound": {
                "totalRowCount": 1,
                "footer": [],
                "columns": [],
                "rows": []
            }
        },
        "time_details": {},
        "exporter_details": {},
        "graph": {}
    }
}
```

Field details:

Field	Sub-field	Description
`table`(will include separate data for inbound and outbound if applicable)	`colum`	`elementName`: Name of the data element in the column`format`: Formatting details for data in the column`label`: Table header label
	`rows`	`rawValue`: Unformatted value (as returned from the database)`label`: Formatted value including bits, bytes, or percent
	`foote`	`[0]`: Represents the **Others** data for a calculated column, which is the sum of the data in all rows not included in the table`[1]`: Represents **Total** for a calculated column, which is the sum of the data in all included rows plus the **Others** value for the same column
	`totalRowCount`	Integer specifying the total number of rows available
`graph`	`all`	Includes data for all graph types
	`pie`	Values for graphing table data as a pie chart
	`timeserie`	Values for graphing table data as a line graph
	`none`	Includes only default graph (pie) data

User account management

The following fields are required for all user account management API requests:

- `authToken` - Admin authentication token generated by Scrutinizer (required for API access)
- `rm` - `user_api` (runmode corresponding to the function set being accessed)
- `action` - One or more of the following *actions* to be initiated by the request:
  - *createUser* - Creates one or more new Scrutinizer user accounts with the option to assign each to user groups
  - *delUsers* - Deletes one or more user accounts
  - *createUsergroup* - Creates one or more user groups with the option to add users to each group
  - *delUsergroups* - Deletes one or more user groups
  - *membership* - Adds and/or removes users to or from specified user groups
  - *prefs* - Edits preferences for a single user
  - *permissions* - Edits permissions for one or more user groups
  - *changeUsername* - Renames an existing user account

Request examples

Below are additional details and request examples for the user account management API call `action` field.

createUser

Creating user accounts using the `createUser` action requires an additional `json` field containing an array (`users`) of the following:

- `name` - Username for the account

- `pass` - Password for the account
- `membership` - Array of one or more user group IDs to assign the user account to

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=user_api' \
--form 'action=createUser' \
--form 'json=
{
    "users": [
        {
            "name": "NewAdmin",
            "pass": "NewAdminPassword",
            "membership": [1]
        },
        {
            "name": "NewGuest",
            "pass": "NewGuestPassword",
            "membership": [2]
        }
    ]

}'
```

Returned JSON object

```
{
    "data": [
       {
           "id": 3,
           "name": "NewAdmin"
       },
       {
           "id": 4,
           "name": "NewGuest"
       }
    ]
}
```

Note

User group IDs are stored in the `plixer.usergroups` table. By default, `1` is the administrators group and `2` is the guest users group.

delUser

Deleting user accounts using the `delUser` action requires an additional `json` field containing an array (`delUsers`) of the usernames and/or user IDs of the accounts to be deleted:

API request

```
curl --location --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--header 'Content-Type: application/json' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=user_api' \
--form 'action=delUser' \
--form 'json=
{
    "delUsers": [
        11,
        "NewGuest",
        207
    ]
}'
```

Returned JSON object

```
{
    "data": [
       "Deleting user id 11 (1 matched)",
       "Deleting user named 'NewGuest' (1 matched)",
       "Deleting user id 207 (0 matched)"
    ]
}
```

createUsergroup

Creating user groups using the `createUsergroup` action requires an additional `json` field containing an array (`usergroups`) of the following:

- `name` - User group name
- `template_usergroup` - Existing user group ID of the existing group to use as the template for the new user group
- `users` - Array of usernames or user IDs to be added to the group (if an empty array is passed, an empty user group will be created)

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=user_api' \
--form 'action=createUsergroup' \
--form 'json=
{
    "usergroups": [
        {
            "name": "My Group",
            "template_usergroup": 1,
            "users": [1,"AnotherUser"]
        },
        {
            "name": "Other Group",
            "template_usergroup": 2,
            "users": ["MyUser",2]
        }
    ]

}'
```

Returned JSON object

```
{
    "data": [
       {
           "id": 5,
           "name": "My Group",
           "members": [1,"AnotherUser"]
       },
       {
           "name": "Other Group",
           "error": "A usergroup already exists with that name"
       }
    ]
}
```

delUsergroups

Deleting user groups using the `delUsergroups` action requires an additional `json` field containing an array (`delUsergroups`) of the names and/or IDs of the user groups to be deleted.

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--header 'Content-Type: application/json' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=user_api' \
```

(continues on next page)

(continued from previous page)

```
--form 'action=delUsergroups' \
--form 'json=
{
    "delUsergroups": [
        3,
        "My User Group"
    ]
}'
```

Returned JSON object

```
{
    "data":[
       "Deleting usergroup id 3 (1 matched)",
       "Deleting usergroup named 'My User Group' (0 matched)",
    ]
}
```

membership

Editing user group membership using the `membership` action requires an additional `json` field containing `add` and/or `remove` arrays to specify the usernames/user IDs and user groups to add/remove them to/from.

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=user_api' \
--form 'action=membership' \
--form 'json=
{
    "membership":
    {
        "add": [
            {
                "user_id": 13,
                "usergroup_id": 5
            },
            {
                "user_name": "NewUser",
                "usergroup_name": "Other Group"
            }
        ],
        "remove": [
            {
                "user_name": "USER3",
                "usergroup_id": 4
            }
        ]
    }
```

(continues on next page)

(continued from previous page)

```
}'
```

Returned JSON object

```
{
    "data":
        "added": [
            "User 13 added to usergroup 5",
            "User 14 added to usergroup 3"
        ],
        "removed": [
            "User 15 removed from usergroup 4"
        ]
}
```

prefs

The `prefs` action modifies one or more user preferences for a single user account and requires an additional `json` field containing an array (`prefs`) of all preference changes.

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=user_api' \
--form 'action=prefs' \
--form 'json=
{
    "user_id": 11,
    "prefs": [
        {
            "pref": "statsTopn",
            "setting": 10
        },
        {
            "pref": "language",
            "setting": "english"
        }
    ]
}
```

Returned JSON object

```
{
    "data": {
        "updated": [
            "statusTopn updated to 10 for user_id 11",
            "language updated to english for user_id 11"
        ],
```

(continues on next page)

(continued from previous page)

```
        "errors": []
    }
}
```

permissions

The permissions action updates permissions for one or more user groups and requires an additional json field containing all user groups names/IDs and permission changes as add and remove arrays.

The following table lists all available permission_type and seccode (for use with the "plixer" permission_type) options in the request:

Permission Type	Value/Code	Description
permission_type	device	IP address of a device in hex (e.g. '0A010107')
	interface	IP address of a device in hex and the interface index separated by a hyphen (e.g. '0A010107-1')
	group	Group ID of a mapping/device group from plixer.groups
	report	saved_id of a saved report from reporting.saved_reports
	gadget	gadget_id of a dashboard gadget from plixer.dash_gadgets (e.g. 'welcomeGadget')
	thirdparty	ID of a third-party link from plixer.third_party
	plixer	Permission code corresponding to different functions/sections within Scrutinizer (see below)
seccode	3rdPartyIntegration	Create, edit, and delete third-party integration links
	ackBBEvent	Acknowledge alarms
	adminTab	Access the Admin tab/section
	alarmSettings	Configure alarm notifications
	alarmsTab	Access the Alarm Monitor tab/section
	allDevices	Access the status of all devices and their interfaces
	allGadgets	Access all gadgets created by any user
	allGroups	Access all mapping/device groups
	allInterfaces	Report on interfaces for any device
	allLogalotReports	All Logalot reports
	allReportFolders	Access all saved report folders
	allReports	Access all saved reports created by any user
	allThirdparty	Access all configured third-party links
	almDelete	Permanently delete alarms
	ApplicationGroups	Configure application groups
	asnames	Configure AS names

continues on next page

Table 2 – continued from previous page

Permission Type	Value/Code	Description
	`auditing`	Access auditing reports containing logs of Scrutinizer user actions
	`auth`	Manage external authentication tokens
	`Authentication`	Manage external authentication types
	`authLdapServers`	Manage LDAP server configuration for Scrutinizer authentication
	`awsSettings`	AWS configuration
	`changeUserPasswords`	Change passwords for other users without needing their credentials
	`createDashTabs`	Create new dashboards
	`createUsers`	Create new local Scrutinizer user accounts
	`dashboardAdmin`	Manage all dashboards created by any user
	`DataHistory`	Configure data history/retention settings
	`deleteReport`	Delete saved reports regardless of owner
	`deleteUsers`	Delete local Scrutinizer user accounts
	`DeviceDetails`	Edit device interface details
	`EmailNotifications`	Configure the mailserver for Scrutinizer reports and emails
	`faExclusions`	Configure flow analytics exclusions
	`fa_mgmt_link`	Configure flow analytics thresholds and settings
	`feedbackForm`	Access the link to send feedback to Plixer
	`FlowAnalyticsSettings`	Access global flow analytics settings
	`helpTab`	Access the Help tab/section
	`HostNames`	Edit hostname information
	`IPGroups`	Configure Scrutinizer IP groups
	`language`	Create and edit language localization settings
	`licensing`	Configure Scrutinizer product licensing and features
	`LogalotPrefs`	Configure global alarm settings
	`MACAddresses`	Configure device MAC address information
	`ManageCollectors`	Manage devices collecting flow data for Scrutinizer
	`ManageExporters`	Manage devices exporting flow data to Scrutinizer
	`mappingGroupConfiguration`	Create and edit mapping/device groups
	`mappingObjectConfiguration`	Create and edit mapping objects
	`mapsTab`	Access the Network Maps page
	`myViewTab`	Access the Dashboards page
	`NotificationManager`	Manage alarm notifications
	`PolicyManager`	Manage alarm policies

continues on next page

Table 2 – continued from previous page

Permission Type	Value/Code	Description
	`protocolExclusions`	Edit protocol exclusions for flow reports
	`proxySettings`	Configure proxy server settings in Scrutinizer
	`radiusConf`	Manage RADIUS server configuration for Scrutinizer authentication
	`ReportDesigner`	Design new custom report types
	`reportFilters`	Update the filters used in reports
	`reportFolders`	Manage saved report folders
	`reportSettings`	Reporting engine configuration options
	`runReport`	Run flow reports
	`saveReport`	Name and save flow reports
	`scheduledReports`	Create, edit, and delete scheduled email reports
	`sf_asa_acls`	Configure ASA ACL descriptions
	`SNMPCredentials`	Manage SNMP credentials for polling device information
	`srCreate`	Schedule saved reports to be emailed on a regular basis
	`sso`	Add, delete, and edit Identity Provider configurations for Single Sign-On integration
	`statusTab`	Access the Status tab
	`syslogNotifications`	Syslog server configuration
	`SystemPreferences`	Administrative access to global Scrutinizer preferences
	`tacacsConf`	Manage TACACS+ server configuration for Scrutinizer authentication
	`tos`	Edit TOS configuration
	`userAccounts`	Admin access to the user management page
	`usergroups`	Manage Scrutinizer user groups
	`viewUserIdentity`	View identity and access information relevant to GDPR restrictions
	`viptelaSettings`	Configure Viptela settings
	`Vitals`	View Scrutinizer server vitals reports
	`wkp`	Edit WKP configuration

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=user_api' \
--form 'action=permissions' \
--form 'json=
{
    "permissions":
    {
```

(continues on next page)

(continued from previous page)

```
        "add": [
            {
                "usergroup_name": "Dashboarders",
                "permission_type": "device",
                "seccode": "0A010107"
            }
        ],
        "remove": [
            {
                "usergroup_name": "ReadOnlyReporters",
                "permission_type": "plixer",
                "seccode": "allGadgets"
            }
        ]
    }

}'
```

Returned JSON object

```
{
    "data":
    {
        "errors": []
        "updated": [
            "Added device permission 0A010107 to usergroup 26",
            "Removed plixer permission allGadgets from usergroup 27"
        ]
    }
}
```

changeUsername

The `changeUsername` action is used to edit the name of an existing user account and requires an additional `json` field specifying the account (by `oldname` or `user_id`) and the new name.

API request

```
curl --location  --insecure --request POST 'https://
↪<SCRUTINIZER_ADDRESS>/fcgi/scrut_fcgi.fcgi' \
--form 'authToken=<AUTH_TOKEN>' \
--form 'rm=user_api' \
--form 'action=changeUsername' \
--form 'json=
{
    "changeUsername":
    {
        "oldname": "MyUser",
        "newname": "OpSCT"
    }

```

(continues on next page)

(continued from previous page)

```
}'
```

Returned JSON object

```
{
    "data":
    {
        "message": "User MyUser successfully renamed to OpSCT"
    }
}
```

4.5.3.2 Backups

The Scrutinizer filesystem includes utilities that automate the process of creating or restoring system backups.

> **Note**
>
> - These utilities are recommended for most long-term backup scenarios, because they include all database configuration and historical data for a Scrutinizer instance. Native snapshots may still be used as a short-term recovery option when there is no need to store the data, e.g., when upgrading the instance.
> - For Scrutinizer instances deployed on AWS, backups should be created and/or restored using native AWS functionality.

These utilities allow several types of backup and restore operations to be performed by the user.

Full backups

Full or comprehensive backups are disaster-recovery-grade images of a Scrutinizer instance and include the following elements of the filesystem:

- Application data and collected NetFlow in the PostgreSQL database
- Host index data in BadgerDB databases
- Scrutinizer's third-party encryption key - `/etc/plixer.key`
- Web Server TLS certificate and key

> **Important**
>
> - The license key (if the instance is a primary reporter) and the TLS certificates and keys generated by Scrutinizer are **not** backed up and **cannot be restored**.
> - Any files not included in full backups must be manually backed up and restored, including:
>   - Custom threat lists created under `/home/plixer/scrutinizer/files/threats`
>   - Custom notifications created under `/home/plixer/scrutinizer/files`
>   - LDAP authentication certificates

Creating full backups

The Scrutinizer filesystem includes the `backup.sh` utility, which automates the creation of full backups. This script is located under `home/plixer/scrutinizer/files`.

> **Note**
>
> The default runmode of `backup.sh` *saves the backup file locally*. Due to the size of full backup files, however, the remote method outlined below is recommended.

The following instructions cover the process of creating and saving full Scrutinizer instance backups to a specified remote host:

View instructions

1. SSH to the Scrutinizer server to be backed up and start a tmux session to prevent timeouts:

   ```
   tmux new -s backup
   ```

2. Allow others to use FUSE mounts:

   ```
   sudo grep -Eq "^user_allow_other" /etc/fuse.conf || \
   sudo sed -i '$ a user_allow_other' /etc/fuse.conf
   ```

3. Create the backup directory locally and mount it to an empty directory on the remote host:

   ```
   BACKUPDIR=/mnt/backup
   sudo mkdir -p $BACKUPDIR
   sudo chown plixer:plixer $BACKUPDIR
   sshfs -o allow_other -o reconnect REMOTE_USER@REMOTE_HOST:REMOTE_DIRECTORY
   ↪$BACKUPDIR
   ```

   > **Important**
   >
   > Verify that the remote directory to be used is empty and there is sufficient storage available, before running the backup script in the next step. For a rough estimate of the backup file size, run the following on the Scrutinizer instance:
   >
   > ```
   > df -h /var/db | awk '!/^Filesystem/ {print "Space Required: "$3}'
   > ```

4. Run `backup.sh` as the `plixer` user, with the mounted remote directory set as the backup file location:

   ```
   BACKUPDIR=/mnt/backup ~plixer/scrutinizer/files/backup.sh
   ```

5. Once the script confirms that the backup file has been saved, unmount the remote backup directory:

   ```
   BACKUPDIR=/mnt/backup
   fusermount -u $BACKUPDIR
   sudo rmdir $BACKUPDIR
   ```

Full backup files are created as `scrutinizer-VERSION-backup-DATE.tar.gz` at the specified location and owned by the `plixer user`.

Note

- A *second Scrutinizer instance* can be used as the remote backup host, provided it has sufficient disk space available and is running the same Scrutinizer version as the instance to be backed up. However, doing so is only recommended for redundancy.
- If no remote hosts are available, backups can be *saved locally on the same Scrutinizer instance*. However, this will limit the amount of storage available for system functions and is not recommended.

For further details or assistance with issues, contact *Plixer Technical Support*.

Backing up additional files

When creating a full backup of a Scrutinizer server, any files not *covered by the script* must be manually backed up and should be stored on an external host/system.

These files should also be manually restored, after running the *restore script*.

Restoring from a full backup

To restore a Scrutinizer instance from a full backup file, use the `restore.sh` utility located under `home/plixer/scrutinizer/files`.

The script will fully restore *all backed up elements* of a Scrutinizer instance, provided the following conditions are met:

- A valid full backup file is accessible by the `plixer` user at the specified ($BACKUPDIR) remote location.
- The Scrutinizer instance to be used for the restore has been freshly deployed.
- The version of the backup matches the version of the fresh Scrutinizer instance to restore *to* (e.g. a 19.3.0 backup can only be restored to a new 19.3.0 instance).

Important

- A restore completely overwrites the state of the target instance and deletes the source backup file. It is highly recommended to always restore from a **copy** of a backup file.
- If the restore target is the primary reporter in a distributed cluster, contact *Plixer Technical Support* for assistance.

The following instructions cover the process of restoring from a backup file on a remote host to a fresh Scrutinizer deployment:

View instructions

1. SSH to the target Scrutinizer server for the restore, and start a tmux session to prevent timeouts:

```
tmux new -s restore
```

2. Allow others to use FUSE mounts:

```
sudo grep -Eq "^user_allow_other" /etc/fuse.conf || \
sudo sed -i '$ a user_allow_other' /etc/fuse.conf
```

3. Create the backup directory locally and mount the remote directory containing the backup file(s):

```
BACKUPDIR=/mnt/backup
sudo mkdir -p $BACKUPDIR
sudo chown plixer:plixer $BACKUPDIR
sshfs -o allow_other -o reconnect REMOTE_USER@REMOTE_HOST:REMOTE_DIRECTORY
↪$BACKUPDIR
```

4. Run `restore.sh` as the `plixer` user, with the remote directory set as the backup file location:

   ```
   BACKUPDIR=/mnt/backup ~plixer/scrutinizer/files/restore.sh
   ```

5. When prompted, enter `yes` to select the backup file to use for the restore or `no` to have the script continue searching (if the backup file was not previously specified).

   > **Hint**
   >
   > To specify the file to use for the restore, use `BACKUPDIR=/mnt/backup BACKUP=restore_filename.tar.gz ~plixer/scrutinizer/files/restore.sh` at the previous step instead.

6. Once the script confirms that the restore has been completed, unmount the remote backup directory:

   ```
   BACKUPDIR=/mnt/backup
   fusermount -u $BACKUPDIR
   sudo rmdir $BACKUPDIR
   ```

> **Important**
>
> The `restore.sh` utility does not restart Scrutinizer services after it completes running.

Based on the role of the Scrutinizer instance, proceed to finalize setup of the restored server:

- If the restored instance is a **standalone server**, run the following to restart all services and register it:

  ```
  scrut_util --services --name all --switch restart
  scrut_util --set selfregister --reset
  ```

  These commands may take several minutes to complete.

- If the restored instance is a **remote collector** in a *distributed cluster*, run the following on the primary reporter to register it:

  ```
  scrut_util --set registercollector --ip RESTORED_INSTANCE_IP
  ```

- If the restored instance is a **primary reporter** in a distributed cluster or a standalone server, and its Machine ID is different from that of the backup file, contact *Plixer Technical Support* to obtain a new license key.

Alternative backup methods

Because full backup files are extremely large and intended for use in disaster recovery scenarios, saving and storing backup files to remote hosts serving ssh is highly recommended.

In scenarios where this is not possible, the following alternative backup methods can be used:

Backup to a second Scrutinizer instance

If a separate host is not available to save backups to, a second Scrutinizer instance can be used for backup file storage instead. The versions of the two instances must match.

> **Important**
>
> Due to how Scrutinizer is designed to optimize the use of all available disk space, it will likely be necessary to add more storage and/or modify the data retention settings of the second instance. For assistance, contact *Plixer Technical Support*.

The following instructions cover the additional steps required for creating backups on a second Scrutinizer instance (using the default location):

View instructions

1. Set the location/directory to use for backup files:

```
BACKUPDIR=${BACKUPDIR:='/var/db/big/pgsql/restore'}
REMOTE=YOUR_REMOTE_SCRUTINIZER_INSTANCE
```

2. Create the backup directory on both instances:

```
sudo mkdir -p $BACKUPDIR
sudo chown plixer:plixer $BACKUPDIR
ssh plixer@$REMOTE "sudo su -c 'mkdir -p $BACKUPDIR && chown plixer:plixer
↪$BACKUPDIR'"
```

3. Allow other users to use FUSE mounts:

```
sudo grep -Eq "^user_allow_other" /etc/fuse.conf || \
sudo sed -i '$ a user_allow_other' /etc/fuse.conf
```

4. Mount the remote instance's backup directory on the local instance:

```
sshfs -o allow_other -o reconnect plixer@$REMOTE:$BACKUPDIR $BACKUPDIR
```

5. Run `backup.sh` using the directory mounted from the remote instance as the backup file location:

```
BACKUPDIR=/var/db/big/pgsql/restore ~plixer/scrutinizer/files/backup.sh
```

> **Important**
>
> Before running the backup utility, verify that the remote directory to be used is empty and there is sufficient storage available. For a rough estimate of the backup file size, run the command `df -h /var/db | awk '!/^Filesystem/ {print "Space Required: "$3}'` on the Scrutinizer instance.

6. After the backup is complete, unmount the remote Scrutinizer directory:

```
BACKUPDIR=${BACKUPDIR:='/var/db/big/pgsql/restore'}
fusermount -u $BACKUPDIR
```

Local backups

By default, both `backup.sh` and `restore.sh` are set to use `/var/db/big/pgsql/restore` on the local Scrutinizer filesystem for full backup files. However, in most cases, the backup operation will likely fail unless additional disk space is allocated to or created on the Scrutinizer instance. Running the command `df -h /var/db | awk '!/^Filesystem/ {print "Space Required: "$3}'` will provide a rough estimate of the storage required for the backup.

To force a checkpoint, enter `psql plixer -c "CHECKPOINT"` after the script has finished running.

Note

- In v19.2, the backup file path must be defined in the `backup.sh` and `restore.sh` scripts before they are run.
- Storing backup files locally will severely limit the storage Scrutinizer can use for its primary functions. As such, backup files saved to the instance should be transferred to a separate resource as soon as possible.

Configuration backups

For more "lightweight" backup and restore operations, the `scrut_conf_dump.sh` and `scrut_conf_restore.sh` scripts (both located in `/home/plixer/scrutinizer/database/utils`) can be used to target only the application/configuration data of a Scrutinizer instance, including:

- User-added maps
- Dashboards
- IP groups
- Saved reports
- 3rd-party integration settings

Configuration backups do not include any collected flow data.

Note

In *distributed clusters*, the primary reporter regularly syncs application/configuration data to remote collectors. Only the configuration backup of the primary reporter is needed to perform a restore for the cluster.

`scrut_conf_dump.sh` and `scrut_conf_restore.sh` use Postgres's pg_dump and pg_restore utils and respect the same set of environment variables:

Variable	Description	Default
`DUMP`	Location of the backup file	`./conf.dump`
`PGHOST`	IP address or hostname of the PostgreSQL database	`localhost`
`PGUSER`	Role/user used to connect to PGHOST	`plixer`
`PGDATABASE`	The database to access at PGHOST	`plixer`

Backing up configuration data

To create a backup of a Scrutinizer server's current configuration data, follow these steps:

View instructions

1. Run the backup script.

   To save the backup file to the default location:

   ```
   ~/scrutinizer/database/utils/scrut_conf_dump.sh
   ```

   To use a custom location/filename:

   ```
   mkdir /tmp/CONF_BACKUP_DIR
   touch /tmp/CONF_BACKUP_DIR/CONF_BACKUP.dump
   DUMP=/tmp/CONF_BACKUP_DIR/CONF_BACKUP.dump ~/
   ↪scrutinizer/database/utils/scrut_conf_dump.sh
   ```

2. Restart the stopped services:

   ```
   sudo systemctl restart scrutinizer
   ```

Restoring configuration data

To restore configuration data to a Scrutinizer server from a backup file, follow these steps:

View instructions

1. Stop the `plixer_webapp` and `plixer_collector` services:

   ```
   sudo systemctl stop plixer_webapp
   sudo systemctl stop plixer_collector
   ```

2. Run the restore script.

   To restore from the default backup location/file:

   ```
   ~/scrutinizer/database/utils/scrut_conf_dump.sh
   ```

   To restore from a specified location/file:

   ```
   PGHOST=SCRUTINIZER_IP
   DUMP=/tmp/CONF_BACKUP_DIR/CONF_BACKUP.dump ~/
   ↪scrutinizer/database/utils/scrut_conf_restore.sh
   ```

3. Restart the stopped services:

   ```
   sudo /bin/systemctl start plixer_webapp
   sudo /bin/systemctl start plixer_collector
   ```

4. Resync the access table:

   ```
   psql -c "SELECT setval(pg_get_serial_sequence('plixer.access', 'access_id'),␣
   ↪COALESCE(max(access_id) + 1, 1), false) FROM plixer.access;"
   ```

Note

`scrut_conf_restore.sh` should only be used for restoring configuration data for the same Scrutinizer server/appliance. Follow *this guide* to migrate configuration data from one Scrutinizer server to another.

Additional notes

- `pg_restore` errors typically only cause the restore to fail for the table associated with the error. Other tables should still be restored successfully.
- Errors associated with **duplicate keys** usually indicate a conflict between existing rows in the table and the rows being restored.

  ```
  pg_restore: [archiver (db)] Error from TOC entry 51348; 0 17943 TABLE DATA␣
  ↪exporters plixer
  pg_restore: [archiver (db)] COPY failed for table "exporters": ERROR:  duplicate␣
  ↪key value violates unique constraint "exporters_pkey"
  DETAIL:  Key (exporter_id)=(\x0a4d4d0a) already exists.
  ```

  The conflicting keys should be removed from the table before attempting to restore again.

- If you are swapping IP addresses, the database keys should be rotated using `scrut_util --pgcerts --verbose`, because the backed up keys will be associated with the old address.

4.5.3.3 Certificate management

This section contains additional instructions/guides related to certificate management in Scrutinizer.

On this page:

Certificate rotation and regeneration *Certificate rotation and regeneration* Wildcard certificates *Wildcard certificates* Full chain certificates *Full chain certificates* Distributed cluster certificates *Distributed cluster certificates* For further information or assistance with these functions, contact *Plixer Technical Support*.

Certificate rotation and regeneration

The following certificate rotation utilities can be run to re-issue certificates and keys to address database communication issues:

Operations and syntax

Regenerate all certificates on all nodes, including any ML engines, with an optional expiration of the number of `DAYS` specified. If the `--reset` flag is included, the CA and web server certificates on the primary reporter will also be regenerated.

```
scrut_util --rotatecerts [--days <DAYS>] [--reset] [--verbose]
```

Regenerate the web server certificate and key with an optional expiration of the number of `DAYS` specified. The `--csr` option can be used to create a certificate signing request (CSR) using the current private key and user-configured subject in `/home/plixer/scrutinizer/files/scrutinizer.csr` instead.

```
scrut_util --webappcerts [--days <DAYS>] [--csr] [--verbose]
```

Generate a new self-signed certificate (`/etc/pki/tls/certs/ca.crt`), private key (`/etc/pki/tls/private/ca.key`), and CSR (`/etc/pki/tls/private/ca.csr`) with default details; must be run from the *scrut_util prompt*.

```
SCRUTINIZER> set ssl on
```

To run the operation from the shell with user-provided details, use:

```
scrut_util --set ssl --toggle [on|off] --port <TCP_PORT> --country <COUNTRY> --
↪state <STATE/PROVINCE> --city <CITY/LOCALITY> --org <ORG_NAME> --email
↪<CONTACT_EMAIL> --name <COMMON_NAME> --keysize [1024|2048|4096]
```

Regenerate TLS certificates and private keys on the Plixer ML Engine node with the specified `IP_ADDRESS` with an optional expiration of the number of `DAYS` specified. If the `--install` option is included, `setup.sh --reload-certs` will not be executed on the node.

```
scrut_util --mlcerts --ip <IP_ADDRESS> [--days <DAYS>] [--install] [--verbose]
```

Regenerate all certificates used for PostgreSQL connections on all nodes with an optional expiration of the number of `DAYS` specified. If the `--reset` option is included, the CA certificate on the primary reporter will also be regenerated.

```
scrut_util --pgcerts [--days <DAYS>] [--reset] [--verbose]
```

Note

- The optional `DAYS` flag can be used to set an expiration date for the certificate(s) regenerated by each utility. Once they expire, the same command can be run again to re-issue certificates with new expiry dates.
- With the exception of `set ssl on`, the above commands *cannot* be run from the `SCRUTINIZER>` prompt.

Wildcard certificates

If a signed wildcard certificate and key were generated with a passphrase, the passphrase must be removed from the private key to allow Scrutinizer to use the pair.

1. Copy the private key file (`*.key`) to `/etc/pki/tls/private/`.
2. Re-generate the key without a passphrase (replace `ORIGINAL` with the filename of the key):

   ```
   openssl rsa -in /etc/pki/tls/private/ORIGINAL.key -out /
   ↪etc/pki/tls/private/new.key
   ```

3. When prompted, enter the passphrase used for the original key.

This will create a new, unencrypted key named `new.key` in `/etc/pki/tls/private/`, which must be renamed to `ca.key`. If the key pair was originally created without a passphrase, it need only be renamed after being copied into the correct directory.

Full chain certificates

A full chain certificate or chain of trust can be created as follows:

1. Create the file `ca_chain.crt` under `/etc/pki/tls/CA/`.
2. Copy the contents of the intermediate CA `.crt` file into `ca_chain.crt`.
3. Copy the contents of the root CA `.crt` file into `ca_chain.crt` (after the intermediate CA).
4. Set `ssl_certificate` setting in `/etc/nginx/webapp.d/inc/ssl.conf`:

```
ssl_certificate /etc/pkl/tls/CA/ca_chain.crt
```

5. Restart the web server:

```
sudo systemctl restart plixer_webapp
```

After the restart, the full chain certificate will be in use.

Distributed cluster certificates

To generate CSRs and install the signed keys for a distributed cluster, run the following scripts:

Note

- These scripts should be run from the distributed cluster's primary reporter as the `plixer` user and rely on Scrutinizer's default SSH connectivity.
- `scrut_util --rotatecerts --reset` (see above) can be used if either of these scripts causes unexpected issues or DB connection errors. However, *any existing signed certificates will be lost*.

`/` `↪home/plixer/scrutinizer/files/ge`	Generates certificate signing requests (CSRs) for all TLS keys in a distributed cluster *CSRs are saved to subdirectories in* `/tmp/request` *with* `webapp_plixer_client` *being the signing request for the primary reporter's web server.*
`/` `↪home/plixer/scrutinizer/files/ir`	Installs signed TLS certificates to all nodes in a distributed cluster `.cer` *files should be saved to* `/tmp/signed` *following the path and filename conventions used by* `generate_requests.sh` *for the signing requests. The Certificate Authority's root certificate should be saved as* `ca.cer`.

4.5.3.4 Data migration

The `scrut_util --migrate` command can be used to migrate configuration and/or historical data from source Scrutinizer server to a destination server.

Note

Because the steps outlined here can potentially result in an irrecoverable state for the source and/or destination Scrutinizer server, it is highly recommended to contact *Plixer Technical Support* for assistance. In case of issues, the migration utility logs (`/var/log/migrate.log`) can be provided to help with troubleshooting.

Source and destination server support

The migration utility includes separate runmodes for configuration data migration and historical data migration.

The following table shows supported migration modes between Scrutinizer versions:

View table

	19.5.x destination	19.6.x destination	19.7.x destination
18.20 source	Data	Data	Data
19.4.0 source	Configuration and data	Configuration and data	Configuration and data
19.5.x source	Configuration and data	Configuration and data	Configuration and data
19.6.x source		Configuration and data	Configuration and data
19.7.0 source			Configuration (19.7.1+ only) and data

Important

Configuration data **cannot** be migrated from a 19.7.0 source to a 19.7.0 destination.

Note

- If a Scrutinizer 18.20 server is *upgraded to v19.4.0*, its configuration data can be migrated to servers running v19.5.0 and above.
- The utility may report **pg_restore** errors after cross-server configuration migrations. These errors can safely be ignored, and any issues are addressed by re-running the installer as described in these *configuration migration instructions*.

Source and destination server setup

Follow the steps below to prepare source and destination Scrutinizer servers for migration.

Source server

Run the following on the source server to create a temporary migration role and allow connections from the destination server using the role:

Note

Replace `ROLE_PASSWORD` below with the desired password for the temporary migration user/role `migrator`. The password will also be required to prepare the destination server for the migration (see below).

```
DESTINATION=DESTINATION_IP
psql -c "CREATE USER migrator WITH SUPERUSER ENCRYPTED PASSWORD 'ROLE_PASSWORD'"
sudo sed -i -e "1ihost plixer migrator $DESTINATION/32 md5" /
↪var/db/big/pgsql/data/pg_hba.conf
psql -c "SELECT pg_reload_conf()"
```

Important

For Scrutinizer 18.20 source servers, use the following commands instead:

```
DESTINATION=x.x.x.x
psql -Uroot plixer -c "CREATE USER migrator WITH SUPERUSER ENCRYPTED PASSWORD
↪'ROLE_PASSWORD'"
sudo sed -i -e "1ihost plixer migrator $DESTINATION/32 md5" /
↪var/db/big/pgsql/data/pg_hba.conf
psql -Uroot plixer  -c "SELECT pg_reload_conf()"
```

Destination server

Before starting the migration, the destination server must be *deployed, licensed*, and already collecting/receiving data.

Important

- If the source and destination servers are collecting data from the same exporters prior to the migration, reports can be run against the migrated data after the migration.
- Configuration migrations will overwrite the destination server's configuration.
- It is highly recommended to run any type of migration within a tmux session, as the operation may take some time.
- The flow collection service should be running during data migrations.

After verifying that the destination is collecting data, update `/etc/migrate.ini` with the details of the source server as well as the migration role's username (`migrator`) and password (`ROLE_PASSWORD` entered when the role was created).

Configuration migrations

Configuration migrations transfer application configuration and user preference data from the source Scrutinizer server to the destination.

After the source and destination servers have been set up, follow these steps to perform a configuration (and optional historical data) migration:

View instructions

1. Start a new tmux session on the destination server:

   ```
   tmux new -s migration
   ```

2. Transfer the source server's encryption key, pause collection services, and start the migration:

   ```
   SOURCE=SOURCE_SERVER_IP
   scp $SOURCE:/etc/plixer.key /etc/plixer.key
   sudo systemctl stop plixer_flow_collector
   scrut_util --migrate config
   ```

3. Once the migration is complete, reinstall the package for the destination server's current Scrutinizer version and restart the flow collection service:

```
PKG=$(rpm -qa plixer-scrutinizer_pg\*)
sudo yum reinstall -y $PKG
sudo systemctl restart scrutinizer
```

4. Log in to the web interface and verify that all configuration data and user preferences have been successfully migrated.

5. [Optional] Migrate historical data from the source server to the destination:

   1. Run the utility to migrate historical data:

      ```
      scrut_util --migrate data
      ```

   2. After the data migration is complete, return to the web interface and run one or more reports (time ranges must cover migrated data) to verify that the operation was successful.

6. Remove the hba rule and temporary migrator role on the source server:

   ```
   sudo sed -i '/migrator/d' /var/db/big/pgsql/data/pg_hba.conf
   psql -c "SELECT pg_reload_conf()"
   psql -c "REASSIGN OWNED BY migrator TO plixer"
   psql -c "DROP ROLE migrator"
   ```

Note

- Configuration migrations only include Scrutinizer application data and not the system configuration, e.g., if a custom listening port was added via the Scrutinizer admin menu, that setting will be migrated, but the corresponding port will not be opened on the system. In such cases, undo and re-apply the setting in Scrutinizer after the migration to update the configuration.
- The distributed cluster configuration on the destination Scrutinizer server are retained by default. These settings can also be migrated (while still preserving the destination server's IP address and machine ID) by including the `--force_dist` option as follows:

  ```
  scrut_util --migrate config --force_dist
  ```

- To extend the utility to also migrate contributions to the Scrutinizer SQL codebase, update `/home/plixer/scrutinizer/database/utils/scrut_conf_dump.sh` to include them.

Historical data migration

If only historical data needs to be migrated, follow these steps after completing the *setup steps for the source and destination servers*:

Important

- The collector service (`plixer_flow_collector`) should be running on the destination server before and during the migration. The data migration process may take some time, depending on the amount of flow data to be migrated.
- If the source and destination servers are collecting data from the same exporters prior to the migration, reports can be run against the migrated data after the migration.

View instructions

1. Start a new tmux session on the destination server:

```
tmux new -s migration
```

2. Run the migration utility:

```
scrut_util --migrate data
```

3. Confirm to start the migration when prompted.
4. After the data migration is complete, log in to the web interface and run one or more reports (time ranges must cover migrated data) to verify that the historical data was successful.
5. Remove the hba rule and temporary migrator role from the source server:

```
sudo sed -i '/migrator/d' /var/db/big/pgsql/data/pg_hba.conf
psql -c "SELECT pg_reload_conf()"
psql -c "REASSIGN OWNED BY migrator TO plixer"
psql -c "DROP ROLE migrator"
```

4.5.3.5 Database expansion

This section contains guides for adding disk drives to Plixer appliances.

Scrutinizer

Follow the steps below to configure a Scrutinizer virtual appliance to use an additional drive.

> **Note**
>
> For assistance with expanding storage on a hardware appliance, contact *Plixer Technical Support*.

1. Attach the new drive to the Scrutinizer appliance/VM.
2. Log in to the virtual appliance as the `plixer` user.
3. Launch the interactive utility (*scrut_util*):

```
scrut_util
```

4. Inspect and take note of the current size of the database mounted on `/var/db`:

```
show diskspace
```

5. Identify the drive that was added:

```
show partitions
```

6. Make the new drive available to the virtual appliance:

```
set partitions <NEW_PARTITION>
```

7. When prompted, select whether or not a backup is available.

Once the operation is complete, confirm that the drive has been added successfully by running `show diskspace` again and verifying that the new database size includes the drive that was added.

ML Engine

Follow the steps below to extend a volume on a Plixer ML Engine appliance.

1. Add/attach a new hard disk to the hardware appliance or VM, and then restart the machine.
2. Navigate to Admin > Resources > ML Engines in the web interface and wait for the engine's deployment status to switch to *Deployed*.

   > **Note**
   >
   > The ML engine can take up to 30 minutes to fully restart when under heavy load. Refresh the **ML Engines** page every few minutes until the engine is shown as *Deployed*.

3. Log in or SSH to the host using the credentials `plixer:plixer`.
4. Determine the device name of the new disk (usually `/dev/sdb`):

   ```
   lsblk
   ```

5. Extend the volume that requires additional disk space:

   ```
   /home/plixer/ml/tools/mladmin.sh --extend <DEVICE> <VOLUME>
   ```

   Where `DEVICE` is the device name of the new disk and `VOLUME` is one of the following:

   - `root` - root partition
   - `sibyl` - models partition (`/SibylData`)
   - `db` - database partition (`/var/db`)
   - `zookeepers` - Kafka ZooKeeper partition (`/var/kafka/zookeepers`)
   - `brokers` - Kafka brokers partition (`/var/kafka/brokers`)

When done, the selected partition will be extended by the full capacity of the newly added disk.

4.5.3.6 Interactive CLI

The Scrutinizer interactive CLI utility provides access to system-level functions, such as admin operations, configuration/maintenance routines, and integration management.

The interactive prompt (`SCRUTINIZER>`) is accessed by establishing an SSH session with the Scrutinizer server and running:

```
scrut_util
```

On this page:

Configuration & settings *Configuration & settings* Data collection & processing *Data collection & processing* Data management & maintenance *Data management & maintenance* Importing & exporting data *Importing & exporting data* Network & monitoring *Network & monitoring* System management *System management* Third-party integrations *Third-party integrations* User & security management *User & security management*

Note

Most **scrut_util** commands can also be executed using direct shell syntax, which allows them to be used in scripts to automate maintenance tasks. Run the following from the shell to view the equivalent syntax for the top-level interactive commands listed below:

```
scrut_util --help [COMMAND]
```

Configuration & settings

convert

The `convert` command is used to convert different types of data and information.

Command	Description
`converttoaes`	Converts all encrypted information stored by Scrutinizer to use AES 256 encryption

disable

The `disable` commands are used to disable specific functions/features in Scrutinizer.

Note

These commands can alter Scrutinizer functionality and should be used with caution.

Command	Description
`disable baseline <IP_ADDRESS>`	Disables all baselines for the exporter with the specified `IP_ADDRESS` *Historical data associated with the exporter will not be deleted but will still expire following the configured data retention settings.*
`disable elk http://<IP:PORT>`	Disables ELK flows from Scrutinizer to the URL specified by `IP:PORT`
`disable ipv6`	Disables IPv6 for all interfaces in `sysctl.conf`
`disable splunk http://<IP:PORT>`	Disables Splunk flows from Scrutinizer to the URL specified by `IP:PORT`
`disable ssh_root_login`	Prohibits the superuser root account from logging into a Linux shell directly from outside hosts *Instead of allowing remote root SSH login, it is recommended to instead log in as the `plixer` user and use `sudo` for maintenance tasks. This command will not affect root logins from a physical or virtual console.*
`disable unresponsive`	Disables pinging of exporters that have been flagged as unresponsive
`disable user <USERNAME>`	Disables the specified `USERNAME` account with scrut_util access (e.g., for server maintenance)

enable

The `enable` commands are used to enable/configure specific functions in Scrutinizer.

Note

These commands can alter Scrutinizer functionality and should be used with caution.

Command	Description
`enable baseline <IP_ADDRESS> default`	Enables default baselines for the exporter with the specified `IP_ADDRESS`
`enable baseline <IP_ADDRESS> manual <PRIMARY[, SECONDARY] ELEMENT avg|count|min|max|std|sum dailyhr|busday|sameday>`	Enables a custom baseline with the following parameters for the exporter with the specified `IP_ADDRESS`: • `PRIMARY` - IPFIX element to be included in the baseline (e.g., sourceIPv4Address, applicationName, etc.) • `SECONDARY` - Optional secondary IPFIX element to be included in the baseline • `ELEMENT` - Corresponding numeric IPFIX element for the primary and secondary elements to be used to determine the baseline (e.g., packetDeltaCount, octetDeltaCount, etc.) • `AVG` | `COUNT` | `MIN` | `MAX` | `STD` | `SUM` - Selects between average (`AVG`), flow count (`COUNT`), minimum value (`MIN`), maximum value (`MAX`), standard deviation (`STD`), or sum (`SUM`) for measuring the specified `ELEMENT` • `dailyhr` | `busday` | `sameday` - Selects between daily (`dailyhr`), daily on business days (`busday`), or same day weekly (`sameday`) for baseline comparison *When baselining IP addresses, IP groups should be defined for the address ranges and subnets to be included in the baseline. This will prevent addresses that may only talk once from triggering false positives.*
`enable elk http://<IP:PORT>`	Enables ELK flows from Scrutinizer to the URL specified by `IP:PORT`
`enable ipv6`	Enables IPv6 for all interfaces in `sysctl.conf`
`enable splunk http://<SPLUNK_SERVER_IP:PORT> <SYSLOG_PORT> <SPLUNK_FORWARDER_IP>`	Enables Splunk integration using the provided server and forwarder details
`enable ssh_root_login`	Allows the superuser root account to log into a Linux shell directly from outside hosts *Instead of allowing root SSH login, it is recommended to instead log in as the `plixer` user and use `sudo` for maintenance tasks.*
`enable user <USERNAME> <1|2|3>`	Creates a new login account with the specified `USERNAME` and one of the following security levels: • `1` - Only commands that can stop data collection are disabled. • `2` - Commands that can remove integrations or stop data collection are disabled. • `3` - Only commands to collect information about Scrutinizer and the operating system are enabled.

set

The `set` commands are used to manage settings/behaviors related to authentication, networking, and general operation for the Scrutinizer server.

Note

These commands can alter Scrutinizer functionality and should be used with caution.

Command	Description

`set columnmoniker <OLD_NAME>`
`↪<NEW_NAME> [ELEMENT_LIST]`
	Replaces an information element's `OLD_NAME` with the specified `NEW_NAME`
	If the optional `ELEMENT_LIST` of one or more elements (comma-delimited) is included, renaming will be limited to flow templates that also include those elements.
	This command should only be run under the direction of Plixer Technical Support.

`set dns`	Allows use the user to enter one or more new DNS servers for hostname resolution *The operation will overwrite the system's previous DNS server list.*
`set hostinfo <IP_ADDRESS> <FQHN>`	Assigns the specified `FQHN` (fully qualified hostname) to the current Scrutinizer appliance and configures resolution for the provided `IP_ADDRESS`
`set leds_threshold`	Resets the LED warning threshold to 10% of the total storage available on the appliance's data partition *When combined with the Auto History Trimming settings, this function can help prevent Scrutinizer from using up all available storage.*

`set myaddress <IPv4_ADDRESS>`
`↪<NETMASK> <GATEWAY>`
`set myaddress`
`↪<IPv6_ADDRESS/CIDR> <GATEWAY>`
	Assigns the specified `IPv4`/`IPv6_ADDRESS`, `CIDR`/`NETMASK`, and `GATEWAY` to the current appliance
	After the provided IP information has been confirmed to be correct, the previous address of the same type will be overwritten.
	Because an SSH session will automatically be terminated after the new IP address is assigned, it is recommended to run this command from a console connection.

`set partitions <PARTITION>`
`↪<extend>`
	Extends the specified `PARTITION` to expand OS diskspace for the current hardware or virtual (requires the `extend` flag) appliance
	It is highly recommended to create a backup *before running this command.*

`set password plixer`
`set password webui <USER_NAME>`
	Resets the password for the `plixer` OS user/account or the web interface account with the specified `USER_NAME`

`set registercollector`
`↪<IP_ADDRESS> [secondary]`
	Registers the Scrutinizer appliance with the specified `IP_ADDRESS` as a remote collector and, if the `SECONDARY` flag is included, as the *secondary reporter* for the *distributed cluster*
	This command must be run from the distributed cluster's primary reporter/server.

`set salt <SALT>`	Adds the specified `SALT` value to the current appliance's machine details for license key generation
`set selfregister [reset]`	Reinitializes the server and, if the `reset` flag is included, resets all *appliance settings*
`set selfreporter`	Promotes the *secondary reporter* in a *distributed cluster* to the *primary reporter* role *This command must be run on an appliance that was assigned the secondary reporter role (see `registercollector` above).*
`set sshcollectorkeys`	Generates a new SSH key pair and distributes it to all registered appliances *The operation will also overwrite any previous key pairs, which will address any issues that require re-syncing of SSH access.*
`set ssl [ecc]`	Reconfigures Scrutinizer's web server self-signed SSL certificate *If the `ecc` argument is included, a 256-bit Elliptical Curve (EC) public/private key pair will also be generated.* *For further details on Scrutinizer's default SSL settings and behavior, see the* SSL configuration guide.
`set timezone <TIMEZONE>`	Sets the Scrutinizer appliance's time zone to the specified `TIMEZONE` *For a list of time zones, use the* show timezone *command.*
`set tuning`	Re-tunes the appliance by modifying certain Scrutinizer and OS settings in `sysctl.conf`, `postgresql.conf`, and `plixer.ini` as

show

The show commands are used to view various details, settings, and other functional elements for the Scrutinizer server/environment.

Command	Description
`show datasize`	Displays a breakdown of database storage sizes by schema
`show diskspace`	Displays storage allocation and utilization details
`show dns`	Displays a list of all DNS servers used for hostname resolution
`show exporters [FILTER]`	Displays a list of exporters sending data to collectors (using the specified FILTER if included)
`show groups`	Displays a list of all current device/mapping groups
`show interfaces [FILTER]`	Displays a list of interfaces sending data to collectors (using the specified FILTER if included)
`show ipaddresses`	Displays all IP addresses assigned to the current Scrutinizer appliance
`show metering [FILTER]`	Displays a list of interfaces by exporter and their metering direction (using the specified device IP address FILTER if included)
`show partitions`	Displays partition information for the current Scrutinizer appliance
`show task [FILTER]`	Displays a list of all tasks currently configured in Scrutinizer (using the specified task name FILTER if included)
`show timezone`	Displays the timezone configured for the current Scrutinizer appliance
`show tzlist [FILTER]`	Displays a list of timezones that can be configured for the Scrutinizer appliance (via the *set timezone* command)
`show unknowncolumns`	Displays a list of exporter information elements that are unrecognized by Scrutinizer *Contact* Plixer Technical Support *for any information elements that you need supported.*
`show yum_prox`	Displays the current yum proxy settings *To edit these settings, use the* set yum_proxy *command.*

Data collection & processing

check

The check commands can be used to run a check/test against the resource, setting, or function specified by the option used.

Note

The collector should be stopped before running any of the `history_index` commands.

Command	Description
check activeif	Checks for active flows based on interface details and returns the last timestamp and number of interfaces that received flows
check collectorclass <CLASS [SUBSYSTEM]>	Returns running state details for the specified collector CLASS or, if provided, the specified SUBSYSTEM of that class *This command is used by Plixer Technical Support for troubleshooting.*
check data_last_written	Returns activity details for collected flow data written to the database
check dist_info	Returns distributed cluster configuration details for the Scrutinizer server
check hdtest [TRIES]	Tests hard drive performance by running a write-delete operation either 10 times (default) or, if provided, the number of times specified by the TRIES parameter and returns details for the amount of time taken
check heartbeat <database|api>	Test and returns information on internal communications with the specified resource type
check history_index	Checks the history index and returns historical activity information for the 1m interval aggregation table
check history_index_empty_tables	Checks the history index and returns a list of tables with zero rows (collector should be stopped first) *To delete empty tables, use the delete command instead.*
check history_index_orphans	Checks the history index and returns a list of entries for which a table does not actually exist *To delete orphan entries, use the delete command instead.*
check history_index_table_orphans	Checks the history index and returns a list of tables that do not have a history index entry (collector should be stopped first) *To delete orphan tables, use the delete command instead.*
check interfaces [all|cisco|sonicwall|huawei [HOST_IP]]	Uses alternative methods to retrieve interface descriptions (SNMP for Huawei and NetFlow data for Cisco and SonicWall) on the specified HOST_IP *This operation leverages NetFlow data for Cisco and SonicWall devices. Checking Huawei devices relies on SNMP and referencing their vendor-specific MIBs instead.*
check license	Returns license details for the Scrutinizer server
check machine_id	Returns the current Machine ID of the Scrutinizer server
check machine_id_list	Returns all previous, current, and possible Machine IDs for the Scrutinizer server
check rollcall	Checks the current states of data roll-up time buckets and returns a list of states and record counts by bucket
check rollups	Checks the current states of all data roll-ups and returns a list of roll-up counts by status
check route <DEVICE_IP>	Checks the specified DEVICE_IP to determine if its routing data is accessible and returns the result
check simplercv <UDP_PORT>	Checks for UDP traffic on the specified <UDP_PORT> *This command can be used to verify that flows are being received at the top of the stack (i.e., tcpdump -> collector).*
check snmp	Attempts to get *SysObjectID* for all devices and returns the credential object if successful (or an error if the attempt failed)
check ssl	Returns the current settings for SSL parameter *To enable/disable SSL or edit the configuration, use the set ssl command.*
check stats_exporters	Returns an exporter activity time log
check task <TASK_ID>	Returns a list of execution times and error codes for the specified TASK_ID *For a list of all task IDs, use the show task command.*
check tuning	Checks the current system configuration and returns a list of settings that can be modified to improve performance

collect

The `collect` commands are used to manually execute collection processes for data that can be used in various Scrutinizer functions. Many of these processes are run automatically at regular intervals.

Command	Description
`collect asa_acl`	Immediately polls Cisco ASA devices to collect ASA ACL information
`collect baseline`	Collects baseline data and checks for alarms/events
`collect dbsize`	Collects database size information
`collect elk <IP_ADDRESS>`	Collects data from Scrutinizer and forwards it to the ELK server with the specified `IP_ADDRESS`
`collect optionsummary`	Initiates processing of flow option data collected by Scrutinizer
`collect snmp`	Immediately polls SNMP devices to collect data used by Scrutinizer
`collect splunk <IP_ADDRESS>`	Collects data from Scrutinizer and forwards it to the Splunk with the specified `IP_ADDRESS`
`collect supportfiles`	Collects various logs and configuration data that can be used by *Plixer Technical Support* for troubleshooting
`collect topology`	Collects device data to help Scrutinizer understand the network's topological layout
`collect useridentity`	Initiates processing of user identity data collected by Scrutinizer

Data management & maintenance

clean

The `clean` commands are used to manually execute housekeeping processes that are automatically run at regular intervals.

Command	Description
`clean all`	Immediately executes all scheduled housekeeping tasks
`clean baseline`	Resets all configured baselines to the default values *Historical data will not be deleted but will still expire following the configured data retention settings.*
`clean database`	Purges all temporary database entries
`clean ifinfo`	Purges *ifinfo* entries that do not have matching entries in *activeif*
`clean old_logs`	Purges old log files that are set to the *backup* status
`clean tmp`	Purges all temporary files created by the graphing engine

delete

The `delete` commands are used to delete database entries or tables from the Scrutinizer system.

Note

- These commands will permanently delete data and should be used with caution.
- The collector should be stopped before running any of the `history_index` commands.

Command	Description
`delete history_index_empty_tables`	Deletes all tables with zero rows from the history index
`delete history_index_orphans`	Deletes all history index entries for which a table does not actually exist
`delete history_table_orphans`	Deletes all tables that do not have a history index entry

expire

The `expire` commands are used to delete expired historical data following the configured data retention settings.

Note

These commands will permanently delete data and should be used with caution.

Command	Description
`expire dnscache [all]`	Purges expired DNS cache data (based on the *Days of DNS Request Data* setting) or, if the `all` option is included, all DNS cache data
`expire history [trim]`	Purges expired flow data (based on *Flow Historical X Avg* settings) and also deletes older data until the *Minimum Percent Free Disk Before Trimming* is reached if the `trim` option is included
`expire inactiveflows`	Removes expired inactive interfaces (based on the *Inactive Expiration* system preference setting) from interface views
`expire templates`	Purges flow template metadata for templates that haven't been observed for 30 days

optimize

The `optimize` commands are used to manually execute optimization processes that are automatically run at regular intervals.

Note

These commands will modify database tables in Scrutinizer and should be used with caution.

Command	Description
`optimize common`	Optimizes tables that are commonly inserted and deleted to improve database performance
`optimize database <DATABASE>`	Optimizes only tables in the specified `DATABASE`

repair

The `repair` commands are used to run various repair processes related to Scrutinizer functions and databases.

Note

These commands will modify database tables in Scrutinizer and should be used with caution.

Command	Description
`repair business_hour_saved_reports`	Converts saved reports with business hours that were created in older Scrutinizer versions (15.5 and below) to the latest format with the same business hours
`repair history_tables`	Repairs history tables that have the wrong *col* type for `octetDeltaCount` *This command is not used for PostgreSQL installations.*
`repair policy_priority_order`	Repairs irregularities in alarm policy IDs (e.g., duplication)
`repair range_starts`	Repairs history tables without the start time used to identify the range of data they contain *This repair process may take some time to complete and should only be executed under the direction of* Plixer Technical Support.

Importing & exporting data

export

The `export` commands are used to dump data from Scrutinizer for external use.

Command	Description
`export applications ↪<PATH/FILENAME>`	Exports all current application rules/definitions as a CSV file with the specified `PATH` and `FILENAME`
`export ipgroups <PATH/FILENAME>`	Exports all current IP group rules/definitions as a CSV file with the specified `PATH` and `FILENAME`
`export langtemplate <LANG_NAME>`	If `LANG_NAME` keys are defined, creates a CSV file with the English and `LANG_NAME` keys and saves it as `home/plixer/scrutinizer/files/pop_languages_LANGNAME_template.csv`

import

The `import` commands are used to import various types of data (labels, definitions, groupings, etc.) for use in Scrutinizer's functions.

For further information, see *this guide on importing data*.

upload

The `upload supportfiles` command is used to upload the log and configuration data package (after running the *collect supportfiles* command) for use by *Plixer Technical Support*.

Command	Description
`upload supportfiles`	Uploads the log and configuration data package for use by Plixer Technical Support

Network & monitoring

remove

The `remove address ipv6` command is used to delete the current IPv6 address assigned to the server.

Note

- The IPv6 address can only be removed if there is an IPv4 address assigned. To edit IP address settings, use the *set myaddress* command.
- This command will alter Scrutinizer functionality and should be used with caution.

Command	Description
`remove address ipv6`	Deletes the current IPv6 address assigned to the server

snoop

The `snoop` commands are used to listen for traffic at the interface level.

Command	Description
`snoop interface <INTERFACE> ↪<PORT>`	Listens for traffic on the specified `INTERFACE` and `PORT`
`snoop ipaddress <IP_ADDRESS> ↪<PORT>`	Listens for traffic on the specified `IP_ADDRESS` and `PORT`

System management

services

The `services` command is used to `stop`, `start`, or `restart` all services.

```
services all <stop|start|restart>
```

Managing individual services

To stop, start, or restart specific services, run one of the following from the shell instead:

Service	Function
`sudo systemctl <start|stop|restart> scrutinizer`	Start, stop, or restart all Scrutinizer services
`sudo systemctl <start|stop|restart> plixer_collector`	Start, stop, or restart all data collection and processing services
`sudo systemctl <start|stop|restart> plixer_webapp`	Start, stop, or restart all Scrutinizer UI and API-related services
`sudo systemctl <start|stop|restart> plixer_db`	Start, stop, or restart all database, connection pooling, and caching services
`sudo systemctl <start|stop|restart> replicator`	Start, stop, or restart all Replicator-related services when it is licensed to operate on the same machine as Scrutinizer

system

The `system` command is used to reboot or shut down the system.

Command	Description
`system <restart|shutdown>`	Reboots or shuts down the system

version

The `version` command is used to show version information for Scrutinizer.

Command	Description
`version`	Shows version information for Scrutinizer

Third-party integrations

awssync

The `awssync` command can be used to sync IDs and descriptions from AWS when AWS flow log ingestion is enabled.

Command	Description
`awssync`	Syncs IDs and descriptions from AWS when AWS flow log ingestion is enabled

ciscoise

The `ciscoise` commands are used to manage Cisco Identity Services Engine (ISE) node integration in Scrutinizer.

Command	Description
`ciscoise add <IP_ADDRESS TCP_PORT ISE_USER>`	Adds a Cisco ISE node with the specified `IP_ADDRESS`, `TCP_PORT`, and `ISE_USER` (must have API access) to queue to acquire user identities for all active sessions *The `ISE_USER` password will also need to be entered after this command is run.*
`ciscoise check`	Tests node polling and returns the results *This command can be used to verify that Scrutinizer is able to collect user identity information.*
`ciscoise kick <ISE_ID IP_ADDRESS [MAC_ADDRESS]>`	Kicks the `ISE_ID` off the ISE node at the specified `IP_ADDRESS` and optional `MAC_ADDRESS`, forcing re-authentication
`ciscoise nodelist`	Returns a list of all Cisco ISE nodes currently configured
`ciscoise poll`	Forces a poll of all Cisco ISE nodes and returns the results
`ciscoise remove <IP_ADDRESS>`	Removes the Cisco ISE node with the specified `IP_ADDRESS` from Scrutinizer
`ciscoise update <IP_ADDRESS TCP_PORT ISE_USER>`	Updates the current configuration of the Cisco ISE node with the specified `IP_ADDRESS` to use the provided `TCP_PORT` and `ISE_USER` *The `ISE_USER` password will also need to be entered after this command is run.*

endace

The `endace` commands are used to manage EndaceProbe for *Pivot2Packets (P2P) integration*.

Command	Description
`endace add <IP_ADDRESS> <PORT> <USER> <PASSWORD>`	Enable integration with an EndaceProbe with the specified `IP_ADDRESS`, `PORT`, and Endace `USER:PASSWORD` *The default port used by an EndaceProbe is `443`.*
`endace remove <IP_ADDRESS>`	Remove the EndaceProbe with the specified `IP_ADDRESS`
`endace update <IP_ADDRESS> <PORT> <USER> <PASSWORD>`	Update EndaceProbe integration settings with the specified `IP_ADDRESS`, `PORT`, and Endace `USER:PASSWORD`

> **Note**
>
> The above commands will only accept an IP address. Hostnames will not work.

Hint

- More than one EndaceProbe can be configured for P2P integration. All probes added will be available in a dropdown menu in the P2P search.
- *Pivot2Vision integration* can be configured to use a separate EndaceProbe (or probes) from the probe(s) added via the scrut_util CLI for P2P integration.

moloch

The `moloch` command is used to enable or disable integration for the Moloch probe using the specified `IP_ADDRESS` and `PORT`.

Command	Description
`moloch <on|off> <IP_ADDRESS [PORT]>`	Enables or disables integration for the Moloch probe using the specified `IP_ADDRESS` and `PORT`

User & security management

rotate

The `rotate` commands are used to replace the keys and certificates used by Scrutinizer in its functions.

Note

- These commands will alter Scrutinizer functionality and should be used with caution.
- `rotatecerts` can only be run using direct shell/script syntax and not from the `SCRUTINIZER>` prompt (as shown below).

Command	Description
`rotatekeys`	Creates a new encryption key and re-encrypts all encrypted fields in the database
`scrut_util --rotatecerts [--days <DAYS>] [--reset] [--verbose]`	Regenerates all certificates on all nodes (including any Plixer ML Engine deployments) with an optional expiration date in the specified number of `DAYS` *If the `--reset` flag is included, the CA certificate on the primary reporter and the web server certificate will also be regenerated.*

unlock

The `unlock` command is used to unlock a locked `USER` account (due to failed login attempts).

If no authentication method is specified (`ldap`, `radius`, or `tacacs`) the account defaults to local authentication.

<table>
<tr><th>Command</th><th>Description</th></tr>
<tr><td><code>unlock &lt;USER> [ldap|radius|tacacs]</code></td><td>Unlocks a locked <code>USER</code> account (due to failed login attempts)</td></tr>
</table>

4.5.3.7 Platform extension

The additional configuration options below are supported for platform extension

On this page:

MCP server **(BETA)** *MCP server* Reverse-path filtering *Reverse-path filtering* Streaming to data lakes *Streaming to data lakes* Localization *Localization*

MCP server

The Plixer MCP service allows an MCP host application's integrated LLMs to leverage *Scrutinizer reporting APIs* (via the `scrutinizer_report` tool) to run network traffic reports. The server also provides access to full documentation for the APIs (via the `reporting-api-docs` resource).

The MCP server supports both **stdio** (for direct MCP integration) and **HTTP** transport with Server-Sent Events (SSE) for real-time event streaming.

View instructions

Enabling the MCP server

To enable the Plixer MCP server on a Scrutinizer host, follow these steps:

1. Navigate to **Admin** > **Users & Groups** > **Authentication Tokens** in the Scrutinizer web interface and create an authentication token with API access.
2. Set the following environment variables in `/usr/lib/systemd/system/plixer_mcp.service`:

   ```
   Environment=API_SERVER_HOST=<PRIMARY_REPORTER_IP>
   Environment=MCP_ACCESS_TOKEN=<API_AUTH_TOKEN>
   ```

   > **Note**
   >
   > The MCP service can be hosted on any Scrutinizer server in a distributed cluster, including the primary reporter.

3. Reload systemd to update the environment details:

   ```
   sudo systemctl daemon-reload
   ```

4. Start the MCP service:

   ```
   sudo systemctl start plixer_mcp
   ```

5. Verify that the MCP server is running:

   ```
   sudo systemctl status plixer_mcp
   ```

Once the MCP service has been started, the server can be added to an external MCP host application.

Client configuration

To add the Plixer MCP server to the MCP host application, follow these steps:

1. Download `https://files.plixer.com/PlixerMCP.tar` and extract `mcp-proxy.js` to a location that can be accessed by the MCP host
2. Update the host application's settings json with the Plixer MCP server's address (`MCP_SERVER_HOST`) and the API authentication token (`MCP_ACCESS_TOKEN`).

   For example, for Cline in VS Code running from WSL:

   ```
   "mcpServers": {
     "scrutinizer": {
       "type": "stdio",
       "command": "node",
       "args": [
       "/PATH/TO/mcp-proxy.js"
       ],
       "env": {
       "MCP_SERVER_HOST": "SCRUTINIZER_WITH_MCP_SERVICE_IP",
       "MCP_ACCESS_TOKEN": "SCRUTINIZER_WITH_MCP_SERVICE_AUTH_TOKEN"
       }
     }
   }
   ```

After the MCP server has been added, the MCP host application's integrated LLMs will have access to the APIs and documentation for running report types and filters that support AI prompts.

Reverse-path filtering

When reverse-path filtering is enabled, a Scrutinizer collector is able to receive flows from IP addresses that it is unable to route to normally, such as non-local hosts whose traffic data is forwarded by a proxy or replication appliance.

This configuration should only be used when the Scrutinizer server/collector is both **in a secure environment** and **using a single interface**.

Important

In multi-interface/multi-homed scenarios and/or where strict networking practices are observed, the recommendations in RFC 3704 should be followed. This ensures that spoofed/forged packets cannot be used to generate responses that are sent out over a different interface.

Enabling reverse-path filtering

To enable reverse-path filtering on a Scrutinizer collector, find the following line in `/etc/sysctl.conf`:

```
net.ipv4.conf.default.rp_filter = 1
```

And change its value from `1` to `0`.

In addition, the following steps are also recommended:

- To bypass having to restart networking after editing the file, enable reverse-path filtering by running the command:

  ```
  sysctl net.ipv4.conf.default.rp_filter = 0
  ```

- Verify that the routing tables include routing data for all networks to be monitored to ensure that flows can be collected from non-local address spaces.

VRF (Virtual Routing and Forwarding) Mode

In some scenarios, such as when there are special security requirements or if the management network IP addresses overlap with collection-side interfaces, routing tables may need to be isolated from the management network.

Separate routing tables can be created to isolate management traffic to the management interface, so collection and polling traffic only impact their respective interfaces.

Sample routing table configuration

This example outlines the steps to configure two separate routing tables called `plixer` and `public` corresponding to interfaces `eth0` and `eth1` on a Scrutinizer deployment.

1. Add the two routing tables to `/etc/iproute2/rt_tables` after the line `#1 inr.ruhep`:

```
#
# reserved values
#
255 local
254 main
253 default
0 unspec
#
# local
#
#1 inr.ruhep
1 public
2 plixer
```

2. Create the files `route-eth0` and `route-eth1` under `/etc/sysconfig/network-scripts/` containing the following lines to define the default gateway for each table:

   `route-eth0`

```
default via 172.16.2.20 table plixer
```

   `route-eth1`

```
default via 10.1.1.251 table public
```

3. Add the gateway for each interface in `/etc/sysconfig/network-scripts/ifcfg-eth0` and `ifcfg-eth1` (no other changes are necessary) as follows:

   `ifcfg-eth0`

```
DEVICE="eth0"
BOOTPROTO="none"
HWADDR=""
NM_CONTROLLED="yes"
ONBOOT="yes"
BOOTPROTO="none"
PEERDNS=no
TYPE="Ethernet"
```

(continues on next page)

(continued from previous page)

```
NETMASK=255.255.255.0
IPADDR=172.16.2.7
GATEWAY=172.16.2.20
```

ifcfg-eth1

```
DEVICE="eth1"
BOOTPROTO="none"
HWADDR=""
NM_CONTROLLED="yes"
ONBOOT="yes"
BOOTPROTO="none"
PEERDNS=no
TYPE="Ethernet"
NETMASK=255.255.0.0
IPADDR=10.1.4.190
GATEWAY=10.1.1.251
```

4. Reboot the server to restart networking.
5. Verify that networking is functioning and confirm that IP tables are configured to accept or deny the correct traffic on each interface.

Streaming to data lakes

Scrutinizer supports data streaming to customer data lakes.

For assistance with the configuration process, contact *Plixer Technical Support*.

Localization

Scrutinizer supports translation of the web interface for localization purposes.

To add or modify translations of UI elements:

1. Navigate to **Admin > Settings > System/New User Defaults > Language**.
2. Select a language from the dropdown menu.
3. Click on a key type to enter or modify the translation for that UI element.
4. Repeat the process to translate additional UI elements.

Language translations are saved as `/home/plixer/scrutinizer/files/localize_languageName.xls`.

4.5.3.8 Third-party integrations

Log ingestion

AWS VPC logs

Enable/configure AWS VPC log data ingestion

AWS VPC logs **Google Cloud VPC logs**

Enable/configure GCP VPC log data ingestion

Google Cloud VPC logs **Microsoft Azure logs**

Enable/configure Microsoft Azure log data ingestion

Microsoft Azure logs **Oracle Cloud VCN logs**

Enable/configure OCI VCN log data ingestion

Oracle Cloud VCN logs **Zscaler ZIA logs**

Enable/configure ZIA log data ingestion

Zscaler ZIA logs **Zscaler ZPA logs**

Enable/configure ZPA log data ingestion

Zscaler ZPA logs

Network management

Cisco FireSIGHT

Security management platform integration for enhanced threat visibility

Cisco FireSIGHT **Endace**

Packet capture and analysis integration for deep network investigation

Endace **Kubernetes Flow Exporter**

Monitoring and visibility for Kubernetes clusters **(BETA)**

third-kubernetes **SD-WAN log ingestion**

Log data ingestion and visibility for software-defined networks

SD-WAN log ingestion

Analytics & SIEM

Grafana

Enable/configure Grafana integration

Grafana **SolarWinds**

Enable/configure SolarWinds integration

SolarWinds **Splunk**

Enable/configure Splunk integration

Splunk **STIX-TAXII**

Enable/configure STIX-TAXII integration

STIX-TAXII

Enterprise systems

PRTG

Enable/configure PRTG integration

PRTG **ServiceNow**

Enable/configure ServiceNow bi-directional integration

ServiceNow **Username reporting**

Enable user correlation via Microsoft AD or Cisco ISE

Username reporting

AWS VPC logs

When AWS Virtual Private Cloud (VPC) flow log ingestion is configured and enabled, Scrutinizer can report additional insights for network traffic destined for AWS, including top AWS users and applications, as well as traffic load generated by AWS-hosted applications.

The following AWS-specific report types become available to run:

AWS report types

Report Type	Description
Action	Aggregation based on the *Action* (`ACCEPT`, `REJECT`, or `DROP`) associated with the traffic
Action with Interface	Aggregation based on the action applied and the interface associated with the flow
Action with Interface and Dst	Aggregation based on the action applied, the associated interface, and the traffic's destination
Action with Interface and Src	Aggregation based on the action applied, the associated interface, and the traffic's source
Availability Zones	Aggregation based on the AWS Availability Zone associated with the traffic
Dst Service	Aggregation based on the AWS service the traffic was destined for
Interface	Aggregation based on the source or destination interface associated with the traffic
Pair Interface	Aggregation based on the source and destination interfaces associated with the traffic
Pair Interface Action	Aggregation based on the *Action* applied and the source and destination interfaces of the traffic
Src Service	Aggregation based on the AWS service the traffic originates from
Src Service-Dst Service	Aggregation based on AWS services the traffic originated from and was destined for
Traffic Path	Aggregation based on the traffic path used by egress traffic to reach its destination
VPCs	Aggregation based on the VPC ID associated with the traffic

This section covers the prerequisites and setup/configuration steps for AWS VPC flow log ingestion.

Setting up the AWS S3 storage bucket

Before setting up AWS VPC flow log ingestion in Scrutinizer, the Amazon S3 storage bucket(s) that will be used should be configured as follows:

- Set the VPC(s) to be monitored to send flow logs to the bucket. The flow log format *must include* the following fields:
  - `log-status`
  - `vpc-id`
  - `interface-id`
  - `flow-direction`
- The bucket should be reserved for exclusive use by Scrutinizer. If the flow logs need to be archived or used for other purposes, send the flow logs to a separate S3 bucket, and then automate the replication/duplication of those logs to the bucket that will be used by Scrutinizer.
- Versioning should be disabled.

Note

- When upgrading from older versions of Scrutinizer, it may be necessary to delete the existing VPC flow log configuration and create a new one that includes the `interface-id` and `flow-direction` fields.
- When creating a VPC flow log, leaving the *Maximum Aggregation Interval* setting at the default 10 minutes will minimize processing load on the Scrutinizer collector at the cost of longer update times and data spikes. Setting the maximum aggregation interval to 1 minute will result in more granular data but also increase resource utilization.
- After an S3 bucket is first configured for ingestion, Scrutinizer purges all older flow logs from the bucket before starting to collect and delete the most recent 15 minutes of logs as normal. If any historical data needs to be retained, it should be copied off the bucket before the integration is configured. Manually clearing the bucket of any log data older than 15 minutes will also allow Scrutinizer to become current more quickly.

Configuring AWS VPC flow log ingestion in Scrutinizer

To add an S3 bucket as a flow log ingestion source in Scrutinizer, follow these steps:

1. In the Scrutinizer web interface, navigate to **Admin** > **Integrations** > **Flow Log Ingestion**.
2. Click the **+** icon and select *AWS VPC FlowLogs* in the tray.
3. In the secondary tray, configure the bucket details as follows:
   - Enter a name to identify the bucket/source by.
   - Select the Scrutinizer servers to use as log downloader(s) and collector(s) for the bucket (in *distributed clusters*, remote collectors are recommended for these roles).
   - Enter the name of the bucket.
   - Select the AWS region where the bucket is hosted from the dropdown.
   - Enter the credentials to use to access the bucket (AWS access key ID and secret access key; must have full access to the bucket)
4. Click the **Save** button to add the bucket with the current settings.

Once added, the bucket will be listed in the main Admin > Integrations > Flow Log Ingestion view under the configured name. An exporter associated with the VPC will also be added to the device lists for Scrutinizer's various functions (reports, network maps, etc.).

Note

- After a bucket configuration has been saved, click on the name assigned to it in the main view to open the settings tray, and use the **Test** button to confirm that Scrutinizer is able to establish a connection to the bucket with the credentials entered.
- To verify that an AWS VPC flow log source has been successfully added, look for an exporter labeled `vpc-` in the **Explore** > **Exporters** > **By Exporters** view or the **Admin** > **Resources** > **Exporters** page (after ~1 hour).
- Flow log ingestion processes are divided between the *log downloader* (downloads the flow logs from the bucket) and the *flow collector* (collects and processes the downloaded logs). A different Scrutinizer server can be used for each role, and a single bucket can have multiple downloaders and collectors.

Troubleshooting

If the **Admin** > **Resources** > **Exporters** view does not list exporters matching the virtual network(s) set up for flow ingestion, check the following for issues:

- Open the tray for the ingestion source in the **Admin** > **Integrations** > **Flow Ingestion** view and use the **Test** button to verify that the collector/downloader is able to communicate with the data source using the details entered.
- Verify that logs are correctly being sent to the source bucket.
- Check the collector log file in `/home/plixer/scrutinizer/files/logs/` for errors.
- Check `awss3_log.json` for possible source-side issues.

For further assistance, contact *Plixer Technical Support*.

Overloaded collectors/downloaders

The *Unresourced - Enabled* status in the Admin > Resources > Exporters view indicates that a log source is being temporarily disabled/paused due to insufficient resources.

The following are potential solutions for an overloaded collector:

- If the collector is a VM, allocate additional resources (starting with CPU cores) to it.
- If the collector is ingesting logs from only one bucket, distribute the logs across multiple buckets, which can then be assigned to different collectors.
- If the collector is ingesting logs from multiple buckets, distribute the buckets across multiple collectors.
- If the collector license has a flow rate limit, the license may need to be upgraded.

Note

- Sources that are tagged as *Disabled* may have been automatically disabled (last-in/first-out order) due to the license exporter count limit.
- In distributed deployments, it is recommended to start with a 1:1 pairing of sources and collectors.

Enabling role-based IAM for AWS deployments

Role-based IAM can be enabled for Scrutinizer AMI instances by ticking the checkbox in the configuration tray. The role assigned to the EC2 instance should be provisioned with the following permissions:

```
{ "Version": "2012-10-17",
  "Statement": \[
             { "Sid": "VisualEditor0",
               "Effect": "Allow",
               "Action": \[ "s3:GetObject", "s3:DeleteObject" \],
               "Resource": \[ "arn:aws:s3:::<S3BUCKET>/\*" \]
             },
             { "Sid": "VisualEditor1",
               "Effect": "Allow",
               "Action": "s3:\*",
               "Resource": "arn:aws:s3:::<S3_BUCKET_NAME>"
             }
  \]
}
```

Note

Role based authentication is only available when all log downloaders are hosted in AWS.

Importing AWS entity descriptions

To allow description reporting and filtering by AWS entity identifiers (`interface-id`, `vpc-id`, etc.) directly in the Scrutinizer UI, follow these steps:

1. Provision the user or IAM role with the following additional permissions:

```
ec2:DescribeInstances
ec2:DescribeSubnets
ec2:DescribeVpcs
ec2:DescribeNetworkInterfaces
```

2. Start an SSH session with the Scrutinizer server (or the primary reporter in distributed deployments), and run the following command via the **scrut_util** interactive CLI:

```
SCRUTINIZER> awssync
      AWS entities synced!
```

Once entity descriptions have been synced, AWS entity identifiers will automatically be replaced with their descriptions whenever an AWS-specific report is run. The *awssync* task will automatically be run every hour thereafter.

Cisco FireSIGHT

Scrutinizer can be configured to receive flows from a Cisco FireSIGHT system via its Event Streamer (eStreamer) service.

After this integration is enabled, the following reports become available in Scrutinizer:

- App Internet HTTP Host
- Application E-Zone & Sub Type
- Application I-Zone & Sub Type
- Firewall List
- Ingress and Egress Zones
- User App HTTP Host
- User App HTTP URL
- User Application
- Web App & CoS
- Web App Event & Rule Details
- Web App and Source IP

Important

The minimum supported eStreamer version is 5.4.

Registering Scrutinizer with FireSIGHT

Before setting up the integration in Scrutinizer, the server/collector must be registered under the FireSIGHT Defense Center:

1. Log into the FireSIGHT Defense Center.

   For Firepower v5.4: Navigate to **System** > **Local** > **Registration**

   For Firepower v6.x: Navigate to **System** > **Integration** > **eStreamer**

2. Enable all eStreamer Events, and then click **Save**.
3. Click the **Create Client (+)** button, and then enter the IP address of the Scrutinizer collector.
4. [OPTIONAL] Enter a password.
5. Locate the Scrutinizer client in the list, and then click **Download** to download the client certificate.
6. Upload the client certificate to the `/home/plixer/scrutinizer/files/` directory on the Scrutinizer appliance.

Configuring Scrutinizer as an eStreamer client

After the Scrutinizer collector has been registered, it will need to be configured to start receiving FireSIGHT flows:

1. Start an SSH session with the Scrutinizer collector.
2. Edit the `/etc/firesight.ini` file to reflect your Scrutinizer collector and FireSIGHT configuration:
   - `CollectorIp` - Scrutinizer collector IP address
   - `CollectorPort` - Scrutinizer receiving port for FireSIGHT flows
   - `fdi_templates` - Path where export templates are defined (default: `/home/plixer/scrutinizer/files/fdi_templates/firesight.fdit`)
   - `host` - FireSIGHT server address
   - `port` - FireSIGHT server outbound port
   - `pkcs12_file` - Location of the FireSIGHT eStreamer client certificate (default: `/home/plixer/scrutinizer/files/<Plixer_Scrutinizer_IP>.pkcs12`)
   - `pkcs12_password` - Password entered during registration process; leave blank if no password was set
   - `fs_bind_addr` - eStreamer client address (collector IP address)
   - `export_to` - Collector name set at the beginning of the file

   > **Note**
   >
   > - The Scrutinizer eStreamer client configuration will automatically be updated whenever `firesight.ini` is modified.
   > - Editing the provided `firesight.ini` file is recommended, but a new file can also be created in the same directory. A sample file (`firesight.ini.sample`) can be found in `/home/plixer/scrutinizer/files`.
   > - Multiple collectors and FireSIGHT servers with unique names can be set up within the same `firesight.ini` file. A collector can be configured to receive flows from more than one source and a FireSight server can send flows to more than one destination.

3. The eStreamer client will export flows to the collector at CollectorIP and CollectorPort.

4. `fdi_templates` is the path where the export templates are defined. Use the location provided in the example.
5. The eStreamer client will connect to the FireSIGHT at the FireSIGHT host and port.
6. `pkcs12_file` is the location of the updated FireSIGHT eStreamer client certificate.
7. `pkcs12_password` is the certificate password, or blank if a password wasn't specified.
8. `fs_bind_addr` is the eStreamer client address registered with FireSIGHT (Scrutinizer collector IP address). It must be a bindable address that can route to the eStreamer service.
9. `export_to` tells the eStreamer client which collector or collectors will receive exported flows.
10. In the `/home/plixer/scrutinizer/env/local_env file`, change the value for `export PLIXER_NO_FIRESEER=1` to `0`.
11. Restart the Collector using the command: `service plixer_flow_collector restart`

After the restart, Scrutinizer should start receiving FireSIGHT flows within 1 minute. For assistance with the configuration process or troubleshooting help, contact *Plixer Technical Support*.

Endace

When Endace integration is enabled, the following additional host inspection options become available after a report is run:

- **Endace - Pivot-to-Vision**: Opens the Endace Vision view
- **Endace - Pivot-to-Packets**: Downloads a packet capture with user-specified parameters (`*.pcap` or `*.erf`)

These options can be accessed after clicking on an IP address or hostname in the report results view, under **Other Options** in the tray.

Note

- It may be necessary to log in to *EndaceProbe* or *InvestigationManager* when pivoting to *Vision* for the first time.
- Data will only be available for hosts/traffic seen on the EndaceProbe.

Configuration requirements

The following details will be required to enable/configure Endace integration:

- Endace server IP address (and DNS hostname, if desired)
- Port to use to connect to the Endace server (typically 443 or 80)
- Credentials to use for the Endace server
- [Optional] Names of data sources configured on the Endace server

Configuring Endace Pivot-to-Packets

To enable Endace integration, add the EndaceProbe or InvestigationManager by *launching scrut_util* and running the following at the `SCRUTINIZER>` prompt:

Note

This command requires an IP address and will not work with a hostname.

```
endace add <ENDACE_IP_ADDRESS> <PORT> <USER> <PASSWORD>
```

For example:

```
SCRUTINIZER> endace add 10.11.12.13 443 adminuser adminuserpass
```

See *this page* for other scrut_util commands related to Endace integration.

Configuring Endace Pivot-to-Vision

The Endace *Pivot-to-Vision* option can be configured as follows:

Note

- The *Pivot-to-Vision* option can be configured independently of *Pivot-to-Packets*.
- Data sources are defined in the EndaceProbe configuration. For example, to use all available rotation files, replace `DATA_SOURCES` below with `tag%3Arotation-file`.
- When using port 80, it may be necessary to replace `https://` with `http://` in the below URLs.

1. SSH to the Scrutinizer server as the `plixer` user.
2. Configure the EndaceProbe IP address or hostname and data sources to use by adding the following lines to the end of `/home/plixer/scrutinizer/files/applications.cfg`:

   ```
   Endace - Pivot2Vision, https://
   ↪<ENDACEPROBE_IP_OR_HOSTNAME>/vision2/pivotintovision/?datasources=<DATA_SOURCES>&title=Scrutin
   ↪Endace Vision 2 - Investigation
   ```

   Examples:

   ```
   Endace - Pivot2Vision, https://
   ↪endace-probe.company.com/vision2/pivotintovision/?datasources=tag%3Arotation-file&title=Scruti
   ↪Endace Vision 2 - Investigation
   ```

Google Cloud VPC logs

When Google Cloud Platform Virtual Private Cloud (VPC) flow log ingestion is configured and enabled, Scrutinizer can monitor and report on traffic data associated with GCP VPC assets.

This section covers the prerequisites and setup/configuration steps for GCP VPC flow log ingestion.

Setting up the Google Cloud Pub/Sub topic and subscription

Scrutinizer uses the GCP Pub/Sub messaging service as an ingestion source for VPC flow logs.

To set up the Pub/Sub topic that will receive the log entries to be ingested, follow these steps:

Note

To ensure seamless access between components/services, it is highly recommended to set everything up under the project where flow logs will originate.

1. Enable and configure VPC flow logs for the target resources (**all** metadata fields must be included/enabled).
2. Next, navigate to the **Pub/Sub Topics** page and create a new topic with message retention enabled and set to at least one hour (other topic settings can be configured as desired).
3. After the topic has been created, go to the **Subscriptions** page and create a pull subscription for the new topic (note the Subscription ID for later use).
4. Next, go to the **Log Router** page and create a sink to route the log entries to the newly created topic and configure any inclusion/exclusion filters necessary.
5. After adding the Pub/Sub topic as a sink, navigate to the **Service Accounts** page and select a service account associated with the sink/topic.
6. Under the *Keys* tab, click the **Add Key** button and select *JSON* to download a file containing the credentials required to subscribe to the Pub/Sub topic.

Once the above steps have been completed, verify that log entries are being correctly routed to the Pub/Sub topic, and then proceed to configuring ingestion in Scrutinizer.

Configuring GCP VPC flow log ingestion in Scrutinizer

Once the Pub/Sub topic is receiving log entries and the subscription has been set up, it can be added to Scrutinizer as follows:

1. In the Scrutinizer web interface, navigate to **Admin** > **Integrations** > **Flow Log Ingestion**.
2. Click the **+** icon, and then select *Google Cloud Platform* in the tray.
3. In the secondary tray, configure the subscription details as follows:
   - Enter a name to identify the source by.
   - Select the Scrutinizer servers to use as the log downloader(s) and collector(s) (in *distributed clusters*, remote collectors are recommended for these roles).
   - Enter the GCP project ID associated with the topic subscription.
   - Enter the subscription name/ID used.
   - Enter/paste the contents of the service account key JSON file.
4. Click the **Save** button to add the subscription with the current settings.

Note

- After a subscription configuration has been saved, click on the name assigned to it in the main view to open the settings tray, and use the **Test** button to confirm that Scrutinizer is able to establish a connection with the credentials entered.
- To verify that an GCP VPC flow log source has been successfully added, look for an exporter whose hostname matches the GCP VPC in the **Explore** > **Exporters** > **By Exporters** view or the **Admin** > **Resources** > **Manage Exporters** page (after ~1 hour).

- Flow log ingestion processes are divided between the *log downloader* (downloads the flow logs through the topic subscription) and the *flow collector* (collects and processes the downloaded logs). A different Scrutinizer server can be used for each role, and a single subscription can have multiple downloaders and collectors.

Troubleshooting

If the **Admin** > **Resources** > **Exporters** view does not list exporters matching the virtual network(s) set up for flow ingestion, check the following for issues:

- Open the tray for the ingestion source in the **Admin** > **Integrations** > **Flow Ingestion** view and use the **Test** button to verify that the collector/downloader is able to communicate with the data source using the details entered.
- Verify that logs are being published to the source topic.
- Check the collector log file in `/home/plixer/scrutinizer/files/logs/` for errors.
- Check `gcpst_log.json` for possible source-side issues.

For further assistance, contact *Plixer Technical Support*.

Overloaded collectors/downloaders

The *Unresourced - Enabled* status in the Admin > Resources > Exporters view indicates that a log source is being temporarily disabled/paused due to insufficient resources.

The following are potential solutions for an overloaded collector:

- If the collector is a VM, allocate additional resources (starting with CPU cores) to it.
- If the collector is ingesting logs from only one stopic, distribute the logs across multiple topics, which can then be assigned to different collectors.
- If the collector is ingesting logs from multiple topics, distributed the topics across multiple collectors.
- If the collector license has a flow rate limit, the license may need to be upgraded.

Note

- Sources that are tagged as *Disabled* may have been automatically disabled (last-in/first-out order) due to the license exporter count limit.
- In distributed deployments, it is recommended to start with a 1:1 pairing of sources and collectors.

Grafana

Integrating the Grafana plugin with Scrutinizer allows users to monitor systems, applications, and infrastructure through dashboards, charts, and graphs.

To setup the Grafana integration, do the following:

1. Deploy the default Grafana server.
2. Start the server with the default (Production) settings.
3. Adjust the `default.ini` file to run the server in development mode.

Note

If you are deploying a non-default server, edit the `custom.ibj` file instead.

4. Start the server.

For assistance in getting or setting up the Grafana plugin, contact *Plixer Technical Support*.

Microsoft Azure logs

When Azure virtual network flow log ingestion is configured and enabled, Scrutinizer can monitor and run reports on traffic traversing assets in the cloud.

Once flow data for network resources on Azure is being received, the following additional report types can be run:

Azure report types

Report Type	Description
Flow Decisions	Aggregation based on decision (*accept* or *deny*) applied to traffic via configured rules
Flow Decisions Count	Flow count aggregation for each traffic decision
Flow States	Aggregation based on distinct states reported for individual network flows
Flow States Count	Flow count aggregation for each network flow state
All Details	Aggregation based on full range of flow details, including the rule and application associated with the traffic
Resource IDs	Aggregation based on resource IDs

This section covers the prerequisites and setup/configuration steps for Azure flow log ingestion.

Changed in version 19.6.0: Scrutinizer now supports ingestion for VNet flow data in addition to NSG flow data.

Setting up the Azure blob storage container

Before setting up Azure flow log ingestion in Scrutinizer, the virtual networks to be monitored should be configured to send flow logs to the Azure Storage blob container(s) that will be used. Both v1 and v2 flow logs are supported, but the latter is recommended to enable volume-based reports.

Important

- Any containers used for this purpose should have versioning disabled and must be reserved for exclusive use by Scrutinizer. If the flow logs need to be archived or used for other purposes, send the flow logs to a separate blob container, and then automate the replication/duplication of those logs to the container that will be used by Scrutinizer.
- If any historical data needs to be retained, it will need to be copied off the container before ingestion is enabled/configured. Manually clearing the container of inactive log files will also allow Scrutinizer to become current more quickly.

Once a blob countainer is configured as a flow log source, Scrutinizer will periodically collect the most recent 15 minutes of logs and delete all inactive log files not updated in the past ~1 hour.

Configuring Azure flow log ingestion in Scrutinizer

To add an Azure Storage blob container as a flow log source in Scrutinizer, follow these steps:

1. In the Scrutinizer web interface, navigate to **Admin** > **Integrations** > **Flow Log Ingestion**.
2. Click the **+** icon, and then select *Azure FlowLogs* in the tray.
3. In the secondary tray, configure the container details as follows:
   - Enter a name to identify the bucket/source by.
   - Enter the container name:
     - For NSG flow logs, this will typically follow the format of `insights-logs-networksecuritygroupflowevent`
     - For VNet flow logs, this will typically follow the format of `insights-logs-flowlogflowevent`
   - Select the collector(s) to assign to the container from the dropdown (in *distributed clusters*, a remote collector is recommended for this role).
   - Enter the storage account name and key to use to access the container (in most cases, the service URL host name without `.blob.core.windows.net/` or another domain)
   - Enter the service URL for the container (in most cases, formatted as `https://STORAGE-ACCOUNT-NAME.blob.core.windows.net/`).
4. Click the **Save** button to add the container with the current settings.

Once added, the container will be listed in the main Admin > Integrations > Flow Log Ingestion view under the configured name. An exporter associated with the Azure virtual network will also be added to the device lists for Scrutinizer's various functions (reports, network maps, etc.).

> **Note**
>
> - After a container configuration has been saved, click on the name assigned to it in the main view to open the settings tray, and use the **Test** button to confirm that Scrutinizer is able to establish a connection to the container with the credentials entered.
> - To verify that the Azure flow log source has been successfully added, look for an exporter whose hostname matches the virtual network in the **Explore** > **Exporters** > **By Exporters** view or the **Admin** > **Resources** > **Exporters** page (after ~1 hour).

Troubleshooting

If the **Admin** > **Resources** > **Exporters** view does not list exporters matching the virtual network(s) set up for flow ingestion, check the following for issues:

- Open the tray for the ingestion source in the **Admin** > **Integrations** > **Flow Ingestion** view and use the **Test** button to verify that the collector/downloader is able to communicate with the data source using the details entered.
- Verify that logs are correctly being sent to the source container.
- Check the collector log file in `/home/plixer/scrutinizer/files/logs/` for errors.
- Check `azure_log.json` for possible source-side issues.

For further assistance, contact *Plixer Technical Support*.

Overloaded collectors/downloaders

The *Unresourced - Enabled* status in the Admin > Resources > Exporters view indicates that a log source is being temporarily disabled/paused due to insufficient resources.

The following are potential solutions for an overloaded collector:

- If the collector is a VM, allocate additional resources (starting with CPU cores) to it.
- If the collector is ingesting logs from only one container, distribute the logs across multiple containers, which can then be assigned to different collectors.
- If the collector is ingesting logs from multiple containers, distribute the containers across multiple collectors.
- If the collector license has a flow rate limit, the license may need to be upgraded.

Note

- Sources that are tagged as *Disabled* may have been automatically disabled (last-in/first-out order) due to the license exporter count limit.
- In distributed deployments, it is recommended to start with a 1:1 pairing of sources and collectors.

Oracle Cloud VCN logs

When Oracle Cloud Infrastructure Virtual Cloud Network (VCN) flow log ingestion is configured and enabled, Scrutinizer can monitor and report on traffic associated with specified Oracle Virtual Network Interface Cards (VNICs).

This section covers the prerequisites and setup/configuration steps for OCI VCN flow log ingestion.

Setting up the OCI flow log stream

VCN flow log ingestion in Scrutinizer uses the OCI streaming service as the log data source. After being downloaded from a stream, the log data is forwarded to one or more specified collectors as regular flows.

To set up the flow log stream, follow these steps:

1. Create a new stream in any stream pool to publish the flow logs to.
2. Enable flow logs for the VCN, subnet, or VNICs.
3. Configure a new service connector as follows:
   - Source: Compartment, log group, and name associated with the logs enabled in step 2.
   - Target: Compartment and name associated with the stream created in step 1.
4. Create/provision an IAM group with the `use stream-pull` permission and add a user to the group (or select an existing user).
5. Generate an API signing key pair for the user and download the private key as described here.
6. Get the private key fingerprint using this command.

Verify that the flow logs are correctly being published to the stream, and then proceed to configuring Scrutinizer to download/ingest the log data.

Note

If the key pair was not generated via the OCI console, the public key will need to be uploaded for the user.

Configuring OCI VCN flow log ingestion in Scrutinizer

Once the OCI stream has been successfully configured, it can be added to Scrutinizer as a flow log source as follows:

1. In the Scrutinizer web interface, navigate to **Admin** > **Integrations** > **Flow Log Ingestion**.
2. Click the **+** icon, and then select *Oracle Cloud Streams* in the tray.
3. Enter the following details in the secondary tray:
   - Enter a name to identify the stream/source by.
   - Select the Scrutinizer servers to use as log downloader(s) and collector(s) for the stream (in *distributed clusters*, remote collectors are recommended for these roles).
   - Enter the URL for the stream pool containing the flow log stream.
   - Enter the OCID of the stream receiving the VCN flow logs.
   - Enter the OCID of the OCI tenancy.
   - Enter the OCID of the user to be used to access the streams (must have the required permissions).
   - Enter the fingerprint of the private API signing key generated for the user.
   - Enter the passphrase associated with the private key (leave blank if no passphrase was used when the key was generated)
   - Enter the private key in PEM format.
   - Enter the name of the home region of the tenancy.
4. Click the **Save** button to add the stream with the current settings.

Once added, the stream will be listed in the main Admin > Integrations > Flow Log Ingestion view under the configured name. An exporter associated with VCN will also be added to the device lists for Scrutinizer's various functions (reports, network maps, etc.).

Note

- After a stream configuration has been saved, click on the name assigned to it in the main view to open the settings tray, and use the **Test** button to confirm that Scrutinizer is able to establish a connection to the stream with the credentials entered.
- To verify that an OCI VCN flow log source has been successfully added, look for an exporter whose hostname matches the VCN in the **Explore** > **Exporters** > **By Exporters** view or the **Admin** > **Resources** > **Manage Exporters** page (after ~1 hour).
- Flow log ingestion processes are divided between the *log downloader* (downloads the flow logs from the stream) and the *flow collector* (collects and processes the downloaded logs). A different Scrutinizer server can be used for each role, and a single stream can have multiple downloaders and collectors.

Troubleshooting

If the **Admin** > **Resources** > **Exporters** view does not list exporters matching the virtual network(s) set up for flow ingestion, check the following for issues:

- Open the tray for the ingestion source in the **Admin** > **Integrations** > **Flow Ingestion** view and use the **Test** button to verify that the collector/downloader is able to communicate with the data source using the details entered.
- Verify that logs are correctly being sent to the source stream.
- Check the collector log file in `/home/plixer/scrutinizer/files/logs/` for errors.

- Check `ocist_log.json` for possible source-side issues.

For further assistance, contact *Plixer Technical Support*.

Overloaded collectors/downloaders

The *Unresourced - Enabled* status in the Admin > Resources > Exporters view indicates that a log source is being temporarily disabled/paused due to insufficient resources.

The following are potential solutions for an overloaded collector:

- If the collector is a VM, allocate additional resources (starting with CPU cores) to it.
- If the collector is ingesting logs from only one stream, distribute the logs across multiple streams, which can then be assigned to different collectors.
- If the collector is ingesting logs from multiple streams, distribute the streams across multiple collectors.
- If the collector license has a flow rate limit, the license may need to be upgraded.

Note

- Sources that are tagged as *Disabled* may have been automatically disabled (last-in/first-out order) due to the license exporter count limit.
- In distributed deployments, it is recommended to start with a 1:1 pairing of sources and collectors.

PRTG

When PRTG integration is enabled, users can view PRTG-based device information when inspecting exporters in the Scrutinizer web interface.

To set up PRTG integration in Scrutinizer, navigate to **Admin** > **Integrations** > **3rd Party Integration** and follow these steps:

1. Select **PRTG** from the dropdown and untick the **Disabled** checkbox.
2. Fill in the additional fields:
   - **Protocol** - Protocol used by the PRTG server
   - **Server IP** - PRTG server address
   - **Port** - Port used by the PRTG server
   - **User** - Username to be used to log in to the PRTG server
   - **Password** - Password to be used to log in to the PRTG server

   Important

   Default values assume the PRTG server is running on HTTPS. If necessary, modify these values to match what is configured under **PRTG Administration Tool** > **Web Server** on the PRTG server.
3. Click **Save**.

Once configured, the option to view PRTG details becomes available from the **Integrations** menu when inspecting exporters.

Important

In the Scrutinizer Classic UI, PRTG details can be viewed from the exporter trees under the **Status** tab.

SD-WAN log ingestion

Scrutinizer is capable of acting as a collector for NetFlow- or IPFIX-exporting devices/appliances from the following SD-WAN providers:

- HPE Aruba Networking EdgeConnect (*formerly Silver Peak Unity EdgeConnect*)
- Barracuda Secure SD-WAN
- Cisco IOS-XE/Catalyst SD-WAN (*formerly Viptela*)
- Citrix NetScaler SD-WAN (*formerly CloudBridge*)
- F5 SD-WAN
- GFI Software Exinda SD-WAN
- HPE Juniper SD-WAN with Session Smart Routing (*formerly 128 Technology Session Smart SD-WAN*)
- Nuage Networks from Nokia (*formerly Alcatel-Lucent SD-WAN*)
- Prisma SD-WAN / Palo Alto Networks SD-WAN (*formerly CloudGenix*)
- Riverbed SteelConnect with SteelHead
- SonicWall
- Stormshield
- VeloCloud / Arista Networks SD-WAN (*formerly VMware SD-WAN*)

Both standard and vendor-specific information elements are supported, with additional *report types* (search by provider) also available based on provider.

Configuration

In most cases, setting up Scrutinizer as a collector in an SD-WAN environment will involve three main steps (via the environment orchestrator interface):

1. Enable exporting of NetFlow or IPFIX flow data.
2. Add Scrutinizer as a collector using the following details:
   - IP address: IP Address of the Scrutinizer server/collector flow data will be sent to
   - UDP port: UDP listening port to use on the Scrutinizer collector
   - Export interval (or active flow timeout): 1 minute
3. Define the types and sources of flow data for the Scrutinizer collector:
   - Assign the collector to interfaces and/or devices
   - Apply custom traffic data filters

Once set up, verify that Scrutinizer is receiving flow data by checking the Admin > Resources > Exporters page for the SD-WAN exporters. A vendor-specific report type can also be created/run (may take several minutes to become available) to confirm that flow data is being received correctly.

Configuration example

The example below outlines the steps for adding and assigning Scrutinizer as a collector in a VeloCloud environment:

Adding Scrutinizer as a NetFlow collector:

1. Navigate to **Configure** > **Network Services** > **NetFlow Settings**.
2. Under **Collectors**, click *New*, and then enter the following details:
    - Collector name: Name to identify the Scrutinizer collector by
    - Collector IP: IP address of the Scrutinizer collector to export flows to
    - UDP port: UDP port to use on the Scrutinizer collector (default: 2055)
3. [Optional] Under **Filters**, click *New*, and then create one or more traffic filters:
    - Define criteria (source/destination IP, application ID, etc.) as needed.
    - Select to *Allow* or *Deny* matching data.

Assigning the collector(s) to Profiles (up to 8), Segments (up to 2), and/or Edges (up to 8):

- Profiles: Navigate to **Configure** > **Profiles**, and then assign the collector(s) and filters under **Network Settings**
- Edges: Navigate to **Configure** > **Edges**, enable *Edge Override* for the Edge device, and then configure the following:
    - Source interface(s) (e.g., LAN or WAN)
    - Collector(s) to assign
    - Optional filters
    - Export interval (e.g., every 60 seconds)

Configuring global Netflow export settings:

- Select the *source interface(s)* for flow exports (will be shown as the sender IP)
- Set the *export interval* to define flow export frequency (recommended: every 5 minutes)
- Verify that all Edge devices exporting flows are able to reach (firewall rules, routing, etc.) the Scrutinizer collector(s)

Viptela

Viptela flow data is collected via API, which can be enabled in the Scrutinizer web interface as follows:

1. Navigate to **Admin** > **Integrations** > **Viptela Settings**.
2. Tick the checkbox to enable the Viptela integration and enter the following details in the fields provided:
    - Viptela vManage NMS (or Cisco Catalyst SD-WAN Manager) IP address or hostname
    - Maximum number of Viptela API requests that can be processed concurrently (default: 10)
    - Maximum number of records that should be returned by each Viptela API request (default: 1000)
    - Password for the login account to use for API requests
    - Port to use for API requests (default: 8443)
    - Protocol to use for API requests (default: HTTPS)
    - Username for the login account to use for API requests (must have full read access)
3. Click the *Test* button to verify that Scrutinizer is able to access the Viptela SD-WAN API with the account credentials provided.
4. Click **Save**.

Once Viptela flow data collection has been enabled and configured, additional report types will be available to run.

Note

If there are issues using the configured settings, verify that all details were entered correctly and check the collector log for errors. For further assistance, contact *Plixer Technical Support*.

ServiceNow

Bi-directional ServiceNow integration streamlines troubleshooting ticket creation and management by linking incident reports directly to the relevant data in Scrutinizer.

When a collection is flagged for ticketing, ServiceNow generates an incident that links back to more detailed views in Scrutinizer. Alarm policies can also be configured to send notifications with optional JSON parameters for automatic incident generation.

Important

ServiceNow integration requires additional licensing to enable. Contact *Plixer Technical Support* to learn more.

Configuring ServiceNow integration

To configure a ServiceNow instance to Scrutinizer, follow these steps:

1. Navigate to **Admin** > **Integrations** > **ServiceNow**
2. Click the **Add** button and enter the following details for the ServiceNow instance to be added:
   - Unique name for the instance (used only within Scrutinizer)
   - Instance URL
   - Username to be used to connect to the ServiceNow instance
   - Password associated with the username

   Important

   The ServiceNow user registered in Scrutinizer must be assigned the `sn_incident_write` role.

3. Verify that the details entered are correct, and then click **Save**.

   Hint

   The **Test** button can be used to confirm that the ServiceNow instance has been correctly configured.

Once ServiceNow integration has been enabled, the ServiceNow instance name will be added as an option when managing collections or configuring notification profiles for alarm policies.

SolarWinds

When SolarWinds integration is enabled, users can view SolarWinds-based device statistics when inspecting exporters in the Scrutinizer web interface.

To set up SolarWinds integration in Scrutinizer, navigate to **Admin** > **Integrations** > **3rd Party Integration**, and then do the following:

1. Select **SolarWinds** from the dropdown, and then untick the **Disabled** checkbox.
2. Fill in the additional fields:
   - **Server IP** - SolarWinds server IP address
   - **User** - Username to be used to log in to the SolarWinds server
   - **Password** - Password associated with the entered SolarWinds login
   - **API Port** - API port users by the SolarWinds server (default: 17778)
3. Click **Save**.

Once configured, the option to view SolarWinds details will be available from the **Integrations** menu when inspecting exporters.

Important

When accessing SolarWinds details from the Scrutinizer web interface, the username and password are included in the URL used to open the page. The use of HTTPS will protect the integrity of the credentials over the network, but they will still be visible as outlined in this SolarWinds support article.

Scrutinizer integration in SolarWinds

The SolarWinds Network Performance Monitor supports pivoting from the **Node Details** page to a report in Scrutinizer.

Note

This integration was configured for Solarwinds NPM 12.2 and is not guaranteed to work on older installations.

To set up Scrutinizer integration in SolarWinds, follow these steps:

1. Navigate to **Settings** > **All Settings**. Under **Node & Group Management**, select **Manage Custom Properties**.
2. Click **Add Custom Property**, and then select **Nodes** from the dropdown list.
3. Fill out the name and description fields for the property (e.g., Scrutinizer), and then click **Next**.
4. Click **Select Nodes** to assign the property to at least one existing node, and then click the **Add** arrow to add exporters.
5. Fill in the value box for the added node(s) with the following code:

```
<a href=
↪"http://SCRUTINIZER_IP_ADDRESS/search.html?el=${IP_Address}&reportType=conversations
↪">
          <img src=
↪"https://cdn.plixer.com/wp-content/uploads/2016/09/scrutinizer_logo-300x49.png"␣
↪height="49" width="300"></img>
    </a>
```

Hint

The above code block opens the *Conversations WKP* report type as the default, but this can be modified by replacing `conversations` with a different report name API as the value for `reportType`.

6. Click **Submit**.

If the integration has been correctly configured, a custom property widget for all selected nodes/exporters will be added to the **Node Details** page. To run the default report, click on Scrutinizer in the widget.

Splunk

Splunk integration enables the inspection of Scrutinizer flow and event data in the Splunk dashboard via the **Scrutinizer for Splunk** app.

This allows teams already using Splunk to seamlessly leverage Scrutinizer's flow collection and analysis capabilities and quickly jump between the two platforms as needed.

> **Note**
>
> Splunk integration requires Scrutinizer 19.6.0 or higher. The **Scrutinizer for Splunk** app expects both the *Splunk Enterprise* server and *Splunk Forwarder* client software to be pre-installed in the customer environment.

Configuring Splunk integration in Scrutinizer

To set up Scrutinizer for Splunk integration, follow these steps:

1. SSH to the Scrutinizer server as the `plixer` user.
2. Launch the interactive CLI:

   ```
   /home/plixer/scrutinizer/bin/scrut_util
   ```

3. At the `SCRUTINIZER>` prompt, run the following:

   ```
   SCRUTINIZER> enable splunk http://<SPLUNK_SERVER_IP:PORT> <SYSLOG_PORT>
   ↪<SPLUNK_FORWARDER_IP>
   ```

   The default Splunk server port is 8000 (if port 80 is used, no port number is required after the server IP address). The default listening port (`SYSLOG_PORT`) on the Splunk Forwarder is 1514.

After the command is run, Scrutinizer will begin sending flow and event data once the next flow analytics collection and detection cycle is complete.

Installing the Scrutinizer for Splunk app

After configuring Scrutinizer to send data to Splunk, the **Scrutinizer for Splunk** app can be installed as follows:

1. Download the **Scrutinizer for Splunk** app:

   ```
   REPO_HOST=files.plixer.com
   curl -k -o scrutinizer.spl https://
   ↪$REPO_HOST/plixer-repo/scrutinizer/19.7.2/util/scrutinizer.spl
   ```

   If an *offline repo host* was used to install or upgrade Scrutinizer, `REPO_HOST` can be set to the IP address of that host.

2. Log into Splunk.
3. Go to **Apps** > **Manage Apps** in the Splunk dashboard.
4. Click the **Install app from file** button, and then select the `scrutinizer.spl` file downloaded in step 1.

5. After the app is installed, locate the **Scrutinizer for Splunk** app in the **Manage Apps** menu and click *View Objects*.
6. Select *Default*, and then replace the default IP address (`10.42.100.142`) with the address of the Scrutinizer server to connect to Splunk.
7. Click the **Save** button to save the new address.

When done, return to the dashboard and access the **Scrutinizer for Splunk** app from the **Apps** menu. Data and graphs should begin to be filled in after a few minutes.

Note

- If no data appears in the Splunk UI after 5-10 minutes, restart the Splunk service on the Splunk Server by running:

  ```
  sudo /opt/splunk/bin/splunk restart
  ```

  Data should start to appear on the *Scrutinizer Vitals* page in the Splunk UI.

- To upgrade the **Scrutinizer for Splunk** app instead, tick the *Upgrade app* checkbox when selecting the `scrutinizer.spl` in the install dialog.

Visit https://www.plixer.com to learn more or contact *Plixer Technical Support* for further assistance.

Disabling Splunk integration

To disable Splunk integration in Scrutinizer:

1. SSH to the Scrutinizer server as the `plixer` user, and then *launch the interactive CLI*:
2. At the `SCRUTINIZER>` prompt, run the following:

   ```
   SCRUTINIZER> disable splunk http://<SPLUNK_SERVER_IP:PORT>
   ```

STIX-TAXII

STIX-TAXII integration allows Scrutinizer to import comprehensive and up-to-date threat intelligence in the industry-standard Structured Threat Information eXchange (STIX) format via the Trusted Automated eXchange of Indicator Information (TAXII) protocol from external systems and organizations. This greatly enhances Scrutinizer's already robust IP detection capabilities.

Important

STIX-TAXII integration requires additional licensing to enable. Contact *Plixer Technical Support* to learn more.

Importing STIX files via CLI

To have Scrutinizer automatically import IP/domain watchlists, download the files in STIX format (v1 or v2) and copy them to the `/home/plixer/scrutinizer/files/threats` directory on the appliance. The name of the file will also be used as the category.

Important

Domain watchlists are currently only used in AI-based threat detection algorithms and need not be imported for deployments that do not include the Plixer ML Engine.

> **Note**
>
> Scrutinizer supports `.stix`, `.stix1`, and `.stixv1` extensions for v1 (XML) and `.stix2` and `.stxv2` extensions for v2 (JSON).

Configuring STIX-TAXII feeds

To configure a new STIX-TAXII feed in the Scrutinizer web interface, follow these steps:

1. Navigate to **Admin** > **Integrations** > **STIX-TAXII**, and then click **Add** to create a new feed.
2. Fill in the following fields:

- Feed name
- API Root (**not** the Discovery URL)
- Collection ID
- Login credentials for the feed

3. Click **Save**.
4. Use the **Test** button to verify that Scrutinizer can access the feed with the configured settings.

After the feed has successfully been added, Scrutinizer will attempt to pull the lists from the TAXII server every time the host reputation list download service runs.

Once imported, STIX-TAXII threat intelligence will be added to Scrutinizer's (IP only) and the ML Engine's (IP and domain) reputation algorithms for alarm and event reporting under their respective alarm policies.

Additional tips

- Import IP watchlists only. All other indicators will be ignored but can cause the import of IP indicators to fail.
- Don't attempt to import IP watchlists that use complex boolean logic to trigger matches.
- The feature will ingest only independent IP indicators. It will ignore more complex ones.

> **Note**
>
> A complicated indicator included with more basic ones will not prevent them from being imported.

Username reporting

Username reporting allows Scrutinizer to ingest user traffic/activity data to enrich flow data and enhance reporting/filtering functions.

This page covers the setup and configuration steps to enable username reporting on different platforms.

On this page:

Plixer AD Users *Plixer AD Users (for Microsoft Active Directory)* Cisco ISE *Cisco Identity Services Engine (ISE)*

Plixer AD Users (for Microsoft Active Directory)

The **Plixer AD Users** utility enables username reporting for a Microsoft Active Directory server (or *Azure*). Once installed and set up on an AD server or a *remote event collector*, the utility continuously parses authentication events from an event log file and then sends the data to Scrutinizer (or another IPFIX collector).

Scrutinizer is able to leverage the username data to create IP-to-user mappings (viewable under Explore > Entities > Usernames) and apply additional reporting and filtering options.

> **Note**
>
> Only IP addresses included in internal IP groups can be mapped to users.

Setup and configuration

Follow the steps below to install the Plixer AD Users utility and set up username reporting to Scrutinizer:

1. **Enable event auditing for login events on the domain.**

   - Under **Computer Configuration** > **Policies** > **Windows Settings** > **Security Settings** > **Advanced Audit Policy Configuration** > **Audit Policies** > **Logon/Logoff**, enable *Success and Failure* for **Audit Logoff** and **Audit Logon**
   - Under **Computer Configuration** > **Policies** > **Windows Settings** > **Local Policies** > **Security Options**, select **Audit: Force audit policy subcategory settings**, and then tick the checkboxes for *Define this policy setting* and *Enabled*.

2. **Download and install the AD Users utility.**

   1. Download the product package from `https://files.plixer.com/ad-users/ad-users.zip`.
   2. Extract the contents of the archive, and then run `ad-users-installer.exe`.
   3. Follow the prompts to continue the installation.
      - To use a different admin account (instead of running the utility as a *LocalService*), the account must have administrator privileges and the *Log on as a service* permission. It must also be added to the *Event Log Readers* group.
      - Leaving the *Verify AD Users* and *Open config file* checkboxes ticked before clicking **Finish** is recommended.

   This will install the utility as a service, which will run in the background and can be set to automatically restart after the host reboots.

   > **Note**
   >
   > To start/restart the utility via command line, run the following from the command prompt:
   >
   > ```
   > ad-users.exe run
   > ```
   >
   > Running the utility from the command line does not require the package to be installed but is primarily intended for debugging/troubleshooting. Installing the utility as a service is strongly recommended for production environments.

3. **Configure and start the AD Users utility.**

   - Verify that the details in the configuration file (`ad-users.yml`) are correct:

> **Note**
>
> All `log.*****` values are only required if the utility is run from a command prompt instead of as a service.

View file contents

Name	Description	Example Value	Default Value
`chunking`	Number of events to collect and send at a time (Use `0` to send each event as it is parsed)	1000	0
`flush_w`	Number of seconds to wait before all collected events in the buffer are sent (Use `0` to use only the chunking value before sending events)	60	0
`path`	Path to the Windows event log (`Security.evtx` if the utility is running on the AD server or `ForwardedEvents.evtx` if events are being forwarded from a separate host)	C:\Windows\Syster	C:\Windows\System32\winevt\Logs\FwardedEvents.evtx
`collector`	IP address and port (in `IP:PORT` format) of the Scrutinizer server/collector	127.0.0.1:4739	127.0.0.1:4739
`exporter`	[Optional] IP address and port (in `IP:PORT` format) of the Windows server running the utility	8.8.8.8:9996	(Local IP address with port 9996)
`log.name`	[Optional] Log file to use	C:\ad-users\Plixer\ad-users.log	`ad-users.log` in the same directory as `ad-users.exe`
`log.level`	[Optional] Log level to use for the utility log `debug` (default), `info`, `warn`, or `error`	`debug`, `info`, `warn`, or `error`	`debug`
`log.max_siz`	Maximum size (in MB) of the log file before rotation	100	5
`log.max_bac`	[Optional] Maximum number of log entries to retain before rotation	100	0
`log.max_age`	[Optional] Maximum number of days to retain old log entries before rotation	14	0

   - In the Windows **Services** view, right-click on `PlixerADUsers` and select *Properties* to apply the following settings:
     - **General**: Set the startup type to *Automatic (Delayed Start)*.
     - **Recovery**: Select to *Restart the service* for the first, second, and subsequent failures.

4. **Verify that the utility is running and Scrutinizer is receiving the data.**

   - Check log files: Log entries for the utility can be viewed in the Windows Event Viewer and should only include configuration and start/stop info messages (i.e., no errors).

> **Note**
>
> If the utility is not running as a service, log entries will be written to the log file specified in `ad-users.yml` and displayed in the console (stdout).

- Navigate to **Explore** > **Entities** > **Usernames** and verify that usernames and authentication events are being received from the utility. If the **Usernames** view is not being populated, verify that the current license can support the additional exporters (each AD Users instance running will count as one exporter) under **Admin** > **Plixer** > **Scrutinizer Licensing**. An exporter corresponding to the AD Users server/host should also be listed in the **Admin** > **Resources** > **Manage Exporters** view.

> **Note**
>
> - If the AD Users service is stopped, the record ID of the event last sent is saved to `last_recordID.txt`. If this file exists, only events with record IDs greater than the number in this file will be sent to Scrutinizer. This will help avoid duplicate events or lapses in the authentication event processing when the service restarts.
> - While the `chunking` and `flush_wait_seconds` settings should help maintain consistent rates, environments with extremely high volumes of Active Directory authentication events may need to enable export spreading to prevent NetFlow export storms.

Event forwarding

The Plixer AD Users utility can also be run on a remote event collection server that is joined to the same domain as the AD server/domain controller.

Follow these steps to set up event forwarding from an AD server to a collection server:

View instructions

1. On the Active Directory server, start the Windows Remote Management service with the default configuration (requires elevated permissions):

   ```
   winrm quickconfig
   ```

2. On the event collection server, enable the Windows Event Collector Utility service with the default configuration (requires elevated permissions):

   ```
   wecutil qc
   ```

3. **Create a subscription to the AD server log file on the event collection server:**

   1. Launch the Windows Event Viewer as an administrator, and then click **Subscriptions**.
   2. Select **Create Subscription** in the **Actions** pane, and then enter a subscription name.
   3. Select **Computers**, and then enter the address of the AD server.
   4. Under **Destination log** > **Forwarded events**, select *Keep user account as machine account*.
   5. Under **Events** > **Security for event logs**, enter the following event IDs: `4624`, `4634`, `4647`, `6272-6274`, `6278`, and `6279`.

After the subscription has been created/configured, verify that Scrutinizer is receiving username and authentication event data as described in the *AD Users setup instructions* (step 4).

Plixer ML Engine integrations

Plixer AD Users 2.0.0+ is able to forward both AD authentication events and Azure sign-in logs from specified storage containers to the Plixer ML Engine. The engine is then able to leverage the data to build models to detect anomalies, deliver alerts, and generate reports.

Active Directory authentication events

To configure the AD Users utility to send username and authentication event data to a Plixer ML Engine instance for modeling and additional behavior analytics, edit the following flags in the `ad-users.yml` configuration file:

- `ml.send_ad_to_ml`: Set to `true`
- `ml.scrut_auth_token`: Enter the authentication token generated when the ML engine instance was *registered*
- `ml.ml_engine_ip`: Enter the IP address of the ML engine instance

After the changes have been saved, restart the `PlixerADUsers` service on the host running the utility, and then verify that Scrutinizer is receiving username and authentication event data as described in the *AD Users setup instructions* (step 4).

Microsoft Azure sign-in logs

The AD Users utility can collect Azure sign-in logs (for both interactive and non-interactive sign-ins) archived in a specified storage account and then send it to a Plixer ML Engine instance for model generation.

Note

Follow this Microsoft guide to create an Azure storage account and set up archiving of sign-in logs.

Once the storage account has been created and is receiving sign-in logs, follow these steps to configure the AD Users utility to collect the data and forward it to the ML engine instance:

View instructions

Important

The AD Users utility can send AD authentication event logs **and/or** Azure sign-in logs to a Plixer ML Engine instance.

`ml.ml_engine_ip` and `ml.scrut_auth_token` must both be defined for either log type to be sent.

1. Navigate to **Admin** > **Settings** > **ML AD Users** in the Scrutinizer web interface and enter the name and key for the Azure storage account being used to store the sign-in logs.
2. Edit the following flags in the `ad-users.yml` configuration file:
   - `ml.scrut_auth_token`: Enter the authentication token generated when the ML engine instance was *registered*
   - `ml.ml_engine_ip`: Enter the IP address of the ML engine instance
   - `azure.send_azure_to_ml`: Set to `true`
   - `azure.interactive_container`: Enter name of the container used to store interactive sign-in logs (e.g., `insights-logs-signinlogs`)

- `azure.non_interactive_container`: Enter name of the container used to store non-interactive sign-in logs (e.g., `insights-logs-noninteractive-signinlogs`)
- `azure.max_log_age_minutes`: [Optional] Enter the number of minutes to use for backfilling Azure sign-in events (i.e., logs terminated within the last X minutes will be processed; duplicates of both AD and Azure events are deleted for model generation)
- `azure.account_filter_list`: [Optional] Enter the Azure accounts/emails (comma-delimited) whose sign-in events should be processed (event data for all accounts is sent if left blank)
- `azure.app_filter_list`: [Optional] Enter the applications (e.g., `Microsoft Teams`,`Microsoft Office`,`Office 365 Exchange Online`) whose sign-in events should be processed (event data for all applications is sent if left blank)

After the changes have been saved, restart the `PlixerADUsers` service on the host running the utility, and then verify that Scrutinizer is receiving username and authentication event data as described in the *AD Users setup instructions* (step 4).

Cisco Identity Services Engine (ISE)

When Cisco Identity Services Engine (ISE) username reporting is enabled, Scrutinizer is able to retrieve username lists, search flows for specific usernames, and run additional reports related to Cisco ISE user traffic.

Important

Username reporting integration in Scrutinizer supports Cisco ISE versions 1.x, 2.x, and 3.x.

Enabling ERS

Before setting up Cisco ISE username reporting in Scrutinizer, External RESTful Services (ERS) should first be enabled on the ISE appliance as follows:

View instructions

1. On the ISE server, create a new user with the following permissions:
   - ERS Admin
   - ERS Operator
   - Super Admin
   - System Admin
2. Test the configuration from an external host by sending a `GET` request using the URL: `https://[ISE_server_address]/ise/mnt/Session/AuthList/null/null`

   Hint

   - Use any REST client (such as Postman, or similar tools) to perform this test.
   - If using a GUI-based tool like Postman, navigate to the server URL first in a browser and accept the security certificate, then leave that browser window open.

Visit the Cisco website to learn more about enabling ERS for the supported ISE versions.

Configuring Scrutinizer for Cisco ISE

View instructions

1. SSH into the Scrutinizer server as the `plixer` user and run `/home/plixer/scrutinizer/bin/scrut_util` to launch the *scrut_util interactive CLI*.
2. At the `SCRUTINIZER>` prompt, enter:

   ```
   SCRUTINIZER> ciscoise add [ISE_IP] [ISE_TCP_port] [ISE_user]
   ```

   This adds a Cisco ISE node from which username data for active sessions can be retrieved. `ISE_IP` and `ISE_TCP_port` refer to the ISE server's address and TCP port number and `ISE_user` refers to the user previously created on the same server.
3. When prompted, enter the password for the ISE user.

After all configuration steps have been completed, all functions associated with Cisco ISE username reporting will immediately be enabled.

> **Note**
>
> It may take several minutes before usernames are displayed in the web interface.

scrut_util commands for Cisco ISE

Information about other **scrut_util** commands related to Cisco ISE username reporting can be found *here*.

Zscaler ZIA logs

When Zscaler Secure Internet and SaaS Access (ZIA) log ingestion is configured and enabled, Scrutinizer can run both preconfigured and custom reports against the collected ZIA log data.

Follow the steps below to set up a ZIA Cloud or VM-based Nanolog Streaming Service (NSS) feed for firewall logs and configure Scrutinizer to ingest the log flows.

Cloud NSS

To enable log ingestion from a Cloud NSS feed, set up the feed in the ZIA Admin Portal before configuring Scrutinizer to download and consume the logs.

Adding/creating a Cloud NSS feed

To create and configure the feed, log in to the ZIA Admin Portal and follow these steps:

1. Navigate to **Administration** > **Nanolog Streaming Service**.
2. Select the **Cloud NSS Feeds** tab, and then click *Add Cloud NSS Feed*.
3. In the next window, configure the following (other details can be set as needed/desired):
   - NSS Type: Select *NSS for Firewall*.
   - SIEM Type: Select *Other*.
   - OAuth 2.0 Authentication: Not currently supported for *Other* SIEMs and can be left disabled.
   - Max Batch Size: Adjust the value to match log throughput as closely as possible (can be set lower to improve latency at the cost of network overhead).

- API URL: Enter `https://SCRUTINIZER_DOWNLOADER_IP:8888/` (address and port may be different if translation or forwarding is enabled)
- HTTP Headers: Create a `PlixerAccessToken` key and note the corresponding value for later use (recommended for authentication in place of OAuth).
- Log Type: Select *Firewall Logs* or *DNS Logs*.
- Firewall Log Type: Select *Both Session and Aggregate Logs*. Some events are only provided as aggregate logs. Including both ensures that all events are sent.
- Feed Output Type: Select JSON.
- JSON Array Notation: Leave disabled.
- Feed Escape Character: Enter `,\"` (required to avoid unintentional termination of string data values).
- Feed Output Format: See the below section on the *JSON output format* for more details.

To stream both log types (firewall and DNS), create a feed for each type.

Configuring log ingestion in Scrutinizer

After the feed has been created, configure Scrutinizer to receive the logs as follows:

1. In the Scrutinizer web interface, navigate to **Admin** > **Integrations** > **Flow Log Ingestion**.
2. Click the **+** icon, and then select *Zscaler ZIA* in the tray.
3. In the secondary tray, configure the following details:
   - Type: Select *Cloud NSS*.
   - Log Source: Select the log type that was set to be streamed.
   - Log Downloader: IP address of the Scrutinizer server/collector that will download the logs (must match the address entered for the API URL)
   - Log Downloader Port: Port to use to receive logs on the downloader (must match the port entered for API URL)
   - Worker Count: Enter 1 (can be increased if log streaming rate seems slow; recommended maximum of 10).
   - Certificate Authority: Full contents of the public API URL SSL certificate file (see *this section* for troubleshooting help)
   - Private Key: Full contents of the SSL certificate's private key file (will be encrypted at rest)
   - Access Token: Value of the `PlixerAccessToken` created for HTTP header authentication
   - Flow Collectors: Select the Scrutinizer collector(s) that will receive the logs as standard flows (in *distributed clusters*, a remote collector is recommended for this role)
4. Click the **Save** button to add the log stream with the current settings.

Once added, the Cloud NSS feed will be listed in the main Admin > Integrations > Flow Log Ingestion view under the downloader IP address and port configured. If multiple feeds were created to stream different ZIA log types, each one will require a matching configuration in Scrutinizer.

VM-based NSS

When setting up log ingestion via VM-based NSS, deploy and then register the NSS VM/server in the ZIA Admin Portal before creating the feed and configuring Scrutinizer to download and consume the logs.

Adding/creating a VM-based NSS feed

To create and configure the feed, log in to the ZIA Admin Portal and follow these steps:

1. Navigate to **Administration** > **Nanolog Streaming Service**.
2. Select the **NSS Feeds** tab, and then click *Add NSS Feed*.
3. In the next window, configure the following (other details can be set as needed/desired):
   - NSS Server: Select the NSS server to be used to stream logs to Scrutinizer.
   - SIEM Destination Type: Select the method the NSS server should use to address the Scrutinizer downloader/server.
   - SIEM IP Address: Enter the Scrutinizer downloader's IP address or FQDN (address may be different if translation or forwarding is enabled).
   - SIEM TCP Port: TCP port on the downloader to be used to receive the logs (may also be different if translation is enabled)
   - SIEM Rate: Leave on *Unlimited* unless Scrutinizer is overwhelmed.
   - Log Type: Select *Firewall Logs* or *DNS Logs*.
   - Firewall Log Type: Select *Both Session and Aggregate Logs*. Some events are only provided as aggregate logs. Including both ensures that all events are sent.
   - Feed Output Type: Select JSON.
   - Feed Escape Character: Enter `,\"` (required to avoid unintentional termination of string data values).
   - Feed Output Format: See the below section on the *JSON output format* for more details.
   - Duplicate Logs: Select *Disabled* (recommended for reporting accuracy).

To stream both log types (firewall and DNS), create a feed for each type.

Configuring log ingestion in Scrutinizer

After the feed has been created, configure Scrutinizer to receive the logs as follows:

1. In the Scrutinizer web interface, navigate to **Admin** > **Integrations** > **Flow Log Ingestion**.
2. Click the **+** icon, and then select *Zscaler ZIA* in the tray.
3. In the secondary tray, configure the following details:
   - Type: Select *VM NSS*.
   - Log Source: Select the log type that was set to be streamed.
   - Log Downloader: Select the IP address of the Scrutinizer server/collector that will download the logs (must match the *SIEM IP Address* entered for the feed)
   - Log Downloader Port: Port to use to receive logs on the downloader (must match the *SIEM TCP Port* entered for the feed)
   - Worker Count: Enter 1 (can be increased if log streaming rate seems slow; recommended maximum of 10).
   - Flow Collectors: Select the Scrutinizer collector(s) that will receive the logs as standard flows (in *distributed clusters*, a remote collector is recommended for this role).
4. Click the **Save** button to add the log stream with the current settings.

Once added, the Cloud NSS feed will be listed in the main Admin > Integrations > Flow Log Ingestion view under the downloader IP address and port configured. If multiple feeds were created to stream different ZIA log types, each one will require a matching configuration in Scrutinizer.

JSON output format

Event logs streamed to Scrutinizer are expected in the following format:

- One JSON object per event (separated by newlines), comprising two fields:
  - "sourcetype": Must be either "zscalernss-fw" or "zscaler-dns" (other types are discarded)
  - "event": Will contain details of the log event
- JSON field names match names used by Zscaler.
- Unnecessary fields can be omitted for network and storage efficiency.

DNS event fields

The following JSON object example shows all supported fields for ZIA DNS log events. "datacenter" and "epochtime" are significant within Scrutinizer and must be included. However, other fields that will not be needed may be removed to reduce overhead and improve throughput.

The example is broken into lines and indented for readability and to make modification easier. The backslashes (\) before JSON object delimiters are so that Zscaler recognizes them as literal characters and not part of field data placeholders.

It is recommended to remove all newlines and spaces before inputting this into the NSS configuration dialog. The formatting is harmless to leave in, but it will needlessly consume extra bandwidth.

ZIA DNS log event JSON example

```
\{
  "sourcetype": "zscalernss-dns",
  "event": \{
    "cip": "%s{cip}",
    "cloudname": "%s{cloudname}",
    "company": "%s{company}",
    "datacentercountry": "%s{datacentercountry}",
    "datacenter": "%s{datacenter}",
    "datacentercity": "%s{datacentercity}",
    "deviceappversion": "%s{deviceappversion}",
    "devicemodel": "%s{devicemodel}",
    "devicename": "%s{devicename}",
    "deviceostype": "%s{deviceostype}",
    "deviceosversion": "%s{deviceosversion}",
    "deviceowner": "%s{deviceowner}",
    "devicetype": "%s{devicetype}",
    "dnsapp": "%s{dnsapp}",
    "dnsappcat": "%s{dnsappcat}",
    "dnsgw_flags": "%s{dnsgw_flags}",
    "dnsgw_slot": "%s{dnsgw_slot}",
    "dnsgw_srv_proto": "%s{dnsgw_srv_proto}",
    "domcat": "%s{domcat}",
    "durationms": "%d{durationms}",
    "ecs_prefix": "%s{ecs_prefix}",
```

(continues on next page)

(continued from previous page)

```
    "ecs_slot": "%s{ecs_slot}",
    "edepartment": "%s{edepartment}",
    "edevicehostname": "%s{edevicehostname}",
    "eedone": "%s{eedone}",
    "elocation": "%s{elocation}",
    "elogin": "%s{elogin}",
    "epochtime": "%d{epochtime}",
    "error": "%s{error}",
    "http_code": "%s{http_code}",
    "istcp": "%d{istcp}",
    "pcapid": "%s{pcapid}",
    "protocol": "%s{protocol}",
    "recordid": "%d{recordid}",
    "req": "%s{req}",
    "reqaction": "%s{reqaction}",
    "reqrulelabel": "%s{reqrulelabel}",
    "reqtype": "%s{reqtype}",
    "res": "%s{res}",
    "resaction": "%s{resaction}",
    "respipcat": "%s{respipcat}",
    "resrulelabel": "%s{resrulelabel}",
    "restype": "%s{restype}",
    "sip": "%s{sip}",
    "sport": "%d{sport}"
  \}
\}
```

Firewall event fields

The following JSON object example shows all supported fields for ZIA firewall log events. "datacenter" and "epochtime" are significant within Scrutinizer and must be included. However, other fields that will not be needed may be removed to reduce overhead and improve throughput.

The example is broken into lines and indented for readability and to make modification easier. The backslashes (\) before JSON object delimiters are so that Zscaler recognizes them as literal characters and not part of field data placeholders.

It is recommended to remove all newlines and spaces before inputting this into the NSS configuration dialog. The formatting is harmless to leave in, but it will needlessly consume extra bandwidth.

ZIA firewall log event JSON example

```
\{
  "sourcetype": "zscalernss-fw",
  "event": \{
    "time": "%s{time}",
    "epochtime": "%d{epochtime}",
    "csip": "%s{csip}",
    "csport": "%d{csport}",
    "cdip": "%s{cdip}",
    "cdport": "%d{cdport}",
    "cdfqdn": "%s{cdfqdn}",
```

(continues on next page)

(continued from previous page)

```
    "tsip": "%s{tsip}",
    "elocation": "%s{elocation}",
    "ttype": "%s{ttype}",
    "aggregate": "%s{aggregate}",
    "srcip_country": "%s{srcip_country}",
    "threatcat": "%s{threatcat}",
    "ethreatname": "%s{ethreatname}",
    "threat_score": "%d{threat_score}",
    "threat_severity": "%s{threat_severity}",
    "ipsrulelabel": "%s{ipsrulelabel}",
    "ips_custom_signature": "%d{ips_custom_signature}",
    "sdport": "%d{sdport}",
    "sdip": "%s{sdip}",
    "ssip": "%s{ssip}",
    "ssport": "%d{ssport}",
    "ipcat": "%s{ipcat}",
    "avgduration": "%d{avgduration}",
    "durationms": "%d{durationms}",
    "numsessions": "%d{numsessions}",
    "stateful": "%s{stateful}",
    "erulelabel": "%s{erulelabel}",
    "action": "%s{action}",
    "dnat": "%s{dnat}",
    "dnatrulelabel": "%s{dnatrulelabel}",
    "recordid": "%d{recordid}",
    "pcapid": "%s{pcapid}",
    "inbytes": "%ld{inbytes}",
    "outbytes": "%ld{outbytes}",
    "nwapp": "%s{nwapp}",
    "ipproto": "%s{ipproto}",
    "destcountry": "%s{destcountry}",
    "nwsvc": "%s{nwsvc}",
    "eedone": "%s{eedone}",
    "elogin": "%s{elogin}",
    "edepartment": "%s{edepartment}",
    "edevicehostname": "%s{edevicehostname}",
    "devicemodel": "%s{devicemodel}",
    "devicename": "%s{devicename}",
    "deviceostype": "%s{deviceostype}",
    "deviceosversion": "%s{deviceosversion}",
    "deviceowner": "%s{deviceowner}",
    "deviceappversion": "%s{deviceappversion}",
    "external_deviceid": "%s{external_deviceid}",
    "ztunnelversion": "%s{ztunnelversion}",
    "bypassed_session": "%d{bypassed_session}",
    "bypass_etime": "%s{bypass_etime}",
    "flow_type": "%s{flow_type}",
    "datacenter": "%s{datacenter}",
    "datacentercity": "%s{datacentercity}",
    "datacentercountry": "%s{datacentercountry}",
    "rdr_rulename": "%s{rdr_rulename}",
    "fwd_gw_name": "%s{fwd_gw_name}",
```

(continues on next page)

(continued from previous page)

```
    "zpa_app_seg_name": "%s{zpa_app_seg_name}"
  \}
\}
```

Troubleshooting

If the **Admin** > **Resources** > **Exporters** view does not list exporters matching the log stream(s) set up for ingestion, check the following for issues:

- Verify that the downloader was successfully configured by looking for the following message (or something similar) in `/home/plixer/scrutinizer/files/logs/zscaler_log.json` on that server/collector and confirm that the `bind` value matches the configured TCP port:

  ```
  {"level":"warn","pid":2059650,"bind":":10000","time":
  ↪"2025-06-16T14:34:50.916-04:00","caller":"/
  ↪builds/plixer-products/scrutinizer/application/golang/zscaler/sink-tcp-server.go:131
  ↪","message":"listening for connections"}
  ```

- Verify that the source log stream has been correctly configured.
- Check the collector log file in `/home/plixer/scrutinizer/files/logs/` for errors.
- Check `zscaler_log.json` for other possible source-side issues.

For further assistance, contact *Plixer Technical Support*.

SSL certificate

When accessing the Scrutinizer Cloud NSS endpoint using a standard browser, a blank page indicates that there are no issues.

If the page indicates SSL errors, use the option to inspect the certificate to diagnose the issue(s).

Overloaded collectors/downloaders

The *Unresourced - Enabled* status in the Admin > Resources > Exporters view indicates that a log source is being temporarily disabled/paused due to insufficient resources.

The following are potential solutions for an overloaded collector:

- If the collector is a VM, allocate additional resources (starting with CPU cores) to it.
- If the collector is ingesting logs from only one log stream, distribute the logs across multiple streams, which can then be assigned to different collectors.
- If the collector is ingesting logs from multiple log streams, distribute the streams across multiple collectors.
- If the collector license has a flow rate limit, the license may need to be upgraded.

> **Note**
>
> - Sources that are tagged as *Disabled* may have been automatically disabled (last-in/first-out order) due to the license exporter count limit.
> - In distributed deployments, it is recommended to start with a 1:1 pairing of sources and collectors.

Zscaler ZPA logs

When Zscaler Secure Private Access (ZPA) log ingestion is configured and enabled, Scrutinizer can run both preconfigured and custom reports against the collected ZPA log data.

Follow the steps below to set up a ZPA log stream and configure Scrutinizer to ingest the log flows.

Adding Scrutinizer as a log receiver on ZPA

Before configuring Scrutinizer to ingest ZPA logs, add it as a log receiver via **Configuration & Control** > **Private Infrastructure** > **Log Streaming Service** > **Log Receivers** in the ZPA admin interface.

> **Note**
>
> ZPA log ingestion relies on the Log Streaming Service (LSS), which uses an App Connector to forward the logs to a specified Scrutinizer downloader server/collector. If a new App Connector needs to be deployed for this purpose, follow these guides for creating/configuring App Connectors in the ZPA Admin Portal and deploying App Connectors.

After clicking **Add**, configure the following in the next window:

Log Receiver tab

- *Domain or IP Address*: IP address of the Scrutinizer collector that will download the logs
- *TCP Port*: TCP port on the downloader to receive the logs
- *TLS Encryption*: Can be enabled if desired, provided the downloader collector meets these requirements
- *App Connector Groups*: Select the App Connector group(s) whose logs should be streamed to Scrutinizer

Log Stream tab

- *Log Type*: Select a supported log type (*App Connector Status*, *Private Service Edge Status*, *Private Cloud Controller Status*, *Browser Access*, or *User Activity*) to be streamed
- *Log Template*: Select *JSON*
- *Log Stream Content*: `"LogTimestamp"` is required, but other fields can be included/removed as needed (may affect default *report types* but custom reports can still be created). Additionally, all time fields (`"LogTimestamp"`, etc.) must be updated to the format `%J{TIME_FIELD_NAME:epoch}` (with a capital 'J' and the ':epoch' suffix). See below for full stream content examples for each log type.

If more than one log type needs to be streamed to Scrutinizer, create a log receiver configuration for each type. *Each configuration/type must use a unique TCP port.*

Log stream content examples

The following examples have all supported fields (including reformatted time fields) and can be pasted directly into the *Log Stream Content* field:

App Connector Status

```
{
    "LogTimestamp": %J{LogTimestamp:epoch},
    "Customer": %j{Customer},
    "SessionID": %j{SessionID},
```

(continues on next page)

(continued from previous page)

```
    "SessionType": %j{SessionType},
    "SessionStatus": %j{SessionStatus},
    "Version": %j{Version},
    "Platform": %j{Platform},
    "ZEN": %j{ZEN},
    "Connector": %j{Connector},
    "ConnectorGroup": %j{ConnectorGroup},
    "PrivateIP": %j{PrivateIP},
    "PublicIP": %j{PublicIP},
    "Latitude": %f{Latitude},
    "Longitude": %f{Longitude},
    "CountryCode": %j{CountryCode},
    "TimestampAuthentication": %J{TimestampAuthentication:epoch},
    "TimestampUnAuthentication": %J{TimestampUnAuthentication:epoch},
    "CPUUtilization": %d{CPUUtilization},
    "MemUtilization": %d{MemUtilization},
    "ServiceCount": %d{ServiceCount},
    "InterfaceDefRoute": %j{InterfaceDefRoute},
    "DefRouteGW": %j{DefRouteGW},
    "PrimaryDNSResolver": %j{PrimaryDNSResolver},
    "HostStartTime": %J{HostStartTime:epoch},
    "ConnectorStartTime": %J{ConnectorStartTime:epoch},
    "NumOfInterfaces": %d{NumOfInterfaces},
    "BytesRxInterface": %d{BytesRxInterface},
    "PacketsRxInterface": %d{PacketsRxInterface},
    "ErrorsRxInterface": %d{ErrorsRxInterface},
    "DiscardsRxInterface": %d{DiscardsRxInterface},
    "BytesTxInterface": %d{BytesTxInterface},
    "PacketsTxInterface": %d{PacketsTxInterface},
    "ErrorsTxInterface": %d{ErrorsTxInterface},
    "DiscardsTxInterface": %d{DiscardsTxInterface},
    "TotalBytesRx": %d{TotalBytesRx},
    "TotalBytesTx": %d{TotalBytesTx},
    "MicroTenantID": %j{MicroTenantID}
}\n
```

Private Service Edge Status

```
{
    "LogTimestamp": %J{LogTimestamp:epoch},
    "Customer": %j{Customer},
    "SessionID": %j{SessionID},
    "SessionType": %j{SessionType},
    "SessionStatus": %j{SessionStatus},
    "Version": %j{Version},
    "PackageVersion": %j{PackageVersion},
    "Platform": %j{Platform},
    "ZEN": %j{ZEN},
    "ServiceEdge": %j{ServiceEdge},
    "ServiceEdgeGroup": %j{ServiceEdgeGroup},
```

(continues on next page)

(continued from previous page)

```
    "PrivateIP": %j{PrivateIP},
    "PublicIP": %j{PublicIP},
    "Latitude": %f{Latitude},
    "Longitude": %f{Longitude},
    "CountryCode": %j{CountryCode},
    "TimestampAuthentication": %J{TimestampAuthentication:epoch},
    "TimestampUnAuthentication": %J{TimestampUnAuthentication:epoch},
    "CPUUtilization": %d{CPUUtilization},
    "MemUtilization": %d{MemUtilization},
    "InterfaceDefRoute": %j{InterfaceDefRoute},
    "DefRouteGW": %j{DefRouteGW},
    "PrimaryDNSResolver": %j{PrimaryDNSResolver},
    "HostUpTime": %J{HostUpTime:epoch},
    "ServiceEdgeStartTime": %J{ServiceEdgeStartTime:epoch},
    "NumOfInterfaces": %d{NumOfInterfaces},
    "BytesRxInterface": %d{BytesRxInterface},
    "PacketsRxInterface": %d{PacketsRxInterface},
    "ErrorsRxInterface": %d{ErrorsRxInterface},
    "DiscardsRxInterface": %d{DiscardsRxInterface},
    "BytesTxInterface": %d{BytesTxInterface},
    "PacketsTxInterface": %d{PacketsTxInterface},
    "ErrorsTxInterface": %d{ErrorsTxInterface},
    "DiscardsTxInterface": %d{DiscardsTxInterface},
    "TotalBytesRx": %d{TotalBytesRx},
    "TotalBytesTx": %d{TotalBytesTx},
    "MicroTenantID": %j{MicroTenantID}
}\n
```

Private Cloud Controller Status

```
{
    "LogTimestamp": %J{LogTimestamp:epoch},
    "Customer": %j{Customer},
    "SessionID": %j{SessionID},
    "SessionType": %j{SessionType},
    "SessionStatus": %j{SessionStatus},
    "Version": %j{Version},
    "PackageVersion": %j{PackageVersion},
    "Platform": %j{Platform},
    "ZEN": %j{ZEN},
    "PrivateCloudController": %j{PrivateCloudController},
    "PrivateCloudControllerGroup": %j{PrivateCloudControllerGroup},
    "PrivateIP": %j{PrivateIP},
    "PublicIP": %j{PublicIP},
    "Latitude": %f{Latitude},
    "Longitude": %f{Longitude},
    "CountryCode": %j{CountryCode},
    "TimestampAuthentication": %J{TimestampAuthentication:epoch},
    "TimestampUnAuthentication": %J{TimestampUnAuthentication:epoch},
    "CPUUtilization": %d{CPUUtilization},
```

(continues on next page)

(continued from previous page)

```
    "MemUtilization": %d{MemUtilization},
    "InterfaceDefRoute": %j{InterfaceDefRoute},
    "DefRouteGW": %j{DefRouteGW},
    "PrimaryDNSResolver": %j{PrimaryDNSResolver},
    "HostUpTime": %J{HostUpTime:epoch},
    "PrivateCloudControllerStartTime": %J{PrivateCloudControllerStartTime:epoch},
    "NumOfInterfaces": %d{NumOfInterfaces},
    "BytesRxInterface": %d{BytesRxInterface},
    "PacketsRxInterface": %d{PacketsRxInterface},
    "ErrorsRxInterface": %d{ErrorsRxInterface},
    "DiscardsRxInterface": %d{DiscardsRxInterface},
    "BytesTxInterface": %d{BytesTxInterface},
    "PacketsTxInterface": %d{PacketsTxInterface},
    "ErrorsTxInterface": %d{ErrorsTxInterface},
    "DiscardsTxInterface": %d{DiscardsTxInterface},
    "TotalBytesRx": %d{TotalBytesRx},
    "TotalBytesTx": %d{TotalBytesTx},
    "MicroTenantID": %j{MicroTenantID}
}\n
```

Browser Access

```
{
    "LogTimestamp":%J{LogTimestamp:epoch},
    "ConnectionID":%j{ConnectionID},
    "Exporter":%j{Exporter},
    "TimestampRequestReceiveStart":%J{TimestampRequestReceiveStart:epoch},
    "TimestampRequestReceiveHeaderFinish":
↪%J{TimestampRequestReceiveHeaderFinish:epoch},
    "TimestampRequestReceiveFinish":%J{TimestampRequestReceiveFinish:epoch},
    "TimestampRequestTransmitStart":%J{TimestampRequestTransmitStart:epoch},
    "TimestampRequestTransmitFinish":%J{TimestampRequestTransmitFinish:epoch},
    "TimestampResponseReceiveStart":%J{TimestampResponseReceiveStart:epoch},
    "TimestampResponseReceiveFinish":%J{TimestampResponseReceiveFinish:epoch},
    "TimestampResponseTransmitStart":%J{TimestampResponseTransmitStart:epoch},
    "TimestampResponseTransmitFinish":%J{TimestampResponseTransmitFinish:epoch},
    "TotalTimeRequestReceive":%d{TotalTimeRequestReceive},
    "TotalTimeRequestTransmit":%d{TotalTimeRequestTransmit},
    "TotalTimeResponseReceive":%d{TotalTimeResponseReceive},
    "TotalTimeResponseTransmit":%d{TotalTimeResponseTransmit},
    "TotalTimeConnectionSetup":%d{TotalTimeConnectionSetup},
    "TotalTimeServerResponse":%d{TotalTimeServerResponse},
    "Method":%j{Method},"Protocol":%j{Protocol},
    "Host":%j{Host},
    "URL":%j{URL},
    "UserAgent":%j{UserAgent},
    "XFF":%j{XFF},
    "NameID":%j{NameID},
    "StatusCode":%d{StatusCode},
    "RequestSize":%d{RequestSize},
```

(continues on next page)

(continued from previous page)

```
    "ResponseSize":%d{ResponseSize},
    "ApplicationPort":%d{ApplicationPort},
    "ClientPublicIp":%j{ClientPublicIp},
    "ClientPublicPort":%d{ClientPublicPort},
    "ClientPrivateIp":%j{ClientPrivateIp},
    "Customer":%j{Customer},
    "ConnectionStatus":%j{ConnectionStatus},
    "ConnectionReason":%j{ConnectionReason},
    "Origin":%j{Origin},
    "CorsToken":%j{CorsToken}
}\n
```

User Activity

```
{
    "LogTimestamp": %J{LogTimestamp:epoch},
    "Customer": %j{Customer},
    "SessionID": %j{SessionID},
    "ConnectionID": %j{ConnectionID},
    "InternalReason": %j{InternalReason},
    "ConnectionStatus": %j{ConnectionStatus},
    "IPProtocol": %d{IPProtocol},
    "DoubleEncryption": %d{DoubleEncryption},
    "Username": %j{Username},
    "ServicePort": %d{ServicePort},
    "ClientPublicIP": %j{ClientPublicIP},
    "ClientPrivateIP": %j{ClientPrivateIP},
    "ClientLatitude": %f{ClientLatitude},
    "ClientLongitude": %f{ClientLongitude},
    "ClientCountryCode": %j{ClientCountryCode},
    "ClientZEN": %j{ClientZEN},
    "Policy": %j{Policy},
    "Connector": %j{Connector},
    "ConnectorZEN": %j{ConnectorZEN},
    "ConnectorIP": %j{ConnectorIP},
    "ConnectorPort": %d{ConnectorPort},
    "Host": %j{Host},
    "Application": %j{Application},
    "AppGroup": %j{AppGroup},
    "Server": %j{Server},
    "ServerIP": %j{ServerIP},
    "ServerPort": %d{ServerPort},
    "PolicyProcessingTime": %d{PolicyProcessingTime},
    "ServerSetupTime": %d{ServerSetupTime},
    "TimestampConnectionStart": %J{TimestampConnectionStart:epoch},
    "TimestampConnectionEnd": %J{TimestampConnectionEnd:epoch},
    "TimestampCATx": %J{TimestampCATx:epoch},
    "TimestampCARx": %J{TimestampCARx:epoch},
    "TimestampAppLearnStart": %J{TimestampAppLearnStart:epoch},
    "TimestampZENFirstRxClient": %J{TimestampZENFirstRxClient:epoch},
```

(continues on next page)

(continued from previous page)

```
    "TimestampZENFirstTxClient": %J{TimestampZENFirstTxClient:epoch},
    "TimestampZENLastRxClient": %J{TimestampZENLastRxClient:epoch},
    "TimestampZENLastTxClient": %J{TimestampZENLastTxClient:epoch},
    "TimestampConnectorZENSetupComplete":
↪%J{TimestampConnectorZENSetupComplete:epoch},
    "TimestampZENFirstRxConnector": %J{TimestampZENFirstRxConnector:epoch},
    "TimestampZENFirstTxConnector": %J{TimestampZENFirstTxConnector:epoch},
    "TimestampZENLastRxConnector": %J{TimestampZENLastRxConnector:epoch},
    "TimestampZENLastTxConnector": %J{TimestampZENLastTxConnector:epoch},
    "ZENTotalBytesRxClient": %d{ZENTotalBytesRxClient},
    "ZENBytesRxClient": %d{ZENBytesRxClient},
    "ZENTotalBytesTxClient": %d{ZENTotalBytesTxClient},
    "ZENBytesTxClient": %d{ZENBytesTxClient},
    "ZENTotalBytesRxConnector": %d{ZENTotalBytesRxConnector},
    "ZENBytesRxConnector": %d{ZENBytesRxConnector},
    "ZENTotalBytesTxConnector": %d{ZENTotalBytesTxConnector},
    "ZENBytesTxConnector": %d{ZENBytesTxConnector},
    "Idp": %j{Idp},"ClientToClient": %j{c2c},
    "ClientCity": %j{ClientCity},
    "MicroTenantID": %j{MicroTenantID},
    "AppMicroTenantID": %j{AppMicroTenantID},
    "Platform": %j{Platform},
    "Hostname": %j{Hostname}
}\n
```

Configuring Zscaler ZPA log ingestion in Scrutinizer

After adding the log receiver configuration, configure Scrutinizer to receive the logs:

1. In the Scrutinizer web interface, navigate to **Admin** > **Integrations** > **Flow Log Ingestion**.
2. Click the **+** icon, then select *Zscaler ZPA* in the tray.
3. In the secondary tray, configure the following details:
   - *Log Source*: Select the log type that was set to be streamed
   - *Log Downloader*: IP address of the Scrutinizer server/collector that will download the logs (must match the address entered for the log receiver configuration)
   - *Log Downloader Port*: Port to use to receive logs on the downloader (must match the port entered for the log receiver configuration)
   - *Worker Count*: Enter `1` (can be increased if log streaming rate seems slow; recommended maximum of 10)
   - *Flow Collectors*: Select the Scrutinizer collector(s) that will receive the logs as standard flows (in *distributed clusters*, a remote collector is recommended for this role)
4. Click **Save** to add the log stream with the current settings.

Once added, the log stream will be listed in the main **Admin** > **Integrations** > **Flow Log Ingestion** view under the downloader IP address and port configured. If multiple log receivers were created to stream different ZPA log types, each one will require a matching configuration in Scrutinizer.

> **Note**
>
> ZPA log data collection can be tested by running a new report and checking if the *Zscaler ZPA* report type category is available. ZPA log exporters will be listed using the IP address of the App Connector forwarding logs to Scrutinizer.

Troubleshooting

If the **Admin** > **Resources** > **Exporters** view does not list exporters matching the log stream(s) set up for ingestion, check the following for issues:

- Verify that the downloader was successfully configured by looking for the following message (or something similar) in `/home/plixer/scrutinizer/files/logs/zscaler_log.json` on that server/collector and confirm that the `bind` value matches the configured TCP port:

```
{"level":"warn","pid":2059650,"bind":":10000","time":
↪"2025-06-16T14:34:50.916-04:00","caller":"/
↪builds/plixer-products/scrutinizer/application/golang/zscaler/sink-tcp-server.go:131
↪","message":"listening for connections"}
```

- Verify that the source log stream has been correctly configured.
- Check the collector log file in `/home/plixer/scrutinizer/files/logs/` for errors.
- Check `zscaler_log.json` for other possible source-side issues.

For further assistance, contact *Plixer Technical Support*.

Overloaded collectors/downloaders

The *Unresourced - Enabled* status in the Admin > Resources > Exporters view indicates that a log source is being temporarily disabled/paused due to insufficient resources.

The following are potential solutions for an overloaded collector:

- If the collector is a VM, allocate additional resources (starting with CPU cores) to it.
- If the collector is ingesting logs from only one log stream, distribute the logs across multiple streams, which can then be assigned to different collectors.
- If the collector is ingesting logs from multiple log streams, distribute the streams across multiple collectors.
- If the collector license has a flow rate limit, the license may need to be upgraded.

> **Note**
>
> - Sources that are tagged as *Disabled* may have been automatically disabled (last-in/first-out order) due to the license exporter count limit.
> - In distributed deployments, it is recommended to start with a 1:1 pairing of sources and collectors.

4.5.3.9 Upgrades and updates

To ensure a consistently feature-rich and secure experience, all supported versions of Scrutinizer will continuously be updated. When installed, update packages may add new features, improve existing functionality, and/or apply patches for emerging security threats. All update packages will have been applied to Plixer's own QA servers and extensively tested before they are made available.

> **Important**
>
> While it is possible to install Scrutinizer update packages without assistance, it is highly recommended to contact *Plixer Technical Support* and allow our engineers to guide you through the process.

On this page:

Update preparations *Update preparations* Version upgrades *Version upgrades* General and CVE patches *General and CVE patches* Vulnerability patch verification *Vulnerability patch verification*

Update preparations

Before attempting to install any type of update package, the following procedures should be observed:

1. Verify that the version currently installed can be upgraded to the target version (e.g., v18.20 or v19.x -> v19.4.0).
2. Back up the current install:
   - Virtual appliances: Take a snapshot, ideally with the appliance powered off.
   - Hardware appliances: Perform a *full* or *configuration* backup. For further details, see the *Backups* subsection of this documentation or contact *Plixer Technical Support*.
3. **Hardware appliances only** - Log in to iDRAC and perform a hardware health check. Any hardware issues discovered should be escalated to Dell for resolution. A reboot is also recommended as an additional check for underlying hardware issues.
4. Confirm that all Scrutinizer collectors/servers have access to `https://files.plixer.com`. This check can be performed by downloading the checksum file using the following command:

   ```
   curl -O https://
   ↪files.plixer.com/plixer-repo/scrutinizer/19.7.2/scrutinizer-install.run.sha256
   ```

   For Scrutinizer deployments that do not have internet access, download the file from the `REPO_HOST_IP` for the *offline yum/dnf repository* instead.
5. Verify that the Scrutinizer server/appliance meets the target version's *minimum resource requirements* based on the following details:
   - Flows per second
   - Number of active Exporters
   - CPU (number of cores, clock speeds)
   - Amount of RAM
   - Disk speed and RAID type
   - Flow Analytics algorithms enabled

   > **Important**
   >
   > Do **not** proceed with the upgrade if the server is underprovisioned. Contact *Plixer Technical Support* for assistance with tuning system parameters or allocating additional resources to the server.
6. Obtain a valid license key for the upgrade if one has not been acquired.
7. Delete any older versions of `scrutinizer-installer.run` on the Scrutinizer instance. This will prevent them from being used instead of the correct installer.

8. Run `crontab -e` and inspect the table for lines containing `* * * * * /home/Plixer/scrutinizer/files/collector_restart.sh`. These should be commented out by adding a `#` at the beginning of the line to prevent scheduled restarts from interfering with the upgrade process.
9. [*Distributed cluster* upgrades only] - If there are Palo Alto firewalls configured for the cluster, whitelist the connections between the primary reporter and remote collectors. This will prevent the firewall from identifying the ~113 SSH connections created during the collector registration process as a threat. Alternatively, the rate at which the SSH connections are established can be slowed down by adding `sleep 5` to the `/home/plixer/.bashrc` file on each remote collector.
10. [*AWS flow log integration* only] - As of version 19.2, Scrutinizer requires four log fields to be configured for AWS flow log collection: `log-status`, `vpc-id`, `interface-id`, and `flow-direction`. For further details, see the *AWS flow log integration guide*.

These steps are meant to identify and resolve any underlying issues with the current Scrutinizer install and help ensure that the upgrade will be applied without issue.

Once completed, follow the *appropriate upgrade guide* to update Scrutinizer to the latest version.

> **Note**
>
> All install logs will be saved to `/var/log/Scrutinizer-Install.log`.

Changed in version 19.1: `plixer` is the recommended OS user for command line access. The `root` user is no longer required.

Version upgrades

Version upgrades update Scrutinizer to the latest major or minor release (e.g., 19.4) and include significant improvements over the previous version. These upgrades may include additional functionality, performance enhancements, and/or QoL improvements, in addition to implementing fixes for certain types of issues.

Latest Scrutinizer release

After completing the recommended *update preparations*, follow the instructions below to upgrade Scrutinizer to the latest version.

> **Note**
>
> - Only deployments on v19.5.0 and above can be upgraded directly to v19.7.2 and beyond. For versions < 19.5.0, follow the steps in *these guides* to upgrade to the required Scrutinizer 19.5.x release before upgrading to the latest version.
> - After Scrutinizer has been upgraded to v19.7.0 or higher, contact *Plixer Technical Support* for a new Plixer One Core or Plixer One Enterprise key to enable AI features.
> - If the Scrutinizer server being upgraded does not have Internet access, an internal NTP server can be configured by running the following:
>
> ```
> sed -i -e '/^pool/aserver NTP_ADDRESS' -e 's/^pool/#&/' /etc/chrony.conf
> ```
>
> - For AWS deployments, contact *Plixer Technical Support* to obtain the latest AMI installer.

Upgrade procedure

To download and install the latest version upgrade for Scrutinizer, follow these steps:

View instructions

1. SSH to the **primary reporter** as the `plixer` user:

   ```
   ssh plixer@SCRUTINIZER_IP
   ```

2. Start a new tmux session (to maintain the upgrade session if the SSH connection is lost):

   ```
   tmux new -s upgrade
   ```

3. Download the installer and checksum file for the latest version:

   ```
   curl -O https://
   ↪files.plixer.com/plixer-repo/scrutinizer/19.7.2/scrutinizer-install.run
   curl -O https://
   ↪files.plixer.com/plixer-repo/scrutinizer/19.7.2/scrutinizer-install.run.sha256
   ```

   Note

   For Scrutinizer deployments that do not have Internet access, use the `REPO_HOST_IP` for the *offline yum/dnf repository* instead. The `-k` flag can also be added to ignore certificates if necessary.

4. Verify the checksum:

   ```
   sha256sum -c scrutinizer-install.run.sha256
   ```

5. Set the correct permissions for the installer:

   ```
   chmod 755 scrutinizer-install.run
   ```

6. Run the installer as the `plixer` user:

   ```
   ./scrutinizer-install.run
   ```

   For offline upgrades, use:

   ```
   REPO_HOST=REPO_HOST_IP ./scrutinizer-install.run -- -k
   ```

7. [*Distributed cluster* upgrades only] When prompted for the authentication method to use for remote collectors in the cluster, enter either `existing` (recommended) or `passwords`.

8. After the installer finishes running, execute the following heartbeat checks to verify communication between nodes:

   ```
   scrut_util --check heartbeat --type database
   scrut_util --check heartbeat --type api
   ```

If the heartbeat checks are successful, the upgrade is complete.

Offline upgrades

To upgrade Scrutinizer collectors/servers that are unable to access the default yum/dnf repository on `https://files.plixer.com/plixer-repo/scrutinizer/19.7.2`, an offline repository will need to be set up on the local network. The local repository can be hosted on the primary Scrutinizer server or another host on the network.

To set up the offline repository on the primary Scrutinizer server (with IP address `REPO_HOST_IP`), follow these steps:

View instructions

1. Download the offline repo package and checksum file on a host with Internet access:

```
curl -O https://files.plixer.com/plixer-repo/scrutinizer/19.7.2_offline.tgz
curl -O https://files.plixer.com/plixer-repo/scrutinizer/19.7.2_offline.tgz.sha256
```

2. Start an SSH session with the primary reporter as the `plixer` user:

```
ssh plixer@REPO_HOST_IP
```

3. Confirm that `/var/db/big` has at least 84 GB of free disk space:

```
df -h --output='avail' /var/db/big
```

4. Create a new directory for the offline installation files and set the correct permissions to give the plixer user access to it:

```
sudo mkdir -p /var/db/big/offline
sudo chown plixer:plixer /var/db/big/offline
```

5. On the Internet-connected host, copy the offline bundle and checksum file downloaded in step 1 to the repo host:

```
scp 19.7.2_offline.tgz* plixer@REPO_HOST_IP:/var/db/big/offline/
```

6. On the repo host, validate the checksum:

```
(cd /var/db/big/offline/ ; sha256sum -c 19.7.2_offline.tgz.sha256)
```

7. Extract the repository:

```
tar -zxvf /var/db/big/offline/19.7.2_offline.tgz -C /var/db/big/offline
```

8. Create a link to the offline repo in a directory accessible to the web server:

```
sudo -u webapp ln -sf /var/db/big/offline/plixer-repo /home/webapp/html/
```

> **Note**
>
> For versions before 19.7.0, use `ln -sf /var/db/big/offline/plixer-repo /home/plixer/scrutinizer/html/plixer-repo` instead.

9. Export the repo host's IP address:

```
export REPO_HOST=local
export LOCAL_REPO_BASEDIR=/var/db/big/offline
```

Once the offline repository has been set up, follow *these steps* to proceed with the upgrade.

Plixer ML Engine

Review/complete the *recommended update preparations*, and then follow the steps below to upgrade an ML Engine deployment from v19.5.0 to v19.7.0:

Important

Scrutinizer 19.7.2 requires Plixer ML Engine deployments to be upgraded to v19.5.0 or higher. v19.7.0 is recommended.

View instructions

1. SSH to the ML VM (i.e., the host used for management/deployment) as the `plixer` user.
2. Download the installer for the latest version:

```
curl -o plixer-machine-learning-update.run https://
↪files.plixer.com/scripts/plixer-machine-learning/release/19.7.0/plixer-machine-learning-update
```

3. Download the checksum file and validate the integrity of `plixer-machine-learning-update.run`:

```
curl -o plixer-machine-learning-checksums.txt https://
↪files.plixer.com/scripts/plixer-machine-learning/release/19.7.0/plixer-machine-learning-checks
cat plixer-machine-learning-checksums.txt
sha256sum plixer-machine-learning-update.run
```

4. Set the correct permissions for the installer:

```
chmod +x plixer-machine-learning-update.run
```

5. Run the installer as the `plixer` user:

```
STAGE="release"
VERSION="UPGRADE_VER"
STAGE=$STAGE ./plixer-machine-learning-update.run
```

After the installer script completes running, `setup.sh` will automatically be run to pull in any configuration changes and redeploy pods with new images. AWS ML engine deployments will require *these additional steps* to be upgraded.

Note

If any changes were previously made to `pxi-settings.yaml`, `azure.tfvars`, `aws.tfvars`, or `vsphere.tfvars`, the file(s) will be retained even if the upgrade package includes a newer version of the file. The updated file will instead be saved with a `.dpk-dist` extension, and any necessary edits should be migrated before it is used to overwrite the old configuration/tfvars file.

Once the upgrade process is complete, wait for the **rke2-server** service to restart. This sequence can be monitored by running:

```
journalctl -xeu rke2-server -f
```

Upgrading AWS ML engine deployments

After the ML VM (management/deployment host) has been upgraded, follow these steps to upgrade an AWS ML engine cluster to 19.7.0:

View instructions

1. Copy the contents of ~/common/kubernetes/aws/.terraform.lock.hcl to ~/common/kubernetes/aws/.terraform.lock.hcl.bak on the ML VM.
2. Navigate to `/home/plixer/common/kubernetes` and run the Kubernetes cluster deployment script:

   ```
   ./01_aws_infrastructure.sh
   ```

3. Verify that there were no errors during the script.
4. Update the `kube_version` value to `"1.32"` in `/home/plixer/common/kubernetes/aws.tfvars`.
5. Run `01_aws_infrastructure.sh` again and check for errors.
6. Update the `kube_version` value to `"1.33"`.
7. Run `01_aws_infrastructure.sh` again and check for errors.
8. Navigate to the `/home/plixer/ml` and re-initialize the cluster:

   ```
   ./setup.sh --redeploy
   ```

The **rke2-server** service will restart after configurations are updated and pods are re-deployed with new images.

Additional notes for ML Engine upgrades from v19.4.0 to v19.5.0+

- Scrutinizer 19.6.0+ features a new management/configuration UI for the ML engine. Any v19.4.0 ML engine deployments will need to be *re-registered* in the Scrutinizer web interface and then upgraded (not re-deployed) to v19.5.0 or higher.
- If the ML engine is deployed as a standalone/single-node VM, new Docker images will be downloaded (may take several minutes) after the package updates. This step is skipped for cloud deployments.

Pre-19.5.x Scrutinizer deployments

The Scrutinizer 19.5.0 upgrade includes the *migration to Oracle Linux 9*, which will be required for all new versions/releases going forward. Deployments on older versions must first be upgraded to the latest v19.5.x release before being upgraded further.

The following guides provide instructions for the required upgrade(s):

Upgrading from older versions to Scrutinizer 19.4.0 (required to upgrade to v19.5.3/v19.5.4)

View guide

Pre-v19.4.0 Scrutinizer deployments must first be upgraded to v19.4.0 before being upgraded to v19.5.3 (or v19.5.4 for AWS AMI appliances), which includes the migration to Oracle Linux 9.

Follow *these instructions* to download the v19.4.0 installer (replace `19.7.2` with `19.4.0` in the download URLs) and apply the update. Once done, proceed with *upgrading to the latest v19.5.x release*.

Note

- When upgrading an appliance that was previously upgraded from v18.20, the installer script will ask whether to delete the `data.old` backup created during that upgrade. Since a more recent backup *should be created* before the current upgrade process, this file can safely be deleted.
- If a *distributed cluster* is being upgraded from v18.20, the prompt to create a new *Plixer control key* should be left blank unless encrypted keys are required. Additionally, `passwords` should be selected in the next step, when prompted for the login method to use for remote collectors.

Upgrading from 19.4.0 to 19.5.3

View guide

Follow the steps outlined below to upgrade a Scrutinizer deployment on v19.4.0 or above to v19.5.3.

To upgrade an AWS AMI from 19.4.0 to v19.5.4, follow *this guide* instead. For versions < 19.4.0, refer to *this guide* to upgrade to v19.4.0 before proceeding.

Note

- The upgrade will take **at least one hour to complete**.
- The `plixer` user SSH password will be needed during the upgrade. If necessary, it can be reset when the OS upgrade script is run.
- If root SSH login is enabled on the Scrutinizer server, it will be disabled as part of the upgrade.
- If upgrading from v19.5.0 or above, proceed directly to *upgrading to Scrutinizer 19.5.3*.
- If the Scrutinizer server is able to access `files.plixer.com`, the `REPO_HOST` variable should be set to `files.plixer.com` for the steps outlined below. For *offline upgrades*, the IP address of the offline repo should be used instead.

For assistance or clarifications, contact *Plixer Technical Support*.

Upgrade process

The process of upgrading a v19.4.0 Scrutinizer server to v19.5.3 involves the following steps:

- *Backing up the current install's database and server-specific files*
- Downloading the operating system upgrade script, `olmigrate.run`, and running it a total of four times (with a reboot between runs). **This only applies if upgrading from v19.4.0.**
- Downloading and running the Scrutinizer v19.5.3 installation script (`scrutinizer-install.run`)
- Verifying that the current install's data has been successfully migrated after v19.5.3 is installed

Pre-upgrade preparation

- [Hardware appliances] Create a *full backup* of the current Scrutinizer install and store it on an external system/drive.
- [Virtual appliances] Back up the current Scrutinizer install by taking a VM snapshot.
- Review the *general update preparation guide* and complete any steps that apply.
- [Offline upgrades] If the Scrutinizer server does not have access to `files.plixer.com`, *set up an offline repository for this upgrade*.

Distributed cluster upgrades

Remote collectors in *distributed clusters* must be reverted to standalone appliances before being individually upgraded to v19.5.3:

View instructions

1. Navigate to Admin > Resources > Collectors and delete all remote collectors.
2. SSH to each remote collector as the `plixer` user and register it as a standalone appliance:

```
scrut_util --set selfregister --reset
```

3. Verify that each appliance is now running in standalone mode (no other addresses under `collector_ips`):

```
scrut_util --check dist_info
```

When done, proceed with the OS migration and v19.5.3 upgrade for each node, and then *rebuild the distributed cluster*.

OS migration

Once all preparation steps have been completed, follow these steps to migrate the v19.4.0 appliance to the new operating system:

View instructions

Important

- For offline upgrades, `REPO_HOST` should point to the IP address of the *offline repo* instead of `files.plixer.com`.
- In *distributed clusters*, complete the upgrade for all remote collectors before upgrading the primary reporter.
- To verify the current progress of the OS upgrade at any time:

  ```
  cat /etc/motd
  ```

  or check versions between runs (`NAME=` and `VERSION=` lines):

  ```
  cat /etc/os-release
  ```

- If any errors are encountered during the upgrade process, run the following to collect log files:

  ```
  sudo tar -czf /tmp/olmigrate_logs.tar.gz /var/log/olmigration/ /var/log/leapp/␣
  ↪/var/log/messages /var/log/Scrutinizer-Install.log
  ```

  Afterwards, move `/tmp/olmigrate_logs.tar.gz` off the server before reverting. *Plixer Technical Support* will require the logs to better assist you with any issues.

1. SSH to the v19.4.0 server to be upgraded as the `plixer` user.
2. Verify that the current working directory is correct (`plixer`):

```
cd /home/plixer/
```

3. Download the OS upgrade script and its checksum file:

```
REPO_HOST=files.plixer.com
curl -k -o olmigrate.run https://
↪$REPO_HOST/plixer-repo/scrutinizer/19.5.3/olmigrate.run
curl -k -o olmigrate.run.sha256 https://
↪$REPO_HOST/plixer-repo/scrutinizer/19.5.3/olmigrate.run.sha256
```

4. Validate the integrity of `olmigrate.run`:

```
sha256sum -c olmigrate.run.sha256
```

5. Update permissions for the OS upgrade script:

```
chmod a+x olmigrate.run
```

6. Run the `olmigrate.run` script a total of four times:

```
REPO_HOST=files.plixer.com ./olmigrate.run -- -k
```

> **Important**
>
> Reboots between runs of the OS upgrade script (`olmigrate.run`) can take a long time. Before trying to reconnect to the server, start a `PING` to the Scrutinizer IP address and wait for it to become available again. **Do NOT manually reboot the server**.

After the fourth `olmigrate.run` run (there will be no reboot), the OS migration will be complete.

Upgrading to Scrutinizer 19.5.3

Once the appliance is on the *new OS*, Scrutinizer can be upgraded to v19.5.3 as follows:

View instructions

1. Change directories to `/tmp`:

```
cd /tmp/
```

2. Download the Scrutinizer v19.5.3 installation script and its checksum file:

```
REPO_HOST=files.plixer.com
curl -k -o scrutinizer-install.run https://
↪$REPO_HOST/plixer-repo/scrutinizer/19.5.3/scrutinizer-install.run
curl -k -o scrutinizer-install.run.sha256 https://
↪$REPO_HOST/plixer-repo/scrutinizer/19.5.3/scrutinizer-install.run.sha256
```

3. Validate the integrity of `scrutinizer-install.run`:

```
sha256sum -c scrutinizer-install.run.sha256
```

4. Update permissions for the installation script:

```
chmod a+x scrutinizer-install.run
```

5. Run `scrutinizer-install.run` to begin the upgrade to Scrutinizer v19.5.3:

```
REPO_HOST=files.plixer.com ./scrutinizer-install.run -- -k
```

6. After the installation script finishes running, reboot the appliance:

```
sudo shutdown -r now
```

7. After the reboot, run the following commands to verify that the system is in working order:

```
scrut_util --check heartbeat --type database
scrut_util --check heartbeat --type api
```

> **Important**
>
> For *distributed clusters*, the heartbeat checks should only be run on remote collectors *after the primary reporter has been upgraded, and the cluster has been reestablished*.

If the heartbeat checks are successful, then the Scrutinizer appliance has been successfully upgraded to v19.5.3.

Offline upgrades to v19.5.3

The following instructions for setting up an offline repo are intended for upgrading to Scrutinizer v19.5.3 only.

View instructions

1. *Deploy a new Scrutinizer VM* and *assign an IP address to it*.
2. SSH to the VM as the `plixer` user:

```
ssh plixer@SCRUTINIZER_VM_IP
```

3. Create the offline repo directory and assign it the correct permissions:

```
sudo mkdir /var/db/big/offline
sudo chown plixer:plixer /var/db/big/offline
```

4. Download the offline tar file for 19.5.3 and its checksum file:

```
curl -o /var/db/big/offline/19.5.3_offline.tgz https://
↪files.plixer.com/plixer-repo/scrutinizer/19.5.3_offline.tgz
curl -o /var/db/big/offline/19.5.3_offline.tgz.sha256 https://
↪files.plixer.com/plixer-repo/scrutinizer/19.5.3_offline.tgz.sha256
```

5. Validate the integrity of `19.5.3_offline.tgz`:

```
sha256sum -c /var/db/big/offline/19.5.3_offline.tgz.sha256
```

6. Extract the offline tar file:

```
cd /var/db/big/offline
tar xvf 19.5.3_offline.tgz
```

7. Create a symlink in the html directory to the offline repo:

```
ln -s /var/db/big/offline/plixer-repo /home/plixer/scrutinizer/html/plixer-repo
```

After the offline repo has been set up, the VM's IP address should be used in place of `files.plixer.com` for `REPO_HOST` in the *upgrade instructions*.

Upgrading from Scrutinizer 19.4.0 to 19.5.4 (AMI only)

View guide

Follow the steps outlined below to upgrade a Scrutinizer AMI on v19.4.0 or above to v19.5.4.

For versions < 19.4.0, refer to *this guide* to upgrade to v19.4.0 before proceeding.

Note

- The upgrade will take **at least one hour to complete**.
- The `plixer` user SSH password will be needed during the upgrade.
- If root SSH login is enabled on the Scrutinizer server, it will be disabled as part of the upgrade.
- The new v19.5.4 instance must be in the same availability zone as the original v19.4.0 machine. Volumes outside the current availability zone will not be accessible from the AWS console.

Distributed cluster upgrades

Remote collectors in *distributed clusters* must be reverted to standalone appliances before being individually upgraded to v19.5.4:

View instructions

1. Navigate to Admin > Resources > Collectors and delete all remote collectors.
2. SSH to each remote collector as the `plixer` user and register it as a standalone appliance:

```
scrut_util --set selfregister --reset
```

3. Verify that each appliance is now running in standalone mode (no other addresses under `collector_ips`):

```
scrut_util --check dist_info
```

When done, proceed with upgrading each node as described below, and then *rebuild the distributed cluster*.

For assistance or clarifications, contact *Plixer Technical Support*.

Upgrade process

The process of upgrading a Scrutinizer 19.4.0 Scrutinizer AMI to v19.5.4 involves the following steps:

- Backing up the current Scrutinizer install by taking a VM snapshot
- Deploying a new v19.5.4 AMI appliance
- Copying the `dbexport.sh` file from the new v19.5.4 appliance to the current v19.4.0 appliance
- Detaching the storage volume from the v19.4.0 instance (using `dbexport.sh` and running as the `root` user)
- Attaching the storage volume to the new v19.5.4 instance (using `dbimport.sh` and running as the `root` user)
- Verifying that the v19.4.0 data has been successfully migrated after v19.5.4 is installed

Expanding storage

AMI deployments will require additional storage to be upgraded to v19.5.4.

View instructions

To verify whether the Scrutinizer 19.4.0 AMI instance is running on the default sizing, run the following:

```
df -h
```

If the output does not list a line that includes `vg_scrut-lv_db`, contact *Plixer Technical Support* for assistance with expanding storage before proceeding.

Upgrading to Scrutinizer 19.5.4

View instructions

1. Copy the following file from the new v19.5.4 appliance to your current v19.4.0 appliance:

   ```
   /home/plixer/scrutinizer/files/dbimport/dbexport.sh
   ```

2. Run the following command to make `dbexport.sh` executable:

   ```
   sudo chmod +x dbexport.sh
   ```

3. SSH to the 19.4.0 appliance as the `plixer` user, and then navigate to the location where `dbexport.sh` was saved.
4. Run the script to prepare the 19.4.0 storage volume to be detached:

   ```
   sudo ./dbexport.sh exportdb
   ```

5. Shut down the Scrutinizer v19.4.0 instance.
6. In the AWS EC2 management page, navigate to the *Volumes* page.
7. In the *Volume Management* page, select the storage volume, click the **Actions** menu, and then select **Detach volume**. It may take a minute for the storage volume to go from *In use* to *Available*.
8. Once the detached storage volume(s) are marked as *Available* (it may take several minutes), attach it to the Scrutinizer v19.5.4 instance. Refer to STEP 6 of the *storage expansion instructions*.
9. After the storage volume(s) have been attached, SSH to the 19.5.4 instance as the `plixer` user.
10. Run the following to import and set up the database on the 19.5.4 instance.

```
/home/plixer/scrutinizer/files/dbimport/dbimport.sh importdb <device name for the␣
↪storage volume, e.g. /dev/xvdg>
```

Use the `lsblk` and `show partitions` commands to get the correct partition/device name to use. Once the script completes running, Scrutinizer will run a self-register reset that requires user input for verification.

11. *Add a new license key* to fully activate your Scrutinizer v19.5.4 instance.

> **Note**
>
> - If there are multiple volumes listed after `dbexport.sh` completes running, all volumes will need to be detached from the v19.4.0 instance and attached to the v19.5.4 instance.

- At the end of the output from dbexport, the volumes that are part of the volume group for the database are listed. If the volume group contains more than one volume, the output will list all of those volumes, which will need to be detached and then attached to the Scrutinizer v19.5.4 instance.
- When you first log in to the v19.5.4 UI to add a new Scrutinizer license, you must use the UI admin password for the v19.4.0 AWS instance. Alternatively, you can reset the UI admin password in scrut_util first.

General and CVE patches

From time to time, customers may be notified that general and/or CVE patches are available for the Scrutinizer version they are currently running. These patches typically address noncritical system issues and/or improve protections against new security threats.

Note

General and CVE patches do not increment the Scrutinizer version number.

To apply these updates, follow the *version upgrade instructions* to download and run the latest installer for the current Scrutinizer version. Going through the *standard update preparations* is also highly recommended.

When run, the installer will automatically download and apply all available patches.

Vulnerability patch verification

Some vulnerability scanning and auditing solutions may report vulnerabilities that have already been patched in the most recent update. This is typically the combined result of a backported security patch and the tool only scanning for component version numbers.

If this happens, there are two ways to verify the validity of the vulnerability report:

- Check the package changelog for the CVE identifier/number of the vulnerability (e.g., CVE-2017-3169)
- Download and install the latest OVAL definitions from oval.cisecurity.org/repository, which will allow any compatible tools to determine the status of vulnerabilities, even when security patches have been backported.

For additional assistance, contact *Plixer Technical Support*.

4.6 Additional Resources

Frequently asked questions

Answers to frequently asked questions

FAQ

4.6.1 Appendices

Alarm policy list

Alarm policy overview

Alarm policy list **Event details**

Alarm policy violation/event details

Event details **FA algorithm list**

FA algorithm overview and recommendations

FA algorithm list **FA algorithm settings**

Configurtion options by algorithm

Algorithm settings **Functional IDs**

Default functional accounts/IDs

Functional IDs **Report types**

Report type details by category

Report types **Required ports**

Firewall port configuration information

Required ports **User permissions**

User group permission overview

User permissions

4.6.2 Changelogs

Plixer ML Engine

Plixer ML Engine updates and version history

Plixer ML Engine changelogs **Replicator**

Replicator updates and version history

Replicator changelogs **Scrutinizer**

Scrutinizer updates and version history

Scrutinizer changelogs

4.6.3 Other References

Terminal utilities

Quick references for OS functions

resource-terminal **Attributions**

Open source and third-party licenses

Third-party attributions **Glossary**

Glossary of terms used in Scrutinizer and Plixer One

Glossary Plixer technical support Plixer Technical Support is available with an active maintenance contract. Contact our support team at:

- **Phone:** +1 (207) 324-8805 ext 4
- **Website:** https://www.plixer.com/support/

4.6.3.1 Appendices

This section contains additional references/guides for Scrutinizer's functional elements.

On this page:

Alarm policy list *Alarm policy list* Event details *Event details* FA algorithm list *FA algorithm list* Algorithm settings *Algorithm settings* Functional IDs *Functional IDs* User permissions *User permissions* Required ports *Required ports* Report types *Report types*

Alarm policy list

The tables below contain basic information for all Plixer One/Scrutinizer alarm policies sorted by their source technologies/components.

Plixer One Core

Scrutinizer

Policy	Category	Description
DNS Command and Control Detection	Command and Control > Application Layer Protocol > DNS	This algorithm monitors the use of DNS TXT messages traversing the network perimeter as detected by FlowPro Defender. DNS TXT messages provide a means of sending information into and out of your protected network over DNS, even when you have blocked use of an external DNS server. This technique is used by malware as a method of controlling compromised assets within your network and to extract information back out. Additionally, some legitimate companies also use this method to communicate as a means to ‘phone home’ from their applications to the developer site. The algorithm will detect inbound, outbound, and bidirectional communications using DNS TXT messages. Thresholds may be set based either on the number of DNS TXT messages or the number of bytes observed in the DNS TXT messages within a three-minute period. The default setting is for any detected traffic to alarm, and alarm aggregation defaults to 120 minutes.To suppress alarms from authorized applications in your network, you may add the domain generating the alarm message to the ‘trusted.domains’ list on FlowPro Defender.
DNS Hits	Command and Control > Application Layer Protocol > DNS	Triggers an alarm when a host initiates an excessive number of DNS queries. This identifies hosts that perform an inordinate number of DNS lookups. To do this, set the flow threshold to a large value that reflects normal behavior on your network. The default threshold is 2500 DNS flows in three minutes. Either the source or destination IP address can be excluded from triggering this alarm.
DNS Server Detection	Command and Control > Application Layer Protocol > DNS	When used with FlowPro Defender, detects new DNS Servers being used on or by your network through analysis of the DNS packets being exchanged between the client and the server. Exclude DNS servers that are authorized for use on the network.
BotNet Detection	Command and Control > Dynamic Resolution	This alarm is generated when a large number of unique DNS name lookups have failed. When a DNS lookup fails, a reply commonly known as NXDOMAIN is returned. By monitoring the number of NXDOMAINs detected as well as the DNS name looked up, behavior normally associated with a class of malware that uses Domain Generation Algorithms (DGAs) can be detected.The default threshold is 100 unique DNS lookup failures (NXDOMAIN) messages in three minutes. Either the source or destination IP address can be excluded from triggering this alarm.
Domain Reputation	Command and Control > Web Service > Bidirectional Communication	Domain reputation provides much more accurate alarming with a dramatic decrease in the number of false positive alarms as compared to IP based Host Reputation. The domain list is provided by Plixer and is updated each hour and currently contains over 400,000 known bad domains.FlowPro Defender performs the actual monitoring, and when it detects a domain with poor reputation, it passes the information to Scrutinizer for additional processing. The default setting is for any detected traffic to alarm, and alarm aggregation defaults to disabled so that all DNS lookups observed will result in a unique alarm.To suppress alarms from authorized applications in your net-

Plixer One Enterprise

Scrutinizer

Policy	Category	Description
Protocol Misdirection	Command and Control > Data Obfuscation > Protocol Impersonation	This algorithm uses NBAR (Network Based Application Recognition) data to identify when the actual application traffic observed on a network port does not match the application expected for that port. For example, if non-HTTP traffic is detected on port 80, or non-SSH traffic is detected on port 22, this may indicate that an attacker is using protocol impersonation to disguise command and control or data exfiltration traffic as legitimate services. The algorithm compares the application tag reported by the flow exporter against a mapping of standard applications to their well-known ports. Known exceptions, such as SSL/HTTPS on port 443, are automatically excluded from detection. Either the client or server IP address can be excluded from triggering this alarm.
Host Watchlist	Command and Control > Web Service > Bidirectional Communication	This algorithm monitors network traffic from hosts that have been placed on an administrator-defined watchlist. When a watched host generates any traffic matching the configured criteria, an alarm is triggered. The watchlist supports filtering by IP address, protocol, and port, allowing administrators to monitor specific types of activity from specific hosts. This is useful for tracking hosts suspected of compromise during an incident response, monitoring sensitive systems, or verifying that remediation efforts have been effective. The alarm reports the protocol and port used by the watched host. The source IP address can be excluded from triggering this alarm.
Ping Scan (Internal)	Discovery > Remote System Discovery	Alerts when a host is suspected of performing a ping scan. A ping scan uses ICMP Echo Requests (ping) to discover what IP addresses are in use on a network. This behavior is commonly demonstrated by attackers attempting to find targets for compromise or lateral movement. The algorithm counts the number of unique destination IP addresses receiving ICMP Echo Requests from a single source within a three-minute period. The default threshold is 32 unique destination hosts. Detection can be configured to monitor internal-to-external and external-to-internal scan directions. Either the source or destination IP address can be excluded from triggering this alarm.
Lateral Movement Attempt	Discovery > System Network Connections Discovery	This algorithm identifies potential lateral movement by correlating network scanning events over time. When a host is detected performing worm-like scanning activity (e.g., contacting many destinations on the same port), the algorithm tracks the targets of that scan. If any of those targets subsequently begin performing similar scanning activity on the same port and protocol, this indicates that the original scan may have resulted in a successful compromise, and the newly infected host is now attempting to spread further.The algorithm monitors for this propagation pattern over a configurable time range (default 28 days), linking the initial scanning event to the subsequent lateral movement. The alarm identifies the original attacker, the compromised host that is now spreading, and the port, protocol, and worm classification involved. Either the source or destination IP address can be excluded from triggering this alarm.
Reverse SSH Shell	Execution > Com-	This algorithm identifies possible reverse SSH tunnels to

Endpoint Analytics

Policy	Category	Description
Endpoint Analytics Info	Endpoint Data	Informational messages from Endpoint Analytics

FlowPro Defender

Policy	Category	Description
Detection of a non-standard protocol or event	Command and Control > Custom Command and Control Protocol	Detects non-standard protocols or events (e.g. use of deprecated or rarely used protocols)
Generic Protocol Command Decode	Command and Control > Custom Command and Control Protocol	Detects generic protocol command decodes (e.g. malformed DHCP options)
Domain Observed Used for C2 Detected	Command and Control > Dynamic Resolution	Detects domains known to be used for malware command and control
Malware Command and Control Activity Detected	Command and Control > Non-Standard Port	Detects malware communicating with an external command and control server
Successful Credential Theft Detected	Credential Access > Credential Dumping	Detects successful attempts at stealing user credentials
A client was using an unusual port	Defense Evasion > Non-Application Layer Protocol	Detects when a client is using an unusual port for a given well-known protocol (e.g. a client sending HTTP requests over a non-standard port)
A suspicious filename was detected	Defense Evasion > Obfuscated Files or Information	A suspicious filename is detected that is often related to known malware families
Detection of a Network Scan	Discovery > Network Service Scanning	Detects network scanning activities (e.g. a large number of requests to different ports on a single machine or multiple machines)
Device Retrieving External IP Address Detected	Discovery > Remote System Discovery	Detects devices retrieving their external IP addresses (e.g. a device making a request to whatismyip services, commonly used in malware recon and exfiltration)
Exploit Kit Activity Detected	Execution > Exploitation for Client Execution	Detects known exploit kit activities
A system call was detected	Execution > System Services	Detects when a potential system call was made (e.g. x86 shellcode found in a network payload)
Executable code was detected	Execution > System Services	Detects when executable binary shellcode is detected in a network payload
FlowPro Event Capture	FlowPro Event Captured	A user defined FlowPro capture rule.
Detection of a Denial of Service Attack	Impact > Endpoint Denial of Service > Application or System Exploitation	Detects Denial of Service (DoS) attacks
Denial of Service	Impact > Network Denial of Service	A known threat vector has been observed that indicated a DoS attempt has been successful
Crypto Currency Mining Activity Detected	Impact > Resource Hijacking	Detects cryptocurrency mining activities (e.g. traffic to known mining pools)
Possibly Unwanted Program Detected	Initial Access > Drive-by Compromise	Detects potentially unwanted programs (e.g. various spyware applications)
Access to a potentially vulnerable web application	Initial Access > Exploit Public-Facing Application	Detects when there is access to a potentially vulnerable web application (e.g. an apache ?M=D directory list attempt)
Web Application Attack	Initial Access > Exploit Public-Facing Application	Detects when a possible web application attack occurs (e.g. a SQL injection attack on a web application or shellcode found in URI)
Targeted Malicious Activity was Detected	Initial Access > Phishing	Fires when targeted malicious activity is detected (e.g. Advanced Persistent Threats (APTs) that try to remain

Plixer Machine Learning

Policy	Category	Description
Data Accumulation	Collection > Data Staged > Local Data Staging	Identifies when an internal host is receiving a disproportionately large volume of data from another internal source, which may indicate staging prior to exfiltration. The algorithm analyzes all inbound traffic to the suspect host over a 32-minute window and calculates the percentage of total bytes received from each source IP. An alarm is triggered when a single internal source accounts for more than 70% of the host's received traffic and the total data received exceeds a configurable byte threshold (default ~50 MB). Either the source or destination IP address can be excluded from triggering this alarm.
ML Engine command and control alert	Command and Control > Non-Standard Port	Uses a supervised machine learning classification model to detect network traffic patterns that match the signatures of known banking trojan families, including Dridex, Emotet, Qakbot, and Trickbot. The model analyzes time-series flow data including bytes-per-packet ratios, destination counts, and port usage patterns over multiple intervals. Before alerting, the algorithm performs several sanity checks: it verifies that the host is not actively port scanning (which can produce similar traffic patterns), confirms that the prominent communication ports involve external IP addresses, and validates that port usage is consistent across the observation window. The alert includes feature importance details describing which traffic characteristics most closely matched the malware signature.
Tunneling through external DNS host	Command and Control > Proxy > External Proxy	Detects when a host is funneling a disproportionate amount of DNS traffic (port 53) to a single external destination, which may indicate the use of a DNS tunnel for covert communication. The algorithm analyzes all outbound DNS traffic from the suspect host over a 32-minute window and sums the total bytes sent to each destination. An alarm is triggered when a single external destination receives 60% or more of the host's total DNS traffic by volume. Either the source or destination IP address can be excluded from triggering this alarm.
Tunneling through external ICMP host	Command and Control > Proxy > External Proxy	Detects when a host is funneling a disproportionate amount of ICMP traffic to a single external destination, which may indicate the use of an ICMP tunnel for covert data transfer. The algorithm analyzes all outbound ICMP traffic from the suspect host over a 32-minute window and sums the total bytes sent to each destination. An alarm is triggered when a single external destination receives 60% or more of the host's total ICMP traffic by volume. Either the source or destination IP address can be excluded from triggering this alarm.
Tunneling through external SSH host	Command and Control > Proxy > External Proxy	Detects when a host is funneling a disproportionate amount of SSH traffic (port 22) to a single external destination, which may indicate the use of an SSH tunnel for covert communication or data exfiltration. The algorithm analyzes all outbound SSH traffic from the suspect host over a 32-minute window and sums the total bytes sent to each destination. An alarm is triggered when a single external destination receives 60% or more of the host's total SSH traffic by volume. Either the source or destination IP address can be excluded from triggering this alarm.
Tunneling through internal DNS	Command and Con-	Detects when a host is funneling a disproportionate

Plixer Machine Learning + FlowPro Defender

Policy	Category	Description
Encrypted traffic alert	Command and Control > Encrypted Channel	**Plixer Machine Learning + FlowPro Defender** - Detects anomalous encrypted network traffic by combining TLS/JA3 fingerprint analysis from FlowPro Defender with ML Engine behavioral correlation. The algorithm assigns a weighted severity score based on multiple factors: rare client TLS signatures (JA3) and rare server TLS signatures (JA3S) each contribute to the score, with additional weight given when the server is an external IP first seen within the last 5 minutes. The score is further increased if the ML Engine has detected unsupervised HTTPS alerts, other unsupervised anomalies, or classification alerts for the same source IP within the preceding 10 minutes. An alarm is triggered when the combined weight exceeds a threshold of 300 points. The alert includes the JA3 and JA3S hashes along with the specific reasons that contributed to the severity score. Either the source or destination IP address can be excluded from triggering this alarm.
SMB Brute-force Attempt	Credential Access > Brute Force > Password Guessing	**Plixer Machine Learning + FlowPro Defender** - Detects a client attempting to brute-force login credentials on an SMB file server. FlowPro Defender captures SMB authentication metadata, and the ML Engine processes these events to identify repeated failed login attempts. The alarm details include the targeted usernames and the number of failed attempts per username, sorted by frequency. The maximum number of usernames displayed in the alarm is configurable (default 10). Either the source or destination IP address can be excluded from triggering this alarm.
Ransomware Behavior	Impact > Data Encrypted for Impact	**Plixer Machine Learning + FlowPro Defender** - Detects a client accessing an SMB file share and exhibiting file encryption patterns consistent with ransomware activity. FlowPro Defender captures SMB file operation metadata, and the ML Engine processes these events to identify bulk file modification behavior. The alarm details include the names of the targeted files and a count of total files affected. The maximum number of filenames displayed in the alarm is configurable (default 10). Either the source or destination IP address can be excluded from triggering this alarm.

Event details

The table below lists the details reported and *default timeout settings* for alarm policy violations/events in Scrutinizer.

View table

Table 3: Alarm Policies Details

Name	Criteria	Alarm Keys	T/O	Message
Access and Audit Events	violators, message	violators, message	5m	
Access to a potentially vulnerable web application	violators	violators, targets, devices, msg	900	
A client was using an unusual port	violators	violators, targets, devices, msg	900	
An attempted login using a suspicious username was detected	violators	violators, targets, devices, msg	900	
A Network Trojan was detected	violators	violators, targets, devices, msg	900	
A suspicious filename was detected	violators	violators, targets, devices, msg	900	
A system call was detected	violators	violators, targets, devices, msg	900	
Attempted Denial of Service	violators	violators, targets, devices, msg	900	
Attempted Information Leak	violators	violators, targets, devices, msg	900	
Attempted User Privilege Gain	violators	violators, targets, devices, msg	900	
Attempt to login by a default username and password	violators	violators, targets, devices, msg	900	
Auto Investigate	first_violator	violators, targets, host_count, policy_count, chain_count, event_count, start_epoch, end_epoch	86400	The host %{FIRST_VIOLATOR} was seen in %{CHAIN_COUNT} event chains involving %{POLICY_COUNT} policies, %{HOST_COUNT} directly involved hosts, and %{EVENT_COUNT} events.
AutoReplicate Error	failure_type	seed_profile, message	5m	AutoReplicate on %{VIOLATORS} encountered %{FAILURE_TYPE} with %{SEED_PROFILE}. %{MESSAGE}

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
AutoReplicate Exporter Added	exporter	exporter, port, profile_name	5m	AutoReplicate on %{VIOLATORS} added %{EXPORTER} on %{PORT} to the %{PROFILE_NAME} profile.
AutoReplicate Exporter Removed	exporter	exporter, port, profile_name	5m	AutoReplicate on %{VIOLATORS} removed %{EXPORTER} on %{PORT} from the %{PROFILE_NAME} profile.
AutoReplicate Ran	seed_profile, type	minutes, summary	300	AutoReplicate ran on: %{VIOLATORS} with a statistics lookback window %{MINUTES} minutes. %{SUMMARY}
Azure user logged on from many hosts	user_id	user_id, total_hosts	300	In the last 30 minutes, %{USER_ID} has attempted to authenticate from %{TOTAL_HOSTS} hosts, which is more hosts than normal. Hosts performing authentication(s) are %{VIOLATORS}
Azure user logged on from many locations	user_id	user_id, total_locations	300	In the last 30 minutes, %{USER_ID} has attempted to authenticate from %{TOTAL_LOCATIONS} different locations, which is more than normal. Locations performing authentication(s) are %{VIOLATORS}
Azure user logged on many times	user_id	user_id, total_auths	300	In the last 30 minutes, %{USER_ID} has attempted %{TOTAL_AUTHS} authentications, which is more authentications than normal. Hosts performing authentication(s) are %{VIOLATORS}
Bad Exporter Flow	violators, reason_text	reason_text, reason_num, repetition, sequence, set_id, source_id, violators, devices	3600	Exporter %{VIOLATORS} sent a bad flow (source %{SOURCE_ID}, sequence %{SEQUENCE}, set %{SET_ID}): %{REASON_TEXT}
Bad Exporter Packet	violators, reason_text	reason_text, reason_num, repetition, violators, devices	3600	Exporter %{VIOLATORS} sent a bad packet: %{REASON_TEXT}

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Bad Exporter Template	violators, reason_text	reason_text, reason_num, repetition, sequence, source_id, template_id, violators, devices	3600	Exporter %{VIOLATORS} sent a bad template #%{TEMPLATE_ID} (source %{SOURCE_ID}, sequence %{SEQUENCE}): %{REASON_TEXT}
Blocked Malicious Domains	violators	violators, targets, domain	300	
Bogon Attempt	violators	violators, targets, devices	3600	Connections to a bogon network, %{TARGETS}, were seen on %{DEVICES} by %{VIOLATORS}
Bogon Connection	violators	violators, targets, devices	3600	Inbound traffic from a bogon network was seen going to %{TARGETS} on %{DEVICES} by %{VIOLATORS}
BotNet Detection	violators	violators, targets, devices, nxcount	3600	Internal IP %{VIOLATORS} performed %{NXCOUNT} unique DNS lookups using DNS server(s) %{TARGETS} that returned a No Existing Domain (NXDOMAIN) message as seen on %{DEVICES} exporter(s). This may indicate the presence of malware on %{VIOLATORS} that uses a domain generation algorithm (DGA) to communicate with malware C&C servers.
Breach Attempt Detection	violators, breachtype	devices, violators, breachtype, targets	900	Detected %{BREACHTYPE} breach by: %{VIOLATORS} with targets: %{TARGETS}
Brute-force RDP (Client-side)	violators	violators, targets	300	
Brute-force RDP (Server-side TCP)	targets	violators, targets	300	
Brute-force RDP (Server-side UDP)	targets	violators, targets	300	
Brute-force SSH (Client-side)	violators	violators, targets	300	
Brute-force SSH (Server-side)	targets	violators, targets	300	
Collector Alert	error	process, process_id, devices, violators, error	300	

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Collector Message	event_type, priority	process, process_id, message, event_type, violators	300	
Configuration Alert	event_type, priority	process, process_id, message, event_type, violators	300	
Crypto Currency Mining Activity Detected	violators	violators, targets, devices, msg	900	
Cstore Strays	devices	count	86400	Found and removed: %{COUNT} stray cstore files on: %{DEVICES}
Data Accumulation	violators	violators, targets, total_data	300	In the last 30 minutes, %{VIOLATORS} accumulated %{TOTAL_DATA} bytes from %{TARGETS}
Data Exfiltration	violators	violators, targets, total_data	300	In the last 30 minutes, %{VIOLATORS} exfiltrated %{TOTAL_DATA} bytes to %{TARGETS}
DDoS	targets	attacker_count, bytes_std_dev, duration, flow_count, packets_std_dev	300	Possible Inbound DDoS Attack: Within %{DURATION} seconds %{ATTACKER_COUNT} external hosts generated a combined total of %{FLOW_COUNT} flows having bytes within %{BYTES_STD_DEV} standard deviations and packets within %{PACKETS_STD_DEV} standard deviations.
Decode of an RPC Query	violators	violators, targets, devices, msg	900	
Denial of Service	violators	violators, targets, devices, msg	900	
Denied Flows Firewall	violators	devices, violators, target_count, flowcount	900	IP %{VIOLATORS} had %{FLOWCOUNT} connection attempts to %{TARGET_COUNT} external IP addresses denied by the firewall as seen on %{DEVICES} exporter(s)

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Detection of a Denial of Service Attack	violators	violators, targets, devices, msg	900	
Detection of a Network Scan	violators	violators, targets, devices, msg	900	
Detection of a non-standard protocol or event	violators	violators, targets, devices, msg	900	
Device Retrieving External IP Address Detected	violators	violators, targets, devices, msg	900	
Diskspace Alert	disk_error, disk_partition, violators	process, process_id, disk_error, disk_partition, message	300	
DNS Command and Control Detection	violators	violators, targets, devices	900	Possible Command and Control (C&C) Activity. DNS TXT messages are being exchanged between asset %{VIOLATORS} and %{TARGETS} as seen on the %{DEVICES} exporter(s)
DNS Data Leak Detection	violators	violators, totaltextlength, dnsname	900	DNS lookups initiated from asset: %{VIOLATORS} using complex domain name: %{DNSNAME} containing a high number of domain levels and a total of: %{TOTALTEXTLENGTH} characters.
DNS Hits	violators	violators, flowcount, threshold	900	Internal IP %{VIOLATORS} performed %{FLOWCOUNT} DNS lookups in the last 5 minutes exceeding the treshold of %{THRESHOLD}
DNS Server Detection	violators	violators, client_count, flowcount, devices	900	
Domain Observed Used for C2 Detected	violators	violators, targets, devices, msg	900	
Domain Reputation	violators, dnsname	violators, dnsname, category	900	IP %{VIOLATORS} performed a DNS lookup on a black-listed domain: %{DNSNAME} in the %{CATEGORY} category

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
DRDoS	targets, port_name	devices, attacker_count, duration, packet_in_count, packet_io_ratio, packet_out_count port, port_name	900	Possible Inbound DRDoS Attack from common port %{PORT} (%{PORT_NAME}): Within %{DURATION} seconds %{ATTACKER_COUNT} violators generated a combined total of %{PACKET_IN_COUNT} inbound packets in response to %{PACKET_OUT_COUNT} outbound request packets, for a ratio of %{PACKET_IO_RATIO} inbound packets per outbound packet.
Encrypted traffic alert	violators	violators, ja3, ja3s, reason, severity	300	ML generated an encrypted traffic alert for %{VIOLATORS}: %{REASON}
Endpoint Analytics Info	violators	violators, macaddress, risk_score, location	300	Host %{VIOLATORS} has MAC address %{MACADDRESS}, has a risk score of %{RISK_SCORE}, and has location %{LOCATION}.
Event Queue Alert	violators, type	threshold, value	300	Event queue on host: %{VIOLATORS} has breached %{TYPE} threshold: %{THRESHOLD} with value: %{VALUE}
Executable code was detected	violators	violators, targets, devices, msg	900	
Exploit Kit Activity Detected	violators	violators, targets, devices, msg	900	
Exporter Ignored	devices, violators, reason_num	reason_text, repetition, violators	3600	Discarding flows from exporter %{VIOLATORS}: %{REASON_TEXT}
Exporter Paused	violators, exporter_id		1s	Exporter %{EXPORTER_ID} paused by reporter %{VIOLATORS} due to insufficient resources. See the feature sizing interface for more details.
Exporter Resumed	violators, exporter_id		1s	Exporter %{EXPORTER_ID} resumed by reporter %{VIOLATORS} due to additional available resources. See the feature sizing interface for more details.

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Feature Set Paused	violators, feature_set		1s	Feature set %{FEATURE_SET} paused by reporter %{VIOLATORS} due to insufficient resources. See the feature sizing interface for more details.
Feature Set Resumed	violators, feature_set		1s	Feature set %{FEATURE_SET} resumed by reporter %{VIOLATORS} due to additional available resources. See the feature sizing interface for more details.
FIN Scan (External)	violators	devices, violators	900	A FIN Scan was seen on %{DEVICES} by %{VIOLATORS}
FIN Scan (Internal)	violators	devices, violators	900	A FIN Scan was seen on %{DEVICES} by %{VIOLATORS}
Flow Collection Paused	violators		60s	Flow collection paused on collector %{VIOLATORS} due to hardware and/or configuration change. See the feature sizing interface for more details.
Flow Collection Resumed	violators	new_flow_rate	60s	Flow collection resumed at %{NEW_FLOW_RATE} flows/sec on collector %{VIOLATORS}.
Flow Inactivity	violators, collector	last_flow	1200	Exporter %{VIOLATORS} stopped sending flows to the %{COLLECTOR} collector. The last flow was received %{LAST_FLOW}. If this is expected, set the exporter to disabled or delete it in manage exporters to stop these alarms.
FlowPro Event Capture	devices, capture_name	violators, targets, devices, capture_name, lookup	900	Traffic captured for %{CAPTURE_NAME} from %{VIOLATORS} to %{TARGETS} seen on %{DEVICES}
FlowPro Event Capture	violators	violators, targets, devices, lookup	900	Traffic captured from %{VIOLATORS} to %{TARGETS} by %{DEVICES}, access via %{LOOKUP}
Flow Rate Limit Changed	violators	new_flow_rate	60s	Flow collection rate limit changed to %{NEW_FLOW_RATE} flows/sec on collector %{VIOLATORS} due to hardware and/or configuration change. See the feature sizing interface for more details.

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Flows Limited - Licensing	devices, violators, reason_num	reason_text	60s	Collector %{VIOLATORS} license exceeded: %{REASON_TEXT}
Forecast Anomaly	devices, interfaces, applications, type, ts	forecast_id, devices, interfaces, target_quantity, observed_value, mean, forecast_start_time, forecast_end_time	300	Forecast: %{FORECAST_ID} found %{INTERFACES} on %{DEVICES} observed value: %{OBSERVED_VALUE} %{TARGET_QUANTITY} is outside forecast for interval %{FORECAST_START_TIME}-%{FORECAST_END_TIME}, Expected Value: %{LOWER_CONF} <= %{MEAN} <= %{UPPER_CONF}
Forecast Task Complete	devices, interfaces, applications, type	forecast_id	60s	Forecast: %{FORECAST_ID} complete, results available
Forecast Task Error	devices, interfaces, applications, type	forecast_id, error_stage, error	60s	Forecast: %{FORECAST_ID} resulted in an error during %{ERROR_STAGE}. Message: %{ERROR}
Forecast Task Starting	devices, interfaces, applications, type	forecast_id	60s	Forecast: %{FORECAST_ID} received by forecasting module
Generic Protocol Command Decode	violators	violators, targets, devices, msg	900	
HA Exporter switchover event	profile_name, reason	profile_name, reason, source_ip	30s	Replicator Profile (%{PROFILE_NAME}) has changed active exporter to %{SOURCE_IP} as %{REASON}
Hardware Resources Exceeded	violators	drop_rate, flow_limit_period	60s	Collector %{VIOLATORS} incoming flow rate exceeds hardware recommendations. %{DROP_RATE} flows per second dropped over the last %{FLOW_LIMIT_PERIOD} seconds. See the feature sizing interface for more details.
Heartbeat Alert	heartbeat_type, violators	process, process_id, heartbeat_type, devices, violators	300	

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Host Index Disk Availability Error	violators	threshold, current	300	Host Indexing service has reached disk storage volume limit of %{THRESHOLD} percent in use, Currently %{CURRENT} percent in use. Stopping processing and starting garbage collection until under threshold.
Host Index Disk Space Error	violators	threshold, current	300	Host Indexing service has reached disk space usage: %{CURRENT}MB, threshold: %{THRESHOLD}MB. Stopping processing and starting garbage collection until under threshold.
Host Index Disk Space Warning	violators	threshold, current	300	Host Indexing service has reached disk space usage: %{CURRENT}MB, over 75% of threshold: %{THRESHOLD}MB
Host Reputation	violators, targets	violators, targets, devices, category_note	3600	IP %{VIOLATORS} sent traffic to a suspect %{CATEGORY_NOTE} at IP address %{TARGETS} as seen on the %{DEVICES} exporter(s)
Host Watchlist	violators	devices, violators, port, protocol	900	Host Watchlist - %{DEVICES} saw watchlisted host %{VIOLATORS} communicating from %{PROTOCOL} %{PORT}
ICMP Destination Unreachable (External)	violators	flowcount, violators	900	External IP %{VIOLATORS} triggered %{FLOWCOUNT} ICMP Destination Unreachable flows within 5 minutes
ICMP Destination Unreachable (Internal)	violators	flowcount, violators	900	Internal IP %{VIOLATORS} triggered %{FLOWCOUNT} ICMP Destination Unreachable flows within 5 minutes
ICMP Port Unreachable (External)	violators	flowcount, violators	900	External IP %{VIOLATORS} triggered %{FLOWCOUNT} ICMP Protocol Unreachable flows within 5 minutes
ICMP Port Unreachable (Internal)	violators	flowcount, violators	900	Internal IP %{VIOLATORS} triggered %{FLOWCOUNT} ICMP Protocol Unreachable flows within 5 minutes
Information Leak	violators	violators, targets, devices, msg	900	

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Interface Threshold Violation	violators, interface_name, instance	exporter, interface_name, instance, threshold, violation, graphStart, graphEnd	900	Interface %{EXPORTER}: %{INTERFACE_NAME} exceeded the threshold of %{THRESHOLD} %{VIOLATION}
IP Address Violations	violators	devices, violators, targets	900	Traffic on %{DEVICES} between %{VIOLATORS} and %{TARGETS} is outside of allowed subnets
Kafka Lag	topic_lagged	topic_lagged, messages_behind	660	ML Kafka topic %{TOPIC_LAGGED} is lagging %{MESSAGES_BEHIND} messages behind
Large Ping	violators	violators, targets, devices, threshold, avg_ping_size	900	Unexpected ICMP Echo traffic seen from violator %{VIOLATORS} to target %{TARGETS} on exporter %{DEVICES} with an average packet size of %{AVG_PING_SIZE} Bytes which violates the threshold of %{THRESHOLD} Bytes
Large Scale Information Leak	violators	violators, targets, devices, msg	900	
Lateral Movement	violators, targets, worm_type	devices, targets, violators	1200	
Lateral Movement Attempt	violators, worm_type	devices, violators, targets, worm_type, dst_port	1200	
Lateral Movement Behavior	violators	violators	300	
Malware Command and Control Activity Detected	violators	violators, targets, devices, msg	900	
Medianet Jitter Violations	violators	targets, violators, jitter	420	Jitter values of %{JITTER}ms between %{VIOLATORS} and %{TARGETS} exceeds threshold
ML Engine alert	violators, source	source, threshold	300	ML service %{SOURCE} has reached threshold %{THRESHOLD}, throttling until next run
ML Engine coin miner alert	violators	violators, family, probability, threshold	300	ML detected %{VIOLATORS} generating malicious traffic related to %{FAMILY} malware family (%{PROBABILITY}% match, threshold set to %{THRESHOLD}%)

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
ML Engine command and control alert	violators	violators, family, probability, threshold	300	ML detected %{VIOLATORS} generating malicious traffic related to %{FAMILY} malware family (%{PROBABILITY}% match, threshold set to %{THRESHOLD}%)
ML Engine Down	host	host, violators	300	ML Engine %{HOST} is not responding to pings
ML Engine exploit kit alert	violators	violators, family, probability, threshold	300	ML detected %{VIOLATORS} generating malicious traffic related to %{FAMILY} malware family (%{PROBABILITY}% match, threshold set to %{THRESHOLD}%)
ML Engine malware alert	violators	violators, family, probability, threshold	300	ML detected %{VIOLATORS} generating malicious traffic related to %{FAMILY} malware family (%{PROBABILITY}% match, threshold set to %{THRESHOLD}%)
ML Engine remote access trojan alert	violators	violators, family, probability, threshold	300	ML detected %{VIOLATORS} generating malicious traffic related to %{FAMILY} malware family (%{PROBABILITY}% match, threshold set to %{THRESHOLD}%)
ML models still building	violators	violators, schedule	300	ML is still building models for schedule %{SCHEDULE}, but the next schedule is currently expected to start. Increase replica count values in the config.
ML Service Alert	service_name	service_name, unavailable, expected	300	ML service %{SERVICE_NAME} has %{UNAVAILABLE}/%{EXPECTED} instances unavailable
NetFlow Domain Reputation	violators, domain	violators, domain, category	900	Internal IP %{VIOLATORS} performed a lookup of %{DOMAIN}, categorized as %{CATEGORY}
Network Anomaly	violators, interface_id, anomaly_type	violators, interface_id, anomaly_type	300	Exporter %{VIOLATORS} is generating anomalous %{ANOMALY_TYPE} traffic on interface %{INTERFACE_ID}
New user using elevated logon	user_id	user_id	300	A new user, %{USER_ID}, is logging in with elevated privileges. Hosts performing login(s) are %{VIOLATORS}

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
NULL Scan (External)	violators	devices, violators, flowcount, threshold	900	A NULL scan was seen on %{DEVICES} by %{VIOLATORS} in %{FLOWCOUNT} flows violating the threshold of %{THRESHOLD}
NULL Scan (Internal)	violators	devices, violators, flowcount, threshold	900	A NULL scan was seen on %{DEVICES} by %{VIOLATORS} in %{FLOWCOUNT} flows violating the threshold of %{THRESHOLD}
Odd TCP Flags (External)	violators	devices, violators, flags, flowcount	900	Odd TCP flags (%{FLAGS}) were seen in %{FLOWCOUNT} flows on %{DEVICES} by %{VIOLATORS}
Odd TCP Flags (Internal)	violators	devices, violators, flags, flowcount	900	Odd TCP flags (%{FLAGS}) were seen in %{FLOWCOUNT} flows on %{DEVICES} by %{VIOLATORS}
Office 365 user logged in many times	user_id	user_id, total_auths	300	In the last 30 minutes, %{USER_ID} has attempted %{TOTAL_AUTHS} authentications, which is more authentications than normal. Hosts performing authentication(s) are %{VIOLATORS}
Office 365 user logged on from many hosts	user_id	user_id, total_hosts	300	In the last 30 minutes, %{USER_ID} has attempted to authenticate from %{TOTAL_HOSTS} hosts, which is more hosts than normal. Hosts performing authentication(s) are %{VIOLATORS}
Office 365 users logged on from many locations	user_id	user_id, total_locations	300	In the last 30 minutes, %{USER_ID} has attempted to authenticate from %{TOTAL_LOCATIONS} different locations, which is more than normal. Locations performing authentication(s) are %{VIOLATORS}
P2P Detection	violators	devices, violators, dst_host_count, dst_port_count	900	P2P traffic to %{DST_HOST_COUNT} destinations using %{DST_PORT_COUNT} distinct port(s) was seen on %{DEVICES} from %{VIOLATORS}

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Packet Flood	violators	devices, violators, targets, count	3600	Packet flood seen from %{VIOLATORS} to %{TARGETS} comprising of %{COUNT} small packets in a minute by devices: %{DEVICES}
Ping Flood	violators	devices, violators, targets, count	3600	Ping flood seen from %{VIOLATORS} to %{TARGETS} comprising of %{COUNT} pings in a minute by devices: %{DEVICES}
Ping Scan (External)	violators	devices, violators, count	3600	Ping scan seen from %{VIOLATORS} to %{COUNT} hosts by devices: %{DEVICES}
Ping Scan (Internal)	violators	devices, violators, count	3600	Ping scan seen from %{VIOLATORS} to %{COUNT} hosts by devices: %{DEVICES}
Possible Social Engineering Attempted	violators	violators, targets, devices, msg	900	
Possibly Unwanted Program Detected	violators	violators, targets, devices, msg	900	
Privileged user logged on from many hosts	user_id	user_id, total_hosts	300	In the last 30 minutes, %{USER_ID} has attempted to authenticate from %{TOTAL_HOSTS} hosts, which is more hosts than normal. Hosts performing authentication(s) are %{VIOLATORS}
Privileged user logged on many times	user_id	user_id, total_auths	300	In the last 30 minutes, %{USER_ID} has attempted %{TOTAL_AUTHS} authentications, which is more authentications than normal. Hosts performing authentication(s) are %{VIOLATORS}
Protocol Misdirection	violators	violators, traffic_type, port, targets	3600	Mismatched traffic type of %{TRAFFIC_TYPE} to port %{PORT} from %{VIOLATORS} to %{TARGETS}
Ransomware Behavior	violators	violators, targets, file_count, files	900	Observed a possible ransomware encryption attack from %{VIOLATORS} targeting SMB share %{TARGETS}. %{FILE_COUNT} files were both read and written to, including files: %{FILES}

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Replicator Exporter State Change	replicator, exporter_ip, exporter_port, state	replicator, exporter_ip, exporter_port, state	30s	Replicator(%{REPLICATOR}) detected a state change for exporter %{EXPORTER_IP}:%{EXPORTER_PORT} state: %{STATE}
Replicator Exporter State Change	replicator, collector_ip, collector_port, state	replicator, collector_ip, collector_port, state	30s	Replicator(%{REPLICATOR}) detected a state change for collector %{COLLECTOR_IP}:%{COLLECTOR_PORT} state: %{STATE}
Replicator Has Encountered An Error	replicator	replicator, errmsg	300	Replicator (%{REPLICATOR}) has encountered an error: %{ERRMSG}
Replicator High Availability State Changed	replicator	replicator, message, state	30s	Replicator (%{REPLICATOR}) has changed state to %{STATE}: %{MESSAGE}
Report Threshold Violation	saved_report, row_identifier	saved_report, row_identifier, violation, graphStart, graphEnd, src_port, dst_port, violator, violator_username, target, target_username, protocol, app_proto, url	420	The report %{SAVED_REPORT} %{ROW_IDENTIFIER} has exceeded its threshold %{VIOLATION}
Reverse SSH Shell	violators	origin_bytes, bytes_per_packet	3600	Possible reverse SSH tunnel from %{VIOLATORS} to %{TARGETS} seen by devices: %{DEVICES} based on %{ORIGIN_BYTES} origin bytes and %{BYTES_PER_PACKET} average origin bytes per packet
Rogue DHCP Service	violators	violators, targets	300	
Rogue DNS Service	violators	violators, targets	300	
Rogue LDAP Service	violators	violators, targets	300	
RST/ACK Detection (External)	violators	violators, flowcount, targets	900	Anomalous Behavior - Possible - RST/ACK Replies Observed Host %{TARGETS} received %{FLOWCOUNT} packets from %{VIOLATORS} without observing any other flags

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
RST/ACK Detection (Internal)	violators	violators, flowcount, targets	900	Anomalous Behavior - Possible - RST/ACK Replies Observed Host %{TARGETS} received %{FLOWCOUNT} packets from %{VIOLATORS} without observing any other flags
Runtime Overrun	process	process, process_id, threshold, duration, action	300	
Scheduled Task Error	violators, task_name	task_id, command, error_code, start_time, run_time	300	A scheduled task on collector %{VIOLATORS}, %{TASK_NAME} (ID %{TASK_ID}) returned error code: %{ERROR_CODE} running: “%{COMMAND}”. It started at %{START_TIME} AND ran for %{RUN_TIME} seconds. View the collector log and/or run the task manually for more details.
Security Anomaly	violators, anomaly_type	violators, anomaly_type	300	
Setup Problem	issue	message	900	
SIGRed Exploit Attempt	violators	violators, targets	300	
SMB Brute-force Attempt	violators	violators, targets, failed_logins, usernames	900	Observed a possible SMB brute force attack from %{VIOLATORS} targeting SMB share %{TARGETS}. %{FAILED_LOGINS} failed logins observed including usernames: %{USERNAMES}
Source Equals Destination	violators	devices, violators	900	Traffic with source and destination of %{VIOLATORS} was seen on %{DEVICES}
Stream Deactivated	stream	size, threshold	900	The stream: %{STREAM} has breached its configured threshold: %{THRESHOLD} with total size: %{SIZE} and has been deactivated.
Stream Reactivated	stream	minutes, size, threshold	900	The stream: %{STREAM} with total size: %{SIZE} below its configured threshold: %{THRESHOLD} has been reactivated after having been deactivated for: %{MINUTES} minutes.

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Successful Administrator Privilege Gain	violators	violators, targets, devices, msg	900	
Successful Credential Theft Detected	violators	violators, targets, devices, msg	900	
Successful User Privilege Gain	violators	violators, targets, devices, msg	900	
Suspicious Host Communication	violators	violators, targets, protocol_name	300	Based on how these hosts and those around them normally communicate, the communication between %{VIOLATORS} and the host(s) %{TARGETS} on protocol %{PROTOCOL_NAME} is unexpected. Use the explore event traffic link to view these communications in detail.
Suspicious Host Communication	violators	violators, targets, protocol	300	Based on how these hosts and those around them normally communicate, the communication between %{VIOLATORS} and the host(s) %{TARGETS} on protocol %{PROTOCOL} is unexpected. Use the explore event traffic link to view these communications in detail.
SYN IP Scan (External)	violators	devices, violators, targets, scanned_host_cou scanned_port_cou host_thresh, port_thresh	900	A SYN IP Scan by %{VIOLATORS} seen scanning %{SCANNED_HOST_COUNT} hosts which exceeds the threshold of %{HOST_THRESH} and %{SCANNED_PORT_COUNT} ports per host exceeding the threshod of %{PORT_THRESH}
SYN IP Scan (Internal)	violators	devices, violators, targets, scanned_host_cou scanned_port_cou host_thresh, port_thresh	900	A SYN IP Scan by %{VIOLATORS} seen scanning %{SCANNED_HOST_COUNT} hosts which exceeds the threshold of %{HOST_THRESH} and %{SCANNED_PORT_COUNT} ports per host exceeding the threshod of %{PORT_THRESH}

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
SYN Port Scan (External)	violators	devices, violators, targets, scanned_host_cou scanned_port_cou host_thresh, port_thresh	900	A SYN Port Scan by %{VIOLATORS} seen scanning %{SCANNED_HOST_COUNT} hosts which exceeds the threshold of %{HOST_THRESH} and %{SCANNED_PORT_COUNT} ports per host exceeding the threshod of %{PORT_THRESH}
SYN Port Scan (Internal)	violators	devices, violators, targets, scanned_host_cou scanned_port_cou host_thresh, port_thresh	900	A SYN Port Scan by %{VIOLATORS} seen scanning %{SCANNED_HOST_COUNT} hosts which exceeds the threshold of %{HOST_THRESH} and %{SCANNED_PORT_COUNT} ports per host exceeding the threshod of %{PORT_THRESH}
System Capacity	vital_type	vital_type, value	300	ML is using %{VALUE} percent of its %{VITAL_TYPE} capacity
Targeted Malicious Activity was Detected	violators	violators, targets, devices, msg	900	
TCP Half-Open (External)	violators	devices, violators, targets, packets_per_port, scanned_port_cou pkt_thresh, port_thresh	900	A possible SYN Half Open Attack by %{VIOLATORS} seen targeting %{TARGETS}. Port count of %{SCANNED_PORT_COUNT} exceeded the threshold of %{PORT_THRESH} and flows per port of %{PACKETS_PER_PORT} exceed the threshold of %{PKT_THRESH}.
TCP Half-Open (Internal)	violators	devices, violators, targets, packets_per_port, scanned_port_cou pkt_thresh, port_thresh	900	A possible SYN Half Open Attack by %{VIOLATORS} seen targeting %{TARGETS}. Port count of %{SCANNED_PORT_COUNT} exceeded the threshold of %{PORT_THRESH} and flows per port of %{PACKETS_PER_PORT} exceed the threshold of %{PKT_THRESH}.

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
TCP Scan (External)	violators	devices, violators, port_count, dst_count	900	A TCP Scan was seen on %{DEVICES} by %{VIOLATORS} scanning %{DST_COUNT} IPs and %{PORT_COUNT} ports
TCP Scan (Internal)	violators	devices, violators, port_count, dst_count	900	A TCP Scan was seen on %{DEVICES} by %{VIOLATORS} scanning %{DST_COUNT} IPs and %{PORT_COUNT} ports
TLS Certificate Expiry	violators	days	86400	TLS certificates on nodes: %{VIOLATORS} will expire in %{DAYS} days. Contact Plixer Support or see `scrut_util --help certs`.
Token Expiration	username, expires_on	username, expires_on, status	86400	An authentication token for %{USERNAME} %{STATUS} on %{EXPIRES_ON}
Tunneling through external DNS host	violators	violators, targets, tunnel_type	300	
Tunneling through external ICMP host	violators	violators, targets, tunnel_type	300	
Tunneling through external SSH host	violators	violators, targets, tunnel_type	300	
Tunneling through internal DNS host	violators	violators, targets, tunnel_type	300	
Tunneling through internal ICMP host	violators	violators, targets, tunnel_type	300	
Tunneling through internal SSH host	violators	violators, targets, tunnel_type	300	
UDP Scan (External)	violators	devices, violators, dst_count, port_count	900	A UDP Scan was seen on %{DEVICES} by %{VIOLATORS} scanning %{DST_COUNT} IPs and %{PORT_COUNT} ports
UDP Scan (Internal)	violators	devices, violators, dst_count, port_count	900	A UDP Scan was seen on %{DEVICES} by %{VIOLATORS} scanning %{DST_COUNT} IPs and %{PORT_COUNT} ports
Unapproved Protocol	protocol	protocol_name, devices	900	Unapproved network transport: %{PROTOCOL_NAME} was seen on: %{DEVICES}

continues on next page

Table 3 – continued from previous page

Name	Criteria	Alarm Keys	T/O	Message
Unsuccessful User Privilege Gain	violators	violators, targets, devices, msg	900	
Web Application Attack	violators	violators, targets, devices, msg	900	
Worm Activity	violators	violators	300	
Xmas Scan (External)	violators	devices, violators	900	An Xmas Scan was seen on %{DEVICES} by %{VIOLATORS}
Xmas Scan (Internal)	violators	devices, violators	900	An Xmas Scan was seen on %{DEVICES} by %{VIOLATORS}
Zerologon	violators	violators, targets	300	

FA algorithm list

The table below contains general information and recommended applications for all flow analytics algorithms available in Scrutinizer.

View table

Algorithm	Function	Recommended Flow Sources
Bogon Traffic	Alerts if traffic to or from an unallocated public IP space is detected	Edge routers and public IP addresses defined in IP groups
BotNet Detection	Alerts when a large number of unique DNS name lookups have failed (Requires **FlowPro**)	FlowPro
Breach Attempt Detection	Alerts when flow behaviors that may indicate a brute force password attack on an internal IP address are observed	Internal/core routers, edge routers, and public IP addresses defined in IP groups
DDoS Detection	Alerts when a Distributed Denial of Service (DDoS) attack targeting the protected network space is identified	Edge routers and public IP addresses defined in IP groups
Denied Flows Firewall	Alerts when the number of denied flows from an internal to an external IP address exceeds the configured threshold	Internal/core routers
DNS Command and Control Detection	Alerts when the volume or size of DNS TXT messages at the network perimeter exceeds the configured threshold (Requires **FlowPro**)	FlowPro
DNS Data Leak Detection	Alerts when the volume or size of messages with suspicious DNS names exceeds the configured threshold (Requires **FlowPro**)	FlowPro
DNS Hits	Alerts when a host initiates an excessive number of DNS queries	Internal/core routers

continues on next page

Table 4 – continued from previous page

Algorithm	Function	Recommended Flow Sources
DNS Server Detection	Alerts when a new DNS is detected based on packet exchanges between clients and servers (Requires **FlowPro**)	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Domain Reputation	Alerts when traffic associated with a suspicious domain (based on a list maintained by Plixer) is detected (Requires **FlowPro**)	FlowPro
DRDoS Detection	Alerts when a Distributed Reflection Denial of Service attack targeting the protected network space is identified	Edge routers and public IP addresses defined in IP groups
FIN Scan	Alerts when a FIN scan is detected	Internal/core routers and edge routers
Flow Reports Thresholds	Alerts when a custom threshold configured for a saved report is exceeded	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Host Indexing	Monitors traffic to maintain an index of hosts seen on the network that includes additional details, such as conversation direction, throughput, and source (Exporter)	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Host Reputation	Monitors traffic to maintain a list of active, non-whitelisted Tor nodes	Edge routers and public IP addresses defined in IP groups
Host Watchlist	Alerts when a host violating a user-defined IP address blacklist is detected	Edge routers and public IP addresses defined in IP groups
ICMP Destination Unreachable	Alerts when a large number of *ICMP Destination Unreachable* messages are sent to a suspicious IP address	Internal/core routers
ICMP Port Unreachable	Alerts when a large number of *ICMP Port Unreachable* messages are sent to a suspect IP address	Internal/core routers
Incident Correlation	Alerts when multiple Indicator of Compromise (IOC) events for a single host are detected	Internal/core routers, edge routers, and public IP addresses defined in IP groups
IP Address Violations	Alerts when a flow containing a non-authorized IP address as the source or destination is received (Requires authorized subnets to be defined)	Internal/core routers, edge routers, and public IP addresses defined in IP groups
JA3 Fingerprinting	Alerts when software sending suspicious encrypted traffic based on TLS handshake data and known signatures is identified (Requires **FlowPro**)	FlowPro
Large Ping	Alerts when unusually large ICMP Echo Request (ping) packets are observed, which may indicate malicious activity, including possible Denial of Service (DoS) attempts.	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Lateral Movement	Alerts when successful lateral movement is observed	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Lateral Movement Attempt	Alerts when behavior that may indicate attempted lateral movement is observed	Internal/core routers, edge routers, and public IP addresses defined in IP groups

continues on next page

Table 4 – continued from previous page

Algorithm	Function	Recommended Flow Sources
Medianet Jitter Violations	Alerts when jitter values reported by a Medianet flow exceed the configured threshold	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Multicast Violations	Alerts when multicast traffic volume exceeds the configured threshold	Internal/core routers, edge routers, and public IP addresses defined in IP groups
NetFlow Domain Reputation	Alerts when a DNS lookup from a blacklisted IP is reported via NetFlow (Blacklist is maintained on nba.plixer.com but cached locally)	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Network Transports	Alerts when traffic over unapproved transport protocols is observed	Internal/core routers, edge routers, and public IP addresses defined in IP groups
NULL Scan	Alerts when a NULL scan is detected	Internal/core routers and edge routers
Odd TCP Flags Scan	Alerts when a scan using unusual TCP flag combinations is detected	Internal/core routers and edge routers
P2P Detection	Alerts when a P2P session with a host count exceeding the configured threshold is observed	Internal/core routers and edge routers
Packet Flood	Alerts when a large volume of small-sized packets intended to overwhelm the target's processing capabilities is detected	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Persistent Flow Risk	Alerts when a persistent flow is detected	Internal/core routers and edge routers
Persistent Flow Risk - ASA	Alerts when IP communication matching a 5-tuple is observed for a specified duration to identify VPN or proxy traffic, remote desktop technologies, and other means of covert communication across various applications	Internal/core routers and edge routers
Ping Flood	Alerts when a ping flood is detected	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Ping Scan	Alerts when a host suspected of performing a ping scan is observed	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Protocol Misdirection	Alerts when traffic not matching the port being used is detected	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Reverse SSH Shell	Alerts upon the discovery of a potential reverse SSH tunnel to an external destination, which may allow an external entity to access internal, protected resources via the use of an established outbound SSH connection	Internal/core routers, edge routers, and public IP addresses defined in IP groups
RST/ACK Detection	Alerts when the system observes a large number of TCP flows containing only RST and ACK flags being sent to the same destination	Internal/core routers and edge routers
Source Equals Destination	Alerts when traffic with the same source and destination is observed, which is commonly due to network misconfigurations but may also indicate malicious activity	Internal/core routers, edge routers, and public IP addresses defined in IP groups
SYN Scan	Alerts when a SYN scan is detected	Internal/core routers and edge routers

continues on next page

Table 4 – continued from previous page

Algorithm	Function	Recommended Flow Sources
TCP Scan	Alerts when a potential TCP scan is detected from an Exporter that does not provide TCP flag information	Internal/core routers and edge routers
Top Applications	Monitors application traffic	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Top Autonomous Systems	Monitors traffic to and from autonomous systems	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Top Countries	Monitors traffic by country	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Top Hosts	Monitors traffic by host	Internal/core routers, edge routers, and public IP addresses defined in IP groups
Top IP groups	Monitors traffic by IP group (Requires at least one IP group to be defined)	Internal/core routers, edge routers, and public IP addresses defined in IP groups
UDP Scan	Alerts when a potential UDP scan is detected	Internal/core routers and edge routers
XMAS Scan	Alerts when a XMAS scan is detected	Internal/core routers and edge routers

Algorithm settings

The table below lists the additional settings that can be used to tune behavior for individual FA algorithms.

View table

Algorithm Name	Setting	Description
Auto Investigate	Candidate Limit	The maximum number of Violator->Policy->Target links to review for correlation.
	Chain Max	The maximum number of Violator->Policy->Target chains that will be considered for deduplication.
	Length Limit	The maximum length of any chain of Violator->Policy->Target links.
BotNet Detection	Threshold	Number of unique No Existing Domain (NXDOMAIN) replies within a three-minute period to trigger alarm
DDoS Detection	DDoS Bytes Deviation	Maximum number of bytes allowed in a single standard deviation to trigger (default 10)
	DDoS Packet Deviation	Maximum number of packets allowed in a single standard deviation to trigger (default 10)
	DDoS Packets	Number of packets each source must have sent to be counted
	DDoS Unique hosts	Minimum number of unique hosts participating in a DDoS attack
Denied Flows Firewall	Denied Threshold	The number of denied flows from a single host within a three-minute period to trigger an event
DNS Command and Control Detection	DNS Command and Control attempts	DNS Command and Control attempts within a three-minute period to trigger alarm
	DNS Command and Control bytes	DNS Command and Control bytes within a three-minute period to trigger alarm
DNS Data Leak Detection	DNS Data Leak attempts	DNS Data Leak attempts within a three-minute period to trigger alarm

continues on next page

Table 5 – continued from previous page

Algorithm Name	Setting	Description
	DNS Data Leak bytes	DNS Data Leak bytes within a three-minute period to trigger alarm
DNS Hits	Flow Threshold	The number of DNS requests within a three-minute period to trigger an event
DNS Server Detection	Flow threshold to trigger alarm	Number of properly formatted DNS request packets sent to the specified IP address to trigger alarm
DRDoS Detection	CharGen (UDP 19)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	DNS (UDP 53)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	Flow Imbalance Threshold	How many inbound packets per outbound packet to trigger a DRDoS alarm
	LDAP (UDP 389)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	Memcached (UDP 11211)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	NetBIOS Name Server (UDP 137)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	NTP (UDP 123)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	Quote of the Day (UDP 17)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	RPC Portmap (UDP 111)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	Sentinel (UDP 5093)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	SNMP (UDP 161,162)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	SSDP (UDP 1900)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
	Trivial File Transfer Protocol (UDP 69)	Enable/Disable Distributed Reflection DoS (DRDoS) Attack Detection
FIN Scan	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Flow Threshold	The number of FIN flows from a single host within a three-minute period to trigger an event
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
Host Indexing	Days of host index data retention	The host index entries last seen more than this many days ago will be trimmed.
	Host Index Database	File path of Host Index. *Background service must be restart from CLI after update. Service will start clean in new location.
	Host Indexing Domain Socket	File path of Host Indexing Domain Socket
	Host Index Max Disk Space	Maximum combined disk space threshold for host indexing (in MB). Warning events sent at 75%, indexing temporarily suspended at 100% until record expiration frees space.
	Host Index Sync Interval Minutes	The sync interval in minutes for each index update

continues on next page

Table 5 – continued from previous page

Algorithm Name	Setting	Description
	Host-to-Host Index	Toggle Host-to-Host indexing
	Host-to-Host Index Database	File path of Host-to-Host Index. Leave blank to disable Host-to-Host indexing. *Background service must be restart from CLI after update. Service will start clean in new location.
	Window Limit	The maximum number of records considered on each index update
Host Reputation	Aggregate Timeout	Aggregate similar alarms until there are no new alarms for over N minutes (default 2 hours = 120 minutes, zero to disable aggregation)
	Threshold	Number of bytes (octets) within a three-minute period to trigger alarm
ICMP Destination Unreachable	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Flow Threshold	The number flows from a single host triggering an ICMP Destination Unreachable reponse within a three-minute period
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
ICMP Port Unreachable	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
	Threshold	The number flows from a single host triggering an ICMP Port Unreachable reponse within a three-minute period
IP Address Violations	Threshold	Number of bytes (octets) within a three-minute period to trigger alarm
Large Ping	Size Threshold	Average packet threshold for determining a large ping packet.
Lateral Movement Attempt	Backdoor Threshold	Number of destination hosts on backdoor ports to trigger alert
	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
	IOT Threshold	Number of destination hosts on IOT ports to trigger alert
	Remote Access Threshold	Number of destination hosts on remote access ports to trigger alert
	Windows Remote Access Threshold	Number of destination hosts on Windows remote access ports to trigger alert
Medianet Jitter Violations	Jitter by Interface	The millisecond variation in packet delay caused by queuing, contention and/or serialization effects on the path through the network. Default = 80 ms. This is also used for record highlighting in Status reports.
Multicast Violations	Threshold	Number of bytes (octets) within a three-minute period to trigger alarm
NULL Scan	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Flow Threshold	The number of flows from a single host within a three-minute period to trigger an event
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
Odd TCP Flags Scan	External to Internal	Enable/Disable Scan Detection in the direction indicated

continues on next page

Table 5 – continued from previous page

Algorithm Name	Setting	Description
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
	Threshold	The number of flows from a single host with odd TCP flags within a three-minute period to trigger an event
P2P Detection	Threshold	Number of distinct destination IPs in a three-minute period to trigger alarm
Packet Flood	Packet Size Threshold	The Maximum average packet size to be considered a flood packet
	Packet threshold	The number of packets that should be observed within a three-minute period to trigger an event
Persistent Flow Risk	Active Flow Threshold (hours)	How long should a flow be active before an alarm is triggered
	Aggregate Timeout	Aggregate similar alarms until there are no new alarms for over N minutes (default 2 hours = 120 minutes, zero to disable aggregation)
	Inactive Flow Threshold (hours)	How long should a flow be inactive before it no longer is considered the same flow
	PCR Threshold	The ratio of traffic where 1 is a pure upload and -1 is a pure download. Set to 0 to disable
Persistent Flow Risk - ASA	Active Flow Threshold (hours)	How long should a flow be active before an alarm is triggered
	Aggregate Timeout	Aggregate similar alarms until there are no new alarms for over N minutes (default 2 hours = 120 minutes, zero to disable aggregation)
	Inactive Flow Threshold (hours)	How long should a flow be inactive before it no longer is considered the same flow
	PCR Threshold	The ratio of traffic where 1 is a pure upload and -1 is a pure download. Set to 0 to disable
Ping Flood	Ping Flood Threshold	Minimum number of pings from a host to a distinct destination in a minute that should triggeer
Ping Scan	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
	Ping Scan Host Threshold	Minimum number of distinct hosts that a violator must ping to trigger
Reverse SSH Shell	Packet Size Threshold	Maximum average packet size in the SSH session that should be considered for triggering the alert
	Reverse Shell Threshold	The maximum number of outbound bytes on an SSH connection that should be considered for triggering the alert
RST/ACK Detection	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Flow Threshold	The number of flows from a single host within a three-minute period to trigger an event
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
SYN Scan	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Half-Open packet per port	The number of packets per dst port to be considered a half-open flood
	Half-Open port count	The number of distinct destination ports to be considered a half-open flood

continues on next page

Table 5 – continued from previous page

Algorithm Name	Setting	Description
	Host Scan Hosts	The number of distinct destination hosts to be considered a host scan
	Host Scan Ports	The number of distinct destination ports to be considered a host scan
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
	Port Scan Hosts	The number of distinct destination hosts to be considered a port scan
	Port Scan Ports	The number of distinct destination ports to be considered a port scan
TCP Scan	Destination Host Threshold	Number of distinct destination hosts to trigger alarm
	Destination Port Threshold	Number of distinct destination ports to trigger alarm
	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
UDP Scan	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Host threshold	The number of hosts scanned within a three-minute period that will trigger an event
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated
	Port threshold	The number of ports per host scanned within a three-minute period that will trigger an event
XMAS Scan	External to Internal	Enable/Disable Scan Detection in the direction indicated
	Flow Threshold	The number of flows from a single host within a three-minute period to trigger an event
	Internal to External	Enable/Disable Scan Detection in the direction indicated
	Internal to Internal	Enable/Disable Scan Detection in the direction indicated

Functional IDs

Scrutinizer uses the generic functional accounts/IDs and access levels below to manage access to system components and functions.

Accounts/IDs

System Component	Account/ID	Type	Access Level	Function
Operating system	`root`	Interactive	Privileged	Provides root access to the Scrutinizer OS, with unrestricted shell, SSH, and console access
	`plixer`	Interactive	Privileged	Primary user for the interactive `scrut_util` CLI utility and provides access to run all Scrutinizer processes and services
	`pgbouncer`	Non-interactive	Non-privileged	Used to manage remote database access between nodes, e.g. user/role access, load balancing, etc.
	`postgres`	Non-interactive	Non-privileged	Used for database operations during deployment
	`webapp`	Non-interactive	Non-privileged	Primary HTTP services user
Database	`plixer`	Interactive	Privileged	Primary database role used by application processes for both local and remote access
	`postgres`	Non-interactive	Non-privileged	Used for local database access during deployment, upgrades, and scheduled `pg_cron` tasks
Web interface	`admin`	Interactive	Non-privileged	Provides full access to web interface management functions

- *Interactive* - can be used to grant a user all privileges inherent to the ID
- *Non-interactive* - reserved for internal use by the system and cannot be assigned to users

Access levels

Feature	Privileged Access	Non-Privileged Access
Permissions	Elevated, can bypass security controls	Limited, for routine tasks only
Scope of Control	System-wide or extensive	Limited to user's own files/space
Examples	System Administrator, Root User	Standard User, Guest User
Primary Goal	Administration and system management	Daily work and general use
Security Implication	High risk, prime target for attackers	Low risk, restricts potential damage
Best Practice	Used sparingly and for specific tasks	Default for most users

User permissions

The tables below list all individual permissions (by feature/permission set) that can be granted to users through user groups.

Alarms Administrator

Permission	Description
Acknowledge Bulletin Board Event	Ability to acknowledge events on Alarms tab bulletin boards
Delete Alarms	Permission to permanently delete alarms

Alarms User

Permission	Description
Alarms Tab	Access the Alarms tab

Dashboard Administrator

Permission	Description
Dashboard Administrator	Manage all dashboards created by any user

Dashboard User

Permission	Description
Create Dashboards	Create new Dashboards
Dashboards Tab	Access the Dashboards tab

Maps Administrator

Permission	Description
Mapping Groups Configuration	Define and manage device groups for network mapping
Mapping Objects Configuration	Define custom map objects and manage object/group object properties

Maps User

Permission	Description
Maps Tab	Access the Maps tab

Reporting Administrator

Permission	Description
Application Groups	Define custom applications using IP address and port rules
AS Names Configuration	View autonomous system (AS) numbers/properties
Delete Reports	Ability to delete saved reports regardless of owner
FlowPro Administrator	Manage FlowPro configuration
Host Names Configuration	Define custom hostname-to-IP mappings and static subnet labels for reporting
Replicator Administrator	Manage Replicator configuration
TOS Configuration	Add custom labels for Type of Service (ToS) and Differentiated Services Code Point (DSCP) values in reports
Well-known Ports Configuration	Edit WKP Configuration

Reporting Power User

Permission	Description
Add/Edit Report Filters	Permission to update the filters used in Status Tab reports
Report Designer	Design custom report type configurations
Report Folders	Create and manage folders to organize saved reports
Save Reports	Ability to name and save flow reports
Scheduled Report Administrator	Set up and manage scheduled email report configurations
Schedule Emailed Reports	Schedule a saved report to be emailed on a regular basis

Reporting User

Permission	Description
AI User	Access Scrutinizer's AI prompt
Replicator User	View Replicator summary data
Run Reports	Ability to run flow reports
Status Tab	Access the Status Tab

System Administrator

Permission	Description
Admin	Access Scrutinizer's administrative functions
AI Settings	Configure AI Settings including AI server URL, API Key, and which model to use
Alarm Notifications	Configure alarm notifications
Alarm Settings	Configure global alarm message options and Flow Inactivity and Interface Threshold Violation alarm settings
ASA ACL Descriptions	Add/edit ASA firewall credentials for ACL description retrieval
Authentication Tokens	Add and manage user authentication tokens
Authentication Types	Manage external authentication types
AWS Configuration	AWS configuration
Change User Passwords	The ability to change the passwords of other users without needing their credentials
Collectors	Manage Scrutinizer collectors and ML Engines in the environment
Configure SMTP server settings for email notifications and reports	Configure the mailserver Scrutinizer will use to send reports and emails
Create Users	The ability to create new local Scrutinizer user accounts
Data History Settings	Set alarm and flow data history retention durations
Delete Users	The ability to delete local Scrutinizer user accounts
Enable/disable and configure third-party integrations for Explore > Exporters view	Create, edit, and delete third-party integration links
Endpoint Analytics	Configure and enable/disable Endpoint Analytics integration
Enforce Session Timeout	If the system preference for user activity timeout is set, members of user groups with this permission will be timed-out of the UI according to that setting
Exporters	Manage and add protocol exclusions to flow-exporting devices in the environment

continues on next page

Table 6 – continued from previous page

Permission	Description
Flow Analytics Configuration	Configure Flow Analytics thresholds and settings
Flow Analytics Exclusions	Configure Flow Analytics exclusions
Flow Analytics Settings	Configure global settings and enable/disable FlowPro Defender for FA algorithms
Flow Log Ingestion	Third-party Flow Log source configuration
Google Maps Proxy Server Settings	Configure proxy server settings for Google Maps requests
Host Indexing	Host Indexing settings
Interface Details Configuration	Edit device interface details
IP Groups Configuration	Define rule-based IP range/subnet groups for reporting
LDAP Server Configuration	Manage LDAP server configuration used for Scrutinizer authentication
MAC Addresses Configuration	Add and manage custom MAC address labels
Notification Manager	Create and manage profiles to assign notification actions by alarm policy
Policy Manager	Reconfigure, enable/disable, and assign notification profiles to alarm policies
Protocol Exclusions	Define protocol exclusion rules for reporting
RADIUS Server Configuration	Manage RADIUS server configuration used for Scrutinizer authentication
Replicator	Configure and enable/disable Replicator integration
Reporting Configuration	Customize Scrutinizer reporting engine functions
Scrutinizer Audit Report	View logs of Scrutinizer user actions
Scrutinizer Language Configuration	Create and edit language localization settings
Scrutinizer Product Licensing	Add a Scrutinizer license key and view license details
Scrutinizer System Preferences	Configure general Scrutinizer environment preferences/settings
ServiceNow	Configure and manage ServiceNow instances for incident/ticket generation via notifications and collections
Single Sign-On Configuration	Add, Delete, and Edit Identity Provider configuration for Scrutinizer’s Single Sign-On Integration
SNMP Credentials	Manage SNMP credential sets for polling exporters in the environment
STIX-TAXII	Add and manage STIX-TAXII threat intelligence feeds
Syslog Server Settings	Syslog server configuration
TACACS+ Server Configuration	Manage TACACS+ server configuration used for Scrutinizer authentication
User Accounts	Manage user accounts and preferences
User Groups	Set up local user groups and manage access to features and resources
View User Identity Information	View identity and access information relevant to GDPR restrictions
Viptela Settings	Viptela Settings
Vitals Report	View the Scrutinizer server vitals reports

Required ports

Refer to the tables below to configure firewall rules when deploying Scrutinizer and other Plixer One components.

> **Note**
>
> For more information on configuring/defining custom firewall rules, refer to *these instructions*.

Scrutinizer

Source Component	Destination Component	Protocol	Port	Reason
All	NTP	UDP	123	Time Sync
All	DNS Server(s)	UDP	53	DNS
DNS Server(s)	All	UDP	53	DNS
Exporters	Scrutinizer Collector	UDP	2055,2056,4432,4739	Flow Telemetry
Exporters	Scrutinizer Collector	UDP	161	SNMP Polling
AD Users Server	Active Directory Server(s)	TCP	135	RPC Call for Username Collection
AD Users Server	Scrutinizer Collector	UDP	2055	Flow Telemetry
NTP Server	All	UDP	123	Time Sync
Scrutinizer Collector	Scrutinizer Reporter	TCP	22,80,443,5432,6432	Intraplatform Comms
Scrutinizer Collector	ML	TCP	22,30404,32000-32002,30323	Intraplatform Comms
Scrutinizer Collector	Exporters	ICMP	N/A	Up/Down Status Checks
Scrutinizer Collector	AWS S3 Bucket	TCP	443	AWS VPC Flow Log Integration
Scrutinizer Collector	Azure Storage Account	TCP	443	Azure Flow Log Integration
Scrutinizer Collector	Viptela IP	TCP	8443	Viptela Integration
Scrutinizer Collector	Exporters	UDP	161	SNMP Polling
Scrutinizer Reporter	Scrutinizer Collector	TCP	22,80,443,5432,6432	Intraplatform Comms
Scrutinizer Reporter	ML	TCP	22,30404,32000-32002,30323,31111	Intraplatform Comms
Scrutinizer Reporter	Mail Server	TCP	25,587	Mail Notifications
Scrutinizer Reporterr	SIEM	UDP	514	Syslog/CEF Notifications
Scrutinizer Reporter	nba.plixer.com	TCP	443	Signature Updates
Scrutinizer Reporter	LDAP Server	TCP	636	User Authentication
Scrutinizer Reporter	RADIUS Server	TCP	1645,1812	User Authentication
Scrutinizer Reporter	TACACS+ Server	TCP	49	User Authentication
User	Scrutinizer Reporter	TCP	443	Web UI Access (Setup and Usage)
User	Scrutinizer Reporter	TCP	22	CLI Access (Setup and Administration)
User	Scrutinizer Collector	TCP	22	CLI Access (Setup and Administration)
User	ML Engine	TCP	22	CLI Access (Setup and Administration)
User	ML Engine	TCP	31112	Kibana Access (Optional for Admins)
User	ML Engine	TCP	30880	Grafana Access (Optional for Admins)
User	ML Engine	TCP/UDP	53	Advanced DNS Monitoring
User	ML Engine	TCP	80	Advanced DNS Monitoring Landing Page for Blocked Sites

Plixer ML Engine

Source Component	Destination Component	Protocol	Port	Reason
Plixer ML Engine	Scrutinizer reporter	TCP	22	Kafka streaming configuration via SSH
Plixer ML Engine	Scrutinizer reporter	TCP	443	Scrutinizer reporting API access
Plixer ML Engine	Scrutinizer reporter	TCP	5432	PostgreSQL database access
User	Plixer ML Engine	TCP	22	SSH access
All	Plixer ML Engine	TCP	30888	ML engine API access
All	Plixer ML Engine	TCP	31111	Elasticsearch HTTPS endpoint access
User	Plixer ML Engine	TCP	31112	Kibana web interface (if enabled) access
All	Kafka bootstrap server	TCP	30323	Cluster layout discovery
All	Kafka brokers	TCP	32000, 32001, 32002, etc. (one port per replica; default: 3)	Communication with broker endpoints
Plixer ML Engine	Kafka exporter	TCP	30404	Kafka metrics exporter endpoint
All	Plixer ML Engine	UDP & TCP	53 (forwarded to 30053 by cluster load balancer on AWS/Azure)	Safe DNS service (if enabled)
All	Plixer ML Engine	TCP	443 (forwarded to 30443 by cluster load balancer on AWS/Azure)	Safe DNS HTTPS landing page (if Safe DNS is enabled and HTTPS is configured)
All	Plixer ML Engine	TCP	80 (forwarded to 30080 by cluster load balancer on AWS/Azure)	Safe DNS HTTP landing page for blocked domains (if Safe DNS is enabled)

Replicator

Source Component	Destination Component	Protocol	Port	Reason
Exporters	Replicator	UDP	2055,2056,4432,4739	Flow Telemetry
AD Users Server	Replicator	UDP	2055	Flow Telemetry
Replicator	LDAP Server	TCP	636	User Authentication
Replicator	Scrutinizer Collector	UDP	2055	Flow Telemetry
Scrutinizer Reporter	Replicator	TCP	22,443	Intraplatform Comms
User	Replicator	TCP	443	Web UI Access (Setup and Usage)
User	Replicator	TCP	22	CLI Access (Setup and Administration)

FlowPro

Source Component	Destination Component	Protocol	Port	Reason
FlowPro	Flow Collector	UDP	2055	Flow Telemetry
FlowPro	Replicator	UDP	2055	Flow Telemetry
FlowPro	nba.plixer.com	TCP	443	Signature Updates
User	FlowPro Sensor	TCP	22	CLI Access (Setup and Administration)

Endpoint Analytics

Source Component	Destination Component	Protocol	Port	Reason
All Endpoints	Endpoint Analytics	UDP	67	DHCP Helper
Endpoint Analytics	Exporters	UDP	161	SNMP Polling
Endpoint Analytics	SIEM	UDP	514	Syslog Event Notifications
Endpoint Analytics	Active Directory Server(s)	TCP	389,636	LDAP(S) query
Endpoint Analytics	nba.plixer.com	TCP	443	Signature Updates
Endpoint Analytics	Tenable IP	TCP	443	API Integration
Endpoint Analytics	MS Defender	TCP	443	API Integration
Exporters	Endpoint Analytics	UDP	162	SNMP Traps
Exporters	Endpoint Analytics	UDP	161	SNMP Polling
RADIUS Server(s)	Endpoint Analytics	UDP	1813	RADIUS Accounting
Scrutinizer Reporter	Endpoint Analytics	TCP	443	API Calls
User	Endpoint Analytics	TCP	443	Web UI Access (Setup and Usage)
User	Endpoint Analytics	TCP	22	CLI Access (Setup and Administration)

Report types

The tables below list all Scrutinizer report types and their data aggregation parameters by report type category.

Amazon AWS

Report	Description
Action	A grouping of Action trending Flows, Packets, Bytes. Information Elements: aws_action, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Action with Interface	A grouping of Action, Interface trending Flows, Packets, Bytes. Information Elements: aws_action, aws_interface, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Action with Interface and Dst	A grouping of Destination, Action, Interface trending Flows, Packets, Bytes. Information Elements: destinationipaddress, aws_action, aws_interface, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Action with Interface and Src	A grouping of Source, Action, Interface trending Flows, Packets, Bytes. Information Elements: sourceipaddress, aws_action, aws_interface, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Availablity Zones	A grouping of Availability Zone trending Flows, Packets, Bytes. Information Elements: aws_az_id, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Dst Service	A grouping of Destination Service trending Flows, Packets, Bytes. Information Elements: aws_pkt_destination_service, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Interface	A grouping of Interface trending Flows, Packets, Bytes. Information Elements: aws_interface, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Pair Interface	A grouping of Source, Interface, Destination trending Flows, Packets, Bytes. Information Elements: sourceipaddress, aws_interface, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Pair Interface Action	A grouping of Source, Interface, Action, Destination trending Flows, Packets, Bytes. Information Elements: sourceipaddress, aws_interface, aws_action, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Src Service	A grouping of Source Service trending Flows, Packets, Bytes. Information Elements: aws_pkt_source_service, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Src Service-Dst Service	A grouping of Source Service, Destination Service trending Flows, Packets, Bytes. Information Elements: aws_pkt_source_service, aws_pkt_destination_service, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Traffic Path	A grouping of Path trending Flows, Packets, Bytes. Information Elements: aws_traffic_path, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
VPCs	A grouping of VPC trending Flows, Packets, Bytes. Information Elements: aws_vpc_id, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.

AppFlow

Report	Description
Application	A grouping of Application trending Count, Packets, Bytes. Information Elements: appflow_applicationid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Application RTT	A grouping of Application, Destination trending Packets, Bytes, RTT. Information Elements: appflow_applicationid, destinationipaddress, octetdeltacount, packetdeltacount, tcprtt.
Connections	A grouping of Src Port, Source, Connection, Destination, Dst Port trending RTT, Count, Packets, Bytes. Information Elements: sourcetransportport, sourceipaddress, connectionid, destinationipaddress, destinationtransportport, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount, tcprtt.
HTTP Request Cookie	A grouping of Transaction ID, HTTP Request Cookie trending Flow Count, Bytes. Information Elements: transactionid, httprequestcookie, octetdeltacount, plixeraggregatedrecordcount.
HTTP Response Length	A grouping of Source, Src Port, Destination, Dst Port trending Count, Avg. Length. Information Elements: sourceipaddress, sourcetransportport, destinationipaddress, destinationtransportport, httpresponselen, plixeraggregatedrecordcount.
HTTP Response Time to First Byte	A grouping of Source, Src Port, Destination, Dst Port trending Count, Avg. Time. Information Elements: sourceipaddress, sourcetransportport, destinationipaddress, destinationtransportport, httpresponsetimetofirstbyte, plixeraggregatedrecordcount.
HTTP Response Time to Last Byte	A grouping of Source, Src Port, Destination, Dst Port trending Count, Avg. Time. Information Elements: sourceipaddress, sourcetransportport, destinationipaddress, destinationtransportport, httpresponsetimetolastbyte, plixeraggregatedrecordcount.
HTTP Status	A grouping of HTTP Status Code, Source, Src Port, Destination, Dst Port trending Count, Bytes. Information Elements: httpresponsestatus, sourceipaddress, sourcetransportport, destinationipaddress, destinationtransportport, octetdeltacount, plixeraggregatedrecordcount.
Request Host	A grouping of HTTP Request Host trending Count, Packets, Bytes. Information Elements: httprequesthost, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Request URL	A grouping of Request URL trending Count, Packets, Bytes. Information Elements: httprequesturl, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Syslog Message Flow Count	A grouping of syslogPriority, Syslog Message trending Flow Count. Information Elements: syslogpriority, syslogmessage, plixeraggregatedrecordcount.

Astaro

Report	Description
afcprotocol Conversations	A grouping of Source, afcprotocol, Destination trending Packets, Bytes. Information Elements: sourceipaddress, afcprotocol, destinationipaddress, octetdeltacount, packetdeltacount.
Top afcprotocol	A grouping of afcprotocol trending Packets, Bytes. Information Elements: afcprotocol, octetdeltacount, packetdeltacount.

Azure

Report	Description
Azure NSG All Details	A grouping of Rule Name, Application, Flow Decision, Flow State trending Packets, Bytes, Count. Information Elements: nsg_rulename, applicationid, nsg_flowdecision, nsg_flowstate, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure NSG Flow Decisions	A grouping of Flow Decision, Application trending Packets, Bytes, Count. Information Elements: nsg_flowdecision, applicationid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure NSG Flow Decisions Count	A grouping of Flow Decision trending Packets, Bytes, Count. Information Elements: nsg_flowdecision, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure NSG Flow States	A grouping of Flow State, Application trending Packets, Bytes, Count. Information Elements: nsg_flowstate, applicationid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure NSG Flow States Count	A grouping of Flow State trending Packets, Bytes, Count. Information Elements: nsg_flowstate, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure NSG Resource IDs	A grouping of Resource ID, Rule Name trending Packets, Bytes, Count. Information Elements: nsg_resourceid, nsg_rulename, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure VNET All Details	A grouping of Rule Name, Application, Flow State trending Packets, Bytes, Count. Information Elements: vnet_rulename, applicationid, vnet_flowstate, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure VNET Flow States	A grouping of Flow State, Application trending Packets, Bytes, Count. Information Elements: vnet_flowstate, applicationid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure VNET Flow States Count	A grouping of Flow State trending Packets, Bytes, Count. Information Elements: vnet_flowstate, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Azure VNET Resource IDs	A grouping of Target Resource ID, Rule Name trending Packets, Bytes, Count. Information Elements: vnet_targetresourceid, vnet_rulename, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.

Barracuda

Report	Description
Bind and Conn	A grouping of Bind IP , Bind Port, Conn IP, Conn Port trending Flows, Bytes. Information Elements: bindipv4address, bindtransportport, connipv4address, conntransportport, octetdeltacount, plixeraggregatedrecordcount.
FW Rule	A grouping of FW Rule trending Flows, Packets, Bytes. Information Elements: fwrule, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Logop	A grouping of Logop trending Flows, Packets, Bytes. Information Elements: logop, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Pair with Rule and Reason	A grouping of Source IP, Destination IP, FW Rule, Reason trending Flows, Bytes. Information Elements: sourceipaddress, destinationipaddress, fwrule, reasontext, octetdeltacount, plixeraggregatedrecordcount.
Pair with Rule, Reason, Service & Traffic	A grouping of Source IP, Destination IP, FW Rule, Reason, Service, Traffic Type trending Flows, Bytes. Information Elements: sourceipaddress, destinationipaddress, fwrule, reasontext, servicename, traffictype, octetdeltacount, plixeraggregatedrecordcount.
Reason	A grouping of Reason trending Flows, Packets, Bytes. Information Elements: reasontext, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Rule, Reason, Service, Traffic & Logop	A grouping of FW Rule, Reason, Service, Traffic Type, Logop trending Flows, Bytes. Information Elements: fwrule, reasontext, servicename, traffictype, logop, octetdeltacount, plixeraggregatedrecordcount.
Service	A grouping of Service trending Flows, Packets, Bytes. Information Elements: servicename, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Source, Bind, Conn, & Destination	A grouping of Source IP, Bind IP , Conn IP, Destination IP trending Flows, Bytes. Information Elements: sourceipaddress, bindipv4address, connipv4address, destinationipaddress, octetdeltacount, plixeraggregatedrecordcount.
Traffic Type	A grouping of Traffic Type trending Flows, Packets, Bytes. Information Elements: traffictype, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.

Chassis

Report	Description
Line Card	A grouping of Line Card trending Pkts, Bytes. Information Elements: linecardid, octetdeltacount, packetdeltacount.
Line Card Port	A grouping of Interface, Line Card, Port trending Pkts, Bytes. Information Elements: exportinterface, linecardid, portid, octetdeltacount, packetdeltacount.

Cisco AnyConnect

Report	Description
Dest Host IP & Name	A grouping of Destination, Dst Host Name trending Flows, Bytes In. Information Elements: destinationipaddress, nvzflowdestinationhostname, octetdeltacount, plixeraggregatedrecordcount.
DNS suffix	A grouping of DNS Suffix trending Flows, Bytes In. Information Elements: nvzflowdnssuffix, octetdeltacount, plixeraggregatedrecordcount.
Loggedin Source	A grouping of Logged User, Source trending Bytes, Flows. Information Elements: nvzflowloggedinuser, sourceipaddress, octetdeltacount, plixeraggregatedrecordcount.
Loggedin Source & DNS	A grouping of Logged User, Source, DNS Suffix trending Flows, Bytes. Information Elements: nvzflowloggedinuser, sourceipaddress, nvzflowdnssuffix, octetdeltacount, plixeraggregatedrecordcount.
Pair with Host Details	A grouping of Logged User, Source, Destination, Dst Host Name trending Flows, Bytes. Information Elements: nvzflowloggedinuser, sourceipaddress, destinationipaddress, nvzflowdestinationhostname, octetdeltacount, plixeraggregatedrecordcount.
Parent Process Details	A grouping of Parent Proc. Acct., Parent Proc. Name, Parent Proc. Hash trending Flows, Bytes. Information Elements: nvzflowparentprocessaccount, nvzflowparentprocessname, nvzflowparentprocesshash, octetdeltacount, plixeraggregatedrecordcount.
Process Details	A grouping of Process Name, Process Hash trending Flows, Bytes. Information Elements: nvzflowprocessname, nvzflowprocesshash, octetdeltacount, plixeraggregatedrecordcount.
Process to Host	A grouping of Parent Proc. Acct., Destination trending Flows, Bytes In. Information Elements: nvzflowparentprocessaccount, nvzflowdestinationhostname, octetdeltacount, plixeraggregatedrecordcount.
Source with Process	A grouping of Source, Logged User, Process Name, System Type trending Flows, Bytes. Information Elements: sourceipaddress, nvzflowloggedinuser, nvzflowprocessname, nvzflowsystemtype, octetdeltacount, plixeraggregatedrecordcount.
Station Name & Dst IP	A grouping of STA Name, Destination trending Flows, Bytes In. Information Elements: nvz_manu_virtual_station_name, destinationipaddress, octetdeltacount, plixeraggregatedrecordcount.
Station Name & Manufacturer	A grouping of STA Name, Manufacturer trending Flows, Bytes. Information Elements: nvz_manu_virtual_station_name, nvz_manu_system_manufacturer, octetdeltacount, plixeraggregatedrecordcount.
Station Name & Process	A grouping of STA Name, Process Account trending Flows, Bytes In. Information Elements: nvz_manu_virtual_station_name, nvz_manu_process_account, octetdeltacount, plixeraggregatedrecordcount.
Station Name & Src IP	A grouping of STA Name, Source trending Flows, Bytes In. Information Elements: nvz_manu_virtual_station_name, sourceipaddress, octetdeltacount, plixeraggregatedrecordcount.

Cisco AVC

Report	Description
EzPM: Host Jitter by SSRC (Dst)	A grouping of Destination, DSCP, SSRC trending % Pkt Loss, TEPL, Jitter. Information Elements: destinationipaddress, ipdiffservcodepoint, trans_rtp_ssrc, ciscopktlostpercent, rtp_jitter_mean_sum, trans_pkt_lost_count.
EzPM: Host Jitter by SSRC (Src)	A grouping of Source, DSCP, SSRC trending % Pkt Loss, TEPL, Jitter. Information Elements: sourceipaddress, ipdiffservcodepoint, trans_rtp_ssrc, ciscopktlostpercent, rtp_jitter_mean_sum, trans_pkt_lost_count.
EzPM: Host Jitter (Dst)	A grouping of Destination, DSCP trending % Pkt Loss, TEPL, Jitter. Information Elements: destinationipaddress, ipdiffservcodepoint, ciscopktlostpercent, rtp_jitter_mean_sum, trans_pkt_lost_count.
EzPM: Host Jitter (Src)	A grouping of Source, DSCP trending % Pkt Loss, TEPL, Jitter. Information Elements: sourceipaddress, ipdiffservcodepoint, ciscopktlostpercent, rtp_jitter_mean_sum, trans_pkt_lost_count.
EzPM: Host to Host Jitter	A grouping of Source, DSCP, Destination trending % Pkt Loss, TEPL, Max Jitter, Jitter. Information Elements: sourceipaddress, ipdiffservcodepoint, destinationipaddress, ciscopktlostpercent, rtp_jitter_mean_sum, trans_pkt_lost_count.
EzPM: Host to Host Jitter by SSRC	A grouping of Source, DSCP, Destination, SSRC trending % Pkt Loss, TEPL, Jitter. Information Elements: sourceipaddress, ipdiffservcodepoint, destinationipaddress, trans_rtp_ssrc, ciscopktlostpercent, rtp_jitter_mean_sum, trans_pkt_lost_count.
EzPM: Jitter by Interface	A grouping of Exporter, in Int trending % Pkt Loss, Jitter. Information Elements: plixerexporter, ingressinterface, ciscopktlostpercent, rtp_jitter_mean_sum.
EzPM: Metadata Jitter	A grouping of Application trending % Pkt Loss, TEPL, Jitter. Information Elements: applicationtag, ciscopktlostpercent, rtp_jitter_mean_sum, trans_pkt_lost_count.
EzPM: Metadata Jitter by DSCP	A grouping of Application, DSCP trending %

Cisco CTS

Report	Description
ctsDestination Group	A grouping of ctsdestinationgrouptag trending Packets, Bytes. Information Elements: ctsdestinationgrouptag, octetdeltacount, packetdeltacount.
ctsGroups Connections	A grouping of src Port, Group Tag, ctsdestinationgrouptag, dst Port trending Packets, Bytes. Information Elements: sourcetransportport, ctssourcegrouptag, ctsdestinationgrouptag, destinationtransportport, octetdeltacount, packetdeltacount.
ctsGroups Conversations	A grouping of Group Tag, Well Known, ctsdestinationgrouptag, Rate trending Packets, Bytes. Information Elements: ctssourcegrouptag, commonport, ctsdestinationgrouptag, rate, octetdeltacount, packetdeltacount.
ctsGroups Grouped Flows	A grouping of src Port, Group Tag, Type Of Service, ctsdestinationgrouptag, dst Port trending Packets, Bytes. Information Elements: sourcetransportport, ctssourcegrouptag, ipclassofservice, ctsdestinationgrouptag, destinationtransportport, octetdeltacount, packetdeltacount.
ctsSource Group	A grouping of Group Tag trending Packets, Bytes. Information Elements: ctssourcegrouptag, octetdeltacount, packetdeltacount.
ctsSrcGrp to ctsDstGrp	A grouping of Group Tag, ctsdestinationgrouptag trending Packets, Bytes. Information Elements: ctssourcegrouptag, ctsdestinationgrouptag, octetdeltacount, packetdeltacount.

Cisco FW

Report	Description
ACL to ACL	A grouping of Ingress ACL, Egress ACL trending Flows. Information Elements: nf_f_ingress_acl_id, nf_f_egress_acl_id, plixeraggregatedrecordcount.
Egress ACL	A grouping of Egress ACL trending Flows. Information Elements: nf_f_egress_acl_id, plixeraggregatedrecordcount.
Ingress ACL	A grouping of Ingress ACL trending Flows. Information Elements: nf_f_ingress_acl_id, plixeraggregatedrecordcount.

Cisco HSL

Report	Description
Classes	A grouping of Class, Packets trending Bytes. Information Elements: classid, packetdeltacount, octetdeltacount.
Destination-Event	A grouping of Destination, Firewall Event, Extended Event Code, Zone Pair trending Flows. Information Elements: destinationipaddress, firewallevent, fw_ext_event, zonepair_id, plixeraggregatedrecordcount.
Host to Host Events	A grouping of Source, Destination, Firewall Event, Extended Event Code, Zone Pair trending Flows. Information Elements: sourceipaddress, destinationipaddress, firewallevent, fw_ext_event, zonepair_id, plixeraggregatedrecordcount.
Host to Host Events by VRF	A grouping of In VRF, Source, Destination, Out VRF, Firewall Event, Extended Event Code trending Flows. Information Elements: ingressvrfid, sourceipaddress, destinationipaddress, egressvrfid, firewallevent, fw_ext_event, plixeraggregatedrecordcount.
Host to Host with Zone and Class	A grouping of Source, Class, Zone Pair, Destination trending Bytes. Information Elements: sourceipaddress, classid, zonepair_id, destinationipaddress, octetdeltacount.
Source-Event	A grouping of Source, Firewall Event, Extended Event Code, Zone Pair trending Flows. Information Elements: sourceipaddress, firewallevent, fw_ext_event, zonepair_id, plixeraggregatedrecordcount.
Zone Pair	A grouping of Zone Pair trending Bytes. Information Elements: zonepair_id, octetdeltacount.
Zone Pair and Class	A grouping of Zone Pair, Class trending Bytes. Information Elements: zonepair_id, classid, octetdeltacount.
Zone Pair Volume	A grouping of Zone Pair trending Flows. Information Elements: zonepair_id, plixeraggregatedrecordcount.

Cisco IWAN

Report	Description
IWAN Bandwidth Usage	A grouping of Source Site, Path Tag ID, Interface Description trending BW In, Speed In, BW Out, Speed Out. Information Elements: source_site_id, path_tag_id, interfacedescription, egress_bw, ingress_bw, maxof_egress_bw, maxof_ingress_bw.
IWAN Route Changes	A grouping of Site, BR, Path Tag ID, IWAN Circuit trending Routes Changed. Information Elements: source_site_id, ipv4_br_addr, path_tag_id, interfacedescription, plixeraggregatedrecordcount.
IWAN Site to Site Bandwidth	A grouping of BR Router, Src Site, Dst Site, Dst Prefix, Interface ID trending Packets, Avg Bits. Information Elements: ipv4_br_addr, source_site_id, destination_site_id, destination_site_prefix, egressinterface, octetdeltacount, packetdeltacount.
IWAN Traffic Control Alerts	A grouping of Source Site, Destination Site, Interface Description, Interface ID, BR Addr, Path Tag ID, Status trending One way delay, AVG Jitter, PKT Loss, Bytes Lost. Information Elements: source_site_id, destination_site_id, interfacedescription, egressinterface, ipv4_br_addr, path_tag_id, oer_unreach, one_way_delay, rtp_jitter_inter_arrival_mean, trans_pkt_lost_rate, trns_cnt_bytes_lost_rate.

Cisco SLT

Report	Description
Event	A grouping of l2l3switchevent trending Count, Packets, Bytes. Information Elements: l2l3switchevent, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Event-Extevent	A grouping of l2l3switchevent, l2l3switchextevent trending Count, Packets, Bytes. Information Elements: l2l3switchevent, l2l3switchextevent, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Int	A grouping of Exporter, ingressphysicalinterface trending Count, Packets, Bytes. Information Elements: plixerexporter, ingressphysicalinterface, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Int-Vlan-Event	A grouping of Exporter, ingressphysicalinterface, vlanid, l2l3switchevent trending Count, Packets, Bytes. Information Elements: plixerexporter, ingressphysicalinterface, vlanid, l2l3switchevent, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Vlan	A grouping of Exporter, vlanid trending Count, Packets, Bytes. Information Elements: plixerexporter, vlanid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.

Cisco VQM

Report	Description
Connections eMOS	A grouping of Source, Src Port, Destination, Dest Port trending Frame Rate, eMOS Score. Information Elements: sourceipaddress, sourcetransportport, destinationipaddress, destinationtransportport, videoemosscore, vqmframerate.
Connections eMOS Detail	A grouping of Source, Src Port, Destination, Dest Port trending Frame Rate, eMOS Pkt Lost, eMOS Compression, eMOS Score. Information Elements: sourceipaddress, sourcetransportport, destinationipaddress, destinationtransportport, videoemosscore, vqmemoscompressionbitstream, vqmemospacketlostbitstream, vqmframerate.
Destination eMOS	A grouping of Destination trending Frame Rate, eMOS Score. Information Elements: destinationipaddress, videoemosscore, vqmframerate.
Destination eMOS Detail	A grouping of Destination trending Frame Rate, eMOS Pkt Lost, eMOS Compression, eMOS Score. Information Elements: destinationipaddress, videoemosscore, vqmemoscompressionbitstream, vqmemospacketlostbitstream, vqmframerate.
Host to Host eMOS	A grouping of Source, Destination trending Frame Rate, eMOS Score. Information Elements: sourceipaddress, destinationipaddress, videoemosscore, vqmframerate.
Host to Host eMOS Detail	A grouping of Source, Destination trending Frame Rate, eMOS Pkt Lost, eMOS Compression, eMOS Score. Information Elements: sourceipaddress, destinationipaddress, videoemosscore, vqmemoscompressionbitstream, vqmemospacketlostbitstream, vqmframerate.
Source eMOS	A grouping of Source trending Frame Rate, eMOS Score. Information Elements: sourceipaddress, videoemosscore, vqmframerate.
Source eMOS Detail	A grouping of Source trending Frame Rate, eMOS Pkt Lost, eMOS Compression, eMOS Score. Information Elements: sourceipaddress, videoemosscore, vqmemoscompressionbitstream, vqmemospacketlostbitstream, vqmframerate.

Client Server

Report	Description
Client	A grouping of Client IP trending sum_plxr_client_bytes, sum_plxr_server_bytes. Information Elements: plxr_client_ip, plxr_client_bytes, plxr_server_bytes.
Client Apps	A grouping of Client IP, Application ID trending sum_plxr_client_bytes, sum_plxr_server_bytes. Information Elements: plxr_client_ip, applicationid, plxr_client_bytes, plxr_server_bytes.
Client Server	A grouping of Client IP, Server IP trending sum_plxr_client_bytes, sum_plxr_server_bytes. Information Elements: plxr_client_ip, plxr_server_ip, plxr_client_bytes, plxr_server_bytes.
Client Server Apps	A grouping of Client IP, Application ID, Server IP trending Client, Server. Information Elements: plxr_client_ip, applicationid, plxr_server_ip, plxr_client_bytes, plxr_server_bytes.
Client Server Apps Flags	A grouping of Client IP, Application ID, Server IP trending TCP Flags, Client, Server. Information Elements: plxr_client_ip, applicationid, plxr_server_ip, plxr_client_bytes, plxr_server_bytes, tcpcontrolbits.
Client Server Flags	A grouping of Client IP, Server IP trending TCP Flags, sum_plxr_client_bytes, sum_plxr_server_bytes. Information Elements: plxr_client_ip, plxr_server_ip, plxr_client_bytes, plxr_server_bytes, tcpcontrolbits.
Server	A grouping of Server IP trending sum_plxr_client_bytes, sum_plxr_server_bytes. Information Elements: plxr_server_ip, plxr_client_bytes, plxr_server_bytes.
Server Apps	A grouping of Server IP, Application ID trending sum_plxr_client_bytes, sum_plxr_server_bytes. Information Elements: plxr_server_ip, applicationid, plxr_client_bytes, plxr_server_bytes.

Counts

Report	Description
Clients	A grouping of Client trending Flows. Information Elements: clientipv4address, plixeraggregatedrecordcount.
Destination	A grouping of Destination trending Flows. Information Elements: destinationipaddress, plixeraggregatedrecordcount.
Initiator Group with Dst Port	A grouping of Source IP Group, Well Known Port, Destination IP Group, Destination Port trending Packets, Bytes, Flows. Information Elements: srcipgroup, commonport, dstipgroup, destinationtransportport, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Internal External Destinations	A grouping of Destination trending Unique Hosts. Information Elements: dstinternal, destinationipaddress.
Internal External Pairs	A grouping of Source, Destination trending Unique Srcs, Unique Dsts. Information Elements: srcinternal, dstinternal, destinationipaddress, sourceipaddress.
Internal External Sources	A grouping of Source trending Unique Hosts. Information Elements: srcinternal, sourceipaddress.
Pairs	A grouping of Source, Destination trending Flows. Information Elements: sourceipaddress, destinationipaddress, plixeraggregatedrecordcount.
Pair Source post NAT	A grouping of Source, Src Post NAT, Destination trending Flows. Information Elements: sourceipaddress, postnatsourceipv4address, destinationipaddress, plixeraggregatedrecordcount.
Pair Source post NAT and NAP	A grouping of Source, Src Post NAT, Src Port, Src NAP Port, Dst Port, Destination trending Flows. Information Elements: sourceipaddress, postnatsourceipv4address, sourcetransportport, postnaptsourcetransportport, destinationtransportport, destinationipaddress, plixeraggregatedrecordcount.
Protocol	A grouping of Protocol trending Flows. Information Elements: protocolidentifier, plixeraggregatedrecordcount.
Servers	A grouping of Server trending Flows. Information Elements: serveripv4address, plixeraggregatedrecordcount.
Source	A grouping of Source trending Flows. Information Elements: sourceipaddress, plixeraggregatedrecordcount.
VRFID with NAT and Src	A grouping of In VRFID, NAT Event, NAT Pool Name, Source trending Flows. Information Elements: ingressvrfid, natevent, natpoolname, sourceipaddress, plixeraggregatedrecordcount.
Well Known Port	A grouping of Well Known trending Flows. Information Elements: commonport, plixeraggregatedrecordcount.

Destination Reports

Report	Description
Autonomous System by IP	A grouping of Destination AS trending Packets, Bytes. Information Elements: dstipas, octetdeltacount, packetdeltacount.
Autonomous System by Tag	A grouping of Dst AS trending Packets, Bytes. Information Elements: bgpdestinationasnumber, octetdeltacount, packetdeltacount.
Autonomous System by Tag (Peer)	A grouping of bgpnextadjacentasnumber trending Packets, Bytes. Information Elements: bgpnextadjacentasnumber, octetdeltacount, packetdeltacount.
Countries	A grouping of Destination Country trending Packets, Bytes. Information Elements: dstcountry, octetdeltacount, packetdeltacount.
Countries with AS	A grouping of Dest Country, Dest AS, Hosts (Dst) trending Flows, Packets, Bytes. Information Elements: dstcountry, dstipas, sourceipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Customer VLAN	A grouping of postdot1qcustomervlanid trending Flows, Packets, Bytes. Information Elements: postdot1qcustomervlanid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Destination w/Flags	A grouping of Destination IP Address, tcpcontrolbits trending Packets, Bytes, Flows. Information Elements: destinationipaddress, tcpcontrolbits, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Dest. IP Groups	A grouping of Destination IP Group trending Packets, Bytes. Information Elements: dstipgroup, octetdeltacount, packetdeltacount.
dot1q VLAN	A grouping of postdot1qvlanid trending Flows, Packets, Bytes. Information Elements: postdot1qvlanid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Dst IP - Src AS	A grouping of Exporter, Destination IP Address, Src AS trending Packets, Bytes. Information Elements: plixerexporter, destinationipaddress, bgpsourceasnumber, octetdeltacount, packetdeltacount.
Host Flows	A grouping of Destination trending Hosts (Source), Packets, Flows. Information Elements: destinationipaddress, packetdeltacount, plixeraggregatedrecordcount, sourceipaddress.
Hosts	A grouping of Destination trending Packets, Bytes. Information Elements: destinationipaddress, octetdeltacount, packetdeltacount.
ICMP	A grouping of Destination, Code, Type trending Count. Information Elements: destinationipaddress, icmpcodeipv4, icmptypeipv4, plixeraggregatedrecordcount.
L2 Octets	A grouping of Destination trending Packets, L2 Octets. Information Elements: destinationipaddress, layer2octetdeltacount, packetdeltacount.
MAC	A grouping of Destination MAC trending Flows, Packets, Bytes. Information Elements: destinationmacaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
MAC L2	A grouping of Destination MAC trending Packets, L2 Octets. Information Elements: destinationmacaddress, layer2octetdeltacount, packetdeltacount.
MAC Sum of Sq	A grouping of Destination MAC trending Packets, Sum

Exinda

Report	Description
Application - App Group	A grouping of Exporter, Application, GROUP, TRAFFIC_CLASS trending Avg Srv Del, RTT, NULL. Information Elements: plixerexporter, applicationtag, ex_app_group_name, ex_traffic_class, ex_rtt, ex_server_delay, octetdeltacount.
Application Detail	A grouping of Application trending Avg. AQS, Packets, NULL. Information Elements: applicationtag, ex_aqs, octetdeltacount, packetdeltacount.
Application Group	A grouping of Exporter, GROUP, TRAFFIC_CLASS trending Avg Srv Del, RTT, NULL. Information Elements: plixerexporter, ex_app_group_name, ex_traffic_class, ex_rtt, ex_server_delay, octetdeltacount.
Application Performances	A grouping of Application trending Avg. AQS, Bytes Lost, Nwk. Delay, Srv. Delay, RTT. Information Elements: applicationtag, ex_aqs, ex_bytes_lost, ex_net_delay, ex_rtt, ex_server_delay.
Destination User	A grouping of Exporter, dst_user trending Packets, NULL. Information Elements: plixerexporter, ex_user_id_dst, octetdeltacount, packetdeltacount.
Extra Info	A grouping of Exporter, EXTRA_INFO_ID, TRAFFIC_CLASS trending Bytes Lost, Avg Srv Del, RTT, NULL. Information Elements: plixerexporter, ex_extra_info_id, ex_traffic_class, ex_bytes_lost, ex_rtt, ex_server_delay, octetdeltacount.
Pair by Policy	A grouping of Exporter, Source, Destination, Policy trending Packets, NULL. Information Elements: plixerexporter, sourceipaddress, destinationipaddress, ex_policy_id, octetdeltacount, packetdeltacount.
Pair Latency	A grouping of Source, Destination, TRAFFIC_CLASS trending Avg Srv Del, RTT, NULL. Information Elements: sourceipaddress, destinationipaddress, ex_traffic_class, ex_rtt, ex_server_delay, octetdeltacount.
Pair, Ports and Latency	A grouping of Source, Src Port, Dst Port, Destination trending Avg Srv Del, RTT, NULL. Information Elements: sourceipaddress, sourcetransportport, destinationtransportport, destinationipaddress, ex_rtt, ex_server_delay, octetdeltacount.
Pair VoIP Details	A grouping of Source, Destination, TRAFFIC_CLASS trending Avg. mos, Avg. Refactor, Jitter, NULL. Information Elements: sourceipaddress, destinationipaddress, ex_traffic_class, ex_net_jitter, ex_voip_mos, ex_voip_rfactor, octetdeltacount.
Policies	A grouping of Exporter, ex_policy_id, TRAFFIC_CLASS trending Bytes Lost, Avg Srv Del, RTT, NULL. Information Elements: plixerexporter, ex_policy_id, ex_traffic_class, ex_bytes_lost, ex_rtt, ex_server_delay, octetdeltacount.
Source Latency	A grouping of Source, TRAFFIC_CLASS trending Bytes Lost, Avg Srv Del, RTT, NULL. Information Elements: sourceipaddress, ex_traffic_class, ex_bytes_lost, ex_rtt, ex_server_delay, octetdeltacount.
Source User	A grouping of Exporter, src_user trending Packets, NULL. Information Elements: plixerexporter, ex_user_id_src, octetdeltacount, packetdeltacount.

FirePOWER

Report	Description
App Internet HTTP Host	A grouping of Application, FS App, HTTP Host trending Flows, Bytes. Information Elements: applicationname, firesight_application, firesight_http_host, octetdeltacount, plixeraggregatedrecordcount.
Application E-Zone & Sub Type	A grouping of Application, FS App, Egress Zone, Event Subtype, Event Type trending Flows. Information Elements: applicationname, firesight_application, firesight_egress_zone, firesight_event_subtype, firesight_event_type, plixeraggregatedrecordcount.
Application I-Zone & Sub Type	A grouping of Application, FS App, Ingress Zone, Event Subtype, Event Type trending Flows. Information Elements: applicationname, firesight_application, firesight_ingress_zone, firesight_event_subtype, firesight_event_type, plixeraggregatedrecordcount.
Firewall List	A grouping of Firewall trending Flows, Packets, Bytes. Information Elements: firesight_sensor_ipv6, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Ingress and Egress Zones	A grouping of Ingress Zone, Egress Zone, Event Type trending Flows. Information Elements: firesight_ingress_zone, firesight_egress_zone, firesight_event_type, plixeraggregatedrecordcount.
User App HTTP Host	A grouping of Source IP, Username, Application, FS App, HTTP Host trending Flows, Bytes. Information Elements: sourceipaddress, username, applicationname, firesight_application, firesight_http_host, octetdeltacount, plixeraggregatedrecordcount.
User App HTTP URL	A grouping of Source IP, Username, Application, FS App, FS URL trending Flows. Information Elements: sourceipaddress, username, applicationname, firesight_application, firesight_http_url, plixeraggregatedrecordcount.
User Application	A grouping of Source IP, Username, Application, FS App trending Flows, Bytes. Information Elements: sourceipaddress, username, applicationname, firesight_application, octetdeltacount, plixeraggregatedrecordcount.
Web App and Source IP	A grouping of Web Application, Application, Source IP trending Flows, Packets, Bytes. Information Elements: firesight_web_application, applicationname, sourceipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Web App & CoS	A grouping of Web Application, CoS trending Flows, Packets, Bytes. Information Elements: firesight_web_application, ipclassofservice, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Web App Event & Rule Details	A grouping of Web Application, Event Subtype, Event Type, Rule Action, Rule Reason trending Flows. Information Elements: firesight_web_application, firesight_event_subtype, firesight_event_type, firesight_rule_action, firesight_rule_reason, plixeraggregatedrecordcount.

Firewall Events

Report	Description
Destination-Event	A grouping of Destination, Firewall Event trending Flows. Information Elements: destinationipaddress, firewallevent, plixeraggregatedrecordcount.
Destination-Event-Ext	A grouping of Destination, Firewall Event, Extended Event trending Flows. Information Elements: destinationipaddress, firewallevent, nf_f_fw_ext_event, plixeraggregatedrecordcount.
Event-Ext-ACL	A grouping of Firewall Event, Extended Event, Ingress ACL, Egress ACL trending Flows. Information Elements: firewallevent, nf_f_fw_ext_event, nf_f_ingress_acl_id, nf_f_egress_acl_id, plixeraggregatedrecordcount.
Firewall Events	A grouping of Firewall Event trending Count. Information Elements: firewallevent, plixeraggregatedrecordcount.
Firewall Events by Host	A grouping of Source, Firewall Event trending Count. Information Elements: sourceipaddress, firewallevent, plixeraggregatedrecordcount.
Firewall Events Ext	A grouping of FW Event Ext trending Flows. Information Elements: fw_ext_event, plixeraggregatedrecordcount.
Pairs-Event	A grouping of Source, Destination, Firewall Event trending Flows. Information Elements: sourceipaddress, destinationipaddress, firewallevent, plixeraggregatedrecordcount.
Pairs-Event Ext	A grouping of Source, Destination, FW Event Ext trending Flows. Information Elements: sourceipaddress, destinationipaddress, fw_ext_event, plixeraggregatedrecordcount.
Pairs-Event-Ext	A grouping of Source, Destination, Firewall Event, Extended Event trending Flows. Information Elements: sourceipaddress, destinationipaddress, firewallevent, nf_f_fw_ext_event, plixeraggregatedrecordcount.
Protocol-Event	A grouping of Protocol, Firewall Event trending Flows. Information Elements: protocolidentifier, firewallevent, plixeraggregatedrecordcount.
Protocol-Event-Ext	A grouping of Protocol, Firewall Event, Extended Event trending Flows. Information Elements: protocolidentifier, firewallevent, nf_f_fw_ext_event, plixeraggregatedrecordcount.
Source-Event	A grouping of Source, Firewall Event trending Flows. Information Elements: sourceipaddress, firewallevent, plixeraggregatedrecordcount.
Source-Event Ext	A grouping of Source, FW Event Ext trending Flows. Information Elements: sourceipaddress, fw_ext_event, plixeraggregatedrecordcount.
Source-Event-Ext	A grouping of Source, Firewall Event, Extended Event trending Flows. Information Elements: sourceipaddress, firewallevent, nf_f_fw_ext_event, plixeraggregatedrecordcount.
Users-Event	A grouping of Username, Firewall Event trending Flows, Bytes. Information Elements: username, firewallevent, octetdeltacount, plixeraggregatedrecordcount.
Users-Event-Ext	A grouping of Username, Firewall Event, Extended Event trending Flows, Bytes. Information Elements: username, firewallevent, nf_f_fw_ext_event, octetdeltacount, plixeraggregatedrecordcount.

FlowPro APM Reports

Report	Description
Application Latency	A grouping of L7 App trending Client, Server, Appl. Information Elements: l7_proto_name, appl_latency_ms, client_nw_delay_ms, server_nw_delay_ms.
App Priority & Latency	A grouping of L7 App, Priority trending Client, Server, Appl. Information Elements: l7_proto_name, ipclassofservice, appl_latency_ms, client_nw_delay_ms, server_nw_delay_ms.
Defined Application Latency	A grouping of Application trending Appl, Client, Server, Packets, Bytes. Information Elements: applicationid, appl_latency_ms, client_nw_delay_ms, octetdeltacount, packetdeltacount, server_nw_delay_ms.
Host Jitter	A grouping of Source trending Pkt Loss, Jitter, Packets, Bytes. Information Elements: sourceipaddress, octetdeltacount, packetdeltacount, rtp_in_jitter, rtp_in_pkt_lost.
Host Jitter By SSRC (Dst)	A grouping of Destination, SSRC, Codec trending Pkt Loss, Jitter, Packets, Bytes. Information Elements: destinationipaddress, rtp_ssrc, rtp_out_payload_type, octetdeltacount, packetdeltacount, rtp_out_jitter, rtp_out_pkt_lost.
Host Jitter By SSRC (Src)	A grouping of Source, SSRC, Codec trending Pkt Loss, Jitter, Packets, Bytes. Information Elements: sourceipaddress, rtp_ssrc, rtp_in_payload_type, octetdeltacount, packetdeltacount, rtp_in_jitter, rtp_in_pkt_lost.
Hosts Latency (Dst)	A grouping of Destination trending Appl, Client, Server, Packets, Bytes. Information Elements: destinationipaddress, appl_latency_ms, client_nw_delay_ms, octetdeltacount, packetdeltacount, server_nw_delay_ms.
Hosts Latency (Src)	A grouping of Source trending Appl, Client, Server, Packets, Bytes. Information Elements: sourceipaddress, appl_latency_ms, client_nw_delay_ms, octetdeltacount, packetdeltacount, server_nw_delay_ms.
Host to Host Jitter All by SSRC	A grouping of Source, Src Payload, SSRC, Destination, Dst Payload trending Src Pkt Loss, Src Jitter, Dst Pkt Loss, Dst Jitter, Packets, Bytes. Information Elements: sourceipaddress, rtp_in_payload_type, rtp_ssrc, destinationipaddress, rtp_out_payload_type, octetdeltacount, packetdeltacount, rtp_in_jitter, rtp_in_pkt_lost, rtp_out_jitter, rtp_out_pkt_lost.
Host to Host Jitter By SSRC/Codec	A grouping of Source, Destination, SSRC, Codec trending Pkt Loss, Jitter, Packets, Bytes. Information Elements: sourceipaddress, destinationipaddress, rtp_ssrc, rtp_in_payload_type, octetdeltacount, packetdeltacount, rtp_in_jitter, rtp_in_pkt_lost.
Host to Host Jitter By SSRC/ToS	A grouping of Source, Destination, SSRC, Type Of Service trending Pkt Loss, Jitter, Packets, Bytes. Information Elements: sourceipaddress, destinationipaddress, rtp_ssrc, ipclassofservice, octetdeltacount, packetdeltacount, rtp_in_jitter, rtp_in_pkt_lost.
Host to Host Latency	A grouping of Source, Destination trending Appl, Client, Server, Packets, Bytes. Information Elements: sourceipaddress, destinationipaddress, appl_latency_ms, client_nw_delay_ms, octetdeltacount, packetdeltacount, server_nw_delay_ms.
Initiator to Responder	A grouping of sip_calling_party, sip_called_party, Codec trending Jitter, Pkt Loss, Packets, Bytes. Infor-

FlowPro Defender Reports

Report	Description
Alert > All Details	A grouping of Category, Signature, Source, Destination trending Observation Count. Information Elements: nids_category, nids_signature, sourceipaddress, destinationipaddress, plixeraggregatedrecordcount.
Alert > Category	A grouping of Category trending Observation Count. Information Elements: nids_category, plixeraggregatedrecordcount.
Alert > Category & Signature	A grouping of Category, Signature trending Observation Count. Information Elements: nids_category, nids_signature, plixeraggregatedrecordcount.
DNS > Auth	A grouping of Auth Rname trending Observation Count. Information Elements: dns_soa_rname, plixeraggregatedrecordcount.
DNS Client Latency	A grouping of Client trending DNS Requests, Latency. Information Elements: dnsnxclientipv4address, dnsresolvetime, plixeraggregatedrecordcount.
DNS Client / Server Latency	A grouping of Client, Responding DNS Svr trending DNS Requests, Latency. Information Elements: dnsnxclientipv4address, dnsnxserveripv4address, dnsresolvetime, plixeraggregatedrecordcount.
DNS Domain Reputation	A grouping of Source, QName, Resolved Address, DNS Server, Threat Category trending Count. Information Elements: sourceipaddress, dnsname, dnsresolvedipv4address, dnsnxserveripv4address, reputationcategoryid, plixeraggregatedrecordcount.
DNS Exfiltration	A grouping of Source, Destination, QName, DNS Text trending Length, Count. Information Elements: sourceipaddress, destinationipaddress, dnsname, dnstext, dnstextlength, plixeraggregatedrecordcount.
DNS Query Refused	A grouping of Client, DNS Server, FQDN trending Lookup Time. Information Elements: dnsnxclientipv4address, dnsnxserveripv4address, dnsname, flowstartseconds.
DNS > RCodes	A grouping of Rcode trending Observation Count. Information Elements: dnsrcode, plixeraggregatedrecordcount.
DNS Request Latency	A grouping of Client, QName, Resolved to, Responding DNS Svr trending Latency. Information Elements: dnsnxclientipv4address, dnsname, dnsresolvedipv4address, dnsnxserveripv4address, dnsresolvetime.
DNS > Requests	A grouping of Request trending Observation Count. Information Elements: dns_rrname, plixeraggregatedrecordcount.
DNS Request Timeout	A grouping of Client, DNS Query Name trending Count. Information Elements: dnsnxclientipv4address, dnsname, plixeraggregatedrecordcount.

continues on next page

Table 8 – continued from previous page

Report	Description
DNS Server Failure	A grouping of Client, DNS Server, FQDN trending Lookup Time. Information Elements: dnsnxclientipv4address, dnsnxserveripv4address, dnsname, flowstartseconds.
DNS Server Latency	A grouping of Responding DNS Svr trending DNS Requests, Latency. Information Elements: dnsnxserveripv4address, dnsresolvetime, plixeraggregatedrecordcount.
DNS Server Responding Details	A grouping of DNS Server, Client, FQDN, Resolved Address trending Resolve Count. Information Elements: dnsnxserveripv4address, dnsnxclientipv4address, dnsname, dnsresolvedipv4address, plixeraggregatedrecordcount.
DNS Server Responding Summary	A grouping of DNS Server trending Number of Clients, Unique Lookup Count, Minimum Resolution Time. Information Elements: dnsnxserveripv4address, dnsnxclientipv4address, dnsresolvetime, plixeraggregatedrecordcount.
File Info > All File Details	A grouping of Source, Destination, File Name, MD5 Checksum, SHA256 Checksum trending Bytes. Information Elements: sourceipaddress, destinationipaddress, filename, md5_file_checksum, sha256_file_checksum, file_size_octets.
File Info > CheckSums	A grouping of MD5 Checksum, SHA256 Checksum trending File Size. Information Elements: md5_file_checksum, sha256_file_checksum, file_size_octets.
File Info > Filename & CheckSums	A grouping of File Name, MD5 Checksum, SHA256 Checksum trending File Size. Information Elements: filename, md5_file_checksum, sha256_file_checksum, file_size_octets.
HTTP > All Details	A grouping of Source, Destination, Request Host, Request Target, User Agent, Content Type, Request Method, Status Code trending Total Payload. Information Elements: sourceipaddress, destinationipaddress, httprequesthost, httprequesttarget, httpuseragent, httpcontenttype, httprequestmethod, httpstatuscode, ippayloadlength.
HTTP > Content Type	A grouping of Content Type, Request Method, Status Code trending Total Payload. Information Elements: httpcontenttype, httprequestmethod, httpstatuscode, ippayloadlength.
HTTP > Request Target	A grouping of Request Target trending Total Payload. Information Elements: httprequesttarget, ippayloadlength.
HTTP > User Agent	A grouping of User Agent trending Observation Count. Information Elements: httpuseragent, plixeraggregatedrecordcount.
NX-FQDN	A grouping of FQDN trending DNS Clients, Resolve Count. Information Elements: dnsnxqname, dnsnxclientipv4address, plixeraggregatedrecordcount.

continues on next page

Table 8 – continued from previous page

Report	Description
SMB > File Details	A grouping of Source, Destination, Command, Status, File Name, Operation, Permissions, Accessed, Modified, File Size trending Observed Count. Information Elements: sourceipaddress, destinationipaddress, smb_command, smb_status, smb_filename, smb_disposition, smb_access, smb_accessed_time, smb_modified_time, smb_file_size, plixeraggregatedrecordcount.
SMB > NTLMSSP Authentication Details	A grouping of Source, Destination, User, Host, Domain, Status, Version trending Observed Count. Information Elements: sourceipaddress, destinationipaddress, smb_ntlmssp_user, smb_ntlmssp_host, smb_ntlmssp_domain, smb_status, smb_ntlmssp_version, plixeraggregatedrecordcount.
SNMP > All Details	A grouping of Community, User, Vars, PDU Type trending Observation Count. Information Elements: mrtgsnmpcommunity, snmp_usm, snmp_var, snmp_pdu_type, plixeraggregatedrecordcount.
SNMP > Community	A grouping of Community trending Observation Count. Information Elements: mrtgsnmpcommunity, plixeraggregatedrecordcount.
SNMP > PDU Type	A grouping of PDU Type trending Observation Count. Information Elements: snmp_pdu_type, plixeraggregatedrecordcount.
SNMP > User	A grouping of User trending Observation Count. Information Elements: snmp_usm, plixeraggregatedrecordcount.
SNMP > Version	A grouping of Version trending Observation Count. Information Elements: mrtgsnmpversion, plixeraggregatedrecordcount.
Src and # of DNS servers	A grouping of Client, User Name(s) trending # of DNS servers. Information Elements: dnsnxclientipv4address, dnsclientname, dnsnxserveripv4address.
Src and # of NX 2LD	A grouping of Client, User Name(s), DNS Server trending NX Replies. Information Elements: dnsnxclientipv4address, dnsclientname, dnsnxserveripv4address, dnsqname2ld.
Src and # of NX 3LD	A grouping of Client, User Name(s), DNS Server trending NX Replies. Information Elements: dnsnxclientipv4address, dnsclientname, dnsnxserveripv4address, dnsqname3ld.
Src and # of NX Replies	A grouping of Client, User Name(s) trending NX Responses. Information Elements: dnsnxclientipv4address, dnsclientname, dnsnxqname.
Src with NX 2LD	A grouping of Client, User Name(s), 2nd Level Domain, DNS Server trending Count. Information Elements: dnsnxclientipv4address, dnsclientname, dnsqname2ld, dnsnxserveripv4address, plixeraggregatedrecordcount.

continues on next page

Table 8 – continued from previous page

Report	Description
Src with NX 3LD	A grouping of Client, User Name(s), 3rd Level Domain, DNS Server trending Count. Information Elements: dnsnxclientipv4address, dnsclientname, dnsqname3ld, dnsnxserveripv4address, plixeraggregatedrecordcount.
Src with NX FQDN	A grouping of Client, User Name(s), DNS Query Name, DNS Server trending Count. Information Elements: dnsnxclientipv4address, dnsclientname, dnsnxqname, dnsnxserveripv4address, plixeraggregatedrecordcount.
Top 2LD Requests	A grouping of 2nd Level Domains trending Clients Requesting, Resolve Count. Information Elements: request2ld, dnsnxclientipv4address, dnsresolvedipv4address.
Top 3LD Requests	A grouping of 3rd Level Domains trending Clients Requesting, Resolve Count. Information Elements: request3ld, dnsnxclientipv4address, dnsresolvedipv4address.

FQDN Reports

Report	Description
Destination FQDN	A grouping of Destination, FQDN trending Lookups. Information Elements: destinationipaddress, dst_fqdn, fqdn_lookup_count.
Host to Host with Dst FQDN	A grouping of Source, Destination, Dst FQDN trending Lookup. Information Elements: sourceipaddress, destinationipaddress, dst_fqdn, fqdn_lookup_count.

Gigamon

Report	Description
App Intel - DNS	A grouping of App, Src IP, Dst IP, Query, Response, Query Type trending Count. Information Elements: applicationid, sourceipaddress, destinationipaddress, dnsqueryname, gigamondnsresponseipv4address, gm_dns_networkservice_host_type, plixeraggregatedrecordcount.
App Intel - FTP	A grouping of App, Src IP, Dst IP, Filename, User, Pass, File Size trending Bytes. Information Elements: applicationid, sourceipaddress, destinationipaddress, gm_ftp_fileserver_filename, gm_ftp_fileserver_login, gm_ftp_fileserver_password, gm_ftp_fileserver_filesize, octetdeltacount.
App Intel - HTTP	A grouping of App, Src IP, Dst IP, User Agent, HTTP Method, Host, URI, Referrer, User Agent trending Bytes. Information Elements: applicationid, sourceipaddress, destinationipaddress, httpuseragent, gm_http_web_method, gm_http_web_host, gm_http_web_uri, gm_http_web_referer, httpstatuscode, octetdeltacount.
App Intel - SMB	A grouping of App, Src IP, Dst IP, File, SMB Version, NTLM User, NTLM Workstation trending Bytes. Information Elements: applicationid, sourceipaddress, destinationipaddress, gm_smb_fileserver_filename, gm_smb_fileserver_version, gm_smb_fileserver_ntlm_user, gm_smb_fileserver_ntlm_workstation, octetdeltacount.
App Intel - SMTP	A grouping of App, Src IP, Dst IP, Recipient, Sender, Subject, Attachment trending Bytes. Information Elements: applicationid, sourceipaddress, destinationipaddress, gm_smtp_mail_receiver, gm_smtp_mail_sender, gm_smtp_mail_subject, gm_smtp_mail_attach_filename, octetdeltacount.
Destination Name and URL	A grouping of Destination, User Name(s), URL trending Count. Information Elements: destinationipaddress, dstipname, gigamonhttprequrl, plixeraggregatedrecordcount.
DNS All Details	A grouping of Src IP, Dst IP, DNS Request, IP Returned, Authority Name trending Count. Information Elements: sourceipaddress, destinationipaddress, dnsqueryname, gigamondnsresponseipv4address, gigamondnsauthorityname, plixeraggregatedrecordcount.
Hosts with URL	A grouping of Src IP, Destination, URL trending Count. Information Elements: sourceipaddress, destinationipaddress, gigamonhttprequrl, plixeraggregatedrecordcount.
Pair Names and URL	A grouping of Source, Source Username, Destination, Destination Username, URL trending Count. Information Elements: sourceipaddress, srcipname, destinationipaddress, dstipname, gigamonhttprequrl, plixeraggregatedrecordcount.
Return Codes	A grouping of Return Code trending Count. Information Elements: gigamonhttprspstatus, plixeraggregatedrecordcount.
Source Name and URL	A grouping of Source, User Name(s), URL trending Count. Information Elements: sourceipaddress, srcipname, gigamonhttprequrl, plixeraggregatedrecordcount.

Honeynet

Report	Description
Adversary and State	A grouping of Adversary, State trending Count. Information Elements: sourceipaddress, connectionstate, plixeraggregatedrecordcount.
Adversary and String	A grouping of Adversary, String trending Count. Information Elements: sourceipaddress, comments, plixeraggregatedrecordcount.
Adversary, String and State	A grouping of Adversary, String, State trending Count. Information Elements: sourceipaddress, comments, connectionstate, plixeraggregatedrecordcount.
Forensic with Start	A grouping of Start Time, Source, String, State trending Count. Information Elements: flowstartmilliseconds, sourceipaddress, comments, connectionstate, plixeraggregatedrecordcount.
State	A grouping of State trending Count. Information Elements: connectionstate, plixeraggregatedrecordcount.
Strings	A grouping of String trending Count. Information Elements: comments, plixeraggregatedrecordcount.
Strings and State	A grouping of String, State trending Count. Information Elements: comments, connectionstate, plixeraggregatedrecordcount.

HTTP

Report	Description
Host to Host Request Volume	A grouping of Source, Destination trending Requests, Packets, Bytes. Information Elements: httprequesthost, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
HTTP User Agent	A grouping of Source, User Agent trending Flow Count, Bytes. Information Elements: httprequesthost, httpuseragent, octetdeltacount, plixeraggregatedrecordcount.
User Agent	A grouping of pm_cisco_httpuseragent trending Count, Packets, Bytes. Information Elements: pm_cisco_httpuseragent, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.

Juniper

Report	Description
Application Performance	A grouping of Application trending Uplink Pkts, Downlink Pkts, Retrans Uplink, Retrans Downlink, Smooth RTT Up, Smooth RTT Down. Information Elements: applicationname, downlinkpackets, retranstcppacketsdownlink, retranstcppacketsuplink, smoothrttdownlink, smoothrttuplink, uplinkpackets.
Host and Num Inst	A grouping of Host, Num Inst 1, Num Inst 2, Num Inst 3, Num Inst 4, Num Inst 5 trending Flows. Information Elements: host, numinstances_1, numinstances_2, numinstances_3, numinstances_4, numinstances_5, plixeraggregatedrecordcount.
Host and Status Code	A grouping of Host, Status Code 1, Status Code 2, Status Code 3, Status Code 4, Status Code 5 trending Flows. Information Elements: host, statuscode_1, statuscode_2, statuscode_3, statuscode_4, statuscode_5, plixeraggregatedrecordcount.
Host DNS Response Time	A grouping of Source trending Flows, Max DNS Resp., Avg DNS Resp.. Information Elements: host, dnsresponsetime, plixeraggregatedrecordcount.
HTTP Details	A grouping of Host, Method, Referrer, Response Code, URI trending Flows. Information Elements: host, http_method, http_referrer, http_responsecode, http_uri, plixeraggregatedrecordcount.
HTTP Method	A grouping of HTTP Method trending Uplink Pkts, Downlink Pkts, Uplink Octets, Downlink Octets, Flows. Information Elements: http_method, downlinkoctets, downlinkpackets, plixeraggregatedrecordcount, uplinkoctets, uplinkpackets.
HTTP Referrer	A grouping of HTTP Referrer trending Uplink Pkts, Downlink Pkts, Uplink Octets, Downlink Octets, Flows. Information Elements: http_referrer, downlinkoctets, downlinkpackets, plixeraggregatedrecordcount, uplinkoctets, uplinkpackets.
HTTP Response Code	A grouping of Response Code trending Uplink Pkts, Downlink Pkts, Uplink Octets, Downlink Octets, Flows. Information Elements: http_responsecode, downlinkoctets, downlinkpackets, plixeraggregatedrecordcount, uplinkoctets, uplinkpackets.
HTTP URI	A grouping of URI trending Uplink Pkts, Downlink Pkts, Uplink Octets, Downlink Octets, Flows. Information Elements: http_uri, downlinkoctets, downlinkpackets, plixeraggregatedrecordcount, uplinkoctets, uplinkpackets.
IFL and Subscriber Details	A grouping of IFL Name, IP Address, Name, Type, VRF trending Flows, UL Pkts, DL Pkts, UL Octets, DL Octets. Information Elements: iflname, subscriberipaddress, subscribername, subscribertype, subscribervrf, downlinkoctets, downlinkpackets, plixeraggregatedrecordcount, uplinkoctets, uplinkpackets.
IFL Name and Counters	A grouping of IFL Name trending Plixer Flows, Uplink Pkts, DL Pkts, Uplink Octets, DL Octets. Information Elements: iflname, downlinkoctets, downlinkpackets, plixeraggregatedrecordcount, uplinkoctets, uplinkpackets.
NAS Details	A grouping of IP Address, Port ID, Port Type trending UL Pkts, DL Pkts, UL Octets, DL Octets, Flows. Information Elements: nasipaddress, nasportid, nasporttype, downlinkoctets, downlinkpackets, plixeraggregate-

Keysight Reports

Report	Description
App with Latency	A grouping of Application trending RTT, Bytes. Information Elements: applicationid, latency, octetdeltacount.
Browsers	A grouping of Browser trending Packets, Bytes. Information Elements: browsername, octetdeltacount, packetdeltacount.
Connections with Latency	A grouping of Source IP, Source Port, Destination IP, Destination Port trending RTT. Information Elements: sourceipaddress, sourcetransportport, destinationipaddress, destinationtransportport, latency.
Conversation App Latency	A grouping of Source IP, Application, Destination IP trending RTT, Bytes. Information Elements: sourceipaddress, applicationid, destinationipaddress, latency, octetdeltacount.
Device and Location	A grouping of OS Name, Source, City, Country trending Packets, Bytes. Information Elements: osdevicename, sourceipaddress, sourcecityname, sourcecountryname, octetdeltacount, packetdeltacount.
Encryption	A grouping of Source, Destination, connencrypttype, encryptioncipher, encryptionkeylength trending Packets, octets. Information Elements: sourceipaddress, destinationipaddress, connencrypttype, encryptioncipher, encryptionkeylength, octetdeltacount, packetdeltacount.
L7 Application	A grouping of L7 Application trending Packets, Flows, Bytes. Information Elements: l7applicationname, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
OS	A grouping of OS Name trending Packets, Bytes. Information Elements: osdevicename, octetdeltacount, packetdeltacount.
OS Device Name	A grouping of OS Name, Source trending Packets, Bytes. Information Elements: osdevicename, sourceipaddress, octetdeltacount, packetdeltacount.
Source City	A grouping of City trending Packets, Bytes. Information Elements: sourcecityname, octetdeltacount, packetdeltacount.
Source Country	A grouping of Country trending Packets, Bytes. Information Elements: sourcecountryname, octetdeltacount, packetdeltacount.

Kubernetes Reports

Report	Description
K8S Destination Pod Traffic	A grouping of Pod Name trending Bytes, Packets. Information Elements: k8s_dst_pod_name, octetdeltacount, packetdeltacount.
K8S Services	A grouping of K8S Service trending Bytes, Packets. Information Elements: k8s_dst_service_name, octetdeltacount, packetdeltacount.
K8S Source Pod Traffic	A grouping of Pod Name trending Bytes, Packets. Information Elements: k8s_src_pod_name, octetdeltacount, packetdeltacount.
K8S Vitals	A grouping of Name, Type trending CPU, CPU (percent of node), Memory (percent of limit), Memory (percent of node). Information Elements: k8s_vitals_name, k8s_vitals_record_type, k8s_vitals_cpu_percent_of_node, k8s_vitals_cpu_usage, k8s_vitals_memory_percent_of_limit, k8s_vitals_memory_percent_of_node.

NAT

Report	Description
All Details	A grouping of Source, Src Port, NAT Src IP, NAT Src Port, NAT Dst Port, NAT Dst IP, Dst Port, Destination trending Flows, Bytes. Information Elements: sourceipaddress, sourcetransportport, postnatsourceipv4address, postnaptsourcetransportport, postnaptdestinationtransportport, postnatdestinationipv4address, destinationtransportport, destinationipaddress, octetdeltacount, plixeraggregatedrecordcount.
Destination Details	A grouping of Destination, Dst Port, NAT Dst IP, NAT Dst Port trending Flows, Bytes. Information Elements: destinationipaddress, destinationtransportport, postnatdestinationipv4address, postnaptdestinationtransportport, octetdeltacount, plixeraggregatedrecordcount.
Dst Translations	A grouping of Destination, Post Dst IP trending Packets, Bytes. Information Elements: destinationipaddress, postnatdestinationipv4address, octetdeltacount, packetdeltacount.
Post Connections	A grouping of in Int, Post Src Port, Post Src IP, Post Dst IP, post , out Int trending Packets, Bytes. Information Elements: ingressinterface, postnaptsourcetransportport, postnatsourceipv4address, postnatdestinationipv4address, postnaptdestinationtransportport, egressinterface, octetdeltacount, packetdeltacount.
Post Host to Host	A grouping of in Int, Post Src IP, Post Dst IP, out Int trending Packets, Bytes. Information Elements: ingressinterface, postnatsourceipv4address, postnatdestinationipv4address, egressinterface, octetdeltacount, packetdeltacount.
Source Details	A grouping of Source, Src Port, NAT Src Port, NAT Src IP trending Flows, Bytes. Information Elements: sourceipaddress, sourcetransportport, postnaptsourcetransportport, postnatsourceipv4address, octetdeltacount, plixeraggregatedrecordcount.
Src Translations	A grouping of Source, Post Src IP trending Packets, Bytes. Information Elements: sourceipaddress, postnatsourceipv4address, octetdeltacount, packetdeltacount.
Translations	A grouping of Source, Destination, Post Src IP, Post Dst IP trending Packets, Bytes. Information Elements: sourceipaddress, destinationipaddress, postnatsourceipv4address, postnatdestinationipv4address, octetdeltacount, packetdeltacount.

NBAR Reports

Report	Description
Application Categories	A grouping of Application Category trending Packets, Bytes. Information Elements: ciscoappcategoryname, octetdeltacount, packetdeltacount.
Application Compression	A grouping of Application trending % Pkt Comp, % Octet Comp. Information Elements: applicationtag, percentoctetcompression, percentpacketcompression.
Application Groups	A grouping of Application Group trending Packets, Bytes. Information Elements: ciscoappgroupname, octetdeltacount, packetdeltacount.
Applications	A grouping of Application trending Packets, Bytes. Information Elements: applicationtag, octetdeltacount, packetdeltacount.
Application Sub Categories	A grouping of Application Sub Category trending Packets, Bytes. Information Elements: ciscosubappcategoryname, octetdeltacount, packetdeltacount.
Conversations	A grouping of Source, Application, Destination trending Packets, Bytes. Information Elements: sourceipaddress, applicationtag, destinationipaddress, octetdeltacount, packetdeltacount.

Overlay Network

Report	Description
Destination Hosts by Network	A grouping of Network ID, Network Type, Destination trending Count, Packets, Bytes. Information Elements: overlay_net_id, overlay_net_type, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Network ID and Type	A grouping of Network ID, Network Type trending Count, Packets, Bytes. Information Elements: overlay_net_id, overlay_net_type, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Source Hosts by Network	A grouping of Network ID, Network Type, Source trending Count, Packets, Bytes. Information Elements: overlay_net_id, overlay_net_type, sourceipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.

Pair Reports

Report	Description
AS to AS by IP	A grouping of Source AS, Destination AS trending Packets, Bytes. Information Elements: srcipas, dstipas, octetdeltacount, packetdeltacount.

continues on next page

Table 9 – continued from previous page

Report	Description
AS to AS by Tag	A grouping of Src AS, Dst AS trending Packets, Bytes. Information Elements: bgpsourceasnumber, bgpdestinationasnumber, octetdeltacount, packetdeltacount.
AS to AS by Tag (Peer)	A grouping of bgpprevadjacentasnumber, bgpnextadjacentasnumber trending Packets, Bytes. Information Elements: bgpprevadjacentasnumber, bgpnextadjacentasnumber, octetdeltacount, packetdeltacount.
Avg Pkt Size	A grouping of Source, Destination trending Avg. Pkt. Size, Packets, NULL. Information Elements: sourceipaddress, destinationipaddress, avgpacketsize, octetdeltacount, packetdeltacount.
Client to Server	A grouping of Client, Server trending Packets, Bytes. Information Elements: clientipv4address, serveripv4address, octetdeltacount, packetdeltacount.
Connections By Bytes	A grouping of src Port, Source, Protocol, Destination, dst Port trending Packets, Bytes. Information Elements: sourcetransportport, sourceipaddress, protocolidentifier, destinationipaddress, destinationtransportport, octetdeltacount, packetdeltacount.
Connections By Flows	A grouping of src Port, Source, Protocol, Destination, dst Port trending Flows. Information Elements: sourcetransportport, sourceipaddress, protocolidentifier, destinationipaddress, destinationtransportport, plixeraggregatedrecordcount.
Connections w/ Obsrv Pt.	A grouping of Source, src Port, Destination, dst Port, Obsrv Pt trending Packets, Sum of Sq. Octets. Information Elements: sourceipaddress, sourcetransportport, destinationipaddress, destinationtransportport, observationpointid, octetdeltasumofsquares, packetdeltacount.
Conversations App	A grouping of Source, Application, Destination trending Packets, Bytes. Information Elements: sourceipaddress, applicationid, destinationipaddress, octetdeltacount, packetdeltacount.
Conversations w/Flags	A grouping of Source IP Address, Well Known Port, tcpcontrolbits, Destination IP Address trending Packets, Bytes, Flows. Information Elements: sourceipaddress, commonport, tcpcontrolbits, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Conversations WKP	A grouping of Source, Well Known, Destination trending Packets, Bytes. Information Elements: sourceipaddress, commonport, destinationipaddress, octetdeltacount, packetdeltacount.
Conv IP Groups	A grouping of Source IP Group, Well Known, Destination IP Group trending Packets, Bytes. Information Elements: srcipgroup, commonport, dstipgroup, octetdeltacount, packetdeltacount.
Country to Country	A grouping of Source Country, Destination Country trending Packets, Bytes. Information Elements: srccountry, dstcountry, octetdeltacount, packetdeltacount.

continues on next page

Table 9 – continued from previous page

Report	Description
Customer VLAN to VLAN	A grouping of postdot1qcustomervlanid, dot1qcustomervlanid trending Flows, Packets, Bytes. Information Elements: postdot1qcustomervlanid, dot1qcustomervlanid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
dot1q VLAN to VLAN	A grouping of postdot1qvlanid, dot1qvlanid trending Flows, Packets, Bytes. Information Elements: postdot1qvlanid, dot1qvlanid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Flow End Reason	A grouping of Source, Src Port, Dst Port, Destination, Flow End Reason trending Packets, Bytes. Information Elements: sourceipaddress, sourcetransportport, destinationtransportport, destinationipaddress, flowendreason, octetdeltacount, packetdeltacount.
Forensic Audit	A grouping of Flow Start, Source, Destination, Common Port, Protocol trending Pkts, Bytes. Information Elements: flowstartmilliseconds, sourceipaddress, destinationipaddress, commonport, protocolidentifier, octetdeltacount, packetdeltacount.
Grouped Flows (DSCP)	A grouping of src Port, Source, DSCP, Destination, dst Port trending Packets, Bytes. Information Elements: sourcetransportport, sourceipaddress, ipdiffservcodepoint, destinationipaddress, destinationtransportport, octetdeltacount, packetdeltacount.
Grouped Flows (TOS)	A grouping of src Port, Source, Type Of Service, Destination, dst Port trending Packets, Bytes. Information Elements: sourcetransportport, sourceipaddress, ipclassofservice, destinationipaddress, destinationtransportport, octetdeltacount, packetdeltacount.
Host - AS by IP - Host	A grouping of Source, Src AS, Dst AS, Destination trending Flows, Packets, Bytes. Information Elements: sourceipaddress, srcipas, dstipas, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Host - AS - Host	A grouping of Source, Src AS, Dst AS, Destination trending Flows, Packets, Bytes. Information Elements: sourceipaddress, bgpsourceasnumber, bgpdestinationasnumber, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Hosts with Country	A grouping of Source, Source Country, Destination, Destination Country trending Packets, Bytes. Information Elements: sourceipaddress, srccountry, destinationipaddress, dstcountry, octetdeltacount, packetdeltacount.
Host to Host	A grouping of Source, Destination trending Packets, Bytes. Information Elements: sourceipaddress, destinationipaddress, octetdeltacount, packetdeltacount.
Host to Host ICMP	A grouping of Source, Code, Type, Destination trending Count. Information Elements: sourceipaddress, icmpcodeipv4, icmptypeipv4, destinationipaddress, plixeraggregatedrecordcount.

continues on next page

Table 9 – continued from previous page

Report	Description
Host to Host L2	A grouping of Source, Destination trending Packets, L2 Octets. Information Elements: sourceipaddress, destinationipaddress, layer2octetdeltacount, packetdeltacount.
Host to Host Sum of Sq.	A grouping of Source, Destination trending Packets, Sum of Sq. Octets. Information Elements: sourceipaddress, destinationipaddress, octetdeltasumofsquares, packetdeltacount.
Host to Host w/Flags	A grouping of Source IP Address, tcpcontrolbits, Destination IP Address trending Packets, Bytes, Flows. Information Elements: sourceipaddress, tcpcontrolbits, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Host To Host With Next Hop	A grouping of Source, Destination, Next Hop trending packet, octect. Information Elements: sourceipaddress, destinationipaddress, ipnexthopipv4address, octetdeltacount, packetdeltacount.
IP Groups with Apps Defined	A grouping of Src Group, Protocol, Application, Dst Group trending Packets, Bytes. Information Elements: srcipgroup, protocolidentifier, applicationid, dstipgroup, octetdeltacount, packetdeltacount.
IP Group to IP Group	A grouping of Source IP Group, Destination IP Group trending Packets, Bytes. Information Elements: srcipgroup, dstipgroup, octetdeltacount, packetdeltacount.
MAC to MAC Routed	A grouping of Source MAC, Post Source MAC, Destination MAC, Post Destination MAC trending Flows, Packets, Bytes. Information Elements: sourcemacaddress, postsourcemacaddress, destinationmacaddress, postdestinationmacaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
MAC to MAC Switched	A grouping of Source MAC, Destination MAC trending Packets, Bytes. Information Elements: sourcemacaddress, destinationmacaddress, octetdeltacount, packetdeltacount.
Rev 2nd lvl Domain pairs	A grouping of Src Rev 2nd lvl Domain, Dst Rev 2nd lvl Domain trending Packets, Bytes. Information Elements: srcdomain, dstdomain, octetdeltacount, packetdeltacount.
Subnet to Subnet	A grouping of Src Subnet, Dst Subnet trending Packets, Bytes. Information Elements: srcnetwork, dstnetwork, octetdeltacount, packetdeltacount.
TOS to TOS	A grouping of Type of Service, Post Type of Services trending Packets, Bytes. Information Elements: ipclassofservice, postipclassofservice, octetdeltacount, packetdeltacount.
VLAN to VLAN	A grouping of postvlanid, vlanid trending Flows, Packets, Bytes. Information Elements: postvlanid, vlanid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.

Palo Alto Networks

Report	Description
Applications	A grouping of appid_pa trending Packets, Bytes. Information Elements: appid_pa, octetdeltacount, packetdeltacount.
CloudGenix Exporter Path Stats	A grouping of Exporter, Path ID trending Down Jitter, Up Jitter, Down Loss, Up Loss, Down MOS, Up MOS, RTT Latency. Information Elements: plixerexporter, cgnxlqmpathidentifier, cgnxlqmdownlinkjittermilliseconds, cgnxlqmdownlinkmos, cgnxlqmdownlinkpacketloss, cgnxlqmrttlatencymilliseconds, cgnxlqmuplinkjittermilliseconds, cgnxlqmuplinkmos, cgnxlqmuplinkpacketloss.
CloudGenix Path Stats	A grouping of Path ID trending Down Jitter, Up Jitter, Down Loss, Up Loss, Down MOS, Up MOS, RTT Latency. Information Elements: cgnxlqmpathidentifier, cgnxlqmdownlinkjittermilliseconds, cgnxlqmdownlinkmos, cgnxlqmdownlinkpacketloss, cgnxlqmrttlatencymilliseconds, cgnxlqmuplinkjittermilliseconds, cgnxlqmuplinkmos, cgnxlqmuplinkpacketloss.
Users	A grouping of userid_pa trending Packets, Bytes. Information Elements: userid_pa, octetdeltacount, packetdeltacount.

Procera Reports

Report	Description
APN and Base Service	A grouping of Access Point Name, Base Service trending Bytes. Information Elements: proceraapn, procerabaseservice, octetdeltacount.
Base Service RTT	A grouping of Base Service trending Internal RTT, External RTT. Information Elements: procerabaseservice, proceraexternalrtt, procerainternalrtt.
Content Categories	A grouping of Content Categories trending External RTT, Bytes. Information Elements: proceracontentcategories, octetdeltacount, proceraexternalrtt.
HTTP Content Type, Language, and Location	A grouping of Content Type, Language, Location trending Bytes. Information Elements: procerahttpcontenttype, procerahttplanguage, procerahttplocation, octetdeltacount.
HTTP Location, Referrer, and Request Method	A grouping of Location, referer, Request Method trending Bytes. Information Elements: procerahttplocation, procerahttpreferer, procerahttprequestmethod, octetdeltacount.
HTTP URL, Response Status, User Agent	A grouping of procerahttpurl, Response Status, User Agent trending Bytes. Information Elements: procerahttpurl, procerahttpresponsestatus, procerahttpuseragent, octetdeltacount.
Incoming Destination Details	A grouping of Destination IP Address trending Drops, Latency, Packets, Bytes. Information Elements: destinationipaddress, proceraincomingoctets, proceraincomingpackets, proceraincomingshapingdrops, proceraincomingshapinglatency.
Incoming Source Details	A grouping of Source IP Address trending Drops, Latency, Packets, Bytes. Information Elements: sourceipaddress, proceraincomingoctets, proceraincomingpackets, proceraincomingshapingdrops, proceraincomingshapinglatency.
Outgoing Destination Details	A grouping of Destination IP Address trending Drops, Latency, Packets, Bytes. Information Elements: destinationipaddress, proceraoutgoingoctets, proceraoutgoingpackets, proceraoutgoingshapingdrops, proceraoutgoingshapinglatency.
Outgoing Source Details	A grouping of Source IP Address trending Drops, Latency, Packets, Bytes. Information Elements: sourceipaddress, proceraoutgoingoctets, proceraoutgoingpackets, proceraoutgoingshapingdrops, proceraoutgoingshapinglatency.
Property and Service	A grouping of property, service trending In Ext. Qoe, In Int. Qoe, Out Ext. Qoe, Out Int. Qoe, Bytes. Information Elements: proceraproperty, proceraservice, octetdeltacount, proceraqoeincomingexternal, proceraqoeincominginternal, proceraqoeoutgoingexternal, proceraqoeoutgoinginternal.

Queue Drops

Report	Description
Queue Drops By Hierarchy	A grouping of Policy Map Hierarchy, Policy QoS Queue Index trending Flows, Q Drops. Information Elements: policymaphierarchy, policyqosqueueindex, plixeraggregatedrecordcount, plixer_qos_queue_drops.
Queue Drops By Index	A grouping of Policy QoS Queue Index trending Flows, Q Drops. Information Elements: policyqosqueueindex, plixeraggregatedrecordcount, plixer_qos_queue_drops.

Replicator

Report	Description
CPU	A grouping of Replicator trending Min, Avg, Max. Information Elements: sourceipaddress, plixercpuutilizationpercent.
Profile Statistics	A grouping of Profile trending Pkts In, Pkts Out, Bytes In, Bytes Out. Information Elements: observationdomainname, octetdeltacount, packetdeltacount, postoctetdeltacount, postpacketdeltacount.

Riverbed

Report	Description
Conversations RTT	A grouping of in Int, Source, Application, Destination, out Int trending RTT. Information Elements: ingressinterface, sourceipaddress, applicationid, destinationipaddress, egressinterface, tcpconnectionrtt_rvbd.
FE Type RTT	A grouping of FE Type trending Retrans Bytes, Retrans Pkts, RTT, Packets, Bytes. Information Elements: fetype_rvbd, octetdeltacount, packetdeltacount, tcpconnectionrtt_rvbd, tcppacketretransmissioncount_rvbd, tcpretransmissionbytecount_rvbd.
FE Type RTT and Source	A grouping of Source, FE Type trending Retrans Pkts, RTT, Packets, Bytes. Information Elements: sourceipaddress, fetype_rvbd, octetdeltacount, packetdeltacount, tcpconnectionrtt_rvbd, tcppacketretransmissioncount_rvbd.
FE Type RTT and Visibility	A grouping of FE Type, Visibility trending Retrans Pkts, RTT, Packets, Bytes. Information Elements: fetype_rvbd, visibility_rvbd, octetdeltacount, packetdeltacount, tcpconnectionrtt_rvbd, tcppacketretransmissioncount_rvbd.
Inner Connection IPs and RTT	A grouping of Source, Destination, IC CFE IP, IC SFE IP trending RTT. Information Elements: sourceipaddress, destinationipaddress, innerconnectioncfeipv4address_rvbd, innerconnectionsfeipv4address_rvbd, tcpconnectionrtt_rvbd.
Non Optimized Traffic	A grouping of Source, Destination, Common Port, Destination trending Packets, Bytes. Information Elements: sourceipaddress, destinationipaddress, commonport, passthroughreason_rvbd, octetdeltacount, packetdeltacount.
Pair RTT and Retrans	A grouping of Source, Destination trending Retrans Pkts, Retrans Bytes, RTT, Packets, Bytes. Information Elements: sourceipaddress, destinationipaddress, octetdeltacount, packetdeltacount, tcpconnectionrtt_rvbd, tcppacketretransmissioncount_rvbd, tcpretransmissionbytecount_rvbd.
Pair RTT with Ports	A grouping of Source, Src Port, Dst Port, Destination trending Retrans Pkts, RTT, Packets, Bytes. Information Elements: sourceipaddress, sourcetransportport, destinationtransportport, destinationipaddress, octetdeltacount, packetdeltacount, tcpconnectionrtt_rvbd, tcppacketretransmissioncount_rvbd.
Retransmissions	A grouping of in Int, Source, Destination, out Int trending Pckt Retrans, Bytes Retrans. Information Elements: ingressinterface, sourceipaddress, destinationipaddress, egressinterface, tcppacketretransmissioncount_rvbd, tcpretransmissionbytecount_rvbd.
Source RTT	A grouping of Source trending Retrans Pkts, Retrans Bytes, RTT, Packets, Bytes. Information Elements: sourceipaddress, octetdeltacount, packetdeltacount, tcpconnectionrtt_rvbd, tcppacketretransmissioncount_rvbd, tcpretransmissionbytecount_rvbd.
Wan Optimization	A grouping of in Int, Source, Src SFE IP, SFE Port, CFE Port, Dst CFE IP, Destination, out Int, Common Port trending Packets, Bytes. Information Elements: ingressinterface, sourceipaddress, innerconnectionsfeipv4address_rvbd, innerconnectionsfe-

Saisei

Report	Description
Dropped Pkts per Int	A grouping of Ingress Int, Distress, Egress Class trending Dropped Octets, Octets, Dropped Packets, Packets. Information Elements: ingressinterface, distress, egressflowclass, droppedoctettotalcount, droppedpackettotalcount, octetdeltacount, packetdeltacount.
Dropped Pkts per User	A grouping of User, Distress, Egress Class trending Dropped Octets, Octets, Dropped Packets, Packets. Information Elements: username, distress, egressflowclass, droppedoctettotalcount, droppedpackettotalcount, octetdeltacount, packetdeltacount.
Forensic Audit	A grouping of User, Application, Egress Class, Flow Start, Flow End trending RTT. Information Elements: username, applicationname, egressflowclass, flowstartmilliseconds, flowendmilliseconds, rttestimate.
Pair with Dropped Pkts	A grouping of Source IP, Destination IP, Distress, Egress Class trending Dropped Octets, Octets, Dropped Packets, Packets. Information Elements: sourceipaddress, destinationipaddress, distress, egressflowclass, droppedoctettotalcount, droppedpackettotalcount, octetdeltacount, packetdeltacount.
Pair with Retrans & RTT	A grouping of Source IP, Destination IP, Distress, Egress Class trending Retransmits, Retransmit Events, RTT. Information Elements: sourceipaddress, destinationipaddress, distress, egressflowclass, retransmissiondeltacount, retransmissioneventdeltacount, rttestimate.
Retransmits & RTT per Int	A grouping of Ingress Int, Distress, Egress Class trending Retransmits, Retransmit Events, RTT. Information Elements: ingressinterface, distress, egressflowclass, retransmissiondeltacount, retransmissioneventdeltacount, rttestimate.

SNMP

Report	Description
CPU	A grouping of Device trending CPU 1 Min, CPU 5 Min. Information Elements: plixercomponentipaddress, cputotal1min, cputotal5min.
Interface Details	A grouping of Exporter, ingressinterface trending Discards, Errors, Unicast Pkts, Non-Unicast Pkts, sum_snmpoctets. Information Elements: plixerexporter, ingressinterface, snmpdiscards, snmperrors, snmpnucastpkts, snmpoctets, snmpucastpkts.
Interface Stats (64bit)	A grouping of Exporter, ingressinterface trending Broadcast Pkts, Multicast Pkts, Unicast Pkts, sum_snmpoctets. Information Elements: plixerexporter, ingressinterface, snmpbroadcastpkts, snmpmulticastpkts, snmpoctets, snmpucastpkts.
Memory	A grouping of Device trending avg_memoryused, avg_memoryfree. Information Elements: plixercomponentipaddress, memoryfree, memoryused.

SonicWALL Reports

Report	Description
App Conv	A grouping of Source, SonicWALL Application, Destination trending Packets, Bytes. Information Elements: sourceipaddress, swapp, destinationipaddress, octetdeltacount, packetdeltacount.
Applications	A grouping of SonicWALL Application trending Packets, Bytes. Information Elements: swapp, octetdeltacount, packetdeltacount.
Available Memory	A grouping of Exporter trending Available Memory. Information Elements: plixerexporter, mem_avail_ram.
CPU Avg. Utilization	A grouping of Core ID trending AVG Util. Information Elements: core_stat_core_id, core_stat_core_util.
CPU Max. Utilization	A grouping of Core ID trending MAX Util. Information Elements: core_stat_core_id, core_stat_core_util.
Intrusions	A grouping of SonicWALL Intrusion trending Packets, Bytes. Information Elements: flow_to_ips_id, octetdeltacount, packetdeltacount.
Spyware	A grouping of SonicWALL Spyware trending Packets, Bytes. Information Elements: flow_to_spyware_id, octetdeltacount, packetdeltacount.
Urls	A grouping of SonicWALL URL trending Packets, Bytes. Information Elements: swurl, octetdeltacount, packetdeltacount.
User Details	A grouping of SonicWALL User, swuserip, swuserauthtype, swuserdomain trending Packets, Bytes. Information Elements: swuser, swuserip, swuserauthtype, swuserdomain, octetdeltacount, packetdeltacount.
Users	A grouping of SonicWALL User trending Packets, Bytes. Information Elements: swuser, octetdeltacount, packetdeltacount.
Virus	A grouping of SonicWALL Virus trending Packets, Bytes. Information Elements: flow_to_virus_id, octetdeltacount, packetdeltacount.
VoIP Conversations	A grouping of swinitcallid, swrespcallid trending Jitter, Pkt Loss, Packets, Bytes. Information Elements: swinitcallid, swrespcallid, octetdeltacount, packetdeltacount, swvoipavglatency, swvoiplostpkts.
VoIP Initiators	A grouping of swinitcallid trending Jitter, Pkt Loss, Packets, Bytes. Information Elements: swinitcallid, octetdeltacount, packetdeltacount, swvoipavglatency, swvoiplostpkts.
VoIP Responders	A grouping of swrespcallid trending Jitter, Pkt Loss, Packets, Bytes. Information Elements: swrespcallid, octetdeltacount, packetdeltacount, swvoipavglatency, swvoiplostpkts.
VPN Local Address	A grouping of VPN Local IP trending Packets, Bytes. Information Elements: swvpnlocalip, octetdeltacount, packetdeltacount.
VPN Name	A grouping of VPN Tunnel Name trending Packets, Bytes. Information Elements: swvpntunnelname, octetdeltacount, packetdeltacount.
VPN Name Local & Remote Address	A grouping of VPN Tunnel Name, VPN Local IP, VPN Remote IP trending Packets, Bytes. Information Elements: swvpntunnelname, swvpnlocalip, swvpnremoteip, octetdeltacount, packetdeltacount.
VPN Remote Address	A grouping of VPN Remote IP trending Packets, Bytes. Information Elements: swvpnremoteip, octetdeltacount,

Source Reports

Report	Description
Autonomous System by IP	A grouping of Source AS trending Packets, Bytes. Information Elements: srcipas, octetdeltacount, packetdeltacount.
Autonomous System by Tag	A grouping of Src AS trending Packets, Bytes. Information Elements: bgpsourceasnumber, octetdeltacount, packetdeltacount.
Autonomous System by Tag (Peer)	A grouping of bgpprevadjacentasnumber trending Packets, Bytes. Information Elements: bgpprevadjacentasnumber, octetdeltacount, packetdeltacount.
Countries	A grouping of Source Country trending Packets, Bytes. Information Elements: srccountry, octetdeltacount, packetdeltacount.
Countries with AS	A grouping of Source Country, Source AS, Hosts (Dst) trending Flows, Packets, Bytes. Information Elements: srccountry, srcipas, destinationipaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Customer VLAN	A grouping of dot1qcustomervlanid trending Flows, Packets, Bytes. Information Elements: dot1qcustomervlanid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
dot1q VLAN	A grouping of dot1qvlanid trending Flows, Packets, Bytes. Information Elements: dot1qvlanid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Host Flows	A grouping of Source trending Hosts (Destination), Packets, Flows. Information Elements: sourceipaddress, destinationipaddress, packetdeltacount, plixeraggregatedrecordcount.
Host Pkt Length	A grouping of Source trending Length MIN, Length MAX, Length AVG . Information Elements: sourceipaddress, iptotallength.
Hosts	A grouping of Source trending Packets, Bytes. Information Elements: sourceipaddress, octetdeltacount, packetdeltacount.
ICMP	A grouping of Source, Code, Type trending Count. Information Elements: sourceipaddress, icmpcodeipv4, icmptypeipv4, plixeraggregatedrecordcount.
L2 Octets	A grouping of Source trending Packets, L2 Octets. Information Elements: sourceipaddress, layer2octetdeltacount, packetdeltacount.
MAC	A grouping of Source MAC trending Flows, Packets, Bytes. Information Elements: sourcemacaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
MAC L2	A grouping of Source MAC trending Packets, L2 Octets. Information Elements: sourcemacaddress, layer2octetdeltacount, packetdeltacount.
MAC Sum of Sq	A grouping of Source MAC trending Packets, Sum of Sq. Octets. Information Elements: sourcemacaddress, octetdeltasumofsquares, packetdeltacount.
Post MAC	A grouping of Post Src Mac trending Count, Packets, Bytes. Information Elements: postsourcemacaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Rev 2nd lvl Domains	A grouping of Src Rev 2nd lvl Domain trending Packets, Bytes. Information Elements: srcdomain, octetdeltacount, packetdeltacount.

Stormshield

Report	Description
Top Url Categories	A grouping of Url Category trending Packets, Bytes. Information Elements: netasqurlcategory, octetdeltacount, packetdeltacount.
Top Urls	A grouping of Url trending Packets, Bytes. Information Elements: netasqurl, octetdeltacount, packetdeltacount.
Top Users	A grouping of User trending Packets, Bytes. Information Elements: username, octetdeltacount, packetdeltacount.

Top Reports

Report	Description
Applications Defined	A grouping of Application trending Packets, Bytes. Information Elements: applicationid, octetdeltacount, packetdeltacount.
Availability By IP	A grouping of Destination IP Address trending Availability. Information Elements: destinationipaddress, state.
Clients	A grouping of Client trending Packets, Bytes. Information Elements: clientipv4address, octetdeltacount, packetdeltacount.
DSCP	A grouping of DSCP trending Packets, Bytes. Information Elements: ipdiffservcodepoint, octetdeltacount, packetdeltacount.
Exporters	A grouping of Exporter trending Bytes. Information Elements: plixerexporter, octetdeltacount.
ICMP Type IPv4	A grouping of ICMP Type Code trending Count, Packets, NULL. Information Elements: icmptypecodeipv4, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
ICMP Type IPv6	A grouping of ICMP Type Code trending Count, Packets, NULL. Information Elements: icmptypecodeipv6, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
IGMP Type	A grouping of IGMP Type trending Count, Packets, NULL. Information Elements: igmptype, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Interface Compression	A grouping of Exporter, Outbound Interface trending % Pkt Comp, % Octet Comp. Information Elements: plixerexporter, egressinterface, percentoctetcompression, percentpacketcompression.
Interface-IP-MAC	A grouping of in Int, Source, Source MAC trending Flows, Packets, Bytes. Information Elements: ingressinterface, sourceipaddress, sourcemacaddress, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Interfaces	A grouping of Exporter, Inbound Interface, Interface Speed trending Bytes, % Util. Information Elements: plixerexporter, ingressinterface, inifspeed, interfacepercent, octetdeltacount.
Multicast Destinations	A grouping of Destination trending Pkts, Bytes. Information Elements: destinationipaddress, octetdeltacount, packetdeltacount.
Multicast Pairs	A grouping of Source, Destination trending Pkts, Bytes. Information Elements: sourceipaddress, destinationipaddress, octetdeltacount, packetdeltacount.
Next Hop	A grouping of Next Hop trending Packets, Bytes. Information Elements: ipnexthopipv4address, octetdeltacount, packetdeltacount.
Obsrv Pt. Layer 2	A grouping of Obsrv Pt trending Count, Packets, L2 Octets. Information Elements: observationpointid, layer2octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Obsrv Pt. Octet Sum of Squares	A grouping of Obsrv Pt trending Packets, Sum of Sq. Octets. Information Elements: observationpointid, octetdeltasumofsquares, packetdeltacount.
Protocols	A grouping of Protocol trending Packets, Bytes. Information Elements: protocolidentifier, octetdeltacount, packetdeltacount.
Round Trip Time By IP	A grouping of Destination IP Address trending RTT. In-

UEBA Reports

Report	Description
Azure User Logins	A grouping of User, Source, Success, Location trending Count. Information Elements: username, sourceipaddress, ipfixifyloginstatename, ipfixifylogsource, plixeraggregatedrecordcount.
LDAP User Logins	A grouping of User, Source, Admin trending Count. Information Elements: username, sourceipaddress, ipfixifylogintypename, plixeraggregatedrecordcount.
Office 365 User Logins	A grouping of User, Source, Application, Success, Location trending Count. Information Elements: username, sourceipaddress, applicationname, ipfixifyloginstatename, ipfixifylogsource, plixeraggregatedrecordcount.

VeloCloud Reports

Report	Description
Application Flow Path	A grouping of Application, destinationuuid, vcflowpath trending Flows. Information Elements: applicationtag, destinationuuid, vcflowpath, plixeraggregatedrecordcount.
Application Link Policy	A grouping of Application, destinationuuid, vclinkpolicy trending Flows. Information Elements: applicationtag, destinationuuid, vclinkpolicy, plixeraggregatedrecordcount.
Application Policies	A grouping of Application, vclinkpolicy, vcroutetype, Traffic Type trending Packets, Bytes. Information Elements: applicationtag, vclinkpolicy, vcroutetype, vctraffictype, octetdeltacount, packetdeltacount.
Application Priority	A grouping of Application, destinationuuid, vcpriority trending Flows. Information Elements: applicationtag, destinationuuid, vcpriority, plixeraggregatedrecordcount.
Application Route Type	A grouping of Application, destinationuuid, vcroutetype trending Flows. Information Elements: applicationtag, destinationuuid, vcroutetype, plixeraggregatedrecordcount.
Application Traffic Type	A grouping of Application, destinationuuid, Traffic Type trending Flows. Information Elements: applicationtag, destinationuuid, vctraffictype, plixeraggregatedrecordcount.
Conv Dst Edge	A grouping of Source, Application, Destination, destinationuuid trending Flows, Bytes. Information Elements: sourceipaddress, applicationtag, destinationipaddress, destinationuuid, octetdeltacount, plixeraggregatedrecordcount.
Dst Edge	A grouping of destinationuuid trending Flows, Bytes. Information Elements: destinationuuid, octetdeltacount, plixeraggregatedrecordcount.
Flow Path	A grouping of vcflowpath trending Flows, Bytes. Information Elements: vcflowpath, octetdeltacount, plixeraggregatedrecordcount.
Interface Jitter	A grouping of ingressinterface trending countdistinct_destinationipaddress, avg_avgjittertxms. Information Elements: ingressinterface, avgjittertxms, destinationipaddress.
Interface Latency	A grouping of ingressinterface trending Unique Dsts, Avg Latency. Information Elements: ingressinterface, avglatencytxms, destinationipaddress.
Interface Metrics	A grouping of ingressinterface trending Avg Latency, avg_avgjittertxms, avg_avglosstxpct. Information Elements: ingressinterface, avgjittertxms, avglatencytxms, avglosstxpct.
Interface Packet Loss	A grouping of ingressinterface trending countdistinct_destinationipaddress, avg_avglosstxpct. Information Elements: ingressinterface, avglosstxpct, destinationipaddress.
Link Utilization	A grouping of linkuuid trending Packets, Bytes. Information Elements: linkuuid, octetdeltacount, packetdeltacount.
Packet Loss Conv	A grouping of Source, Application, Destination, destinationuuid trending Retransmission, Lost Packets. Information Elements: sourceipaddress, applicationtag, destinationipaddress, destinationuuid, lostpacketsrxdeltacount,

Viptela Reports

Report	Description
Health	A grouping of Device Name, Device Model, System IP trending Memory Used, CPU System(%), Disk Used. Information Elements: vtla_host_name, vtla_device_model, vtla_system_ip, vtla_cpu_system, vtla_disk_used, vtla_mem_used.
Local Color Performance	A grouping of vEdge Host, Local Color trending Avg. Latency, Avg. Loss, Avg. Jitter. Information Elements: vtla_vdevice_host_name, vtla_local_color, vtla_mean_jitter, vtla_mean_latency, vtla_mean_loss.
Policies Added	A grouping of vEdge, Policies Added trending Record Count. Information Elements: vtla_host_name, vtla_policies_added, plixeraggregatedrecordcount.
Policies Removed	A grouping of vEdge, Policies Removed trending Record Count. Information Elements: vtla_host_name, vtla_policies_removed, plixeraggregatedrecordcount.
Remote Color Performance	A grouping of vEdge Host, Remote Color trending Avg. Latency, Avg. Loss, Avg. Jitter. Information Elements: vtla_vdevice_host_name, vtla_remote_color, vtla_mean_jitter, vtla_mean_latency, vtla_mean_loss.
SLA Events	A grouping of Event ID, vEdge, Policies Added, Policies Removed trending Event Count. Information Elements: vtla_id, vtla_host_name, vtla_policies_added, vtla_policies_removed, plixeraggregatedrecordcount.
Tunnel Applications	A grouping of vEdge Host, Local Color, Remote System, Remote Color, Application trending Packets, Bytes. Information Elements: vtla_host_name, vtla_local_color, vtla_remote_system_ip, vtla_remote_color, vtla_application, octetdeltacount, packetdeltacount.
Tunnel Performance	A grouping of vEdge Host, Local Color, Remote System, Remote Color trending Avg. Latency, Avg. Loss, Avg. Jitter. Information Elements: vtla_vdevice_host_name, vtla_local_color, vtla_remote_system_ip, vtla_remote_color, vtla_mean_jitter, vtla_mean_latency, vtla_mean_loss.
vEdge Host Performance	A grouping of vEdge Host trending Avg. Latency, Avg. Loss, Avg. Jitter. Information Elements: vtla_vdevice_host_name, vtla_mean_jitter, vtla_mean_latency, vtla_mean_loss.

Vitals

Report	Description
CPU	A grouping of Server trending CPU. Information Elements: plixercomponentipaddress, plixercpuutilizationpercent.
CPU per Process	A grouping of Process trending min, avg, max. Information Elements: processcommandline, processpercentcpu.

continues on next page

Table 10 – continued from previous page

Report	Description
Data Ages	A grouping of Source IP Address, timingtest trending Sent. Information Elements: sourceipaddress, timingtest, dataageseconds.
Database	A grouping of Server trending Connections By Bytes, Read Req, Write Req, Cache Free, Queries, Threads, Buffers Used. Information Elements: plixercomponentipaddress, plixerdbconnections, plixerdbkeybufferused, plixerdbkeyreadreq, plixerdbkeywritereq, plixerdbqcachefreemem, plixerdbquestions, plixerdbthreadsconnected.
Database	A grouping of Server trending txid, Connections By Bytes, Queries, Timed Checkpoints, Requested Checkpoints, Shared Buffers, Buffers Written. Information Elements: plixercomponentipaddress, buffers_allocd, buffers_written, checkpoints_requested, checkpoints_timed, plixerdbconnections, plixerdbquestions, postgresql_txid.
Dir Sizes	A grouping of Server, Directory trending Bytes. Information Elements: plixercomponentipaddress, plixerstoragedrive, plixerstorageusedbytes.
Disk Requests	A grouping of Server, Drive trending Backlog, Request Wait, Read Merges/Sec, Read Requests/Sec, Request Sectors/Sec, Write Octets/Sec, Write Requests/Sec. Information Elements: plixercomponentipaddress, hddlabel, plixerdiskaveragerequestbacklog, plixerdiskaveragerequestwait, plixerdiskreadrequestmergesps, plixerdiskreadrequestsps, plixerdiskrequestsectorsps, plixerdiskwriterequestmergesps, plixerdiskwriterequestsps.
Disk Utilization	A grouping of Server, Drive trending % Utilization, Read Wait, Write Wait, Read Octets/Sec, Write Octets/Sec. Information Elements: plixercomponentipaddress, hddlabel, plixerdiskaveragepercentutilization, plixerdiskaveragereadwait, plixerdiskaveragewritewait, plixerdiskreadoctetsps, plixerdiskwriteoctetsps.
Distributed Heartbeat	A grouping of Server, Plixer Server, Type, Status trending Time. Information Elements: plixercomponentipaddress, ipv4polled, plixerheartbeattype, plixerheartbeatstatus, plixereventdurationmilliseconds.
Distributed Synchronization	A grouping of Source, Destination, Caller, DB Table trending Avg Time, Records. Information Elements: syncsourceipv4addr, syncdestinationipv4addr, plixersubroutine, plixertablename, plixereventdurationmilliseconds, plixerrowcount.
Event Queue Statistics	A grouping of Collector, DB Table trending Data Age, Total Rows, Disk Used. Information Elements: plixercomponentipaddress, plixertablename, plixerdataageseconds, plixerrowcount, plixerstorageusedbytes.

continues on next page

Table 10 – continued from previous page

Report	Description
FA Counts	A grouping of Collector, algorithm trending Min, Avg, Max. Information Elements: plixercomponentipaddress, faalgorithmid, faviolationcount.
FA Times	A grouping of Collector, algorithm trending Min Dur., Avg Dur., Max Dur.. Information Elements: plixercomponentipaddress, faalgorithmid, plixereventdurationmilliseconds.
Flow Metrics/Collector	A grouping of Collector trending MFSN, Packets, Flows. Information Elements: plixercomponentipaddress, plixerflowcount, plixerflowpacketcount, plixermfsncount.
Flow Metrics/Exporter	A grouping of Collector, Exporter trending MFSN, Packets, Flows. Information Elements: plixercomponentipaddress, plixerexporterid, plixerflowcount, plixerflowpacketcount, plixermfsncount.
Flow Metrics/Port	A grouping of Collector, Port trending MFSN, Packets, Flows. Information Elements: plixercomponentipaddress, plixerlisteningport, plixerflowcount, plixerflowpacketcount, plixermfsncount.
Frozen XIDs Age	A grouping of plixercomponentipaddress trending max_postgresql_datfrozenxid_age. Information Elements: plixercomponentipaddress, postgresql_datfrozenxid_age.
Frozen XIDs Age by DB	A grouping of plixercomponentipaddress, postgresql_datname trending max_postgresql_datfrozenxid_age. Information Elements: plixercomponentipaddress, postgresql_datname, postgresql_datfrozenxid_age.
Memory	A grouping of Server trending Available. Information Elements: plixercomponentipaddress, plixermemavailablebytes.
Memory per process	A grouping of Process trending Shared, Resident, Virtual. Information Elements: processcommandline, processresidentmemorysize, processsharedmemorysize, processvirtualmemorysize.
ML Engine Heartbeat	A grouping of ML Engine, Plixer Server, Type, Status trending Response Time, Data Age. Information Elements: plixercomponentipaddress, plixerexporteripv6address, plixerheartbeattype, plixerheartbeatstatus, dataageseconds, plixereventdurationmilliseconds.
ML Engine Index Document Count	A grouping of ML Engine, Elasticsearch Index trending Avg Count. Information Elements: plixercomponentipaddress, plixermlelasticsearchindexname, plixermlelasticsearchindexcount.
ML Engine Kafka Lag	A grouping of ML Engine, Kafka Topic trending Avg Lag. Information Elements: plixercomponentipaddress, plixermlkafkatopicname, plixermlkafkalag.
ML Engine Model Count	A grouping of ML Engine trending Avg Model Count. Information Elements: plixercomponentipaddress, plixermlmodelfilecount.

continues on next page

Table 10 – continued from previous page

Report	Description
PG Lock Count	A grouping of Collector trending Locks. Information Elements: exporteripv4address, postgresql_locks.
Replicator Exporter Stats	A grouping of Replicator, Exporter, Receiving Port trending Packets Received, Bytes Received. Information Elements: plixercomponentipaddress, replicator_exporteraddress, replicator_exporterreceivingport, replicator_exporteroctetdeltacount, replicator_exporterpktdeltacount.
Replicator Exporter to Collectors	A grouping of Replicator, Exporter, Exporter Port, Collector, Collector Port trending Packets, Bytes. Information Elements: plixercomponentipaddress, replicator_exporteraddress, replicator_exporterreceivingport, replicator_collectoraddress, replicator_collectorport, replicator_collectoroctetdeltacount, replicator_collectorpktdeltacount.
Replicator Input by Listening Port	A grouping of Replicator, Listening Port trending Packets, Bytes. Information Elements: plixercomponentipaddress, replicator_exporterreceivingport, replicator_exporteroctetdeltacount, replicator_exporterpktdeltacount.
Replicator Input by Replicator	A grouping of Replicator trending Packets, Bytes. Information Elements: plixercomponentipaddress, replicator_exporteroctetdeltacount, replicator_exporterpktdeltacount.
Replicator Output by Collector	A grouping of Replicator, Collector, Collector Port trending Packets, Bytes. Information Elements: plixercomponentipaddress, replicator_collectoraddress, replicator_collectorport, replicator_collectoroctetdeltacount, replicator_collectorpktdeltacount.
Replicator Output by Listening Port	A grouping of Replicator, Listening Port trending Packets, Bytes. Information Elements: plixercomponentipaddress, replicator_exporterreceivingport, replicator_collectoroctetdeltacount, replicator_collectorpktdeltacount.
Replicator Output by Profile	A grouping of Replicator, Profile trending Packets, Bytes. Information Elements: plixercomponentipaddress, replicator_profilename, replicator_collectoroctetdeltacount, replicator_collectorpktdeltacount.
Replicator Output by Replicator	A grouping of Replicator trending Packets, Bytes. Information Elements: plixercomponentipaddress, replicator_collectoroctetdeltacount, replicator_collectorpktdeltacount.
Replicator Profiles to Collectors	A grouping of Replicator, Profile, Collector, Collector Port trending Packets, Bytes. Information Elements: plixercomponentipaddress, replicator_profilename, replicator_collectoraddress, replicator_collectorport, replicator_collectoroctetdeltacount, replicator_collectorpktdeltacount.

continues on next page

Table 10 – continued from previous page

Report	Description
Report Request Time	A grouping of Server, reportrequestid, Report Type trending duration. Information Elements: plixercomponentipaddress, reportrequestid, reporttype, plixereventdurationmilliseconds.
Report Type Data Time	A grouping of Report Type trending Count, Min Dur., Avg Dur., Max Dur.. Information Elements: reporttype, plixeraggregatedrecordcount, plixereventdurationmilliseconds.
Report Type Query Time	A grouping of Report Type trending Count, Min Dur., Avg Dur., Max Dur.. Information Elements: reporttype, plixeraggregatedrecordcount, plixereventdurationmilliseconds.
Rollup Counts	A grouping of Exporter, Message Info trending Max Rows. Information Elements: plixerexporterid, message_info, plixerrowcount.
Rollup Data Ages	A grouping of Exporter, Template trending Min, Avg, Max. Information Elements: plixerexporterid, plixertemplateid, plixerdataageseconds.
Spool Counts	A grouping of Collector, Directory trending Spool Mins. Information Elements: exporteripv4address, plixerstoragedrive, plixerspoolcount.
Storage	A grouping of Server, Drive/Mount trending Avail Bytes. Information Elements: plixercomponentipaddress, plixerstoragedrive, plixerstorageavailablebytes.
Stream Age	A grouping of Collector, Stream trending Min Age, Avg Age, Max Age. Information Elements: plixercomponentipaddress, plixertablename, plixerdataageseconds.
Stream Statistics	A grouping of Collector, Stream trending Data Age, Total Rows, Disk Used. Information Elements: plixercomponentipaddress, plixertablename, plixerdataageseconds, plixerrowcount, plixerstorageusedbytes.
Syslogs	A grouping of Agent trending Processed, Received. Information Elements: exporteripv4address, plixersyslogsprocessed, plixersyslogsreceived.
Task Runtime	A grouping of Server, Task trending Count, Min Dur., Avg Dur., Max Dur.. Information Elements: plixercomponentipaddress, plixertaskname, plixeraggregatedrecordcount, plixereventdurationmilliseconds.
Totals / Rollups Times	A grouping of Exporter, Template, Event, Interval trending Min Rows, Avg Rows, Max Rows, Min Dur., Avg Dur., Max Dur.. Information Elements: plixerexporterid, plixertemplateid, plixereventid, plixerdstintervallength, plixereventdurationmilliseconds, plixerrowcount.

VMware DFW

Report	Description
Destination IP, vNIC, FW Event	A grouping of Destination, UUID, vNIC, FW Event, Rule ID trending Flow Count, Packets, Bytes. Information Elements: destinationipaddress, vmuuid, vnicindex, firewallevent, ruleid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
Source IP, vNIC, FW Event	A grouping of Source, UUID, vNIC, FW Event, Rule ID trending Flow Count, Packets, Bytes. Information Elements: sourceipaddress, vmuuid, vnicindex, firewallevent, ruleid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.
UUID, vNIC, FW Event	A grouping of UUID, vNIC, FW Event, Rule ID trending Flow Count, Packets, Bytes. Information Elements: vmuuid, vnicindex, firewallevent, ruleid, octetdeltacount, packetdeltacount, plixeraggregatedrecordcount.

VMware VDS

Report	Description
Pairs with Tenants	A grouping of Source, Src Tenant, Dst Tenant, Destination, vxLan ID trending Packets, Bytes. Information Elements: sourceipaddress, tenantsourceipv4, tenantdestipv4, destinationipaddress, overlay_net_id, octetdeltacount, packetdeltacount.
Tenant Conversations	A grouping of Src Tenant, Src Tenant Port, Tenant Protocol, Dst Tenant Port, Dst Tenant, vxLan ID trending Packets, Bytes. Information Elements: tenantsourceipv4, tenantsourceport, tenantprotocol, tenantdestport, tenantdestipv4, overlay_net_id, octetdeltacount, packetdeltacount.
Top Destination	A grouping of Destination, Dst Tenant, Egress Attribute, vxLan ID trending Packets, Bytes. Information Elements: destinationipaddress, tenantdestipv4, egressinterfaceattr, overlay_net_id, octetdeltacount, packetdeltacount.
Top Interfaces	A grouping of Ingress Interface, vxLan ID trending Packets, Bytes. Information Elements: ingressinterfaceattr, overlay_net_id, octetdeltacount, packetdeltacount.
Top Source	A grouping of Source, Src Tenant, Ingress Attribute, vxLan ID trending Packets, Bytes. Information Elements: sourceipaddress, tenantsourceipv4, ingressinterfaceattr, overlay_net_id, octetdeltacount, packetdeltacount.

Volume Reports

Report	Description
Availability	A grouping of Time Stamp, Total trending . Information Elements: goodtime, total, .
Flow Volume	Flow rate. As a volume report, the table represents values per time bucket
Host Count (dst)	The number of distinct destination hosts. As a volume report, the table represents values per time bucket
Host Count (src)	The number of distinct source hosts. As a volume report, the table represents values per time bucket
Pair Volume	The number of distinct source/destination pairs. As a volume report, the table represents values per time bucket
Round Trip Time	A grouping of Time Stamp, Total trending . Information Elements: goodtime, total, .
Traffic Volume	Utilization in bits or bytes along with peak values and 95th percentile. As a volume report, the table represents values per time bucket

Wireless Reports

Report	Description
Applications by Wireless Host	A grouping of Host(s), Application trending Packets, octets. Information Elements: staipv4address, applicationtag, octetdeltacount, packetdeltacount.
Applications by Wireless Host with DSCP	A grouping of Host(s), Application, DSCP, Post DSCP trending Packets, octets. Information Elements: staipv4address, applicationtag, ipdiffservcodepoint, postipdiffservcodepoint, octetdeltacount, packetdeltacount.
Applications Downstream	A grouping of Application trending Avg. Pkt. Size, Packets, octets. Information Elements: applicationtag, avgpacketsize, octetdeltacount, packetdeltacount.
Applications Upstream	A grouping of Application trending Avg. Pkt. Size, Packets, octets. Information Elements: applicationtag, avgpacketsize, octetdeltacount, packetdeltacount.
Clients per AP	A grouping of AP trending Clients. Information Elements: wtpmacaddress, stamacaddress.
Clients per SSID	A grouping of WLAN SSID trending Clients. Information Elements: wlanssid, stamacaddress.
Hosts by SSID	A grouping of Host(s), WLAN SSID trending Packets, octets. Information Elements: staipv4address, wlanssid, octetdeltacount, packetdeltacount.
Hosts with MAC	A grouping of Host(s), STA Mac Addr, AP Mac Addr trending Packets, octets. Information Elements: staipv4address, stamacaddress, wtpmacaddress, octetdeltacount, packetdeltacount.
Hosts with User Name	A grouping of Source, User Name(s) trending Packets, Bytes. Information Elements: staipv4address, staipname, octetdeltacount, packetdeltacount.
Host to Host with AP Mac	A grouping of Source, Destination, AP Mac Addr trending Packets, octets. Information Elements: sourceipaddress, destinationipaddress, wtpmacaddress, octetdeltacount, packetdeltacount.
Host to Host with SSID	A grouping of Source, Destination, WLAN SSID trending Packets, octets. Information Elements: sourceipaddress, destinationipaddress, wlanssid, octetdeltacount, packetdeltacount.
SSID List	A grouping of WLAN SSID trending Packets, octets. Information Elements: wlanssid, octetdeltacount, packetdeltacount.
Usage by SSID and AP	A grouping of AP MAC, WLAN SSID trending Packets, octets, Clients. Information Elements: wtpmacaddress, wlanssid, octetdeltacount, packetdeltacount, staipv4address.
Usage by SSID and AP (Src IP)	A grouping of AP MAC, WLAN SSID trending Packets, octets, Clients. Information Elements: wtpmacaddress, wlanssid, octetdeltacount, packetdeltacount, sourceipaddress.
User and Controller Details	A grouping of AP MAC, Source, User Name(s), WLAN SSID trending Packets, octets. Information Elements: wtpmacaddress, staipv4address, staipname, wlanssid, octetdeltacount, packetdeltacount.

Ziften

Report	Description
App Details	A grouping of Application, Version, App Description, Internal Name, File Name, CMD, MD5 trending Flows, Bytes. Information Elements: zflowverproductname, zflowverproductversion, zflowverfiledescription, zflowverinternalname, zflowveroriginalfilename, zflowcommandline, zflowmd5, octetdeltacount, plixeraggregatedrecordcount.
Base File and User	A grouping of User Name, Base File, OS trending Flows, Bytes. Information Elements: username, zflowparentimagebasefilename, zflowosname, octetdeltacount, plixeraggregatedrecordcount.
Command Line by Src	A grouping of Source, Command Line, PID trending Flows, Bytes. Information Elements: sourceipaddress, zflowcommandline, zflowpid, octetdeltacount, plixeraggregatedrecordcount.
Machine Details	A grouping of Machine, User Name, MD5, OS Name, OS Version, Agent UUID trending Flows, Bytes. Information Elements: zflowmachinename, username, zflowmd5, zflowosname, zflowosversion, zflowagentguid, octetdeltacount, plixeraggregatedrecordcount.
Machines	A grouping of Machine trending Flows, Bytes. Information Elements: zflowmachinename, octetdeltacount, plixeraggregatedrecordcount.
MD5	A grouping of Parent MD5, Parent Product Name, MD5, zflowverproductname trending Flows. Information Elements: zflowparentmd5, zflowparentverproductname, zflowmd5, zflowverproductname, plixeraggregatedrecordcount.

Zscaler ZIA

Report	Description
Data Center	A grouping of zsc_data_center, zsc_dns_app_cat trending sum_plixeraggregatedrecordcount. Information Elements: zsc_data_center, zsc_dns_app_cat, plixeraggregatedrecordcount.
Pairs By Application	A grouping of Client Tunnel IP, Client Hostname, Source Address, Application, Destination Address trending Ingress Octet Count, Egress Octet Count. Information Elements: zsc_client_tun_ip, zsc_cc_device_hostname, sourceipaddress, applicationid, destinationipaddress, egress_octetdeltacount, ingress_octetdeltacount.
Rules	A grouping of zsc_rule_name, zsc_rule_action trending Ingress, Egress. Information Elements: zsc_rule_name, zsc_rule_action, egress_octetdeltacount, ingress_octetdeltacount.
Server Categories	A grouping of zsc_server_ip_category, zsc_rule_name trending sum_plixeraggregatedrecordcount. Information Elements: zsc_server_ip_category, zsc_rule_name, plixeraggregatedrecordcount.
Threats	A grouping of zsc_cc_device_hostname, zsc_threat_name, zsc_threat_score, zsc_threat_severity trending Ingress Octet Count, Egress Octet Count. Information Elements: zsc_cc_device_hostname, zsc_threat_name, zsc_threat_score, zsc_threat_severity, egress_octetdeltacount, ingress_octetdeltacount.
Traffic by Tunnel IP	A grouping of zsc_client_tun_ip trending sum_egress_octetdeltacount, sum_ingress_octetdeltacount. Information Elements: zsc_client_tun_ip, egress_octetdeltacount, ingress_octetdeltacount.

Zscaler ZPA

Report	Description
App Connector: Interfaces Received	A grouping of App Connector, Interface Default Route, Num Interfaces trending Bytes Received, Packets Received, Discards Received, Errors Received, Total Bytes Received. Information Elements: zsc_app_connector_name, interfacename, zsc_app_connector_num_interfaces, zsc_bytes_received_interface, zsc_discards_received_interface, zsc_errors_received_interface, zsc_packets_received_interface, zsc_total_bytes_received.
App Connector: Interfaces Transmitted	A grouping of App Connector, Interface Default Route, Num Interfaces trending Bytes Transmitted, Packets Transmitted, Discards Transmitted, Errors Transmitted, Total Bytes Transmitted. Information Elements: zsc_app_connector_name, interfacename, zsc_app_connector_num_interfaces, zsc_bytes_transmitted_interface, zsc_discards_transmitted_interface, zsc_errors_transmitted_interface, zsc_packets_transmitted_interface, zsc_total_bytes_transmitted.
App Connector Status	A grouping of Name, Start Time, Private IP, ZEN, Group, Customer, Country, Session Status, Default Route GW, Primary DNS, CPU, Memory, Services Monitored trending Count. Information Elements: zsc_app_connector_name, zsc_app_connector_start_time, zsc_app_connector_private_ip, zsc_app_connector_zen, zsc_app_connector_group, zsc_app_connector_customer, zsc_app_connector_country, zsc_app_connector_session_status, zsc_app_connector_def_route_gw, zsc_app_connector_primary_dns_resolver, processpercentcpu, processpercentmemory, zsc_app_connector_service_count, plixeraggregatedrecordcount.
Browser Access	A grouping of Username, Client IP, Client port, Host, URL, Application port, Protocol, Status code, User Agent trending Count. Information Elements: username, zsc_client_ip, destinationtransportport, zsc_cc_device_hostname, urlpath, zsc_application_port, requestprotocol, httpstatuscode, useragent, plixeraggregatedrecordcount.
Private Cloud Controller: Interfaces Received	A grouping of Private Cloud Controller, Interface Default Route, Num Interfaces trending Bytes Received, Packets Received, Discards Received, Errors Received, Total Bytes Received. Information Elements: zsc_private_cc_name, interfacename, zsc_app_connector_num_interfaces, zsc_bytes_received_interface, zsc_discards_received_interface, zsc_errors_received_interface, zsc_packets_received_interface, zsc_total_bytes_received.

4.6.3.2 FAQ

Note

For additional questions or concerns, contact *Plixer Technical Support*.

To try out Scrutinizer or any other Plixer product, contact *Plixer Technical Support* and ask about an evaluation license.

To add a new license or view the details of the currently applied Scrutinizer license, navigate to the **Admin** > **Plixer** > **Scrutinizer Licensing** page.

Evaluation keys will cease to function after their expiry date. Scrutinizer subscription keys include a 60-day grace period where data collection continues on, but access to the data is unavailable until a new key is added. Legacy perpetual licenses will never expire, but deployments they are applied to cannot be upgraded.

The Scrutinizer web interface supports localization to other languages via the **Admin** > **System/New User Defaults** > **Language** page.

An unexpected inconsistency error during the ESXi virtual appliance startup indicates that the server clock is not correctly set, resulting in the disk checks failing. To resolve this issue, set your ESXi host to sync with an NTP server and then redeploy the Scrutinizer OVF.

Use the *services* command at the *scrut_util* (`SCRUTINIZER>`) prompt to stop/start/restart all services or the *systemctl* command to manage individual services.

If you have not changed the default password for the `plixer` user and are unable to log in with the default password `plixer`, check if caps lock is turned on.

When an account is locked due to multiple failed logins, there are two methods that an `Admin` user can use unlock it:

- In the web interface, go to **Admin** > **Users & Groups** > **User Accounts**. Select the locked username/account, click the **Authentication Method** tab in the *Edit User* tray, and then change the authentication method from 'locked' to the appropriate method.
- Use the *unlock* command at the *scrut_util* (`SCRUTINIZER>`) prompt.

In addition to overwriting the Scrutinizer instance the backup was restored to, the restore script is also set to delete the backup file used to perform the operation. As such, it is best to always create a copy of the backup file before initiating a restore.

Not all FA algorithms support (or benefit from) Aggregated Alarms. The *Aggregated Alarm Timeout* setting is only available for algorithms with continuous alarm events that can be combined.

If this happens when deploying virtual appliance to a distributed cluster, contact *Plixer Technical Support*. to obtain a new license key.

Historical data can be trimmed to free up disk space. You can do either of the following:

- In the web interface, go to **Admin** > **Settings** > **Data History**, and then adjust the current retention settings.
- Use the *expire history* command at the *scrut_util* (`SCRUTINIZER>`) prompt.

These packages are automatically installed and enabled as part of the initial configuration for new Scrutinizer deployments. If they were previously disabled, they can be re-enabled by running the following commands:

VMware Tools:

```
sudo systemctl --quiet enable vmtoolsd.service
sudo systemctl --quiet start vmtoolsd.service
```

Hyper-V daemons:

```
sudo systemctl --quiet enable hypervfcopyd.service␣
↪hypervkvpd.service hypervvssd.service
sudo systemctl --quiet start hypervfcopyd.service hypervkvpd.service␣
↪hypervvssd.service
```

4.6.3.3 Glossary

This glossary is meant to serve as a reference for general networking concepts as well as terms specific to Scrutinizer, and the Plixer ML Engine.

Scrutinizer

View content

Alarm policy
: Rule sets that define what types of network behavior or activity should be monitored as events and trigger alarms

Collectors
: SIEMs, flow collectors, SNMP trap receivers, and other network management systems that capture, analyze, and report on flow data sent by exporters

Exporters
: Network devices, such as routers, switches, or servers that can send traffic/activity logs as flows to external systems, such as Replicator and Scrutinizer

Flow analytics
: A library of field-tested algorithms used to analyze network behavior, detect unexpected activity, and report events and alarms

IPFIXify
: A software agent that reads text-based logs, syslog messages, Windows EventLogs and various other types of data sources and sends the information in flows using the IPFIX protocol

Plixer ML Engine
: A software component providing AI capabilities to allow the ingestion and processing of extremely large volumes of flow data for intelligent anomaly and threat detection

Protocol exclusions
: Defines protocols to exclude during the collection process per exporter, exporter interface, and/or all exporters and interfaces

Reverse-path filtering
: Allows collectors to receive non-local traffic that may have been forwarded by a proxy or flow replication solution, such as Replicator

SAF (Summary and Forensic)
: An optimized flow data aggregation method that uses summary tables to condense collected information without compromising transparency or accuracy

TI (Threat Index)
: A single value comprised of events with different weights that age out over time

Plixer ML Engine

View content

Deep learning
A progression of supervised and unsupervised learning to create an artificial neural network that can learn and make intelligent decisions on its own

K-means clustering
An algorithm that groups behaviors into common clusters

Link prediction
A method that detects anomalies and analyzes a device's interactions with other devices rather than just a particular behavior

Supervised learning
The process of training a machine learning algorithm using labeled data sets

Unsupervised learning
The process of training a machine learning algorithm to identify patterns or classifications in untagged data sets

General networking

View content

2LD (Second-level Domain)
Part of the naming convention for domain names. For example, in example.com, *example* is the second-level domain of the .com TLD (top-level domain)

3LD (Third-level Domain)
For example, in www.mydomain.com, *www* is the third-level domain

ACK (Acknowledgment Code)
A unique signal sent by a computer to show that it has successfully transmitted data

ACL (Access Control List)
A set of rules governing access to a particular object or system resource

Active Directory / AD
A proprietary directory service offered by Microsoft, which allows for centralized management of users, devices, and other IT assets

API (Application Programming Interface)
A software component that allows applications to share data and functionality

ARP (Address Resolution Protocol)
A protocol that maps a dynamic IP address to a physical machine's permanent MAC address in a local area network (LAN)

CA (Certification Authority)
A trusted entity that issues, signs, and stores digital certificates

CDP (Cisco Discovery Protocol)
A protocol used by Cisco devices to allow neighboring networking devices to learn about each other

CIDR (Classless Inter-Domain Routing)
An IP addressing method that improves the efficiency of allocating IP addresses

CLI (Command-line Interface)
A text-based interface for applications and operating systems that allows a user to enter commands

Collector
SIEMs, flow collectors, SNMP trap receivers, or other network management systems that analyze data forwarded from networked devices

DHCP (Dynamic Host Configuration Protocol)
A network management protocol used to automatically assign IP addresses and other communication parameters to devices on an Internet protocol network

DNS (Domain Name System)
A system by which computers and other devices on the Internet or Internet protocol networks are uniquely identified using names matched to their IP addresses

Egress
Traffic that exits a device or network

Endpoint
An entity (device, service, node, etc.) at the end of a network communication channel

Encapsulated Remote SPAN (ERSPAN)
Encapsulates mirrored traffic in GRE (Generic Routing Encapsulation) and sends it over Layer 3 networks

ESX (Elastic Sky X)
A pre-configured, ready-to-deploy virtual machine (VM) designed to run on VMware ESX or ESXi

EULA
End-user license agreement

Exporter
A networked device such as a router, switch, or server that generates data and sends it to the flow collector device

Fault tolerance
A system's ability to continue operating without interruptions in the event of hardware or software failure

FQDN (Fully Qualified Domain Name)
The complete address of a computer, host, or any other entity on the Internet

GRE (Generic Routing Encapsulation)
A tunneling protocol developed by Cisco Systems

Hyper-V
A pre-configured, ready-to-deploy virtual machine designed to run on Microsoft Hyper-V, typically packaged in VHD/VHDX format

ICMP (Internet Control Message Protocol)
A protocol used for devices within the network to determine possible network issues

Identity Provider (IdP)
A third-party entity and/or service that stores and manages identities and credentials for use by other websites, applications, or other digital resources

IP address
A unique numerical label assigned to a networked device

IPFIX (Internet Protocol Flow Information Export)
A protocol intended to collect and analyze the flow data from supported network devices

KVM (Kernel-based Virtual Machine)
A pre-configured virtual machine designed to run on KVM hypervisors, packaged in formats like QCOW2 or OVA for easy deployment in Linux-based virtualization environments

Latency
The latency of a network is the time it takes for a data packet to be transferred from its source to the destination

LDAP (Lightweight Directory Access Protocol)
An open, cross-platform protocol used to access and maintain directory services for assets in an Internet protocol network

LLDP (Link Layer Discovery Protocol)
A vendor-neutral protocol used by devices on IEEE 802 networks to advertise their identity, capabilities, and other information

MAC (Media Access Control) address
A unique hardware identifier typically assigned by manufacturers to network adapters and devices

MIB (Management Information Base)
A database that stores information used for managing a network

MTTR (Mean Time to Resolution)
The average amount of time between the detection and remediation of a security threat or incident

NDR (Network Detection and Response)
A cybersecurity solution that uses machine learning to detect cyber threats and aid remediation

Network interface
A (physical or software-based) point of connection between a network entity and the rest of the network

NIC (Network Interface Card)
An adapter that provides devices network connections, either wired or wireless

NID (Network Infrastructure Device)
Any device, such as an access point, router, or switch, that provide the means for entities to communicate with each other over a network

NTP (Network Time Protocol)
A networking protocol used to synchronize device clocks over the Internet

NXDOMAIN (No Existing Domain)
An error message that means that a domain mentioned in the Domain Name System (DNS) query does not exist

Open port
A TCP or UDP port that has been configured to accept packets

OUI (Organizationally Unique Identifier)
A unique 24-bit number in a MAC address that identifies the vendor or the manufacturer of the device

OVF (Open Virtualization Format)
An open source standard for packaging and distributing virtual machines and software applications

Packet
A block of data transmitted across a network

PDU (Protocol Data Unit)
An individual unit of information exchanged by entities on a network using the same protocol

PostgreSQL
An open-source relational database management system (RDBMS) that supports both SQL and JSON querying

PXE (Preboot Execution Environment)
A network booting protocol that allows computers to boot from a network rather than a local storage device like a hard drive or USB

RADIUS (Remote Authentication Dial-In User Service)
A client-server AAA (authentication, authorization, accounting) protocol used to manage remote user access to a network

Redundancy
The state of having duplicate or alternative services as backups to allow for continuous availability

REST API (Representational State Transfer Application Programming Interface)
A set of rules that allows systems to communicate over the web using standard HTTP methods

Router
A device that forwards or routes data packets to devices on a network

Server
A system or device that provides resources, data, services, or applications to other devices over a network

Single Sign-On (SSO)
A technology that allows the integration of third-party authentication services for user access to the Scrutinizer web interface

SIP/RTP (Session Initiation Protocol/Real Time Protocol)
SIP is the control protocol, and RTP is the payload protocol used to send and receive Voice over IP (VoIP)

SNMP (Simple Network Management Protocol)
An IP network protocol used to collect data related to state and/or behavior from devices on a network

SNMP trap
An alert message that is initiated by an SNMP-enabled device to notify the management system of significant events or changes in status

Software agent
A persistent piece of software that performs certain actions and/or interacts with its environment on behalf of a user or another program

SPAN (Switched Port Analyzer)
A dedicated port on a switch that takes a mirrored copy of network traffic from within the switch to be sent to a destination

SSDP (Simple Service Discovery Protocol)
A network protocol used for advertising and discovering network services

SSH (Secure Shell Protocol)
A network communication protocol that allows network services to be used securely over an unsecured network

SSL (Secure Sockets Layer)
A protocol for establishing secure connections between networked devices

STIX (Structured Threat Information eXchange)
An industry-standard file format for the exchange of threat information between organizations and platforms

Suricata
A network threat detection engine used to analyze network traffic and identify potential security threats

Switch
A device that connects devices in a network and allows them to communicate with each other

SYN scan
A port scanning technique that allows for the discovery of the status of a communications port without establishing a full connection

Syslog
A cross-platform network logging protocol used to send and/or receive alerts between different devices on a network

TACACS+ (Terminal Access Controller Access-Control System)
A protocol where the remote access server and the authentication server provide validation for users attempting to access the network

TAXII (Trusted Automated eXchange of Indicator Information)
A protocol that allows the transmission of threat information, primarily in STIX format, between systems and organizations

TCP (Transmission Control Protocol)
A connection-oriented protocol that enables the bidirectional exchange of messages between devices on the same network

TLS handshake
The process that starts secure communication between a client and a server

TSIG (Transaction Signature)
A protocol that secures DNS packets and allows a Domain Name System to authenticate updates to the DNS database

TTL (Time To Live)
A field in the IP packet header that specifies the maximum number of hops (or router passes) a packet can take before being discarded

UDP (User Datagram Protocol)
A communication protocol for transmitting messages between applications and programs in a network

Virtual appliance
A pre-configured virtual machine image with pre-installed software that is meant to serve a specific function

VoIP (Voice over Internet Protocol)
A technology that allows voice calls using an internet connection

VPC (Virtual Private Cloud)
A secure and private cloud hosted in a public cloud

VRF (Virtual Routing and Forwarding)
A technology that separates routing tables to isolate management traffic to the management interface

Web server banner
A text-based greeting message, which includes information like open ports, services, and version numbers, returned by a web host

4.6.3.4 Plixer ML Engine changelogs

Changelog entries are displayed in the format **DESCRIPTION (Ticket Number)**.

> **Note**
>
> - For more information on the Plixer ML Engine and the Plixer One platform, visit www.plixer.com or contact *Plixer Technical Support*.
> - Please refer to our End of Life Policy for EOL schedule details.

Plixer ML Engine v19.7.0 - November 2025

Changelog

New features

- 19.5.0 ML engine HyperV image requires a gen1 VM but docs state v2
- Add SNMPD to ML Repo
- Support for IPv6 subnets in ML Engine rules

Enhancements

- Configure which IP Groups Deep Graph Learning is monitoring

Fixes

- Addressed various security issues
- engine.lock file persists where the engine pod restarts before all the models are moved (1025)
- Upgrade script does not remove hashicorp.list from /etc/apt/sources.d/ (1003)
- Vitals model count report is reporting fewer models than are built (1026)

Plixer ML Engine v19.5.0 - March 2025

Changelog

New features

- Detect brute force activity using failed SMB logon attempts
- Detect remote ransomware attacks using SMB read/write data
- Support ability to modify AI Engine deployment settings from the user interface
- Support custom thresholds for data accumulation detections
- Support KVM
- Support offline vSphere cluster deployments

Enhancements

- Optimize processing to support larger workloads
- Prompt user for deployment information rather than use configuration files
- Support ability to modify AI engine application settings from the user interface
- Support ability to modify AI Engine seasonality settings from the user interface
- Support child groups for IP Group exclusions
- Support extending expiration date of ssl certificate used by Kubernetes cluster
- Verify application pods are updated during an online system update

Fixes

- Addressed various security issues
- Define violator and target correctly for brute force events (889)
- Update rogue DHCP detection logic (954)

Plixer ML Engine v19.4.0 - August 2024

Changelog

New features

- User behavior analytics for O365/Azure AD
- Encrypted Traffic Analytics (ETA)

- New base operating system: Ubuntu
- Added LDAP rogue service detection
- Added support for Hyper-V deployments
- Added new alert type in Scrutinizer for Suricata TLS alerts
- Added support for IP groups in exclusions

Fixes

- Addressed various security issues
- Behavior tab missing after upgrading to 19.3 (745)
- Time zone set as UTC causing training data to not be associated correctly with workdays/weeknights/weekends (750)
- An ML engine can now pair with a multi-collector distributed Scrutinizer (757)
- Certain utilities not functioning properly with 19.3 (786)
- Improved ML event messages by including additional details (810)

4.6.3.5 Replicator changelogs

Changelog entries are displayed in the format **DESCRIPTION (Ticket Number)**.

> **Note**
>
> - For more information on Replicator and the Plixer One platform, visit www.plixer.com or contact *Plixer Technical Support*.
> - Please refer to our End of Life Policy for EOL schedule details.

Replicator v20.0.2 - January 2026

Changelog

Fixes

- Replicator configuration migration version checking (243)
- Replicator hardware upgrade failing from v19.1.1 > v20.0.1 (244)
- Replicator UI freezes upon loading of the topology view with a high exporter count (248)

Replicator v20.0.1 - November 2025

Changelog

Fixes

- Addressed various security issues
- Headless replicator failing to integrate when created with space characters in name (233)
- Replicator upgrade fails (234)

Replicator v20.0.0 - October 2025

Changelog

New features

- Add Support for IPv6
- HA dual exporter profile type
- Manage Auto Replicate through the UI
- Single Sign On (SAML)
- Support giant packets / Fragmented packets

Enhancements

- Auto Replicate: Ability to support only new device discovery
- Auto Replicate: Add logic so that if a policy is modified in the Seed profile, that policy change is applied in any collector profiles that include the same policy.

Fixes

- Issues with more than 103 collectors (19)
- User configurable timezones (8)

4.6.3.6 Scrutinizer changelogs

Changelog entries are displayed in the format **DESCRIPTION (Ticket Number)**.

> **Note**
>
> - For more information on Scrutinizer and the Plixer One platform, visit www.plixer.com or contact *Plixer Technical Support*.
> - Please refer to our End of Life Policy for EOL schedule details.

Scrutinizer v19.7.2 - January 2026

Changelog

New features

- Add local replication to P1, P1-Core, and P1-Enterprise keys

Fixes

- Allow outbound connection to Cisco Secure FMC eStreamer endpoints (5381)
- Duplicate rows in Admin > Exporters and Admin > Interfaces when exporters sent to multiple collectors (5354)
- Entering invalid subnet for IP Group rule causes errors (5386)
- Flow collector can hang on /tmp/aws permissions (5309)
- Replicator upgrade can stop on package dependency failure with custom packages installed (5367)
- scrut_tmp/collector_plixer being unmounted in AWS (5369)

- SNMP details duplicated in admin views (5353)
- SNMP discovery not always detecting new devices (5332)
- SNMP names/speeds sometimes missing after upgrade to 19.7.1 from 19.6 (5330)
- SNMP service panic when sysuptime is not returned (5361)
- Temp directory cleanup can cause UI errors (5396)

Scrutinizer UI fixes

- Interface descriptions missing for filters (3029)
- Save As not working correctly for summary reports (3021)

Scrutinizer v19.7.1 - November 2025

Changelog

Fixes

- Addressed various security issues
- 19.7.0 configuration migrator wipes historical data on the destination (5255)
- Issues with report menu from Host Indexing (5244)
- Protocol incorrect for undefined WKPs (5247)
- Remove duplicate plixer.column_index entries on upgrade (5261)

Scrutinizer v19.7.0 - October 2025

Changelog

New features

- Ability to filter ‘Ungrouped’ Device Group from the Explore Tab
- Ability to Specify custom LDAP attributes (support for eDirectory LDAP)
- Admin: Resources: SNMP Credentials
- AI Assistant (Requires Plixer One Core or Plixer One Enterprise licensing and updated key)
- Capture Netflow Info From Kubernetes Workloads
- Experimental multi-tier storage support
- Flow Hopper to the New UI
- General Components: Context menu with copy & new window functions
- Support for AES256 in SNMPv3
- Support for Zscaler ZIA and ZPA
- Topology: Primary view for beta feature

Enhancements

- Ability to add multiple IP group definitions all at once
- Ability to export the entire list of manage exporters as a CSV
- Add firewall rules for outbound connections

- Added details for report buttons in the Host entity view
- Admin: Authentication Tokens filters
- Admin: Direct links from the Configuration Checklist into the application
- Always show what the report menu will filter on
- Display timezone in the date selector
- Endace Pivot for host pairs
- General Components: Change branding and show licenses
- Remember table column selects per user
- Simplify the AI Settings

Fixes

- Addressed various security issues
- –set timezone writes to wrong configuration file (5059)
- Admin > Alarm Monitor > ML Dimensions page table shows Protocol value of "ALL" as "HOPOPT" (2827)
- Admin: Exporters and Interfaces - can disable all columns in table (2792)
- Admin: FlowPro Probes - cannot save edits to settings without editing APM License Key for table entries (2702)
- Admin: Mapping Groups 'Make Default' slider in Settings doesn't update the table (2759)
- Admin: Menu search bug (2660)
- After removing a report gadget from all dashboards, the "Edit Gadget" menu in the Export tray changes to "Add to Dashboard" (2744)
- Allow port numbers over for 45435 for syslog notifications (5060)
- Auto-created Endace pivot (EndaceProbe P2P) does not work in New UI (1240)
- AWS integration: Fix plxr_awss3 process memory leak (4975)
- AWS integration: Scrutinizer Integration page bug with multiple S3 buckets (2722)
- Bidirectional CSV Export only exporting single direction (5023)
- Can't see the exporter IP in a Report filter (2903)
- Change default report time to "last X" instead of "custom" (2398)
- Cloud icons on maps are chopped off on the left and right (2742)
- Collections: Adding an unsaved report to a collection causes all future unsaved reports to appear as added (2890)
- CSV Exported Client Server report displays EPOCH time (4422)
- Customer Interface Gadget issue with filtering (4758)
- Dashboard Differences - font sizes and text box colors not changeable in Maps (2679)
- Dashboards: Custom Gadget External URLs allowed by default (2791)
- Dashboards: Disable iframe gadget input if external URLs aren't enabled (2846)
- Dashboards: Problems with Double Sankey gadgets (2723)
- Excessive stray files (5176)
- Existing report names can be edited to include invalid characters (2752)

- Explore: Exporters | Exporter tray Interfaces section broken link and errors (2898)
- Explore>Entities: Filters do not stay cleared after clicking the active table again (2629)
- Explore>Exporters: Packets and Flows columns sort incorrectly (2626)
- Exported csv file name differences between hosts tab and policies tab (1807)
- Filters are too persistent (2496)
- Flow Rate under manage exporters doesn't match vitals (4988)
- FlowView isn't hiding the last graph (2843)
- Forecast Table Data Understated when reporting on Rate (2802)
- Forecasts: Duplicate forecast creation is not restricted (1536)
- gettingstarted.sh doesn't use the custom SSL details entered (219)
- Google Maps - View child map button not working (5149)
- Interface Threshold Alarm Reports expiring quickly (5047)
- Investigate>Alarm: Risk column sorting incorrectly (2616)
- Issues with Syslog, SNMP Trap, and CEF test buttons (2819)
- Investigate>Hosts: Underscores in IP Group names are replaced with spaces in src/dst charts (2079)
- lang_key issue with designed reports "Custom_XXX" in report type drop down (4775)
- Manage exporters flow rate does not match vitals reports (4980)
- Mapping: Ampersand in report name displays as encoded "&" (2832)
- Mapping: Remove duplicate refresh button (2717)
- Mapping: Reports fail to load if report name contains parentheses (2845)
- Mapping: Thresholds missing units and column information (2831)
- Mapping: Usability issues with saved report connections (2850)
- MappingObjects: Link for icon type objects cannot be modified (2704)
- MappingObjects: Map does not update after Apply changes to a Text Box object (2777)
- Missing checkbox for IAM authentication in 19.6.0 (4835)
- Monitor > Dashboards: Cannot modify existing dashboard to default or read-only without renaming (1842)
- Monitor: Alarms by Hosts - Show Host Names toggle not respected (2669)
- Naming AWS vpc exporters reverts to vpc Id (4937)
- NAT All Details report needs a column removed to work with Palo Alto (4563)
- Old Host Indexing "first seen" data unavailable after upgrade (4011)
- Reindexing error doesn't halt upgrade (5179)
- Report wizard "Run Report" button doesn't activate (2649)
- Reporting: Incorrect sorting on report name (2892)
- Reporting: Summary Reports save differences (2813)
- Reports requiring collector to collector communication (4940)
- Reports that should default to stacked graph loading unstacked on first load (4647)

- Reports | Other Options links open in new tab and new window, broken Report to ISP link (2900)
- Reports: Avg Pkt Size report changes graph type when the report is re-run. (2678)
- Reports: Cannot switch between Report Type if report has parentheses in the title (2863)
- Reports: Sankey graph doesn't render correctly on the last few pages of report (1781)
- Reports: Sankey graphs are not adjusting correctly to container height (2712)
- Reports: Should not be able to create a gadget from a Summary Report. (2653)
- Reports: Summary reports do not populate/update Last Run timestamp. (2652)
- Saved report contents disappear on dashboard refresh (2708)
- Source Hosts and Destination Hosts reports both named hosts reports in the tray menu (2774)
- Terminal Icons (on maps) display unkonwn icon when changing status (984)
- Top Interfaces click on sparkline is broken (2902)
- Top Interfaces Report: exclude filters shouldn't be an option (2206)
- UI allows emailing and scheduling reports with no email server configured (2849)
- Unable to ingest VPC flow logs from AWS bucket that contains "reject-reason" information field (4858)
- VA disk could fill with logs when deployed, but not configured for weeks (5130)

Scrutinizer v19.6.1 - June 2025

Changelog

Fixes

- Reparser crashes when sFlow is missing L2 header data in sample (4842)
- PDF report generation not working in 19.6.0 (4863)
- Some combinations of protocol exclusions can result in collector crash (4866)
- Device group filter is not displayed correctly (4878)
- New deployments don't default to slim navigation (4889)
- Saving a user with some missing preferences results in losing all preferences (4915)
- Unreadable map labels in dark theme (2754)

Scrutinizer v19.6.0 - March 2025

Changelog

New features

- "DHCP Servers" and "LDAP Servers" IP Groups for ML Exclusion management
- Ability to run reports menu from the host entity view
- Ability to specify number of rows in a report gadget
- Add support for Azure VNet Flow Logs
- Add/Update support for Cisco VXlan IEs
- Added additional CloudGenix / PaloAlto SDWAN reports
- Additional support for additional Keysight

- Admin UI for FlowPro Capture Rules
- An interfaces tab to the host entity view when host is an exporter
- Audit Report to Admin UI
- Collect VPC Flow Logs from Google Cloud Platform
- Direct links to Exporters and Interfaces in Explore
- Disk space calculator as part of data settings under admin
- Edit gadget features from Dashboards
- External NAT filter
- Full screen mode for Dashboards
- Interactive configuration checklist
- Interface entity view
- LIKE/NOT LIKE filters to Alarm Monitor, Explore, and Admin views
- Lollipop chart reporting graph type
- MITRE ATT&CK dashboard gadget
- ML behavior data in Alarm Monitor workflows
- ML Exclusions Admin View
- Move feature resources under system performance so it is per server
- New Dashboard workflows
- New report folder management workflows
- New top interfaces dashboard gadget
- Option to include full Interface names in reports
- Oracle flow log ingestion
- Reporting on VXLAN from sFlow samples
- Reporting: Line Item gadget type
- Ridgeline graph type
- Ring Gauge reporting graph type
- Scheduled Reports view
- Slim navigation mode with vertical navigation bar
- Support for TLS v1.3 with LDAP integration
- Top Exporter report type
- Top N dashboard gadgets
- User behavior reports
- User configurable horizon in report forecasting
- Zabbix client package in our repositories

Enhancements

- Add “Last Year” & “This Year” options to report custom time ranges
- Add exclusion workflow from alarm monitor - front end
- Add Google API key from mapping UI
- Add link to host entity view in “Other Options” under report menu
- Add link to user settings in the user menu
- Add option to report on all interfaces bi-directionally
- Admin menu search
- Always search for report types in all report groups
- Auto-expand active filter sections in tray
- Entities: filter on click from host to alarm
- External custom gadget URLs preference
- FA setting to exclude internal or external communication for lateral movement
- General components: improved severity display
- General components: add plixTips when slimNav is collapsed and title tags when expanded
- Include technology in the alarm monitor view somehow
- Informative detailed error messages when UI can’t communicate with a reporter
- Lateral movement FA algorithms and preferences
- Mapping automatic grid layout
- Mapping icons updated
- Mapping workflows
- New mapping workflows
- Provide interface names in Sankey graph tooltips
- Recategorized system preferences
- Remember selected columns in alarm monitor
- Rename “Favorites” to “Recent” in report menu
- Report description tooltip issues
- Report menu search by description and information element
- Reports: hidden graph button shouldn’t be there if we don’t have a graph available
- Reports: make the saved reports title clickable
- Saved reports view
- sFlow 801.2ah header support
- Show active filter status for alarm monitor and reporting
- Support newer versions of Cisco ISE for user name reporting
- Workflows in manage exporters
- Workflows in manage interfaces

Fixes

- Addressed various security issues
- ‘Client Server’ report failure while filtering for domain (4068)
- Ability to edit Flow Analytics Configuration rules in the new UI (4179)
- Ability to save a key with an unsupported feature_set (4091)
- Adding IP Group with Subnet Rules didn’t save mask selection on initial save (4439)
- Admin: Guest users need these routes inaccessible (2243)
- Admin: Set LEDs to refresh every 30 seconds and on click (2220)
- Admin: Users & Usergroups not respecting routing (2080)
- Apache server version is shown in header responses (4326)
- Apply button for Single Host-Index search not functioning (1981)
- automatic template naming (3154)
- Azure NSG exporter naming for distributed collection (3989)
- Azure NSG flow log bi-flow support (3993)
- Changes made in the oldUI manage interfaces tab are not saved (4301)
- Cleaned up Host Index Max Disk Space error when increasing too much (4219)
- Collections: No Results Found still shown after creating a collection (2582)
- Collector won’t run if the reporter is down (4147)
- Copying or refreshing reporting URLs displays error (2108)
- Dashboards: map resize on gadget resize (2075)
- Date selector doesn’t always allow for shifting the date forward (2236)
- Destination AS filter fails on top interfaces report (4309)
- Editing an applied filter does not provide enough space (1996)
- Expire history failing in some cases after upgrade (4376)
- Explore > Exporters > Interfaces should prefer ifAlias over ifDesc (4397)
- Exported PDF report has Subscription ending message (1359)
- External links using the search route (2105)
- Flow Analytics Admin workflow issues (2153)
- Flow Version is not visible in new UI (2196)
- FlowHopper can’t find flow starting from sFlow (4711)
- Full Interface name in reports, sporadically displays (4230)
- General Components: Adjust ML graph to take in entire time period, not just data extents (2211)
- General Components: app-table pagination skip buttons (2116)
- General Components: Inconsistent table header behavior (2047)
- Gigamon tcpcontrolbits exceeding smallint value (4374)
- Grafana Plugin (443)

- Having a "/" in a report name breaks the ability to run that report from an alarm (4327)
- Hidden interfaces showing in reports (4100)
- Internal Server Error when editing Network Map connections (4063)
- Investigate > Host: Learn More button "Host Details" option has no function, related observations (2491)
- IP Groups, adding a child group displays wrong selection (4440)
- IP V6 import hostfile is broken (4142)
- IP/DNS in Flow Analytics Configuration (2014)
- IPv6 Exporters don't retain snmp configuration (2865)
- Issues when graphing Silverpeak performance reports (microsecond values) (4657)
- LDAP login slow with 100K+ group definitions (4038)
- Less Than & Greater Than options missing from Advanced Filters (2299)
- Manage Exporters & Explore Exporters – Slow / Not loading (4080)
- Missing some country codes (4532)
- Newline characters in report threshold alarm messages (4037)
- Nightly clean all task is removing valid snmp credentials (4419)
- Non-Admin users not able to run reports from alarm pages (4263)
- Old Host Indexing "first seen" data is unavailable after upgrading. - Import From History Option (4011)
- Other options menu opens new window (1965)
- Out of file descriptor errors (3941)
- Provide description of timeframes for top interfaces and exporters view in Explore (2224)
- Recent and Recommended report groups need to show the group details (2536)
- Report JSON link returns unnecessary data (2199)
- Reports: Saving a report as 'testSave' results in 'Test and Save' in header name (2445)
- Restore Manage Exporters view in the Classic UI (4317)
- Saved Reports - Host filters get removed when pivoting (2322)
- Scheduled email reports have the license subscription ending soon warning (4082)
- Scheduled Traffic Volume reports revert to Line graph when set to Step (3962)
- SSL langkey is blank in serverprefs after running set ssl on (4019)
- Sync Primary taking too long in some cases (3384)
- sysbench package should be installed (4682)
- Targets CSV file has 'violators' in the name (2194)
- Threats Domains temp directory is not always being cleaned out (4395)
- Unable to zoom in on TopN report graph (2043)
- Update certificate scripts for Oracle Linux (4468)
- Usability issues with usergroup permissions in the new UI (1264)
- User able to set 'unlicensed' in Manage Exporters (4362)

- Usergroup Permissions do not carry over to new UI when editing saved reports (1316)
- web certificate paths changed back to pre-19.5 location (4359)
- When changing an alarm notification frequency to "Rate", it reverts back to "Each Observation" (3906)

Deprecated

- Remove "Additional notes" input from new object form

Scrutinizer v19.5.4 - November 2024

Changelog

Note

- This release addresses CentOS going EOL. To migrate the OS to Oracle Linux 9, Scrutinizer must be on version 19.4.0. Please contact *Plixer Technical Support* with any questions.

New features

- Support for AWS OL9 AMI
- Support for additional Palo Alto Prisma information elements
- Support for additional Keysight information elements
- FlowPro 20.1 compatibility

Fixes

- Addressed various security issues
- CyberArk Dependencies missing (4497)
- Collector fails to start when the primary reporter is down (4552)
- ML Heartbeat can prevent registering of Plixer ML Engine (4556)

Scrutinizer v19.5.3 - October 2024

Changelog

Note

- This release addresses CentOS going EOL. To migrate the OS to Oracle Linux 9, Scrutinizer must be on version 19.4.0. Please contact *Plixer Technical Support* with any questions.

New features

- Oracle Linux v9.4
- System Migration Utility

Fixes

- Addressed various security issues
- Filtering issue with NSG FlowLogs (4225)
- Slow load times for Admin > Manage Exporters (4080)
- Primary server being down prevented collector services from restarting (4147)

Scrutinizer v19.5.2 - July 2024

Changelog

> **Note**
>
> - This release addresses CentOS going EOL. To migrate the OS to Oracle Linux 9, Scrutinizer must be on version 19.4.0. Please contact *Plixer Technical Support* with any questions.
> - AWS instances of Scrutinizer use Amazon Linux 2 and do not need to be updated to 19.5.2. A later release, which will include new features and bug fixes, will be made available for Scrutinizer deployments on AWS.

New features

- Proxmox support

Fixes

- Addressed various security issues
- Memory leak (4363)
- Missing AS and country names (4353)
- SNMP polling issue (4364)
- Data migration fails when destination expires history (4364)

Scrutinizer v19.5.1 - June 2024

Changelog

> **Note**
>
> This release addresses CentOS going EOL. To migrate the OS to Oracle Linux 9, Scrutinizer must be on version 19.4.0. Please contact *Plixer Technical Support* with any questions.

New features

- Check for supported CPU architecture in olmigrate
- Automatic disabling of root login in olmigrate
- Check for multiple interfaces in olmigrate

Fixes

- Addressed an issue where a recursive directory is created if olmigrate is run more than once for the same upgrade stage

Scrutinizer v19.5.0 - May 2024

Changelog

> **Note**
>
> This release addresses CentOS going EOL. To migrate the OS to Oracle Linux 9, Scrutinizer must be on version 19.4.0. Please contact *Plixer Technical Support* with any questions.

New features

- Oracle Linux v9.4
- System Migration Utility

Fixes

- Addressed various security issues
- Filtering issue with NSG FlowLogs (4225)
- Slow load times for Admin > Manage Exporters (4080)
- Primary server being down prevented collector services from restarting (4147)

Scrutinizer v19.4.0 - October 2023

Changelog

New features

- AWS Flowlog consumption 35x faster
- AWS Flowlog consumption and processing can be spread across multiple collectors
- Azure flow log ingestion
- Azure NSG Reports
- Security Groups for enabling groups of Exporters in Flow Analytics
- Userpreferences Modifiable Template
- Include Custom Designed Reports in Scrutinizer Configuration Backup
- Support 18.20 -> 19.X offline upgrades where the repo server is the Scrutinizer server
- sFlow vlan/sub-interface report
- Merged target and violator alarm views into consolidated hosts view
- Host entity alarm timeline view
- Endpoint Analytics Risk and details into Alarm Monitor views
- Multiple new Alarm Monitor visualizations
- Connections graph type in reporting

- New Admin interfaces
- Default Flow Analytics Exclusion Groups under IP Groups
- Include port name in DrDoS alarm messages
- Support for FlowPro version 20

Fixes

- Addressed various security issues
- On demand PDF/email/csv use server time zone when they should use user time zone (1069)
- Optimize TCP/UDP FA algorithms (2695)
- Turning SSL off breaks the UI (2728)
- Double quotes in SSL serverprefs (2927)
- Distributed upgrades should have collectors run a curl check for Internet access (2958)
- Exporters Not Deleting with Domain Exclusions (3110)
- Event severity timeline (3329)
- Editing FA host exclusions doesn't update caches (3332)
- Distributed Upgrade Installer handle proxy configuration prompt (3339)
- System Performance View shows red when resources exceed the matrix (3351)
- Implement sFlow version 4 (3357)
- Filter all FA sliding windows by streamexporter (3377)
- set myaddress fails on Hardware appliances (3386)
- CSV export column header shifted by one position for Connection reports (3400)
- Reporting - Source / Destination Port EXCLUDE Port Range - Error: "report failed" (3402)
- Setting timezone can pause alarms (3405)
- Issue with units label for application latency report threshold messages (3471)
- Added paged requests to LDAP authentication to handle large lists of Active Directory Security Groups (3481)
- Fix overstated utilization when sFlow counters are dropped (3485)
- Optimize Explore By Exporters view (3541)
- Clean history table orphans in batches (3555)
- RADIUS shared secret needed to be re-entered after v19.3 upgrade (3591)
- scrut_util 'set ssl on/off' requires root - but should not be run as root (3624)
- Escape special characters in interface details (3636)
- Fix a logs-based disk space leak (3667)
- Store AWS interface in the aws_interface element - don't map to ingressinterface (3687)
- Optimize FlowPro FA algorithms (3695)
- Optimize Packet Flood FA algorithm (3699)
- Optimize Slow Port Scan Algorithm (3700)
- Legacy Baselining is now EOL (3704)

- Made UDP receive buffers configurable (3711)
- Mixing include and exclude advanced filters could restrict more results than necessary (3757)
- Don’t allow “Host Index Max Disk Space” setting to exceed available disk space (3779)
- Manage Exporters and Manage Collectors were removed from the classic Admin UI (3808)
- Monitor.top_stdout Parsing Errors (3828)
- AWS Upgrade package dependency problem (3862)
- scrut_util check heartbeat database as root user error (3873)
- Move slog directory out from under html (3882)
- No packet or octet values for exporter sending samplingpacketspace of 0 (3903)
- distributed_stats_exporters wasn’t being cleaned out (3931)

Scrutinizer UI fixes

- Reports: Restructure to allow proper placement of app-page-toolbar and tray (1001)
- Dashboards: Too much air in vitals (1054)
- Dashboard Recent Alarms gadget is out of sync with current alarms (1114)
- Alarms: Show DNS & IP information in messages (1252)
- Acknowledged Alarms View Doesn’t auto-refresh (1417)
- Explore Event Traffic links do not respect PlixCal filter (1441)
- Exported CSV from Entities page displays host names with ‘Show Host Names’ deselected. (1463)
- Spatial Map: Modified timestamp gets wrongly updated to all the existing maps (1498)
- Explore>Entities: “dbQueryError” seen in console when applying filters (1574)
- Admin: Default Status, Tab & View (1591)
- Default map defined for user does not open when accessing Network Maps in new UI (1598)
- Change Endpoint “Identity Score” to “Profile Match” (1617)
- Reporting: Phantom selected select box (1712)
- Issue with displaying child groups with a parent group filter (1877)

Scrutinizer v19.3.2 - September 2023

Changelog

Fixes

- Addressed various security issues
- AWS Upgrade package dependency problem (3826)

Scrutinizer v19.3.1 - April 2023

Changelog

Fixes

- Addressed various security issues

- AWS interface IDs no longer used as observation domain (3568)
- Deleting collector log wouldn't always return diskspace (3667)
- Reduced output to logfile for Feature Resources (3675)
- Optimized query for Explore exporter view (3684)
- Upgrades needed a forced reboot for chromium (3692)
- Update LDAP login to get the `defaultRoute` preference (3697)
- Changed default view for Explore to Top Interfaces (3703)

Scrutinizer v19.3.0 - December 2022

Changelog

New features

- MITRE ATT&CK visualization
- MITRE ATT&CK details for notification profiles
- Support for using hostname when configuring an ML Engine
- Support for redirecting to a proxy address after single sign-on
- LRFM: no audit trail from manual enable/disable
- sFlow: add support for VLAN tags in sampled Ethernet headers
- sFlow: support for sampled IPv6 headers
- Ability to pass custom parameters when opening ServiceNow issues

Fixes

- Moloch Integration Link not clickable in the new UI (1035)
- Admin Tab permission is required to logout (1269)
- Report selection stuck open without selection (1372)
- Report Data Source Values Show Twice (1374)
- FA Configuration > DRDoS > Settings is missing details (1391)
- Top Interfaces are duplicated for exporters in multiple device groups (3204)
- Undefined Error when modifying Guest Permissions (3219)
- CSV export of Volume reports shows incorrect rate data when resolution doesn't match datasource (3226)
- Error when filtering alarms by violator (3230)
- Add search.html type route to the new UI (3234)
- S3 Integration: Fix a crash when the database disappears at certain times (3235)
- Adding Show Interface option to a report shows outbound exporter as NA (3263)
- LDAP Authentication Fails due to Primary Key Duplicate Restraints (3281)
- Flow Collection Resumed Message Displays First Message instead of Last Message (3292)
- Host Index searches show 'first_seen' as the date of the host_index import (3334)
- Totals values could be doubled when an interface is metered both ingress and egress (3370)

- Severity card time frames don't match date selector (3434)
- Kafka logging can crash server processes (3437)
- Report links from Host Index would pop up a broken window (3483)
- Host Index cleanup tasks fail if H2H Index is turned off (3498)

Plixer Scrutinizer UI fixes

- Entities: Alarms: Events: Incidence correlation resize scrollbar (1336)
- Top Src/Dst Host pivot from an IP Group entity view opens a Username Entity view (1412)
- Setting custom interface speed to 0 to override displaying as percent utilization (1416)
- Dashboard issues: Excessive scroll bars on Windows and report gadget graph legends difficult to read (1602)
- CEF: timestamps for start/end times (3369)
- Support multiple usernames per host in alarms (3372)

Scrutinizer v19.2.2 - September 2022

Changelog

Fixes

- Addressed various security issues
- AWS Upgrade package dependency problem (3826)

Scrutinizer v19.2.0 - May 2022

Changelog

New features

- Added option to toggle how device group hierarchy is displayed (153)
- Prioritize exporters that get disabled last in the event that a license overage causes some exporters to be disabled (203)
- Ship Scrutinizer with sysbench and a test script in files (1269)
- Expand CEF message content to include ports and usernames (2001)
- Improve messaging on "Unapproved Transport Protocols" alarm page (2161)
- AWS flowlogs: add support for new version 5 fields (2410)
- Workflow issue: unapproved protocol policy report pivot should include protocol filter (2426)
- AWS S3 test button: test the required permissions (2428)
- Improved alarm policies report link filters (2468)
- Run report on packet flood event does not filter on the traffic that triggered the alert (2499)
- Don't use unencrypted connections for upgrades (port 80) (2607)
- Include shortened report URL in report threshold policy (2636)
- Create some new AWS reports for v5 elements (2651)
- Audit log entries for key management/encryption changes (2723)

- Ability to set a key lifetime (2724)
- VPC flow logs now require interface-id and flow-direction (2817)

Fixes

- Addressed various security issues
- Fixed issue where configuration wouldn't synchronize when all settings are removed (473)
- Admin > settings > proxy server has been renamed 'Google Maps Proxy Server' (941)
- PDFs for large reports show the "painting a Plixer" screen for the report screen shot (1054)
- Device tree hierarchy doesn't carry over to user groups with explicit device group permissions (1500)
- Restore username details to alarm notifications (1999)
- Distributed data expiry errors without events/trends (2190)
- Deactivate sliding windows when FA algos are disabled (2310)
- ACL 'like' filters don't work for ACL descriptions (2312)
- DDoS and DRDoS alarms no longer present CSV access to the offender source list (2343)
- AWS S3 test button: test from the specified collector (2355)
- Improved incident correlation algorithm (2380)
- Emailed reports from report threshold alert sometimes have incomplete report images (2413)
- ipfixify-template filepath updated in manual (2445)
- Unable to export report to PDF or email report for SSL not using port 443 (2463)
- "Report Direct Link" doesn't work for on-demand emailed reports (2485)
- Run report option in report threshold violation event list does not use the saved report filters (2491)
- Unable to export saved reports to CSV with space in saved report name (2506)
- Report threshold violation email's URL should load the timeframe of the violation (2539)
- inserter.pm stops polling for SAFs, sampled SAFs, totals if the database is temporarily unavailable (2556)
- Graph and table show in different timezones (2562)
- Top asn overstates exporter count (2595)
- Proxy server support needed for online upgrades (2608)
- Remove ICMP ping check from upgrades and pass through variables (2609)
- Enable SSL as the default for offline repo servers (2618)
- SonicWALL IPFIX extension templates not being read correctly in v19.X (2622)
- AWS flow reports - can't filter on the interface (2630)
- AWS flowlogs temp dir missing after upgrade to 19.1.0 (2670)
- Allowed transports aren't sync'd to all collector nodes (2675)
- FA NULL scan algo doesn't exclude destinations (2681)
- scrut_util –enable ram_spools blows away /etc/fstab (2684)
- Sflow inserting - extra data after last expected column (2697)
- Latency value ingesting from Ixia not show up properly on Scrutinizer UI (2709)

- Special case sFlow interface instances missing (2712)
- FA worm algos don't exclude hosts (2732)
- Update docs.plixer.com to reflect how syslog alerts are configured (2773)
- events.backfill_summaries() crashing with ddos events (2774)
- FA breach algo doesn't exclude servers (2805)
- An offline update server with self signed certificates may try http (rather than https) and fail (2812)
- Host index is now configured in flow analytics (2856)
-
- Reparser will not redefine templates without hard restart (2882)
- Running single direction report via the top interfaces view returns 'No Template' (2883)
- Scrutinizer device inactivity threshold is not triggering violations (2890)
- Remove plixer_syslogd from systemctl on upgrade (2892)
- FCGI timeout settings removed after upgrade (2893)
- Install fails with dependency error on 'device-mapper-multipath' (2905)
- Distributed upgrade hanging at TASK [Gathering Facts] (2907)
- Disabling an algorithm does not remove its exporters from plixer.streams_config (2944)
- FA reverse shell doesn't exclude source (2952)
- Low spool disk space "FA Streaming was disabled" does not disable FA streaming (2979)
- Event policy customization improvements (2985)
- Events with empty target/violator lists crash the policy view (3010)

Plixer Scrutinizer UI new features

- Unapproved Protocol Policy third donut chart now has top hosts using protocol (966)
- Include Time Zone in the report date/time display (1012)
- Monitor -> Network Maps Grid view delete option (1030)
- Better DNS Resolve Setting description (1053)
- Latest alarm message to events table (1199)
- CSV links in Policy entity (1207)

Plixer Scrutinizer UI fixes

- Naming a dashboard "Network" in v19.0.2 renames it to "Subnet" (909)
- History Navigation shows Alarms by ID instead of English Description (924)
- Navigating into alarm monitor sometimes throws an ExpiredRequestID error (975)
- inbound and outbound interface reports from explore device tab do not apply the correct filter (988)
- Regression: Traffic %, Other, and Total displaying for sFlow reports (1004)
- New UI doesn't use the time zone user preference in reports (1013)
- Time Stamps on Line and Step Stacked 1m data source, 1m resolution overlap (1017)

- Deleting the default collection causes “notExists” error when trying to add to the default collection (1027)
- Host Entity View -Top Alarms bell icon mouseover text does not align with click action. (1029)
- Reports against an exporter with no current flow data does not allow for timeframe changes. (1031)
- New UI | Explore -> Interfaces -> Refresh Rate is not saved (1033)
- Changing Report Options triggers direction back to INBOUND when bidirectional is allowed (1038)
- Clicking the add or remove selected buttons keeps the tooltip on screen (1050)
- Recent Alarms Dashboard gadget shows UTC timestamp for Last Event and Last Notification (1112)
- Explore: Devices not using User Default Unit setting - Shows Percent always (1113)
- Toggling Hostname resolution does not change IPs to hostnames in alarm policy views (1135)
- Device/Interface report filter inconsistent with the Show DNS or IP modes (1216)
- Host to Host Index search doesn’t render a report menu when clicking exporter hyperlinks (1218)
- Alarms Monitor Filtering Option by Violators/Targets returning “noDataAvailable” (1221)
- CSV export of a report loses DNS names (1241)
- PDF export of report only shows 10 lines (1242)
- Peak and 95th Percentile not showing on saved reports (1244)
- Report filters not showing up in the “Additional Filters” drop down (1259)
- Show Others displaying when set to No (1267)

Machine learning engine new features

- Add ML Engine metrics to Vitals reports (338)
- Support high availability (419)
- Support Zerologon detection (446)
- Support SIGRed detection (447)

Scrutinizer v19.1.1 - September 2021

Changelog

New features

- Automatically shut down non-critical features when systems are overwhelmed (2703)

Fixes

- Addressed various security issues
- “Sizing your environment” guide
- Timeout when migrating large historical host_index tables (2337)
- Upgrades didn’t stop on database upgrade error (2638)
- Full alarm message not getting into ServiceNOW tickets (2640)
- AMI didn’t have spools on RAM disk / tuning didn’t run on AMI deployment (2646)
- Resizing disks with AWS C5 instances (2696)

- Performance issues with host_index process (2701)
- Inefficiency in building TopN view (2710)
- Max locks wasn't set high enough for some upgrades from v18 (2751)
- Report links from threshold violations had the wrong timeframe (2785)
- Registering a new collector could overwrite meta data on the primary (2788)
- Character encoding issues synchronizing binary data (2794)
- Pulling STIX TAXII threat list (2831)

Scrutinizer UI fixes

- URL too long error from report wizard with large exporter counts (852)
- Line and step graphs wouldn't load after switch from a Traffic Volume report (1032)
- Graph and tables in a report could show different timezones (1059)
- Changing Report Options triggers direction back to INBOUND (1060)
- Flow data with a single direction could break the gear menu (1062)

Scrutinizer v19.1.0 - May 2021

Changelog

New features

- Scrutinizer services not required to run as root (187)
- Client - Server reports (261)
- Encrypt stored keys (516)
- Copy to clipboard button to api json tab (733)
- Option to toggle Show System Policies (786)
- Expanded and reworked Host Index and H2H Search (883)
- Target / Violator views and filtering in Alarm Monitor(898)
- Show Host Names and Show Acknowledged Events for Alarms(948)
- Include collector IP address in all vitals reports for grouping and filtering(1971)
- Refactor Alarms backend for better performance (2053)
- Flexible notification policies based on event criteria (2060)
- Autoreplicate support for multiple replicators (encrypt multiple passwords) (2111)
- Ability to set Alarm policies to inactive or store (2231)
- root login disabled on new deployments (2361)
- Cisco SDWan (Viptela) integration updated to support version 20 (2374)

Fixes

- Addressed various security issues
- Mapping: add checks and errors for duplicate map connections (313)
- Sorting by bytes does not account for units in Entity Views (724)
- New UI reports do not display Host Names (793)
- PDF Export of Summary Reports Top N and Overview failure (805)
- Classic View option from user menu doesn't work (893)
- Fix scrolling issues for Exporter Details list in Report Settings (939)
- Alarms takes too long to load and acknowledge (1586)
- Reverse DNS exclusions for alarms (1798)
- Reparser crash when Linux ARP cache filled (1970):
- Adding a notification profile to a saved report threshold doesn't work (1977):
- Child Groups not enforced for FA exclusion (2030):
- Vitals process crashing with extremely high MFSNs in flow streams (2090):
- Custom URL Dashboard Gadgets not working (2214):
- Valid licenses with Expired PNI/PSI eval's prevent the upgrade from running (2217):
- Stream bloat on heavily loaded systems could cause disk space problems (2235):
- Running out of file descriptors on heavily loaded systems (2250):
- Invalid certificates in distributed upgrades (2273):
- TopN views are not always populated (2279):
- LDAP login takes too long with a very large list of security groups (2300):
- P2P Alarm report link not working (2307):
- Improve handling of truncated sFlow sampled headers (2336):
- Flow collection doesn't resume at the end of a network outage (2346):
- Set webui_timeout not working (2358):
- Scheduled report tasks called wrong binary name after upgrade (2379):
- IP exclusion only checking source IP for RST/ACK and Host Reputation (2382):
- Fix incorrect or missing sFlow interface numbers for instances above 63 (2393):
- AES key not syncing on upgrade affecting SNMP, AWS, and other credentials needed on a collector (2401):
- License Exceeded alarm detail shows no data in Alarm Monitor (2414):
- Addressed CVE-2021-28993 (2457):

Scrutinizer v19.0.2 - January 2021

Changelog

Fixes

- Disabling User Does Not Invalidate Session (2075)

- Input validation needed in some forms (2076)
- Session cookie value stored in local storage (2080)
- Postgres log noise from unnecessary scheduled analytics command (2118)
- Distributed upgrade issue coming from 19.0.0 (2198)
- pg_cron memory leak (2202)
- Fresh v19.0.1 OVA does not use the 19.0.1 repository (2205) F

Scrutinizer v19.0.1 - December 2020

Changelog

New features

- DDOS: Support IPv6 (12)
- Add AWS Role Based Authentication for use in AWS (377)
- Allow AWS flowlog polling at 1m frequency (940)
- Enforce password policy on password change and restrict from using last four values (1235)
- Summary Reports added to new UI (1459)
- Add "scrut_util –show datasize" to enumerate DB schemas and their disk usage. (1539)
- Define Allegro IEs (1633)
- Support for new format of VPC flow logs (1890)
- Provide descriptions for AWS entity IDs (1891)
- Add Velocloud 4.0 IEs (tcpRttMs and tcpRetransmits) (1899)
- Document new AWS integration requirements (1992)

Fixes

- Mapping: Show Utilization only works for percent (54)
- Not excluding protocols by default (304)
- Secondary reporters show incorrect clock drift (696)
- Apache HTTP Server 2.4.0 - 2.4.39 Remote Open Redirect Vulnerability in mod_rewrite (739)
- Cannot Filter on S3 Bucket Element aws_account_id in a designed report (765)
- Internal Server Error when emailing PDF report name includes / (1065)
- Unable to Exclude IP address from DDoS algorithim (1316)
- Collector log error sflow buffer overrun at ./protocol/sflow/buffer.hpp line 146 (1480)
- VPC Flow Logs should be cleaned up more aggressively (1482)
- The plixer.idp.login_url field appears to be vestigial (1579)
- Other Options > GeoIP links not working (1592)
- Login banners are not working (1660)
- Interface names with special characters cause errors when triggering thresholds (1728)
- Alarm when disabling algorithms or ML stream (1734)

- Group Labels retain original input on Maps Dashboard Widget (1743)
- Host2host and host index lookups to work in distributed setup (1744)
- pgbouncer wont start after yum update (1796)
- Some reports were unable to display in percent interface view (1797)
- Reparser freezes on error during minutely exporter status updates (1812)
- No drillp-down into Connection on Maps (1813)
- Reparser memory leak in sFlow parser (1817)
- Devices blue after upgrade to version 19 (1840)
- ServiceNow Integration doesn't work when server response is too large (1842)
- Reporting: No Data for Timeframe automatically sends to start report wizard (1879)
- Sliding windows falling behind after upgrade to v19 (1911)
- Fix rollup issue for droppedPacketDeltaCount<unsigned64> (1912)
- Closing the report modal doesn't keep the report open (1917)
- Entity Views: sorting by bytes does not account for units (1918)
- Using LDAP user is authenticated but never added to a group when group list was too long (1920)
- Unable to disable unlicensed FA features (1930)
- Unrecognized key type: AWSLogs/xxxxxxxxxxx/ inc/lib/plixer/scrutinizer/awss3.pm line 547 (1941)
- Awss3.pm:373 – get_flowlogs() encountered an error while processing s3_connection_list: Invalid data Invalid data(unknown) for aws_account_id (1942)
- get_flowlogs() encountered an error while processing s3_connection_list: Invalid data (-) @ 1084 for transform (1945)
- Alarm Report data interval default empty for large time frame events (1946)
- NetFlow v5 sampling crashes postgres (1969)
- Too many open files (1981)
- multicast send failure 22: Invalid argument (1984)
- CEF notifications missing 'Device Version' (1988)
- Set 'ssl_prefer_server_ciphers' by default (1994)
- Missing sflow records after an upgrade (2002)
- Report values as rates in tables are incorrect after drilling in on a graph (2021)
- Distributed: AWS S3 secret failing when assigned to remote collector (2029)
- The application is running a vulnerable version of Apache (2068)
- The application is running a vulnerable version of Perl (2069)
- XSS Vulnerability in old UI mechanism to create groups (2070)
- Local file inclusion (2072)
- Autoreplicate support for multiple replicators (encrypt multiple passwords) (2111)
- Formula injection vulnerability in the ability to create third-party CrossCheck methods (2071)

Scrutinizer UI new features

- Entities: Hosts: Anomaly Chart (652)
- Summary Reports: Filtering (692)

Scrutinizer UI fixes

- Report filter descriptions don't always fill in (657)
- Dashboards not deleted (685)
- Drilling into Policy from Collection loses consistency vs Monitor View (688)
- Apache httpd: CWE-345: Insufficient verification of data authenticity (693)
- Reporting: Summary reports not stretching on page (744)
- Stop 'topping' the graphs (765)

Scrutinizer v19.0.0 - August 2020

Changelog

Important

Custom alarm policies are no longer supported. The Report Threshold Violation policy can be assigned one notification profile only.

New features

- New workflow-based user interface (9)
- DDOS: Support IPv6 (12)
- Address data encryption in Scrutinizer (370)
- Initial Collections implementation (371)
- magicbus_fdw: Avro serialization (476)
- Advanced threat intelligence feeds (481)
- SNMP Enterprise MIB support for Viptela (717)
- Support for new VeloCloud information elements (727)
- Use tenant_id for db ROLE (740)
- Require a license key for free mode (780)
- Support for content updates (781)
- Streaming support for customer data lakes (782)
- Host to host flow connection search (783)
- Plixer Replicator integration (784)
- Update the Silverpeak IPFIX information elements (874)
- Advanced security algorithms (903)
- STIXV1 IP watchlist import (1006)

- STIXV2 IP watchlist import (1007)
- TAXII 2 feed support for IP indicators (1008)
- Domain reputation checking (1142)
- JA3 fingerprinting support (1144)
- Machine learning for security-specific events (1152)
- Machine learning for network-specific events (1153)
- New licensed features (1215)
- ML forecasting in Scrutinizer (1256)
- ServiceNow integration (1258)
- CEF notification action (1411)

Fixes

- Failed “system updates” report “no updates available” (541)
- scrut_util.exe –collect asa_acl gives error Use of uninitialized value $debug in concatenation (614)
- Saved Reports Folder changes are not audited (636)
- Insecure Direct Object Reference (749)
- Vitalser Memory Leak (767)
- Define missing Cisco IEs (unknown_9_20000) (820)
- Define the unknown_elements for Viptela IPFIX exports (865)
- scrut_util –collect db_size is timing out (1196)

Scrutinizer v18.20 - April 2020

Changelog

New features

- Optimized sFlow collection (496)
- New VeloCloud information elements (2073)
- Security updates (2154)
- SNMP Enterprise MIB support for Viptela (2164)
- Updated Silverpeak IPFIX information elements (2165)
- CentOS 7 : kernel update (2176)
- PostgreSQL security release 10.12 (2177)
- Change default eval key to 14 days (2190)

Fixes

- sFlow traffic discrepancies (2156)
- Saved report dashboard gadgets always display in totals (2167)
- Reporting issues when 0 byte flows are excluded (2179)

- Fixed issue with totals when both ingress and egress flows are exported (2196)

Scrutinizer v18.18 - December 2019

Changelog

New features

- New VeloCloud reports (1939)
- Set admin password to instance_id for AMIs (2036)
- Add SSO authentication method to the manual (2039)
- Many updates, improvements, and clarifications in documentation (2051)
- New Viptela reports (2124)
- Option template based descriptions for VeloCloud LinkUUID (2133)

Fixes

- Create scheduled reports was also requiring admin tab permission (421)
- Auto refreshing pages would prevent session timeout (1441)
- Resolve timeout for FA reverse DNS exlusions wasn't using setting from admin tab (1405)
- We now exclude 0 byte flows biFlow records for reporting and FA (1536)
- Protocol exclusions were not audited (1756)
- 255 character limitation for 'Security Groups Allowed' when configuring LDAP integration (1816)
- Improved column naming in some VeloCloud reports (1936)
- Resolve a harmless UDP receive buffer error (1985)
- Viptela reports would sometimes not show all vEdge hosts (1992)
- Session timeout based on backend activity, not frontend activity (2030)
- PDF report displays no data when data is present (2040)
- Expand Disk scrut_util commands now support NVME drives (2041)
- If an IdP certificate is not provided, SAMLRequests should be unsigned (2106)
- SSO - Submitting metadata XML via the admin view form incorrectly parses out tags (2107)
- Fixed memory leak in vitalser (2041)

Scrutinizer v18.16 - September 2019

Changelog

New features

- Viptela SD-WAN reports (16)
- Permission configuration on a role basis (270)
- Changed AWS Flow Log collection to use S3 buckets and added support for multiple regions and customer IDs (378)
- VeloCloud SD-WAN reports (550)

- Service Now Notification support (569)
- Appliance self migration from CentOS 6 to CentOS 7 (826)
- Ability to Add/remove/update Defined Applications via the API (891)
- Single-Sign-On support through SAML 2.0 (897)
- Alarm when authentication tokens will expire in 30 days or have expired (937)
- Deleting an exporter doesn't block collection (992)
- Removed device specific status notifications (1099)
- Audit logs can now be expired after a configurable duration (1171)
- FDW option to Database migrator for faster PostgreSQL migrations (1205)
- Flow inactivity alarms are now checked across a distributed cluster and are per exporter rather than per interface (1254)
- Support for Fortinet application names (1425)
- Support Nokia (formerly 'Alcatel-Lucent') IPFIX (1735)
- Support for Gigamon Application Intelligence (1832)

Fixes

- Schedule emails will now use the theme from Admin > Settings > System Preferences (185)
- The ability to use an auth token with any URL (308)
- UTF8 issue with Japanese characters in email alert notifications (636)
- 'Truncate map labels' was grabbing an extra character sometimes (700)
- Addressed an issue with flow class sequence numbers with distributed upgrades (753)
- Removed admin restriction on running group level reports (841)
- Clarify several log error messages, and reduce their volume (846)
- Some Scrutinizer custom gadgets break the ability to add any gadget for all users (900)
- AMI: set partitions doesn't remount pg_stat_tmp as a RAM drive (1066)
- Issue where deleted exporters may not be cleared out of LED stats table (1079)
- Issue where system updates could revert a setting causing "Panic: Can't find temp dir" errors and the interface failing to load (1082)
- Higher default timeouts for collect asa_acl task (1085)
- Issue with special characters in PRTG integration (1117)
- Warnings when an exporter sends the same multiplier data two different ways as long as what it sends is consistent (1120)
- UNION SELECT errors in migrator (1132)
- Autofilling IP on host search from report tables (1140)
- Scheduled reports last sent time used incorrect (1142)
- SQL GROUP BY ERROR in the collector log (1145)
- Issue with Auto SNMP Update not disabling all SNMP calls (1158)
- PostgreSQL logs using too much disk space (1209)

- Special characters in notification profile breaks threshold's 'save & edit policy' option (1229)
- Added stray columnar file check and alarm policy (1231)
- Monitor association of /var/db/fast and RAM spools (1239)
- Issue with running yum update on AWS EC2 instances (1249)
- Issue with load time of Admin > Host names view (1272)
- Defined application changes now realized on distributed collectors w/o a collector restarts (1297)
- Issue with alarm details and FQDN data for clusters using DB encryption (1314)
- DB disk usage stats did not always expire on distributed installs (1322)
- Collect support files includes the PostgreSQL log (1385)
- Allow snmpSystem details longer than 255 characters (1392)
- Errors from set tuning when two changes require a collector restart (1422)
- Getting Internal Server Error (500) when trying to access Maps > CrossCheck and Service Level Reports (1431)
- Some administrative changes for authentication did not generate audit events (1440)
- Addressed issue with ASA ACL collection when the reporter can not communicate with all firewalls (1447)
- - Issue with LDAP/TACACS usernames being case sensitive (1458)
- LDAP authentication was not failing over to try other servers (1489)
- Backup method documentation on docs.plixer.com (1506)
- Advanced TCP flag filters using strings would generate log noise (1527)
- Improved performance of Persistent Flow Risk algorithm (1536)
- Developer tasks_view hours filter causes Internal Server Error (500) (1542)
- Dashboards with multiple saved report gadgets cause oops errors (1544)
- Reporting across migrated data and new data doesn't use the migrated totals tables (1553)
- Migrated totals tables have the wrong scrut_templateid (1556)
- Peak values being less then the total values in the volume -> traffic volume reports (1588)
- Some English values in foreign language themes were out of date (1599)
- New reparser performance (1632)
- Migration from 16.3 mysql to 18.14 removed dashboard gadget permissions (1663)
- LDAP group checking was using sAMAccountName instead of the value specified in the configuration page (1668)
- Map object icons change colors based on polling availability (1691)
- The default group was not being set correctly for new users (1731)
- Payload size preventing CSV rendering of reports (1733)
- Saved reports belonging to users that no longer exist would not show up in report folders (1789)

NOTE: (1458)*

User accounts are no longer case sensitive when being checked on login. If multiple user accounts existed in Scrutinizer prior to the upgrade which were identical except for case, the excess accounts should be deleted from the interface.

Scrutinizer v18.14 - May 2019

Changelog

New features

- Now including cstore table conversion script in utils (873)
- Improved default work_mem settings (951)

Fixes

- DB process needs priority over other processes when system runs out of memory (640)
- Acknowledging Multiple Pages of an Alarm, acknowledges all alarms (676)
- ‘unhandled multicast message’ in the collector log (714)
- Report Designer not saving added row (778)
- Drilling into Palo Alto User Report generates a blank pop up (780)
- Top Interfaces summarization timing out with high interface count (784)
- Issue when upgrading from version 16.7 (790)
- Issue where exporters sending bad timestamps would freeze spool file processing (793)
- “Save password” error when navigating from group membership (832)
- Large number of DrDOS violations could crash process (849)
- Error when changing exporter status (850)
- Backup exporters count against licensing even if same IP is already active (851)
- Interface thresholds would only violate if there was both inbound and outbound traffic (872)
- IP group detection not working for v6 addresses (894)
- Cleanup logging for sFlow exports from Cumulus Router (895)
- Not all interface names are collected from FireSIGHT (896)
- Issue with business hours ending at midnight (903)
- First time LDAP authentication would fail if local authentication is disabled (904)
- Scheduled reports attaching wrong pdf to email (956)
- Drilling in on an interval from volume reports could display the wrong timeframe (963)
- A slow connection could impact API latency LED for other collectors (971)
- Issue with NTP daemon not starting automatically on some installs (990)
- Updated DRDOS thresholds to be ratios instead of fixed packet counts (1004)
- TACACS authentication would work if disabled but configured (1009)
- Issue with the scale APM outbound jitter was displayed in (1019)
- Reparser could not connect to the DB with a space in the password (1063)
- One exporter not collecting when at maximum license count for exporters (1130)

Scrutinizer v18.12.14 - January 2019

Changelog

New features

- Realtime DDOS and DRDOS detection before data is written to disk (10)
- FQDN reports are back and better performing (87)
- Interface threshold checks are now done once a minute and check one minute of data (105)
- FireSIGHT integration includes username support (111)
- FireSIGHT integration includes interface names (112)
- Group reports now include members of child groups (274)
- “User Accounts” permission to allow restriction of Scrutinizer user account creation (299)
- Added option to disable CrossCheck threshold notifications (447)

Fixes

- Faster report CSV generation (132)
- FireSIGHT integration detects connection loss and attempts to reconnect to FirePOWER (167)
- Top interfaces values were understated for sFlow exporters sending multiple totals flows per minute (177)
- PostgreSQL log rotation (263)
- Rate values for Trend reports are now based on graph interval (267)
- Link Back Host set to the wrong port on a deployed AMI (301)
- Installer no longer displays post install script errors (319)
- Add Audit messages when connections to LDAP servers fail (26415)
- Fixed username filtering when name is based on IPv6 address (26768)
- Faster Defined Application tagging (26874)

Scrutinizer v18.9 - September 2018

Changelog

Fixes

- Fixed issue with multiple defined applications on the same IP (26874)
- Improved contrast for some icons in dark themes (26511)
- System user was counting against licensing limits (26536)
- Fixed issue with top N gadgets and exporters only sending egress flows (26550)
- Fixed the Analytics Violation Overview link on the Alarms tab (26557)
- Fixed issue using Gmail to send emails (26579)
- Fixed issue with emailing table views (26587)
- Fixed issue with TopN subnets gadget and SAF aggregation (26600)
- Fixed issue with editing designed reports (26602)

- Backslash in LDAP passwords caused issue on upgrade (26613)
- Fixed issue with map labels in dashboards (26619)
- Multiple subnet filters issue in MySQL (26629)
- Fixed issue with threshold details not being cleared out when switching reports (26632)
- Fixed issue editing designed reports with some manufactured columns in them (26650)
- Fixed issue with interface permissions in mapping (26652)
- Fixed issue with row limiting in CSV files (26655)
- Fixed issue with flow vitals when packets contain multiple flow sets for the same template (26699)
- Reporting: Top 10 rows on any page are now color coded as the graph (26731)
- Postgres installs - improved reporting temp table performance (26735)

Scrutinizer v18.7 - July 2018

Changelog

New features

- Added QRadar Integration (23542)
- Changed dashboard gadget behavior to improve usability and clearly display gadget titles (26194)
- Numerous improvements to the manual (26310)

Fixes

- Flickering issue with report graphs when loading a report (24546)
- Formatting issues in Maps Tab alerts (25156)
- Double tooltip when mousing over report graph (25504)
- Audits from IPv6 hosts are now correctly received and recorded (26042)
- Issues with input parameters for the Users API (26298)
- Optimized rollups (26317)
- Decreased time necessary to run upgrades (26318)
- Links from alarms heatmap were not working (26342)
- Tuning would too aggressively set roller memory (26345)
- Addressed upgrade issue related to DB locking (26350)
- Improved dashboard gadget behavior based on customer feedback (26358)
- Reparser: Fix understatement of NetFlow v9 flow volume in vitals report (26360)
- AWS instances would not upgrade if on Postgres 9.5 (26370)
- Maps couldn't be saved in dashboard gadgets (26371)
- Could not generate PDFs of reports in Japanese (26372)
- Fixed issue with Japanese characters in emailed reports (26373)
- Other Options > Search link not working (26395)
- Peaks in totals tables were 5 minute byte counts rather than 1 minute byte counts (26399)

- Forensic filters were not forcing change to forensic data (26406)
- Fixed filtering on AS number under Admin > Definitions > Autonomous Systems (26431)
- Fixed issue with making dashboards visible to a user group (26451)

* This is the last supported release for the CentOS 6 and MariaDB platforms

Scrutinizer v18.6 - June 2018

Changelog

New features

- Test button for LDAP/RADIUS/TACACS setup (9911)
- Ability to acknowledge alarms with any combination of filters (15154)
- scrut_util command to disable ping for devices that have not responded (16826)
- Manufactured columns can be included in the report designer (17589)
- Full back button support (18291)
- Automatically detect which SNMP credentials to use for exporters (19981)
- Ability to manage interface details via API (20068)
- Ability to filter on a port range (21522)
- All interface reports now account for metering on each interface in the report (21744)
- Host -> AS -> Host reports for additional BGP reporting (21770)
- Major release upgrade to PostgreSQL 9.6 and 10 (22220)
- scrut_util command to enable/disable ipv6 (22773)
- User can be locked out after n failed login attempts (23267)
- Full foreign datastore support in collection and rollups (23478)
- Ability to exclude domain names from flow analytics (23924)
- Ability to edit URLs for custom gadgets (24134)
- Milliseconds now included with formatted timestamps where applicable (24164)
- Columnar store support for AWS Scrutinizers (24297)
- Ability to customize the login page (24452)
- Improved support for configuration of multiple LDAP servers and domains (24600)
- Ability to grant dashboards to other users / groups (24661)
- Default PostgreSQL datastore is columnar. Better disk space utilization and IO performance. (24781)
- Performance improvements for flow class lookups (24948)
- Support IPv4-mapped IPv6 addresses in subnet and ipgroup filters (PostgreSQL) (25077)
- Report IP Group with protocol and defined applications (25216)
- Support for Flowmon probe elements (25289)
- DrDoS detection for memcached and CLDAP attacks (25396)
- Ability to schedule operating system updates (26187)

Fixes

- Flow metrics vitals times now align with ingestion time (12972)
- Ungrouped now visible by non-admin users (22530)
- Tidy up loose ends when deleting exporters. Deleted exporters will stay deleted. (22588)
- Stop showing disabled exporters in the exporters LED (22654)
- Some timezones were duplicated in the selector (24107)
- Latency reports per exporter (24115)
- Addressed issue reporting on multiple interfaces with different metering configured (24659)
- Issue with generating PDF with device group filters (24703)
- Restrict PaloAlto username collection to only internal IPs (24790)
- Donut/Pie Graph not available in Top -> Interfaces report (24875)
- Map interface utilization arrows always pointed in the same direction (24893)
- 'cancel report' button truly cancels backend reporting requests. (24899)
- Device menu in Google maps (24993)
- Cleaned up log noise from Cisco ISE data collection (25027)
- Scheduled reports font issue on AWS (25111)
- Remove memcached external exposure CVE-2017-9951 (25317)
- FlowPro APM jitter report (25323)
- Audit report times now display as clients timezone (25399)
- Addressed CVE-2014-8109 (25419)
- Issue with Queue Drops >> Queue Drops By Hierarchy (25660)

Scrutinizer v17.11 - November 2017

Changelog

New features

- Support for Oracle cloud (24685)

Fixes

- Vitals errors when a user with a long UID is created (24500)
- Save button for filters would go away if field was selected, but not changed (24560)
- Localhost Unlicensed after upgrade to 17.10 (24586)
- Collector appears down after Daylight Savings Time change (24616)
- Potential short gap in rollups after collector restart (24647)

4.6.3.7 Third-party attributions

Certain open source or other third-party software components are integrated and/or redistributed with Scrutinizer software and Plixer Machine Learning software. The licenses are reproduced here in accordance with their licensing terms.

These terms only apply to the libraries themselves, not Scrutinizer software and/or Machine Learning software.

Scrutinizer

Apache 2.0 License

- **Apache Giraph** (http://giraph.apache.org/) - Copyright (c) 2011-2016, The Apache Software Foundation
- **Apache Kafka** (http://kafka.apache.org/) - Copyright (c) 2016 The Apache Software Foundation
- **Bean Validation** (http://beanvalidation.org/) - Copyright (c) 2007-2013 Red Hat, Inc.
- **code-prettify** (https://github.com/google/code-prettify) - Copyright (c) 2006 Google Inc.
- **cstore_fdw** (https://github.com/citusdata/cstore_fdw) - Copyright (c) 2016-2017 Citus Data, Inc.
- **Explorer Canvas** (https://github.com/arv/ExplorerCanvas) - Copyright (c) 2006 Google Inc.
- **fonts** (http://code.google.com/p/fonts) - Copyright (c) 2009 Google Inc.
- **Guava** (https://github.com/google/guava) - Copyright (c) Google, Inc.
- **hogan.js** (https://github.com/twitter/hogan.js) - Copyright (c) 2011 Twitter, Inc.
- **Jackson JSON Processor** (https://github.com/FasterXML/jackson) - Copyright (c) Jackson Project
- **Javassist** (https://github.com/jboss-javassist/javassist) - Copyright (c) 1999-2013 Shigeru Chiba All Rights Reserved
- **Javax Inject** (http://code.google.com/p/atinject) - Copyright (c) 2010-2015 Oracle and/or its affiliates
- **Jetty** (https://github.com/eclipse/jetty.project) - Copyright (c) 2008-2016 Mort Bay Consulting Pty. Ltd.; Copyright (c) 1996 Aki Yoshida, modified April 2001 by Iris Van den Broeke, Daniel Deville
- **Keyczar** (http://code.google.com/p/keyczar/) - Copyright (c) 2008 Google Inc.
- **Log4j** (http://logging.apache.org/log4j/) - Copyright (c) 2007 The Apache Software Foundation
- **LZ4 Java** (https://github.com/jpountz/lz4-java) - Copyright (c) 2001-2004 Unicode, Inc.
- **RocksDB** (http://rocksdb.org/) - deflate 1.2.8 Copyright (c) 1995-2013 Jean-loup Gailly and Mark Adler, inflate 1.2.8 Copyright (c) 1995-2013 Mark Adler
- **Snappy for Java** (https://github.com/xerial/snappy-java) - Copyright (c) 2011 Taro L. Saito
- **WenQuanYi Micro Hei fonts** (https://github.com/anthonyfok/fonts-wqy-microhei) - Copyright (c) 2005-2010 WenQuanYi Board of Trustees
- **ZkClient** (https://github.com/sgroschupf/zkclient) - Copyright (c) 2009 Stefan Groschupf
- **ZooKeeper** (https://zookeeper.apache.org) - Copyright (c) 2009-2014 The Apache Software Foundation

Artistic 1.0 License

- **business–isbn** (https://github.com/briandfoy/business-isbn/) - Copyright (c) 2001-2013, Brian D Foy
- **Common-Sense** (http://search.cpan.org/~mlehmann/common-sense/) - Terms of Perl - No Copyright Author - Marc Lehmann
- **Compress-Raw-Zlib** (http://search.cpan.org/~pmqs/Compress-Raw-Zlib/) - Copyright (c) 2005-2009 Paul Marquess

- **Compress-Zlib** (http://search.cpan.org/~pmqs/IO-Compress-2.066/lib/Compress/Zlib.pm) - Copyright (c) 1995-2009 Paul Marquess
- **crypt-ssleay** (https://github.com/gisle/crypt-ssleay/) - Copyright (c) 2006-2007 David Landgren; Copyright (c) 1999-2003 Joshua Chamas; Copyright (c) 1998 Gisle Aas; Copyright (c) 2010-2012 A. Sinan Unur
- **DBD-mysql** (http://search.cpan.org/dist/DBD-mysql/) - Large Portions Copyright (c) 2004-2013 Patrick Galbraith, 2004-2006 Alexey Stroganov, 2003-2005 Rudolf Lippan, 1997-2003 Jochen Wiedmann, with code portions Copyright (c) 1994-1997, their original authors
- **Digest-MD5** (http://search.cpan.org/dist/Digest-MD5/) - Copyright (c) 1995-1996 Neil Winton; Copyright (c) 1990-1992 RSA Data Security, Inc.; Copyright (c) 1998-2003 Gisle Aas
- **Encode-Locale** (http://search.cpan.org/dist/Encode-Locale/) - Copyright (c) 2010 Gisle Aas
- **ExtUtils-MakeMaker** (http://search.cpan.org/~bingos/ExtUtils-MakeMaker/) - Terms of Perl - No Copyright
- **extutils-parsexs** (https://github.com/dagolden/extutils-parsexs/) - Copyright (c) 2002-2009 by Ken Williams, David Golden and other contributors
- **HTML::Template::Pro** (http://search.cpan.org/~viy/HTML-Template-Pro-0.9510/) - Copyright (c) 2005-2009 by I. Yu. Vlasenko; copyright (c) 2000-2002 Sam Tregar
- **HTML-Parser** (http://search.cpan.org/dist/HTML-Parser/) - Copyright (c) 1995-2009 Gisle Aas; Copyright (c) 1999-2000 Michael A. Chase
- **HTML-Tagset** (http://search.cpan.org/~petdance/HTML-Tagset/) - Copyright (c) 1995-2000 Gisle Aas; Copyright (c) 2000-2005 Sean M. Burke; Copyright (c) 2005-2008 Andy Lester
- **HTTP::Cookies** (http://search.cpan.org/~oalders/HTTP-Cookies-6.04/lib/HTTP/Cookies.pm) - Copyright (c) 1997-2002 Gisle Aas; Copyright (c) 2002 Johnny Lee
- **HTTP::Daemon** (http://search.cpan.org/~gaas/HTTP-Daemon-6.01/lib/HTTP/Daemon.pm) - Copyright (c) 1996-2003 Gisle Aas
- **HTTP::Date** (http://search.cpan.org/~gaas/HTTP-Date-6.02/lib/HTTP/Date.pm) - Copyright (c) 1995-1999 Gisle Aas
- **HTTP::Negotiate** (http://search.cpan.org/~gaas/HTTP-Negotiate-6.01/lib/HTTP/Negotiate.pm) - Copyright (c) 1996, 2001 Gisle Aas
- **http-message** (https://github.com/php-fig/http-message) - Copyright 1995-2008 Gisle Aas
- **IO-Compress** (http://search.cpan.org/dist/IO-Compress/) - Copyright (c) 2005-2009 Paul Marquess
- **IO-HTML** (http://search.cpan.org/~cjm/IO-HTML-1.001/lib/IO/HTML.pm) - Copyright (c) 2012-2013 Christopher J. Madsen
- **IO-Socket-IP** (http://search.cpan.org/~pevans/IO-Socket-IP-0.37/lib/IO/Socket/IP.pm) - Copyright (c) 2010-2013 Paul Evans
- **IO-Socket-SSL** (http://search.cpan.org/~sullr/IO-Socket-SSL/) - Copyright (c) 1999-2002 Marko Asplund; Copyright (c) 2002-2005 Peter Behroozi; Copyright (C) 2006-2014 Steffen Ullrich
- **JSON** (http://search.cpan.org/~makamaka/JSON/) - Copyright (c) 2005-2013 by Makamaka Hannyaharamitu
- **JSON::XS** (http://search.cpan.org/~mlehmann/JSON-XS/) - Copyright (c) 2008 Marc Lehmann
- **libwww-perl** (http://search.cpan.org/dist/libwww-perl/) - Copyright (c) 1995-2009 Gisle Aas, 1995 Martijn Koster, 2002 James Tillman, 1998-2004 Graham Barr, 2012 Peter Marschall
- **libxml-perl** (http://perl-xml.sourceforge.net/libxml-perl/) - Copyright (c) 2001-2003 AxKit.com Ltd., 2002-2006 Christian Glahn, 2006-2009 Petr Pajas

- **Log::Log4perl** (http://search.cpan.org/~mschilli/Log-Log4perl/) - Copyright (c) 2002-2013 Mike Schilli and Kevin Goess
- **LWP::MediaTypes** (http://search.cpan.org/~gaas/LWP-MediaTypes-6.02/lib/LWP/MediaTypes.pm) - Copyright (c) 1995-1999 Gisle Aas
- **Net::Flow** (http://search.cpan.org/~acferen/Net-Flow-1.003/lib/Net/Flow.pm) - Copyright (c) 2007-2008 NTT Information Sharing Platform Laboratories
- **Net-HTTP** (http://search.cpan.org/~oalders/Net-HTTP-6.17/lib/Net/HTTP.pm) - Copyright (c) 2001-2003 Gisle Aas
- **Net-LibIDN** (http://search.cpan.org/~thor/Net-LibIDN/_LibIDN.pm) - Copyright (c) 2003-2009, Thomas Jacob
- **Net-SNMP Perl** (http://search.cpan.org/~dtown/Net-SNMP-v6.0.1/) - Copyright (c) 2001-2009 David M. Town
- **Net-SSLeay** (http://search.cpan.org/~mikem/Net-SSLeay/) - Copyright (c) 1996-2003 Sampo Kellomaki; Copyright (C) 2005-2006 Florian Ragwitz; Copyright (c) 2005 Mike McCauley
- **Perl** (http://www.perl.org) - Copyright (c) 1993-2005, by Larry Wall and others
- **Perl Object Environment** (http://search.cpan.org/~rcaputo/POE-1.367/lib/POE.pm) - Copyright (c) 1998-2013 Rocco Caputo
- **perl-digest-sha1** (http://search.cpan.org/~gaas/Digest-SHA1-2.13/SHA1.pm) - Copyright (c) 2003-2008 Mark Shelor
- **perl-File-Listing** (https://centos.pkgs.org/7/centos-x86_64/perl-File-Listing-6.04-7.el7.noarch.rpm.html) - Copyright (c) 1996-2010, Gisle Aas
- **perl-ldap** (http://ldap.perl.org) - Copyright (c) 1997-2004 Graham Barr
- **perl-REST-Client** (https://centos.pkgs.org/6/epel-i386/perl-REST-Client-272-1.el6.noarch.rpm.html) - Copyright (c) 2008-2010 by Miles Crawford
- **perl-XML-NamespaceSupport** (http://search.cpan.org/~perigrin/XML-NamespaceSupport-1.11/lib/XML/NamespaceSupport.pm) - Copyright (c) 2001-2005 Robin Berjon
- **Pod-Escapes** (http://search.cpan.org/~neilb/Pod-Escapes/) - Copyright (c) 2001-2004 Sean M. Burke
- **Pod-Simple** (http://search.cpan.org/~dwheeler/Pod-Simple-3.26/lib/Pod/Simple.pod) - Copyright (c) 2002 Sean M. Burke
- **TimeDate** (http://search.cpan.org/dist/TimeDate/) - Copyright (c) 1995-2009 Graham Barr
- **Types::Serialiser** (http://search.cpan.org/~mlehmann/Types-Serialiser-1.0/Serialiser.pm) - Terms of Perl - No Copyright Author - Marc Lehmann
- **URI** (http://search.cpan.org/~ether/URI/) - Copyright (c) 1998 Graham Barr, 1998-2009 Gisle Aas
- **WWW-RobotRules** (http://search.cpan.org/~gaas/WWW-RobotRules-6.02/lib/WWW/RobotRules.pm) - Copyright (c) 1995, Martijn Koster, 1995-2009, Gisle Aas
- **XML-LibXML** (http://search.cpan.org/~shlomif/XML-LibXML/) - Copyright (c) 2001-2003 AxKit.com Ltd., 2002-2006 Christian Glahn, 2006-2009 Petr Pajas
- **XML-SAX** (http://search.cpan.org/~grantm/XML-SAX/) - No Copyright listed - Terms of Perl
- **Xml-sax-base** (http://search.cpan.org/~grantm/XML-SAX-Base-1.08/BuildSAXBase.pl) - No Copyright listed - Terms of Perl
- **yaml-perl-pm** (http://search.cpan.org/dist/YAML-Perl/) - Copyright (c) 2001, 2002, 2005. Brian Ingerson; Copyright (c) 2005, 2006, 2008. Ingy döt Net; Some parts Copyright (c) 2009 Adam Kennedy

Artistic 2.0 License

- **NetPacket::** (http://search.cpan.org/~cganesan/NetPacket-LLC-0.01/) - Copyright (c) 2001 Tim Potter and Stephanie Wehner; Copyright (c) 1995-1999 ANU and CSIRO on behalf of theparticipants in the CRC for Avanced Computational Systems ('ACSys')

BSD 2-Clause Simplified License

- **JabberWerxC** (https://github.com/cisco/JabberWerxC) - Copyright (c) 2010-2013 Cisco Systems, Inc.

BSD 3-Clause License

- **Babel** (http://babel.pocoo.org/) - Copyright (c) 2007-2008 Edgewall Software
- **Crypt-DES** (http://search.cpan.org/~dparis/Crypt-DES/) - Copyright (c) 1995, 1996 Systemics Ltd, Modifications are Copyright (c) 2000, W3Works, LLC
- **D3.js** (http://d3js.org/) - Copyright (c) 2010-2014 2010-2017 Mike Bostoc
- **Jinja2** (http://jinja.pocoo.org/) - Copyright (c) 2008-2011 Armin Ronacher; Copyright 2007-2011 by the Sphinx team, 2006-2010 the Jinja Team; Copyright 2010, John Resig; Copyright 2010, The Dojo Foundation
- **libevent** (http://libevent.org/) - Copyright (c) 2000-2007 Niels Provos; Copyright (c) 2007-2012 Niels Provos and Nick Mathewson
- **MarkupSafe** (http://github.com/mitsuhiko/markupsafe) - Copyright (c) 2010 by Armin Ronacher
- **memcached** (http://code.google.com/p/memcached/) - Copyright (c) 2000-2003 Niels Provos; Copyright (c) 2003, Danga Interactive, Inc.
- **Netcast** (http://freshmeat.sourceforge.net/projects/netcast) - Copyright (c) Stanislaw Pasko
- **Net-SNMP** (http://www.net-snmp.org/) - Copyright: See licenses/net-snmp.txt
- **PhantomJS** (http://phantomjs.org/) - Copyright (c) 2011 Ariya Hidayat
- **pyasn1** (http://sourceforge.net/projects/pyasn1/) - Copyright (c) 2005-2017, Ilya Etingof
- **RequireJS** (http://requirejs.org/) - Copyright (c) 2010-2012, The Dojo Foundation
- **Scala** (http://www.scala-lang.org/) - Copyright (c) 2002-2010 EPFL, Lausanne, unless otherwise specified
- **SNMP::Info** (http://freshmeat.net/projects/snmp-info) - Copyright (c) 2002-2003, Regents of the University of California; Copyright (c) 2003-2010 Max Baker and SNMP::Info Developers
- **strace** (http://sourceforge.net/projects/strace/) - Copyright (c) 1991, 1992 Paul Kranenburg; Copyright (c) 1993 Branko Lankester; Copyright (c) 1993 Ulrich Pegelow; Copyright (c) 1995, 1996 Michael Elizabeth Chastain; Copyright (c) 1993, 1994, 1995, 1996 Rick Sladkey; Copyright (c) 1998-2001 Wichert Akkerman; Copyright (c) 2001-2017 The strace developers
- **sudo** (http://www.sudo.ws/sudo/) - Copyright (c) 1994-1996, 1998-2018 Todd C. Miller
- **uthash** (http://sourceforge.net/projects/uthash/) - Copyright (c) 2008-2017 Troy D. Hanson
- **Yahoo! User Interface Library** (http://developer.yahoo.com/yui) - Copyright (c) 2007, Yahoo! Inc.
- **yuicompressor** (http://developer.yahoo.com/yui/compressor/) - Copyright (c) 2013 Yahoo! Inc.

CDDL 1.0 License

- **Java Servlet API** (http://java.sun.com/products/servlet/index.jsp) - Copyright (c) 1997-2003 Oracle and/or its affiliates
- **JAX-RS Specification** (https://java.net/projects/jax-rs-spec) - Copyright (c) 1996-2014 Oracle and/or its affiliates
- **Jersey** (http://jersey.java.net/) - Copyright (c) 2010-2016 Oracle and/or its affiliates, 2000-2011 INRIA, France Telecom, 2004-2011 Eugene Kuleshov
- **jsr250-api** (https://jcp.org/aboutJava/communityprocess/final/jsr250/index.html) - Copyright (c) 1999-2013 Oracle and/or its affiliates

CDDL 1.1 License

- **HK2** (https://javaee.github.io/hk2/) - Copyright (c) 2010-2017 Oracle and/or its affiliates

CURL License

- **cURL** (http://curl.haxx.se) - Copyright (c) 1998-2013, Daniel Stenberg

GPL & MIT Licenses

- **coResizable 1.6** (http://www.bacubacu.com/colresizable/) - Copyright (c) 2012 Alvaro Prieto Lauroba
- **jQuery Accordion** (http://docs.jquery.com/UI/Accordion) - Copyright (c) 2007 Jörn Zaefferer
- **jQuery Ajaxmanager** (http://github.com/aFarkas/Ajaxmanager) - Copyright (c) 2010 Alexander Farkas
- **jQuery Autocomplete** (http://bassistance.de/jquery-plugins/jquery-plugin-autocomplete/) - Copyright (c) 2009 Jörn Zaefferer
- **jQuery blockUI** (http://malsup.com/jquery/block/) - Copyright (c) 2007-2013 M. Alsup
- **jQuery Checkboxes** (https://github.com/SamWM/jQuery-Plugins) - Copyright (c) 2006-2008 Sam Collett
- **jQuery Form** (http://malsup.com/jquery/form/) - Copyright (c) 2017 jquery-form
- **jQuery Select Boxes** (https://github.com/SamWM/jQuery-Plugins) - Copyright (c) 2006-2008 Sam Collett

GPL 2.0 License

- **CSSTidy** (http://csstidy.sourceforge.net) - Copyright (c) 2005, 2006, 2007 Florian Schmitz
- **Filesystem in Userspace** (http://fuse.sourceforge.net/) - Copyright (c) 1989, 1991 Free Software Foundation, Inc.
- **filterlist.js** (http://www.barelyfitz.com/projects/filterlist/index.php) - Copyright (c) 2003, Patrick Fitzgerald
- **Iotop** (http://freshmeat.net/projects/iotop) - Copyright (c) 2007, 2008 Guillaume Chazarain, 2007 Johannes Berg
- **jQuery Pagination** (https://github.com/gbirke/jquery_pagination) - Copyright (c) Gabriel Birke
- **libdbi-drivers** (http://freshmeat.net/projects/libdbi-drivers) - Copyright (c) 2001-2007, David Parker, Mark Tobenkin, Markus Hoenick
- **Nmap Security Scanner** (http://nmap.org/) - Copyright (c) 1996–2016 Insecure.Com LLC
- **sshpass** (http://freshmeat.net/projects/sshpass)
- **sysstat** (http://sebastien.godard.pagesperso-orange.fr/) - Copyright (c) 1999-2009 Sebastien Godard

GPL 3.0 License

- **Ansible** (http://www.ansible.com/) - Copyright (c) 2017, Ansible Project
- **MariaDB** (http://mariadb.org/) - Copyright (c) The MariaDB Foundation

LGPL 2.1 License

- **DHTMLGoodies** (http://www.dhtmlgoodies.com/index.html?page=termsOfUse) - Copyright (c) 2005-2007 Alf Magne Kalleland, www.dhtmlgoodies.com
- **Dynarch DHTML Calendar** (http://www.dynarch.com/jscal/) - Copyright (c) 2002-2005 Mihai Bazo
- **jFeed** (https://github.com/jfhovinne/jFeed) - Copyright (c) 2007-2011 Jean-François Hovinne dual mit/gpl
- **libmspack** (http://freshmeat.net/projects/libmspack) - Copyright (c) 1991, 1999, 2003-2004 Stuart Caie
- **Open Virtual Machine Tools** (http://open-vm-tools.sourceforge.net) - Copyright (c) 2010-2015 VMware, Inc. All rights reserved
- **paramiko** (https://github.com/paramiko/paramiko/) - Copyright (c) 2003-2009 Robey Pointer
- **whatever_hover** (https://github.com/jasoncheow/whatever_hover/) - Copyright (c) 2005 - Peter Nederlof

LGPL 3.0 License

- **GNU Libidn** (http://www.gnu.org/software/libidn/) - Copyright (c) 2004-2012 Simon Josefsson

MIT License

- **Argparse4j** (http://argparse4j.sourceforge.net/) - Copyright (c) 2011, 2015, Tatsuhiro Tsujikawa
- **Backbone.js** (https://github.com/jashkenas/backbone) - Copyright (c) 2010-2017 Jeremy Ashkenas, DocumentCloud Copyright (c) 2013 Charles Davison, Pow Media Ltd.
- **base2** (http://code.google.com/p/base2/) - copyright (c) 2007-2009, Dean Edwards
- **c3.js** (http://c3js.org/) - Copyright (c) 2013 Masayuki Tanaka
- **Cocktail.js** (https://github.com/onsi/cocktail) - Copyright (c) 2012 Onsi Fakhouri
- **d3pie.js** (http://d3pie.org/) - Copyright (c) 2014-2015 Benjamin Keen
- **dshistory.js** (http://code.google.com/p/dshistory/) - Copyright (c) Andrew Mattie
- **Expat** (http://expat.sourceforge.net) - Copyright (c) 1998, 1999, 2000 Thai Open Source Software Center Ltd.
- **Flotr2** (https://github.com/HumbleSoftware/Flotr2) - Copyright (c) 2012 Carl Sutherland
- **gridstack.js** (http://troolee.github.io/gridstack.js/) - Copyright (c) 2014-2016 Pavel Reznikov, Dylan Weiss
- **hoverIntent** (http://cherne.net/brian/resources/jquery.hoverIntent.html) - Copyright (c) 2011 Brian Cherne
- **httplib2** (https://github.com/jcgregorio/httplib2) - Copyright (c) 2006 by Joe Gregorios, Thomas Broyer, James Antills, Xavier Verges Farreros, Jonathan Feinbergs, Blair Zajacs, Sam Rubys, Louis Nyffeneggert, Dan-Haim, 2007 Google Inc.
- **JOpt Simple** (http://jopt-simple.sourceforge.net/) - Copyright (c) 2004-2015 Paul R. Holser, Jr.
- **jQuery** (http://jquery.com/) - Copyright (c) 2007-2011, John Resig
- **jQuery Fixed Header Table** (http://fixedheadertable.com) - Copyright (c) 2013 Mark Malek
- **jQuery Form Plugin** (https://github.com/malsup/form) - Copyright (c) Mike Alsup

- **jQuery Live Query** (https://github.com/brandonaaron/livequery) - Copyright (c) 2010 Brandon Aaron
- **jQuery Migrate** (https://plugins.jquery.com/migrate/) - Copyright (c) jQuery Foundation and other contributors
- **jQuery Plugin: Superfish** (https://superfish.joelbirch.co/) - Copyright (c) 2008 Joel Birch
- **jQuery Plugin: tablesorter** (http://tablesorter.com/docs/) - Copyright (c) 2014 Christian Bach
- **JQuery Plugin: Treeview** (http://bassistance.de/jquery-plugins/jquery-plugin-treeview/) - Copyright (c) 2007 Jörn Zaefferer
- **jQuery qtip.js** (http://craigsworks.com/projects/qtip/) - Copyright (c) 2009 Craig Thompson
- **jQuery UI** (http://jqueryui.com/) - Copyright (c) 2014, 2015 jQuery Foundation and other contributors
- **JQuery Validation Plugin** (http://bassistance.de/jquery-plugins/jquery-plugin-validation/) - Copyright (c) Jörn Zaefferer
- **jQuery-metadata** (https://github.com/jquery-orphans/jquery-metadata) - Copyright (c) 2001-2010. Matteo Bicocchi (Pupunzi)
- **jQuery-mousewheel** (https://github.com/brandonaaron/jquery-mousewheel) - Copyright (c) 2011 Brandon Aaron
- **Logalot** (https://www.npmjs.com/package/logalot) - Copyright (c) Kevin Mårtensson
- **Moment Timezone** (http://momentjs.com/timezone/) - Copyright (c) JS Foundation and other contributors
- **Moment.js** (http://momentjs.com/) - Copyright (c) JS Foundation and other contributors
- **pbox.js** (http://www.ibegin.com/labs/)
- **Python Six** (https://pypi.python.org/pypi/six/) - Copyright (c) 2010-2015 Benjamin Peterson are therefore Copyright (c) 2001, 2002, 2003 Python Software Foundation
- **PyYAML** (http://pyyaml.org/wiki/PyYAML) - Copyright (c) 2006 Kirill Simonov
- **Raphael** (https://github.com/DmitryBaranovskiy/raphael) - Copyright (c) 2008-2013 Dmitry Baranovskiy; Copyright (c) 2008-2013 Sencha Labs
- **setuptools** (https://github.com/pypa/setuptools) - Copyright (C) 2016 Jason R Coomb
- **Simple AJAX Code-Kit** (https://github.com/abritinthebay/simpleajaxcodekit) - Copyright (c) 2005 Gregory Wild-Smith
- **simplejson** (https://github.com/simplejson/simplejson) - Copyright (c) 2008, Bob Ippolito
- **SLF4j** (http://www.slf4j.org) - Copyright (c) 2004-2017 QOS.ch
- **sqlify** (https://www.npmjs.com/package/sqlify) - Copyright (c) 2017 Vajahath Ahmed
- **Underscore JS** (http://underscorejs.org/) - Copyright (c) 2009-2015 Jeremy Ashkenas, DocumentCloud and Investigative Reporters & Editors
- **wickedpicker.js** (http://github.com/wickedRidge/wickedpicker) - Copyright (c) 2015-2016 Eric Gagnon

MIT Old Style License

- **c-ares** (http://c-ares.haxx.se/) - Copyright (c) 1998, 2009 by the Massachusetts Institute of Technology; Copyright (c) 2004-2011, Daniel Stenberg with many contributors

Mozilla Public License 1.1

- **Rhino** (https://github.com/mozilla/rhino)

OpenSSL License & SSLeay License (conjunctive)

- **OpenSSL** (http://www.openssl.org) - Copyright (c) 1998-2011 The OpenSSL Project; Copyright (C) 1995-1998 Eric Young This product includes software written by Tim Hudson; Copyright (C) 1998-2011 The OpenSSL Project

Oracle BCL License

- **Oracle Java** (http://www.oracle.com/technetwork/java/index.html) - Copyright (c) 1993-2015, Oracle and/or its affiliates

PostgreSQL License

- **PostgreSQL** (http://www.postgresql.org/) - Portions Copyright (c) 1996-2018, The PostgreSQL Global Development Group; Portions Copyright (c) 1994, The Regents of the University of California

Unicode, Inc. License Agreement

- **International Components for Unicode (ICU)** (http://www.icu-project.org/) - Copyright (c) 2010 Yahoo Inc.; Copyright (c) 1996-2012, International Business Machines Corporation and Others

Machine learning

Apache Software License

- **Cython** (https://cython.org/) - Copyright (c) Robert Bradshaw, Stefan Behnel, Dag Seljebotn, Greg Ewing, et al.
- **asyncpg** (https://github.com/MagicStack/asyncpg) - Copyright (c) MagicStack Inc
- **python-dateutil** (https://github.com/dateutil) - Copyright (c) Gustavo Niemeyer
- **requests** (https://docs.python-requests.org/en/latest/) - Copyright (c) MMXVIX. A Kenneth Reitz Project. Kenneth Reitz

BSD License

- **idna** (https://github.com/kjd/idna) - Copyright (c) Kim Davies
- **joblib** (https://joblib.readthedocs.io/en/latest/) - Copyright (c) Gael Varoquaux
- **numpy** (https://numpy.org/) - Copyright (c) 2021 NumPy. All rights reserved. Travis E. Oliphant et al.
- **pandas** (https://pandas.pydata.org/) - No Copyright listed
- **patsy** (https://github.com/pydata/patsy) - Copyright (c) Nathaniel J. Smith
- **scikit-learn** (https://scikit-learn.org/stable/) - No Copyright listed
- **scipy** (https://scipy.org/) - Copyright (c) 2021 SciPy.
- **statsmodels** (https://www.statsmodels.org) - Copyright (c) 2009-2019, Josef Perktold Skipper Seabold, Jonathan Taylor, statsmodels-developers

LGPL GNU License

- **chardet** (https://github.com/chardet/chardet) - Copyright (c) Daniel Blanchard

MIT License

- **pmdarima** (http://alkaline-ml.com/pmdarima/) - Copyright (c) 2017-2021, Taylor G Smith
- **pytz** (https://github.com/stub42/pytz) - Copyright (c) Stuart Bishop
- **six** (https://github.com/benjaminp/six/tree/65486e4383f9f411da95937451205d3c7b61b9e1) - Copyright (c) Benjamin Peterson
- **urllib3** (https://github.com/urllib3/urllib3) - Copyright (c) Andrey Petrov

Mozilla Public License 2.0

- **certifi** (https://certifi.io/en/latest/) - Copyright (c) 2020 Kenneth Reitz