Deploy Software Agents on Workloads

A Secure Workload software agent is a lightweight piece of software that you install on your workloads. The purpose of the agent is to:

- Collect host information such as network interfaces and active processes running in the system.
- Monitor and collect network flow information.
- Enforce security policies by setting firewall rules for hosts on which the software agent is installed and enabled.

Agents automatically update the Secure Workload inventory when interface addresses change. You do not need to install agents on end-user (employee) computers.

Deploy Software Agents

Note Installer scripts downloaded from LDAP or AD accounts with automatic role mapping fail once you are logged out. To give the installer scripts uninterrupted access to the cluster, enable Use Local Authentication.

On deployment, the agent is assigned a unique identity by the Secure Workload cluster based on a set of parameters specific to the host where the agent is running. If the host name and the BIOS UUID are a part of the set of parameters, you may encounter the following issues:

1. Registration failure when cloning a virtual machine and retaining the BIOS UUID and host name, and when instant cloning a VDI. The registration failure happens because Secure Workload already has a registered software agent using the same parameters set. You can delete the registered agent using OpenAPI. In some cases, a duplicate BIOS UUID configured during startup is changed by VMware after a certain period of time. Agent registration recovers once the Cisco Secure Workload services are restarted.
2. A new identity is generated for the agent if the host name is changed and the host rebooted. The redundant or the old agent is marked as inactive after a certain period of time. For more information, see Frequently Asked Questions section.

Supported Platforms and Requirements

For information on supported platforms and additional requirements for software agents, see:

- The release notes for your release, see Release Notes.
- The agent install wizard in the Secure Workload web portal: In the navigation menu, click **Manage** > **Workloads** > **Agents**, then click the **Installer** tab. Choose an installation method, a platform, and if applicable, an agent type to see supported platform versions.
- Support Matrix for additional dependencies.
- The following sections for details on additional requirements for each platform and agent type.

Installing Linux Agents for Deep Visibility and Enforcement

Requirements and Prerequisites to Install Linux Agents

- See Supported Platforms and Requirements.
- Root privileges to install and execute the services.
- 1-GB storage space for agent and log file.
- Security exclusions are configured on the security applications that are monitoring the host to prevent these applications from blocking agent installation or agent activity. For more information, see Security Exclusions.
- A special user, **tet-sensor**, is created in the host where the agent is installed. If PAM or SELinux is configured on the host, tet-sensor user must be granted appropriate privileges for executing the tet-sensor process and making connections to collectors. If an alternative install directory is provided and SELinux is configured, ensure that execution is allowed for that location.
- You must be able to use the unzip command, if the agent is installed using the AutoInstall (installer script) method.

Supported Methods to Install Linux Agents

Methods to install a Linux agent for deep visibility and enforcement:

Install Linux Agent using the Agent Image Installer Method

We recommend the automated installer script method for installing Linux agents. Use the image installer method if you have a specific reason for using this manual method..

Prerequisite:

Configure the ACTIVATION_KEY and HTTPS_PROXY in the user.cfg file for SaaS clusters and when you are installing the agent on a non-default tenant of on-premises clusters with multiple tenants. For more information, see (Manual Installations Only) Update the User Configuration File.

To install a Linux agent using the agent image method:

Procedure

Step 1 Navigate to Agent Installation Methods:
- If you are a first-time user, launch the Quick Start wizard and click **Install Agents**.
- In the navigation pane, choose **Manage** > **Agents**, and select the **Installer** tab.

Step 2 Click **Agent Image Installer**.

Step 3 In the **Platform** field, enter Linux.

Step 4 Enter the required agent type and the version of the agent, and then from the results, download the required version of the agent.

Step 5 Copy the RPM package to all the Linux hosts for deployment.

Note If the agent is already installed on the host, do not reinstall the agent. To upgrade the agent, see Upgrading Software Agents section.

Step 6 Based on your platform, run the RPM commands with root privileges.
- For RHEL/CentOS/Oracle platforms, run the command: `rpm -ivh <rpm_filename>`
- For Ubuntu platform:
  - To retrieve the dependency list and ensure all dependencies are met, run the command: `rpm -qpR <rpm_filename>`
  - Install the agent with "–nodeps" option by running the command: `rpm -ivh \\--nodeps <rpm filename>`

Install Linux Agent Using the Agent Script Installer Method

We recommend the installer script method to deploy Linux agents for deep visibility and enforcement.

Note

- The installed Linux agent supports both deep visibility and enforcement.
- By default, enforcement is disabled. To enable enforcement, see Create an Agent Configuration Profile.

To install a Linux agent using the script installer method:

Procedure

Step 1 Navigate to Agent Installation methods:

- If you are a first-time user, launch the **Quick Start Wizard** and click **Install Agents**.
- From the navigation pane, choose **Manage** > **Agents**, and select the **Installer** tab.

Step 2 Click **Agent Script Installer**.

Step 3 From the **Select Platform** drop-down list, choose **Linux**.

To view the supported Linux platforms, click **Show Supported Platforms**.

Step 4 Choose the tenant to install the agents.

Note Secure Workload SaaS clusters do not require selecting a tenant.

Step 5 If you want to assign labels to the workload, choose the label keys and enter label values.

When the installed agent reports IP addresses on the host, the installer CMDB labels selected here, along with other uploaded CMDB labels that have been assigned to IPs reported by this host, would be automatically assigned to the new IP address. If there are conflicts between uploaded CMDB labels and installer CMDB labels:

- Labels assigned to an exact IP address take precedence over labels assigned to the subnet.
- Existing labels assigned to an exact IP address take precedence over installer CMDB labels.

Step 6 If an HTTP proxy is required to communicate with Secure Workload, choose **Yes**, and then enter a valid proxy URL.

Step 7 In the **Installer expiration** section, select an option:

- No expiration: The installer script can be used multiple times.
- One time: The installer script can be used only once.
- Time bound: You can set the number of days for which the installer script can be used.
- Number of deployments: You can set the number of times the installer script can be used.

Step 8 Click **Download** and save the file to the local disk.

Step 9 Copy the installer shell script on Linux hosts and run the following command to grant execute permission to the script: `chmod u+x tetration_installer_default_sensor_linux.sh`

Note The script name may differ depending on the selected agent type and scope.

Step 10 To install the agent, run the following command with root privileges:

```
./tetration_installer_default_sensor_linux.sh
```

Note If an agent is already installed on the tenant, you cannot proceed with the installation.

We recommend running the precheck, as specified in the script usage details.

Linux installer script usage details:

```
bash tetration_linux_installer.sh [--pre-check] [--skip-pre-check=<option>] [--no-install]
 [--logfile=<filename>] [--proxy=<proxy_string>] [--no-proxy] [--help] [--version]
[--sensor-version=<version_info>] [--ls] [--file=<filename>] [--save=<filename>] [--new]
[--reinstall] [--unpriv-user] [--force-upgrade] [--upgrade-local]
[--upgrade-by-uuid=<filename>] [--basedir=<basedir>] [--logbasedir=<logbdir>]
[--tmpdir=<tmp_dir>] [--visibility] [--golden-image]
  --pre-check: run pre-check only
  --skip-pre-check=<option>: skip pre-installation check by given option; Valid options
include 'all', 'ipv6' and 'enforcement'; e.g.: '--skip-pre-check=all' will skip all
pre-installation checks; All pre-checks will be performed by default
  --no-install: will not download and install sensor package onto the system
  --logfile=<filename>: write the log to the file specified by <filename>
  --proxy=<proxy_string>: set the value of CL_HTTPS_PROXY, the string should be formatted
as http://<proxy>:<port>
  --no-proxy: bypass system wide proxy; this flag will be ignored if --proxy flag was
provided
  --help: print this usage
  --version: print current script's version
  --sensor-version=<version_info>: select sensor's version; e.g.: '--sensor-version=3.4.1.0';
 will download the latest version by default if this flag was not provided
  --ls: list all available sensor versions for your system (will not list pre-3.1 packages);
 will not download any package
  --file=<filename>: provide local zip file to install sensor instead of downloading it
from cluster
  --save=<filename>: download and save zip file as <filename>
  --new: remove any previous installed sensor; previous sensor identity has to be removed
from cluster in order for the new registration to succeed
  --reinstall: reinstall sensor and retain the same identity with cluster; this flag has
higher priority than --new
  --unpriv-user=<username>: use <username> for unpriv processes instead of tet-sensor
  --force-upgrade: force sensor upgrade to version given by --sensor-version flag; e.g.:
'--sensor-version=3.4.1.0 --force-upgrade'; apply the latest version by default if
--sensor-version flag was not provided
  --upgrade-local: trigger local sensor upgrade to version given by --sensor-version flag:
 e.g.: '--sensor-version=3.4.1.0 --upgrade-local'; apply the latest version by default if
--sensor-version flag was not provided
  --upgrade-by-uuid=<filename>: trigger sensor whose uuid is listed in <filename> upgrade
to version given by --sensor-version flag; e.g.: '--sensor-version=3.4.1.0
--upgrade-by-uuid=/usr/local/tet/sensor_id'; apply the latest version by default if
--sensor-version flag was not provided
  --basedir=<base_dir>: instead of using /usr/local use <base_dir> to install agent. The
full path will be <base_dir>/tetration
  --logbasedir=<log_base_dir>: instead of logging to /usr/local/tet/log use <log_base_dir>.
 The full path will be <log_base_dir>/tetration
  --tmpdir=<tmp_dir>: instead of using /tmp use <tmp_dir> as temp directory
  --visibility: install deep visibility agent only; --reinstall would overwrite this flag
if previous installed agent type was enforcer
  --golden-image: install Cisco Secure Workload Agent but do not start the Cisco Secure
Workload Services; use to install Cisco Secure Workload Agent on Golden Images in VDI
environment or Template VM. On VDI/VM instance created from golden image with different
host name, Cisco Secure Workload Services will work normally
```

Note

- Ubuntu uses the native .deb package, and new installations and reinstallations switch to this package type. Upgrades from previous versions continue with the .rpm package.
- Ubuntu .deb package is installed under `/opt/cisco/tetration`.
- There is no relocation support for the .deb package and so the –basedir option is not supported for Ubuntu.

Agent Support for NVIDIA Bluefield Networking Platform

A data processing unit (DPU) is a programmable processor that is designed to manage data-centric tasks, including but not limited to data transfer, power optimization, security, compression, analytics, and encryption.

The NVIDIA DPU is a smart network interface card (SmartNic) with excellent network performance. It delivers a high-speed Ethernet NIC capability and it enables the execution of software directly on the NIC itself, allowing for interception, monitoring, and manipulation of network traffic passing through the NIC.

NVIDIA facilitates the functionality through the provision of the DOCA SDK. Leveraging virtualization technology based on PCIe Single Root I/O Virtualization (SR-IOV), the DPU establishes a mechanism for virtual machines (VMs) to communicate directly without hypervisor involvement. The DPU incorporates an OpenVSwitch-based hardware-accelerated eSwitch for network control, enhancing overall efficiency.

Requirements and Prerequisites

- Ensure that Ubuntu 22.04-based DOCA is installed on the BlueField networking platform.
- Set up the DPU card network to enable an agent's connection to the cluster through one of the out-of-band interfaces. Options include oob_net0, tmfifo_net0, or the in-band connection through enp3s0f0s0.

Agent Installation

The installation follows a Linux-like process.

1. Navigate to Agent Installation Methods:
   - If you are a first-time user, launch the **Quick Start** wizard and click **Install Agents**.
   - From the navigation pane, choose **Manage** > **Workloads** > **Agents**.
2. Under the **Installer** tab, click **Agent Script Installer**.
3. From the **Select Platform** drop-down list, choose **Linux**.

   To view the supported Linux platforms, click **Show Supported Platforms**.

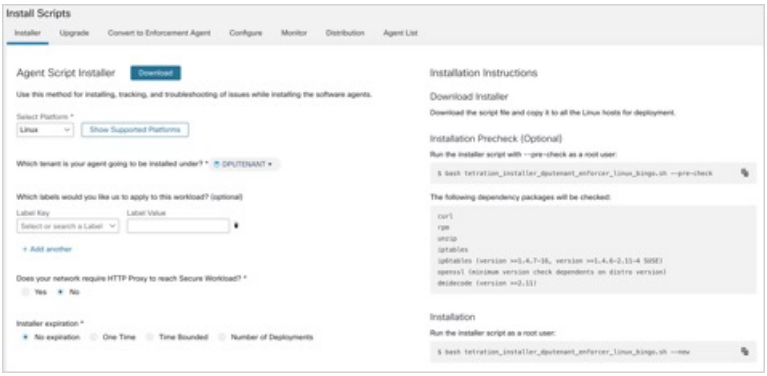
Note Secure Workload Agent is only supported on the Ubuntu 22-based DOCA SDK.

4. Choose the tenant to install the agents.

Note Selecting a tenant is not required for Secure Workload SaaS clusters.

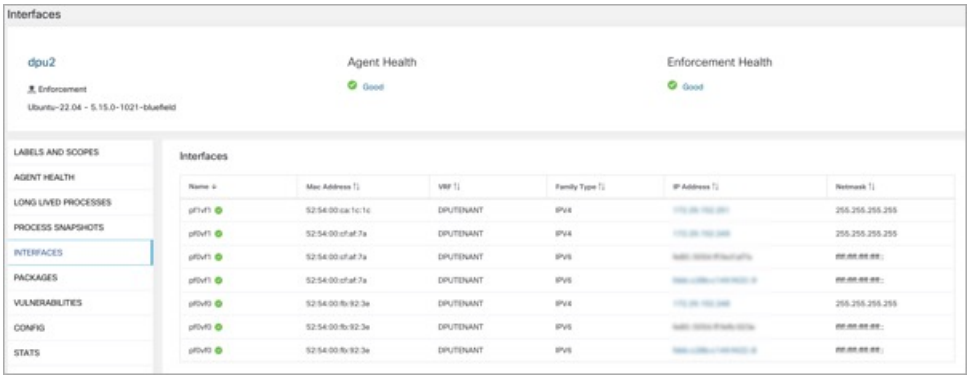
5. (Optional) If you want to assign labels to the workload, choose the label keys and enter label values.
6. If an HTTP proxy is required to communicate with Secure Workload, click **Yes**, and then enter a valid proxy.
7. In the **Installer expiration** section, select one of the available options:
   - **No expiration**: The installer script can be used multiple times.
   - **One time**: The installer script can be used only once.
   - **Time-bound**: You can set the number of days for which the installer script can be used.
   - **Number of deployments**: You can set the number of times the installer script can be used.
8. Click **Download** to download the Linux installer script on to DPU using one of the network devices.
9. Run the installer script. For more information, see Install Linux Agent using the Agent Script Installer Method.

Figure 1: Install Script



Choose **Software Agents** > **Agent List** and click a **Hostname**. Under **Interfaces**, you can view the current mapping of interfaces with the associated IP addresses.

Figure 2: Interface Mapping



Choose **Investigate** > **Traffic** to monitor the network traffic between virtual machines (VMs) when those are utilizing the SR_IOV virtual network interfaces provided by the DPU. The agent on the DPU enables the segmentation of network traffic between these virtual network interfaces.

Verify Linux Agent Installation

Procedure

Run the command sudo rpm -q tet-sensor `sudo rpm -q tet-sensor`.

```
sudo rpm -q tet-sensor
```

A single entry as output confirms that a Linux agent is installed on the host.

Sample output: `tet-sensor-3.1.1.50-1.el6.x86_64`

The specific output may differ depending on the platform and architecture.

Installing Windows Agents for Deep Visibility and Enforcement

Requirements and Prerequisites for Installing Windows Agent

- See the Supported Platforms and Requirements section.
- Administrator privileges are required for installation and service execution.
- Npcap must be installed on workloads running Windows 2008 R2 or when the installed agent version is earlier than version 3.8. If the Npcap driver is not already installed, the recommended Npcap version is installed in the background by the agent after the service starts. For more information, see the Npcap version information.
- One GB storage space for agent and log files.
- Enable the Windows services required for agent installation. Some of the Windows services could have been disabled if your Windows hosts have been security hardened, or have deviated from the default configurations. For more information, see the Required Windows Services section.

- Security exclusions configured on security applications that are monitoring the host and that could block agent installation or agent activity. For more information, see Security Exclusions.

Supported Methods to Install Windows Agents

There are two methods to install Windows agents for deep visibility and enforcement.

- Install Windows Agent using the Agent Script Installer Method, on page 9
- Install Windows Agent using the Agent Image Installer Method, on page 11

You can also install using a golden image. For more information, see Deploying Agents on a VDI Instance or VM Template (Windows).

Install Windows Agent using the Agent Script Installer Method

We recommend the script installer method to deploy Windows agents for deep visibility and enforcement.

Note

- The installed Windows agent supports both deep visibility and enforcement.
- By default, enforcement is disabled. To enable enforcement, see Create an Agent Configuration Profile, on page 59.

To install a Windows agent using the script installer method:

Procedure

Step 1 Navigate to Agent Installation Methods:

- If you are a first-time user, launch the Quick Start wizard and click **Install Agents**.
- From the navigation pane, choose **Manage** > **Agents**, and select the **Installer** tab.

Step 2 Click **Agent Script Installer**.

Step 3 From the **Select Platform** drop-down menu, choose **Windows**.

To view the supported Windows platforms, click **Show Supported Platforms**.

Step 4 Choose the tenant to install the agents.

Note Selecting a tenant is not required for Secure Workload SaaS clusters.

Step 5 If you want to assign labels to the workload, choose the label keys and enter label values.

When the installed agent reports IP addresses on the host, the installer CMDB labels selected here, along with other uploaded CMDB labels that have been assigned to IPs reported by this host, would be assigned to the new IP address. If there are conflicts between uploaded CMDB labels and installer CMDB labels:

- Labels assigned to an exact IP address take precedence over labels assigned to the subnet.
- Existing labels assigned to an exact IP address take precedence over installer CMDB labels.

Step 6 If HTTP proxy is required to communicate with Secure Workload, choose **Yes**, and then enter a valid proxy URL.

Step 7 Under the **Installer expiration** section, select one from the available options:

- No expiration: The installer script can be used multiple times.
- One time: The installer script can be used only once.
- Time bound: You can set the number of days for which the installer script can be used.
- Number of deployments: You can set the number of times the installer script can be used.

Step 8 Click **Download** and save the file to the local disk.

Step 9 Copy the installer PowerShell script to all the Windows hosts for deployment and run the script with administrative privileges.

Note
- Depending on the system settings, the command `Unblock-File` may need to be run before other commands.
- The script does not run if the agent is already installed on the tenant.

We recommend running the pre-check, as specified in the script usage details.

Windows installer script usage details:

```
# powershell -ExecutionPolicy Bypass -File tetration_windows_installer.ps1 [-preCheck]
[-skipPreCheck <Option>] [-noInstall] [-logFile <FileName>] [-proxy <ProxyString>] [-noProxy]
 [-help] [-version] [-sensorVersion <VersionInfo>] [-ls] [-file <FileName>] [-save <FileName>]
 [-new] [-reinstall] [
-npcap] [-forceUpgrade] [-upgradeLocal] [-upgradeByUUID <FileName>] [-visibility]
[-goldenImage] [-installFolder <Installation Path>]
  -preCheck: run pre-check only
  -skipPreCheck <Option>: skip pre-installation check by given option; Valid options include
 'all', 'ipv6' and 'enforcement'; e.g.: '-skipPreCheck all' will skip all pre-installation
 checks; All pre-checks will be performed by default
  -noInstall: will not download and install sensor package onto the system
  -logFile <FileName>: write the log to the file specified by <FileName>
  -proxy <ProxyString>: set the value of HTTPS_PROXY, the string should be formatted as
http://<proxy>:<port>
  -noProxy: bypass system wide proxy; this flag will be ignored if -proxy flag was provided

  -help: print this usage
  -version: print current script's version
  -sensorVersion <VersionInfo>: select sensor's version; e.g.: '-sensorVersion 3.4.1.0.win64';
 will download the latest version by default if this flag was not provided
  -ls: list all available sensor versions for your system (will not list pre-3.1 packages);
 will not download any package
  -file <FileName>: provide local zip file to install sensor instead of downloading it from
 cluster
  -save <FileName>: downloaded and save zip file as <FileName>
  -new: remove any previous installed sensor; previous sensor identity has to be removed
from cluster in order for the new registration to succeed
  -reinstall: reinstall sensor and retain the same identity with cluster; this flag has
higher priority than -new
  -npcap: overwrite existing npcap
  -forceUpgrade: force sensor upgrade to version given by -sensorVersion flag; e.g.:
'-sensorVersion 3.4.1.0.win64 -forceUpgrade'; apply the latest version by default if
-sensorVersion flag was not provided
```

```
  -upgradeLocal: trigger local sensor upgrade to version given by -sensorVersion flag; e.g.:
 '-sensorVersion 3.4.1.0.win64 -upgradeLocal'; apply the latest version by default if
-sensorVersion flag was not provided
  -upgradeByUUID <FileName>: trigger sensor whose uuid is listed in <FileName> upgrade to
version given by -sensorVersion flag; e.g.: '-sensorVersion 3.4.1.0.win64 -upgradeByUUID
"C:\\Program Files\\Cisco Tetration\\sensor_id"'; apply the latest version by default if
-sensorVersion flag was not provided
  -visibility: install deep visibility agent only; -reinstall would overwrite this flag if
 previous installed agent type was enforcer
  -goldenImage: install Cisco Secure Workload Agent but do not start the Cisco Secure
Workload Services; use to install Cisco Secure Workload Agent on Golden Images in VDI
environment or Template VM. On VDI/VM instance created from golden image with different
host name, Cisco Secure Workload Services will work normally
  -installFolder: install Cisco Secure Workload Agent in a custom folder specified by
-installFolder e.g.: '-installFolder "c:\\custom sensor path"'; default path is "C:\Program
 Files\Cisco Tetration"
```

Install Windows Agent using the Agent Image Installer Method

We recommend the automated installer script method for installing Windows agents. Use the image installer method if you have a specific reason for using this manual method.

Note Do not manually deploy an older MSI agent version when an existing agent is already running on the host.

Site-related files that are in the package:

- **ca.cert**—Mandatory—CA certificate for sensor communications.
- **enforcer.cfg**—Mandatory only when installing enforcement sensor—Contains configuration of enforcement endpoints.
- **sensor_config**—Mandatory—Configuration for deep visibility sensor.
- **sensor_type**—Type of the sensor (enforcement or deep visibility).
- **site.cfg**—Mandatory—Global site endpoint configuration.
- **user.cfg**—Mandatory for SaaS—Sensor activation key and proxy configuration.

Prerequisite:

Configure the ACTIVATION_KEY and HTTPS_PROXY in the user.cfg file for SaaS clusters and when you are installing the agent on a non-default tenant of on-premises clusters with multiple tenants. For more information, see (Manual Installations Only) Update the User Configuration File.

To install a Windows agent using the agent image method:

Procedure

Step 1 Navigate to Agent Installation Methods:

- If you are a first-time user, launch the Quick Start wizard and click **Install Agents**.
- From the navigation pane, choose **Manage** > **Agents**, and select the **Installer** tab.

Step 2 Click **Agent Image Installer**.

Step 3 In the **Platform** field, enter Windows.

Step 4 Enter the required agent type and the version of the agent, and then from the results, download the required version of the agent.

Step 5 Copy the *tet-win-sensor<version>.win64-<clustername>.zip* file to all the Windows hosts for deployment.

Step 6 Ensure that you have administrative privileges and extract the ZIP file.

Step 7 In the extracted folder, run the following command to install the agent: `msiexec.exe /i TetrationAgentInstaller.msi`

Additionally, the following options are available for MSI installer.

Table 1: Available Options for MSI Installer

Options	Description
agenttype=<AgentType>	*AgentType* should be either *sensor* or *enforcer*, depending on whether enforcement is required. By default, the installer checks the content of the sensor_type file in the same folder and uses the content to overwrite the passed parameter. However, if agent is installed in */quiet* mode, the option is required.
overwritenpcap=yes	For Windows 2008 R2, by default, the agent does not attempt to upgrade Npcap if Npcap already exists. Pass this parameter to upgrade the existing Npcap. If this option is used, subsequent agent auto-upgrades also upgrade Npcap to newer supported versions.
nostart=yes	Pass this parameter, when installing the agent using a golden image in a VDI environment or VM template, to prevent agent service— CswAgent from starting automatically. On VDI/VM instances created using the golden image and with a different host name, these services, as expected, start automatically.
installfolder=<FullPathCustomFolder>	Use this parameter, at the end of the install command, to install the agent in a custom folder.
serviceuser=<Service UserName>	Use this parameter, at the end of the install command, to configure the service user. The default service user is “LocalSystem”. For local user, serviceuser=.\<Service UserName> For domain user, serviceuser=<domain_name>\<samaccount name> Service user must have local administrative privileges.
servicepassword=<Service UserPassword>	Use this parameter, at the end of the install command, to configure the password for the service user. The password must be in plain-text format.

Options	Description
proxy="<proxy_address>"	Use this parameter to set the HTTPS proxy for accessing the Secure Workload cluster.
activationkey=<activation Key>	Use this parameter to specify the tenant if agent is not being installed under the default tenant.

Note

- If activation key and proxy options are used during manual installation, you do not need to manually configure *user.cfg*.
- For Windows OS other than Windows 2008 R2, when you upgrade to version 3.8, the installed Npcap is automatically uninstalled by the Windows agent.
- If the agent is already installed on the host, do not reinstall the agent. To upgrade the agent, see Upgrading Software Agents section.

Verify Windows Agent Installation

Procedure

Step 1 Ensure that the folder `C:\\Program Files\\Cisco Tetration` (or the custom folder) exists.

Step 2 Ensure that the service— *CswAgent*, for deep visibility and enforcement, exists and is in the running state. Run command `cmd.exe` with administrative privileges.

Run the command `sc query cswagent`

Check if the status is **Running**

Run the command `sc qc cswagent`

Check if the DISPLAY-NAME is **Cisco Secure Workload Deep Visibility**

OR

Run the command **services.msc**

Find the name **Cisco Secure Workload Deep Visibility**

Check if the status is **Running**

Verify Windows Agent in the Configured Service User Context

1. Ensure that the service CswAgent running in the configured service user context. CswAgent runs in the same service user context.

   Run the command `cmd.exe` with **Admin** privileges

Run the command `sc qc cswagent`

Check SERVICE_START_NAME **<configured service user>**

OR

Run the command `services.msc`

Find the name **Cisco Secure Workload Deep Visibility**

Check **Log On As** for the <configured service user>

Find the name **Cisco Secure Workload Enforcement**

Check **Log On As** for the <configured service user>

OR

Run the command `tasklist /v | find /i "cswengine"`

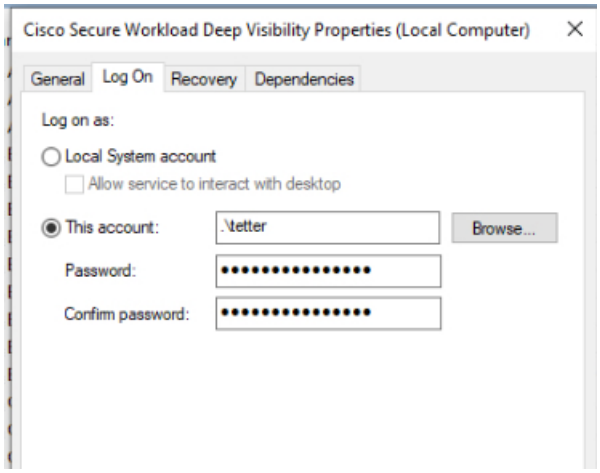
Check the user context for the running processes (5th column)

Modify Service Account

After installing Windows Agents, use one of the following methods to modify the existing Deep Visibility and Enforcement services.

- Use **services.msc**.

Figure 3: Modify Service Account based on services.msc Account



- Use any third party application to configure the services.
- Use the following commands:

1. Run cmd as an administrator.
2. Modify the services using the service account name by running the following commands:
   - sc config cswagent obj= <service user name> password= <password>
3. Verify service configurations by running the following commands:
   - `sc qc cswagent`

4. Restart the CswAgent service by running the following commands:
   a. sc stop cswagent
   b. sc start cswagent

Deploying Agents on a VDI Instance or VM Template (Windows)

By default, agent services start automatically after agents are installed. When installing on a golden image, you must use installer flags to prevent these services from starting. When instances are cloned from the golden image, agent services, as expected, start automatically.

Agent will not install Npcap on golden VMs, but will be automatically installed if needed on VM instances cloned from a golden image. For more information, see Windows Agent Installer and Npcap—For Windows 2008 R2.

Install the agent on a golden image in a VDI environment or VM template

Procedure

Step 1 Install the agent on a golden image in a VDI environment or VM template using an MSI installer or PowerShell installer script:

Use MSI installer with **nostart=yes**

- For more information, see Install Windows Agent using the Agent Image Installer Method, on page 11.
- msiexec.exe /<MSI installer> nostart=”yes” /quiet /norestart /l*v <installer_log_file> OR

OR

Use PowerShell installer with the **-goldenImage** flag.

- For more information, see Install Windows Agent using the Agent Script Installer Method, on page 9.

Step 2 Ensure that the folder `C:\\Program Files\\Cisco Tetration` (or the custom folder) exists.

Step 3 Ensure that the service CswAgent exists and is stopped:

Run the command `cmd.exe` with **Admin** privileges.

Run the command `sc query cswagent`

Check if the STATE is **Stopped**.

Step 4 The VM template is now configured.

Step 5 Shut down the VM template.

Create a new VDI instance VM

Procedure

Step 1 Create a new VDI instance VM by cloning the VM template.

Step 2 Reboot the VDI instance VM.

Step 3 After rebooting the VDI instance VM, ensure that the service CswAgent is running in the configured service context. See Verify Windows Agent Installation.

Step 4 On the VDI instance VM, ensure that the NPCAP driver is installed and running:

Run the command `cmd.exe` with Admin privileges

Run the command `sc query npcap`

Check if STATE is **Running**

Step 5 On the VDI instance VM, ensure that the agent is registered using a valid sensor_id:

- Check the sensor_id file in the installation folder.
- If the sensor_id starts with “uuid”, it is not a valid sensor_id.
- If the agent fails to register but the Secure Workload web interface shows that the agent is registered:
- Delete the agent using OpenAPI. For more information, see Deploy Software Agents.

Note

- Do not change the host name of the golden image or VM template.
- If the golden image or VM template is rebooted after installing the agent, Secure Workload services start running after the reboot.
- If the VDI instance VM fails to report network flows, see the *VDI Instance VM in Network Flows* section.

.

Windows Agent Installer and Npcap—For Windows 2008 R2

1. For supported Npcap versions, see the Support Matrix at https://www.cisco.com/go/secure-workload/requirements/agents.
2. Installation:

   If Npcap is not installed, the agent installs the supported version ten seconds after the service starts. If User has Npcap installed but the version is older than the supported version, Npcap is not be upgraded. Manually upgrade or uninstall Npcap, run the agent installer with the option **overwritenpcap=yes**, or run installer script with **-npcap** to get the supported Npcap version. If Npcap driver is in use by any application, the agent upgrades Npcap at a later time.
3. Upgrade:

   If Npcap is installed by Windows Agent and the version is older than the supported version, Npcap is upgraded to the supported version ten seconds after the service starts. If Npcap driver is in use by any

application, the agent upgrades Npcap at a later time. If Npcap is not installed by Windows Agent, Npcap is not upgraded.

4. Uninstall:

   If Npcap is installed by the Windows Agent, the agent uninstalls Npcap. If Npcap is installed by the user, but upgraded by the agent installer with **overwritenpcap=yes**, Npcap is not uninstalled. If Npcap driver is in use by any application, the agent does not uninstall Npcap.

Windows Agent Flow Captures: For All Windows OS Excluding Windows Server 2008 R2

From the latest version of Windows, the agent uses ndiscap.sys (Microsoft in-built) driver and Events Tracing using Windows (ETW) framework to capture the network flows.

During the upgrade to the latest version:

- The agent switches to ndiscap.sys from npcap.sys.
- The agent installer uninstalls Npcap if:
  - Npcap is installed by the agent.
  - Npcap is not in use.
  - OS version is not Windows Server 2008 R2.

After the agent services are started, the agent creates ETW sessions, CSW_MonNet, and CSW_MonDns (for DNS data), and initiates the capture of network flows.

Note

- On Windows Server 2012, network packets are parsed for DNS data.
- The Windows agent on hosts with Windows Server 2012 and later capture consumer and provider usernames and the usernames are available in the flow observations. This feature is not supported on Windows Server 2008 R2 because of limitations in the OS. In the agent configuration profile, configure the following to capture the usernames:
  - Enable PID/ User Lookup.
  - Set Flow Analysis Fidelity to Detailed.

Installing AIX Agents for Deep Visibility and Enforcement

Note Process tree, Package (CVE), and Forensic Event reporting features are not available on AIX. Additionally, some aspects of those features may not be available on specific minor releases of otherwise supported platforms due to OS limitations.

Requirements and Prerequisites for Installing AIX Agents

- See Supported Platforms and Requirements.

- Additional requirements for deep visibility:
  - Root privileges to install and execute the services.
  - Storage requirement for agent and log files: 500 MB.
  - Security exclusions configured on any security applications that are monitoring the host. These exclusions are to prevent other security applications from blocking agent installation or agent activity. For more information, see Security Exclusions.
  - AIX supports flow capture of only 20 network devices (6 network devices if version is AIX 7.1 TL3 SP4 or earlier). The deep visibility agent captures from a maximum of 16 network devices, leaving the other 4 capture sessions available for exclusive generic system usage (For example, tcpdump).
  - The deep visibilty agent does the following to ensure flow capture of 20 network devices:
    - The agent creates 16 bpf device nodes under the agents directory (/opt/cisco/tetration/chroot/dev/bpf0 - /opt/cisco/tetration/chroot/dev/bpf15)
    - tcpdump and other system tools using bpf will scan through the system device nodes (/dev/bpf0-/dev/bpf19) until they find an unused node (!EBUSY)
    - The bpf nodes created by the agent and the system bpf nodes share the same major/minor, with each major or minor being opened only by one instance (either tcpdump or agent).
    - The agent does not access the system device nodes nor does it create them as the tcpdump does (tcpdump-D creates /dev/bpf0. . . /dev/bpf19 if they do not exist).
  - Running iptrace on the system prevents, in certain scenarios, flow capture from tcpdump and the deep visibilty agent. This is a known design issue and needs to be checked with IBM.
    - To check if this scenario exists, before installing the agent, run tcpdump. If error message is **tcpdump: BIOCSETIF: en0: File exists** the iptrace is blocking flow capture. Stop iptrace to resolve the issue.
  - All deep visibility functions are not supported in AIX. Package and process accounting are among the ones not supported.
- Additional requirements for policy enforcement:
  - If IP Security Filter is enabled (that is, smitty IPsec4), agent installation fails in pre-check. We recommend you to disable IP Security Filter before installing the agent.
  - If IP Security is enabled when the Secure Workload enforcer agent is running, an error is reported and the enforcer agent stops enforcing. Contact support to safely disable the IP Security filter when the enforcer agent is running.

Install AIX Agent using the Agent Script Installer Method

Deep visibility and enforcement AIX agents can only be installed using the Agent Script Installation method.

Note

- The installed AIX agent supports both deep visibility and enforcement.
- By default, enforcement is disabled. To enable enforcement, see Create an Agent Configuration Profile, on page 59.

To install an AIX agent:

Procedure

Step 1 Navigate to Agent Installation Methods:

- If you are a first-time user, launch the Quick Start wizard and click **Install Agents**.
- From the navigation pane, choose **Manage** > **Agents**, and select the **Installer** tab.

Step 2 Click **Agent Script Installer**.

Step 3 From the **Select Platform** drop-down menu, choose **AIX**.

To view the supported AIX platforms, click **Show Supported Platforms**.

Step 4 Choose the tenant to install the agents.

Note Selecting a tenant is not required for Secure Workload SaaS clusters.

Step 5 If you want to assign labels to the workload, choose the label keys and enter label values.

When the installed agent reports IP addresses on the host, the installer CMDB labels selected here, along with other uploaded CMDB labels that have been assigned to IPs reported by this host, would be automatically assigned to the new IP address. If there are conflicts between uploaded CMDB labels and installer CMDB labels:

- Labels assigned to an exact IP address take precedence over labels assigned to the subnet.
- Existing labels assigned to an exact IP address take precedence over installer CMDB labels.

Step 6 If HTTP proxy is required to communicate with Secure Workload, choose **Yes**, and then enter a valid proxy URL.

Step 7 Under the **Installer expiration** section, select one from the available options:

- No expiration: The installer script can be used multiple times.
- One time: The installer script can be used only once.
- Time bound: You can set the number of days for which the installer script can be used.
- Number of deployments: You can set the number of times the installer script can be used.

Step 8 Click **Download** and save the file to the local disk.

Step 9 Copy the installer shell script to all the AIX hosts for deployment.

Step 10 To grant execute permission to the script, run the command: `chmod u+x tetration_installer_default_sensor_aix.sh`

Note The script name may differ depending on the agent type and scope.

Step 11 To install the agent, run the following command with root privileges:

```
./tetration_installer_default_sensor_aix.sh
```

Note If an agent is already installed on the host, you cannot proceed with the installation.

We recommend running the pre-check, as specified in the script usage details.

AIX installer script usage details:

```
ksh tetration_installer_default_enforcer_aix.sh [--pre-check] [--pre-check-user]
[--skip-pre-check=<option>] [--no-install] [--logfile=<filename>] [--proxy=<proxy_string>]
 [--no-proxy] [--help] [--version] [--sensor-version=<version_info>] [--ls]
[--file=<filename>] [--osversion=<osversion>] [--save=<filename>] [--new] [--reinstall]
[--unpriv-user] [--libs=<libs.zip|tar.Z>] [--force-upgrade] [--upgrade-local]
[--upgrade-by-uuid=<filename>] [--logbasedir=<logbdir>] [--tmpdir=<tmp_dir>] [--visibility]
 [--golden-image]
  --pre-check: run pre-check only
  --pre-check-user: provide alternative to nobody user for pre-check su support
  --skip-pre-check=<option>: skip pre-installation check by given option; Valid options
include 'all', 'ipv6' and 'enforcement'; e.g.: '--skip-pre-check=all' will skip all
pre-installation checks; All pre-checks will be performed by default
  --no-install: will not download and install sensor package onto the system
  --logfile=<filename>: write the log to the file specified by <filename>
  --proxy=<proxy_string>: set the value of HTTPS_PROXY, the string should be formatted as
http://<proxy>:<port>
  --no-proxy: bypass system wide proxy; this flag will be ignored if --proxy flag was
provided
  --help: print this usage
  --version: print current script's version
  --sensor-version=<version_info>: select sensor's version; e.g.: '--sensor-version=3.4.1.0';
 will download the latest version by default if this flag was not provided
  --ls: list all available sensor versions for your system (will not list pre-3.3 packages);
 will not download any package
  --file=<filename>: provide local zip file to install sensor instead of downloading it
from cluster
  --osversion=<osversion>: specify osversion for --save flag;
  --save=<filename>: download and save zip file as <filename>; will download package for
osversion given by --osversion flag; e.g.: '--save=myimage.aix72.tar.Z --osversion=7.2'
  --new: remove any previous installed sensor; previous sensor identity has to be removed
from cluster in order for the new registration to succeed
  --reinstall: reinstall sensor and retain the same identity with cluster; this flag has
higher priority than --new
  --unpriv-user=<username>: use <username> for unpriv processes instead of tet-snsr
  --libs=<libs.zip|tar.Z>: install provided libs to be used by agents
  --force-upgrade: force sensor upgrade to version given by --sensor-version flag; e.g.:
'--sensor-version=3.4.1.0 --force-upgrade'; apply the latest version by default if
--sensor-version flag was not provided
  --upgrade-local: trigger local sensor upgrade to version given by --sensor-version flag:
 e.g.: '--sensor-version=3.4.1.0 --upgrade-local'; apply the latest version by default if
--sensor-version flag was not provided
  --upgrade-by-uuid=<filename>: trigger sensor whose uuid is listed in <filename> upgrade
to version given by --sensor-version flag; e.g.: '--sensor-version=3.4.1.0
--upgrade-by-uuid=/usr/local/tet/sensor_id'; apply the latest version by default if
--sensor-version flag was not provided
  --logbasedir=<log_base_dir>: instead of logging to /opt/cisco/tetration/log use
<log_base_dir>. The full path will be <log_base_dir>/tetration
  --tmpdir=<tmp_dir>: instead of using /tmp use <tmp_dir> as temp directory
  --visibility: install deep visibility agent only; --reinstall would overwrite this flag
if previous installed agent type was enforcer
```

```
  --golden-image: install Cisco Secure Workload Agent but do not start the Cisco Secure
Workload Services; use to install Cisco Secure Workload Agent on Golden Images in VDI
environment or Template VM. On VDI/VM instance created from golden image with different
host name, Cisco Secure Workload Services will work normally
```

Verify AIX Agent Installation

Procedure

Run command `lslpp -c -l tet-sensor.rte`, confirm that there is one entry as follows.

Note The specific output may differ depending on the version

`$ sudo lslpp -c -l tet-sensor.rte` /usr/lib/objrepos:tet-sensor.rte:3.4.1.19::COMMITTED:I:TET tet sensor package:

```
$ sudo lssrc -s tet-sensor
```

Subsystem Group PID Status tet-sensor 1234567 active

```
$ sudo lssrc -s tet-enforcer
```

Subsystem Group PID Status tet-enforcer 7654321 active

Installing Kubernetes or OpenShift Agents for Deep Visibility and Enforcement

Requirements and Prerequisites

Operating system support information is available at Agent OS support matrix.

Requirements

- The install script requires Kubernetes or OpenShift administrator credentials to start privileged agent pods on the cluster nodes.
- Secure Workload entities are created in the **tetration** namespace.
- The node or pod security policies must permit privileged mode pods.
- busybox:1.33 images must either be preinstalled or be downloadable from Docker Hub.
- For containerd run time, if the config_path is not set, modify your config.toml (default location: /etc/containerd/config.toml) as follows:

  ```
  ```
      [plugins."io.containerd.grpc.v1.cri".registry]
      config_path = "/etc/containerd/certs.d"
  ```
  ```

  Restart the containerd daemon.

- To run on Kubernetes or OpenShift control plane nodes, the *–toleration* flag can be used to pass in a toleration for the Secure Workload pods. The toleration that is usually passed is the NoSchedule toleration that normally prevents pods from running on control plane nodes.
- For Windows worker nodes:
  - Supported Windows worker node container runtime: ContainerD.
  - ContainerD config: Configure the following containerd change.

    ```
    [plugins."io.containerd.grpc.v1.cri".registry]
    config_path = "/etc/containerd/certs.d"
    ```

    Remove configurations under **registry.mirrors**. The default configuration file location is `C:\Program Files\containerd\config.toml`.

    Restart the containerd daemon after the configuration changes.
  - The image **mcr.microsoft.com/oss/kubernetes/windows-host-process-containers-base-image:v1.0.0** must either be preinstallated or downloadable on the Windows worker node.
  - The existing Kubernetes agent which is upgrading to the newer version includes the Windows DaemonSet agent automatically. However, the previous script does not uninstall the Windows DaemonSet agent. Download the latest installer script to uninstall the Windows DaemonSet agent.
  - Supported on:
    - Microsoft Windows Server 2022
    - Windows Server 2019
    - Kubernetes 1.27 and later

Requirements for Policy Enforcement

IPVS-based kube-proxy mode is not supported for OpenShift.

These agents should be configured with the Preserve Rules option that is enabled. For more information, see Create an Agent Configuration Profile.

For enforcement to function properly, any installed CNI plug-in must:

- Provide flat address space (IP network) between all nodes and pods. Network plug-ins that masquerade the source pod IP for intracluster communication are not supported.
- Not interfere with Linux iptables rules or marks that are used by the Secure Workload Enforcement Agent (mark bits 21 and 20 are used to allow and deny traffic for NodePort services)

The following CNI plug-ins are tested for the above requirements:

- Calico (3.13) with the following Felix configurations: *(ChainInsertMode: Append, IptablesRefreshInterval: 0)* or *(ChainInsertMode: Insert, IptablesFilterAllowAction: Return, IptablesMangleAllowAction: Return, IptablesRefreshInterval: 0)*. All other options use their default values.

For more information on setting these options, see the Felix configuration reference.

Install Kubernetes or OpenShift Agent using the Agent Script Installer Method

Note The agent script installer method automatically installs agents on nodes included later.

Procedure

Step 1 Navigate to the Agent Installation Methods:

- If you are a first-time user, launch the Quick Start wizard and click **Install Agents**.
- From the navigation pane, choose **Manage** > **Agents**, and select the **Installer** tab.

Step 2 Click **Agent Script Installer**.

Step 3 From the **Select Platform** drop-down menu, choose **Kubernetes**.

To view the supported Kubernetes or OpenShift platforms, click **Show Supported Platforms**.

Step 4 Choose the tenant to install the agents.

Note Selecting a tenant is not required for Secure Workload SaaS clusters.

Step 5 If HTTP proxy is required to communicate with Secure Workload, choose **Yes**, and then enter a valid proxy URL.

Step 6 Click **Download** and save the file to the local disk.

Step 7 Run the installer script on a Linux machine which has access to the Kubernetes API server and a kubectl configuration file with administrative privileges as the default context/cluster/user.

The installer attempts to read the file from its default location (~/.kube/config). However, you can explicitly specify the location of the config file using the --kubeconfig command.

The installation script provides instructions for verifying the Secure Workload Agent Daemonset and the Pods that were installed.

Note The HTTP Proxy configured on the agent installer page prior to download only controls how Secure Workload agents connect to the Secure Workload cluster. This setting does not affect how Docker images are fetched by Kubernetes or OpenShift nodes, because the container runtime on those nodes uses its own proxy configuration. If the Docker images are not fetched from the Secure Workload cluster, debug the image pulling process of the container and add a suitable HTTP proxy.

Installing Solaris Agents for Deep Visibility

Requirements and Prerequisites for Installing Solaris Agents

- See Supported Platforms and Requirements.
- Root privileges to install and execute the services.
- One GB storage space for agent and log files.
- Configuration of security exclusions on security applications that are monitoring the host, to prevent other security applications from blocking of agent installation or agent activity. For more information, see Security Exclusions.

Install Solaris Agent using the Agent Script Installer Method

The installed Solaris agent supports both deep visibility and process or package visibility.

Procedure

Step 1 Navigate to Agent Installation Methods:

- If you are a first-time user, launch the Quick Start wizard and click **Install Agents**.
- From the navigation pane, choose **Manage** > **Agents**, and select the **Installer** tab.

Step 2 Click **Agent Script Installer**.

Step 3 From the **Select Platform** drop-down menu, choose **Solaris**.

To view the supported Solaris platforms, click **Show Supported Platforms**.

Step 4 Choose the tenant to install the agents.

Note Tenant selection is not required for Secure Workload SaaS clusters.

Step 5 If you want to assign labels to the workload, choose the label keys and enter label values.

When the installed agent reports IP addresses on the host, the installer CMDB labels selected here, along with other uploaded CMDB labels that have been assigned to IPs reported by this host, would be assigned automatically to the new IP address. If there are conflicts between uploaded CMDB labels and installer CMDB labels:

- Labels assigned to an exact IP address take precedence over labels assigned to the subnet.
- Existing labels assigned to an exact IP address take precedence over installer CMDB labels.

Step 6 If HTTP proxy is required to communicate with Secure Workload, choose **Yes**, and then enter a valid proxy URL.

Step 7 Under the **Installer expiration** section, select one from the available options:

- No expiration: The installer script can be used multiple times.
- One time: The installer script can be used only once.

- Time bound: You can set the number of days for which the installer script can be used.
- Number of deployments: You can set the number of times the installer script can be used.

Step 8 Click **Download** and save the file to the local disk.

Step 9 Copy the installer shell script on Solaris hosts and run the following command to grant execute permission to the script: `chmod u+x tetration_installer_default_sensor_solaris.sh`

Note The script name may differ depending on the selected agent type and scope.

Step 10 To install the agent, run the following command with root privileges:
`./tetration_installer_default_sensor_solaris.sh`

Note If an agent is already installed on the tenant, you cannot proceed with the installation.

We recommend running the precheck, as specified in the script usage details.

Solaris installer script usage details:

```
tetration_installer_default_sensor_solaris.sh [--pre-check] [--skip-pre-check=<option>]
[--no-install] [--logfile=<filename>] [--proxy=<proxy_string>] [--no-proxy] [--help]
[--version] [--sensor-version=<version_info>] [--ls] [--file=<filename>] [--save=<filename>]
 [--new] [--reinstall] [--unpriv-user] [--force-upgrade] [--upgrade-local]
[--upgrade-by-uuid=<filename>] [--basedir=<basedir>] [--logbasedir=<logbdir>]
[--tmpdir=<tmp_dir>] [--visibility] [--golden-image]
  --pre-check: run pre-check only
  --skip-pre-check=<option>: skip pre-installation check by given option; Valid options
include 'all', 'ipv6' and 'enforcement'; e.g.: '--skip-pre-check=all' will skip all
pre-installation checks; All pre-checks will be performed by default
  --no-install: will not download and install sensor package onto the system
  --logfile=<filename>: write the log to the file specified by <filename>
  --proxy=<proxy_string>: set the value of CL_HTTPS_PROXY, the string should be formatted
as http://<proxy>:<port>
  --no-proxy: bypass system wide proxy; this flag will be ignored if --proxy flag was
provided
  --help: print this usage
  --version: print current script's version
  --sensor-version=<version_info>: select sensor's version; e.g.: '--sensor-version=3.4.1.0';
 will download the latest version by default if this flag was not provided
  --ls: list all available sensor versions for your system (will not list pre-3.1 packages);
 will not download any package
  --file=<filename>: provide local zip file to install sensor instead of downloading it
from cluster
  --save=<filename>: download and save zip file as <filename>
  --new: remove any previous installed sensor; previous sensor identity has to be removed
from cluster in order for the new registration to succeed
  --reinstall: reinstall sensor and retain the same identity with cluster; this flag has
higher priority than --new
  --unpriv-user=<username>: use <username> for unpriv processes instead of nobody
  --force-upgrade: force sensor upgrade to version given by --sensor-version flag; e.g.:
'--sensor-version=3.4.1.0 --force-upgrade'; apply the latest version by default if
--sensor-version flag was not provided
  --upgrade-local: trigger local sensor upgrade to version given by --sensor-version flag:
 e.g.: '--sensor-version=3.4.1.0 --upgrade-local'; apply the latest version by default if
--sensor-version flag was not provided
  --upgrade-by-uuid=<filename>: trigger sensor whose uuid is listed in <filename> upgrade
to version given by --sensor-version flag; e.g.: '--sensor-version=3.4.1.0
--upgrade-by-uuid=/usr/local/tet/sensor_id'; apply the latest version by default if
--sensor-version flag was not provided
```

```
  --logbasedir=<log_base_dir>: instead of logging to /opt/cisco/secure-workload/log use
<log_base_dir>. The full path will be <log_base_dir>/secure-workload
  --tmpdir=<tmp_dir>: instead of using /tmp use <tmp_dir> as temp directory
  --visibility: install deep visibility agent only; --reinstall would overwrite this flag
if previous installed agent type was enforcer
  --golden-image: install Cisco Secure Workload Agent but do not start the Cisco Secure
Workload Services; use to install Cisco Secure Workload Agent on Golden Images in VDI
environment or Template VM. On VDI/VM instance created from golden image with different
host name, Cisco Secure Workload Services will work normally
```

Verify Solaris Agent Installation

Procedure

Step 1 Run the command: `sudo pkg list tet-sensor`

Step 2 A single entry as output confirms that a Solaris agent is installed on the host.
Sample output:

NAME (PUBLISHER)	VERSION	IFO
tet-sensor (cisco)	3.8.1.1	i--

Note The specific output may differ depending on the platform and architecture.

(Manual Installations Only) Update the User Configuration File

The following procedure is required only for installations involving *all* of the following:

- Secure Workload SaaS, or on-premises clusters with multiple tenants (on-premises clusters that use only the default tenant do NOT need this procedure)
- Manual installation
- Linux or Windows platform

Agents require an activation key to register to the Secure Workload cluster. they require a cluster activation key. Additionally, they might need an HTTPS proxy to reach the cluster.

Note In Windows Environment, you do not need to manually configure user.cfg, if activationkey and proxy options are used during manual installation.

Before installation, configure the required variables in the user configuration file:

Procedure

Step 1 To retrieve your activation key, navigate to **Manage** > **Agents**, click the **Installer** tab, click **Manual Install using classic packaged installers**, then click **Agent Activation Key**.

Step 2 Open the `user.cfg` file in the Secure Workload Agent installation folder. (Example: `/usr/local/tet` on Linux or `C:\\Program Files\\Cisco Tetration` on Windows). The file contains a list of variables in the form of "key=value", one on each line.

Step 3 Add the activation key to the **ACTIVATION_KEY** variable. Example:
`ACTIVATION_KEY=7752163c635ef62e6568e9e852d07bd21bfd60d0`

Step 4 If the agent requires an HTTPS proxy, add the **http** protocol proxy server and port using the **HTTPS_PROXY** variable. Example: `HTTPS_PROXY=http://proxy.my-company.com:80`

Other Agent-Like Tools

AnyConnect Agents

No Secure Wrokload agent is required for platforms supported by Cisco AnyConnect Secure Mobility agent with Network Visibility Module (NVM). AnyConnect connector registers these agents and exports flow observations, inventories, and labels to Secure Workload. For more information, see AnyConnect Connector.

For Windows, Mac, or Linux platforms, see Cisco AnyConnect Secure Mobility Client Data Sheet.

ISE Agents

A Secure Workload agent on the endpoint is not required for endpoints registered with Cisco Identity Service Engine (ISE). ISE connector collects metadata about endpoints from ISE through pxGrid service on ISE appliance. It registers the endpoints as ISE agents on Secure Workload and pushes labels for the inventories on these endpoints. For more information, see ISE Connector.

SPAN Agents

SPAN agents work with the ERSPAN connector. For more information, see ERSPAN Connector.

Third-Party and Additional Cisco Products

- For integrations using external orchestrators configured in Secure Workload,

  See External Orchestrators in Secure Workload.

- For integrations using connectors configured in Secure Workload.

  See What are Connectors.

Connectivity Information

In general, when the agent is installed on the workload, it makes several network connections to the back-end services hosted on the Secure Workload cluster. The number of connections will vary depending on the agent type and its functions.

The following table captures various permanent connections that are made by various agent types.

Table 2: Agent Connectivity

Agent type	Config server	Collectors	Enforcement backend
visibility (on-premises)	CFG-SERVER-IP:443	COLLECTOR-IP:5640	N/A
visibility (SaaS)	CFG-SERVER-IP:443	COLLECTOR-IP:443	N/A
enforcement (on-premises)	CFG-SERVER-IP:443	COLLECTOR-IP:5640	ENFORCER-IP:5660
enforcement (SaaS)	CFG-SERVER-IP:443	COLLECTOR-IP:443	ENFORCER-IP:443
docker images	CFG-SERVER-IP:443	N/A	N/A

Legends:

- CFG-SERVER-IP is the IP address of the config server.
- COLLECTOR-IP is the IP address of the collector. Deep visibility and enforcement agents connect to all available collectors.
- ENFORCER-IP is the IP address of the enforcement endpoint. The enforcement agent connects to only one of the available endpoints.
- For Kubernetes/Openshift agent deployments, the installation script does not contain the agent software - Docker images containing the agent software are pulled from the Secure Workload cluster by every Kubernetes/Openshift node. These connections are established by the container run time image fetch component and directed at CFG-SERVER-IP:443.

Navigate to **Platform** > **Cluster Configuration** to know the config server IP and collector IP.

- **Sensor VIP** is for the config server IP: The IP address that has been set up for the config server in this cluster.
- **External IPs** are for collectors IPs and enforcer: If this is populated, when assigning external cluster IP addresses, the selection process is restricted to only IP addresses defined in this list, that are part of the external network.

Note

- The Secure Workload agent always acts as a client to initiate the connections to the services hosted within the cluster, and never opens a connection as a server.
- Agents, for which upgrade is supported, periodically perform HTTPS requests (port 443) to the cluster sensor VIP to query for available packages.
- An agent can be located behind a NAT server.

Connections to the cluster might be denied if the workload is behind a firewall, or if the host firewall service is enabled. In such cases, administrators must create appropriate firewall policies to allow the connections.

Security Exclusions

Software agents continuously interact with the host operating system during their normal operations. This operation may cause other security applications installed on the host such as antivirus, security agents, and others, to raise alarms or block the actions of Secure Workload agents. Therefore, to ensure that agents are installed successfully and are functioning, you must configure the necessary security exclusions on the security applications that are monitoring the host.

Table 3: Security Exclusions for Agent Directories

Host OS	Directories
AIX	/opt/cisco/tetration
Linux	/usr/local/tet or /opt/cisco/tetration or <user chosen inst dir>
	/var/opt/cisco/secure-workload
Windows	C:\Program Files\Cisco Tetration
	C:\ProgramData\Cisco Tetration
Solaris	/opt/cisco/secure-workload

Table 4: Security exclusions for Agent Processes

Host OS	Processes
AIX	csw-agent
	tet-sensor
	tet-enforcer
	tet-main
Linux	csw-agent
	tet-sensor
	tet-enforcer
	tet-main
	enforcer
Windows	CswEngine.exe
	TetEnfC.exe

Host OS	Processes
Solaris	csw-agent
	tet-sensor
	tet-enforcer
	tet-main

Table 5: Security Exclusions for Agent Actions

Host OS	Actions
AIX	Access /dev/bpf*, /dev/ipl, /dev/kmem
	Invokes cfg_ipf, curl, ipf, ippool, ipfstat lslpp, lsfilt, prtconf
	Scan /proc
Linux	Invokes curl, ip[6]tables-save, ip[6]tables-restore, rpm/dpkg
	Scan /proc, open netlink sockets
Windows	Access registry
	Register to firewall events
	Invokes c:\windows\system32\netsh.exe
Solaris	Invokes curl, lspp, pkg, smbios
	Scan /proc

Table 6: Security Exclusions for Agents Scripts or Binaries Executions

Host OS	Invoked scripts/binaries
AIX	-
Linux	-
Windows	dmidecode.exe
	npcap-installer.exe
	sensortools.exe
	signtool.exe
Solaris	-

Service Management of Agents

Software agents are deployed as a service in all supported platforms. This section describes methods to manage the services for various functions and platforms.

Note Unless specified otherwise, all the commands in this section require root privileges on Linux or Unix, or administrative privileges on Windows to run.

Service Management for RHEL, CentOS, OracleLinux-6.x, and Ubuntu-14

Run the following commands for:

- **Starting a service**: `start csw-agent`
- **Stopping a service**: `stop csw-agent`
- **Restarting a service**: `restart csw-agent`
- **Checking service status**: `status csw-agent`

Service Management for RHEL, CentOS, OracleLinux-7.x and Later

The commands are also applicable to:

- AlmaLinux, Rocky Linux- 8.x and later
- Amazon Linux 2 and later
- Debian 8 and later
- SLES-12SPx and later
- Ubuntu-16.04 and later

Run the following commands for:

- **Starting a service**: `systemctl start csw-agent`
- **Stopping a service**: `systemctl stop csw-agent`
- **Restarting a service**: `systemctl restart csw-agent`
- **Checking service status**: `systemctl status csw-agent`

Service Management for Windows Server or Windows VDI

Run the following commands for:

- **Starting a service**: `net start <service-name>`

  Example: **net start cswagent** for deep visibility and enforcement service

- **Stopping a service**: `net stop <service-name>`

  Example: **net stop cswagent** for deep visibility and enforcement service

- **Restarting a service**:

  1. `net stop <service-name>`
  2. `net start <service-name>`

- **Checking service status**: `sc query <service-name>`

  Example: **sc query cswagent** for deep visibility and enforcement service

Service Management for AIX

Run the following commands for:

- **Starting a service**: `startsrc -s csw-agent`
- **Stopping a service**: `stopsrc -s csw-agent`
- **Restarting a service**:

  1. `stopsrc -s csw-agent`
  2. `startsrc -s csw-agent`

- **Checking service status**: `lssrc -s csw-agent`

Service Management for Kubernetes Agent Installations

- **Starting or stopping a service**: It is not possible to start or stop the agents on a specific node because they are not installed as individual services, but as a cluster-wide daemon set.
- **Restarting an agent on a node**: Locate the Secure Workload agent pod on the node and run the appropriate Kubernetes command to kill it. The pod is automatically restarted.
- **Checking the status of pods**: `kubectl get pod -n tetration` or `oc get pod -n tetration` (for OpenShift) lists the status of all Secure Workload agent pods in the Kubernetes cluster.

Service Management for Solaris

Run the following commands for:

- **Starting a service**: `svcadm enable csw-agent`
- **Stopping a service**: `svcadm disable csw-agent`
- **Restarting a service**: `svcadm restart csw-agent`
- **Checking service status**: `svcs -l csw-agent`

Policy Enforcement with Agents

By default, agents that are installed on your workloads have the capability to enforce policy, but enforcement is disabled. When you are ready, you can enable these agents to enforce policy on selected hosts that are based on the configured intent.

When an agent enforces a policy, it applies an ordered set of rules that specify whether the firewall should ALLOW or DROP specific network traffic that is based on parameters such as the source, destination, port, protocol, and direction. For more information on policies, see Manage Policies Lifecycle in Secure Workload.

Enforcement using agents

- Agents receive policies over a secured TCP or SSL channel.
- Agents run in a privileged domain. On Linux machines, the agent runs as root; on Windows machines, the agent runs as SYSTEM.
- Depending on the platform, when policy enforcement is enabled, agents can completely control the firewall or work with existing configured rules.
- For details about enforcement options and to enable and configure agents to enforce policies, see Create an Agent Configuration Profile, on page 59.

Advanced details

When you enable enforcement, golden rules are formulated to allow the agent to connect to the controller. Agents communicate with the Enforcement Front End (EFE) of the controller through a bidirectional and secure channel using the TLS or SSL protocol. Messages from the controller are signed by the policy generator and verified by the agent.

The agent receives policies in a platform-independent schema from the controller. The agent converts these platform-independent policies into platform-specific policies and programs the firewall on the endpoint.

The agent actively monitors the firewall state. If the agent detects any deviation in the enforced policies, it enforces the cached policies into the firewall again. The agent also monitors its own consumption of system resources such as CPU and memory.

The agent periodically sends a status and stats report to the controller using EFE. The status report includes the status of the latest programmed policies such as success, failure, or error, if any. The stats report includes the policy stats such as allowed and dropped packets, and byte count depending on the platform.

Agent Enforcement on the Linux Platform

On the Linux platform, the agent uses the iptables, ip6tables, or ipset to enforce network policies. After the agent is enabled on the host, by default, it controls, and programs iptables. If the IPv6 network stack is enabled, then the agent controls the IPv6 firewall using ip6tables.

Linux iptables or ip6tables

The Linux kernel has iptables and ip6tables that are used to set up, maintain, and inspect the tables of IPv4 and IPv6 packet filter rules. The iptables and ip6tables consist of many predefined tables. Each table contains predefined chains and can also contain user-defined chains. These chains contain sets of rules and each of these rules specifies the match criteria for a packet. Predefined tables include raw, mangle, filter, and NAT. Predefined chains include INPUT, OUTPUT, FORWARD, PREROUTING, and POSTROUTING.

The Secure Workload agent programs a filter table that contains rules to allow or drop packets. The filter table consists of the predefined chains INPUT, OUTPUT, and FORWARD. Along with these, the agent adds custom TA chains to categorize and manage the policies from the controller. These TA chains contain Secure Workload rules that are derived from the policies along with rules that are generated by the agent. When the agent receives platform-independent rules, it parses and converts them into iptable, ip6table, or ipset rules and inserts these rules into TA defined chains in the filter table. After programming the firewall, the agent monitors the firewall for any rule or policy deviation and if so, reprograms the firewall. It keeps track of the policies that are programmed in the firewall and reports their stats periodically to the controller.

Here is an example to depict this behavior:

A typical policy in a platform-independent network policy message consists of:

```
source set id: "test-set-1"
destination set id: "test-set-2"
source ports: 20-30
destination ports: 40-50
ip protocol: TCP
action: ALLOW
. . .
set_id: "test-set-1"
     ip_addr: 1.2.0.0
     prefix_length: 16
     address_family: IPv4
set_id: "test-set-2"
     ip_addr: 3.4.0.0
     prefix_length: 16
     address_family: IPv4
```

Along with other information, the agent processes this policy and converts it into platform-specific ipset and iptables rule:

```
ipset rule:
Name: ta_f7b05c30ffa338fc063081060bf3
Type: hash:net
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16784
References: 1
Members:
1.2.0.0/16
Name: ta_1b97bc50b3374829e11a3e020859
Type: hash:net
Header: family inet hashsize 1024 maxelem 65536
Size in memory: 16784
References: 1
Members:
3.4.0.0/16
iptables rule:
TA_INPUT -p tcp -m set --match-set ta_f7b05c30ffa338fc063081060bf3 src -m set --match-
‹→set ta_1b97bc50b3374829e11a3e020859 dst -m multiport --sports 20:30 -m multiport --
‹→dports 40:50 -j ACCEPT
```

Caveats

ipset Kernel Module

When enforcement is enabled and preserve rules is disabled in the Agent Config profile, the agents running on Linux hosts ensures that the ipset kernel module has a sufficiently large *max_sets* configuration. In case a change is needed, the agent reloads the ipset kernel module with a new *max_sets* value. If Preserve Rules is

enabled, the agents check the current ipset module *max_sets* value, but does not make any change. The current configured *max_sets* value can be found in `cat /sys/module/ip_set/parameters/max_sets`.

Host Firewall Backup

The first time that enforcement is enabled in the Agent Config profile, the agents running on Linux hosts, store the current content of ipset and ip[6]tables in `/opt/cisco/tetration/backup` before taking control of the host firewall.

Successive disable or enable transitions of enforcement configuration do not generate backups. The directory is not removed after agent uninstallation.

Agent Enforcement on the Windows Platform in WAF mode

On the Windows platform, the Secure Workload agent uses the Windows Firewall to enforce network policies.

Windows Firewall with Advanced Security

A native component on Windows, the Windows Firewall with Advanced Security, regulates network traffic that is based on the following types of settings:

- Rules that regulate inbound network traffic.
- Rules that regulate outbound network traffic.
- Override rules that is based on the authentication status of the source and destination of the network traffic.
- Rules that apply to IPsec traffic and to Windows services.

The Secure Workload Network Policy is programmed using inbound and outbound firewall rules.

Secure Workload Rules and the Windows Firewall

On the Windows platform, the Secure Workload Network Policy is enforced as follows:

1. The platform-independent firewall rules from the Secure Workload Network policy are translated into Windows Firewall rules.
2. The rules are programmed in Windows Firewall.
3. The Windows Firewall enforces the rules.
4. The Windows Firewall and its ruleset are monitored. If a change is detected, the deviation is reported and the Secure Workload Network policy is reset in the Windows Firewall.

Security Profiles

Windows Firewall groups the rules based on the network that the host is connected to. These rule groups are called Profiles and there are three such profiles:

- Domain Profile
- Private Profile
- Public Profile

The Secure Workload rules are programmed into all the profiles, but only rules in active profiles are continuously monitored.

Effective Setting and Mixed-List Policies

The set of rules in the Windows Firewall is not ordered based on precedence. When multiple rules match a packet, the most restrictive of those rules take effect meaning that DENY rules take precedence over ALLOW rules. For more information, see the article on Microsoft TechNet.

Consider the mixed-list, both allow and deny, policy example from the Enforcement Agent section:

```
1. ALLOW 1.2.3.30 tcp port 80
2. ALLOW 1.2.3.40 udp port 53
3. BLOCK 1.2.3.0/24 ip
4. ALLOW 1.2.0.0/16 ip
5. Catch-all: DROP ingress, ALLOW egress
```

When a packet headed for the host 1.2.3.30 TCP port 80 reaches the firewall, it matches all the rules, but the most restrictive of them all, Rule number 3, is the one that will be enforced and the packet will be dropped. This behavior is contrary to the expectation that the rules will be evaluated in order, Rule 1 is the rule that is enforced, and that the packet will be allowed.

This difference in behavior is expected in the Windows platform owing to the design of the Windows Firewall described above. This behavior can be observed in mixed-list policies with overlapping rules that have different rule actions.

For example,

```
1. ALLOW 1.2.3.30 tcp
2. BLOCK 1.2.3.0/24 tcp
```

Interference from Other Firewalls or Policies

We recommend that you grant the agent full and exclusive control of the Windows Firewall to enforce the Secure Workload Network Policy as intended. Agents cannot reliably enforce the policy if:

- A third-party firewall is present. (The Windows Firewall is required to be the active firewall product on the host.)
- The Firewall is disabled for the current profiles.
- Conflicting firewall settings are deployed using Group Policy. Some of the conflicting settings are:
  - Firewall rules.
  - Default inbound or outbound actions in the current profiles that differ from the catch-all rules of the policy.
  - Firewall disabled for the current profiles.

Stateful Enforcement

Windows Advanced Firewall is considered as a **stateful** firewall, that is for certain protocols such as TCP, the firewall maintains internal state tracking to detect if a new packet hitting the firewall belongs to a known connection. Packets belonging to a known connection are allowed without the firewall rules having to be examined. A stateful firewall enables bidirectional communication without rules having to be established in the INBOUND and OUTBOUND tables.

For example, consider the following rule for a web server: **Accept all TCP connections to port 443**

The intention is to accept all TCP connections on port 443 to the server, and allow the server to communicate back to the clients. In this case, only one rule is inserted in the INBOUND table, allowing TCP connections on port 443. No rule is required to be inserted in the OUTBOUND table. Inserting a rule in the OUTBOUND table is implicitly done by the Windows Advanced Firewall.

Note Stateful tracking applies only to protocols that establish and maintain explicit connections. For other protocols, both INBOUND and OUTBOUND rules must be programmed to enable bidirectional communication.

When enforcement is enabled, a given concrete rule is programmed as **stateful** when the protocol is TCP (the agent decides, based on the context, whether the rule is to be inserted in the INBOUND table or the OUTBOUND table). For other protocols (including **ANY**), both INBOUND and OUTBOUND rules are programmed.

Caveats

Host firewall backup

When enforcement is enabled for the first time in the Agent Config profile, the agents running on Windows hosts, before taking control of the host firewall, export the current Windows Advanced Firewall content to `ProgramData\Cisco\Tetration\backup`. Successive disable or enable transitions of Enforcement configuration do not generate backups. The directory is not removed upon agent uninstallation.

Agent Enforcement on the Windows Platform in WFP Mode

On the Windows platform, the agent enforces the network policies by programming Windows Filtering Platform (WFP) filters. Windows Advanced Firewall is not used to configure the network policy.

Windows Filtering Platform

Windows Filtering Platform (WFP) is a set of APIs provided by Microsoft to configure filters for processing network traffic. Network traffic processing filters are configured using kernel-level APIs and user level APIs. WFP filters can be configured at various layers, Network Layer, Transport Layer, Application Layer Enforcement (ALE). Secure Workload WFP filters are configured at the ALE layer, similar to Windows firewall rules. Each layer has several sublayers, ordered by weight, from highest to lowest. Within each sublayer, filters are ordered by weight, from highest to lowest. A network packet traverses through all the sublayers. At each sublayer, the network packet traverses through the matching filters that are based on weight, from highest to lowest and returns the action: Permit or Block. After passing through all the sublayers, the packet is processed based on the rule that Block action overrides Permit.

Advantages of WFP over WAF

- Avoids Windows Firewall configuration dependencies.
- Overcomes GPO restrictions.
- Ensures ease of migration and policy reversion.
- Allows you to control policy ordering.

- Avoids strict block-first policy order of Windows Firewall.
- Reduces CPU overhead on policy update.
- Creates an efficient 1:1 policy rule filter.
- Ensures a faster single-step update.

Agent Support for WFP

When enforcement is configured to use WFP, Secure Workload filters override Windows Firewall rules.

In WFP mode, the agent configures the following WFP objects:

- Provider has a GUID and name, is used for filter management, and does not affect packet filtering
- Sublayer has a GUID, name, and weight. The Secure Workload sublayer is configured with higher weight than the Windows Advanced Firewall sublayer.
- Filter has name, GUID, ID, weight, layer ID, sublayer key, action (PERMIT/ BLOCK), and conditions. WFP filters are configured for Golder rules, Self Rules, and Policy Rules. The agent also configures the port scanning prevention filters. Secure Workload filters are configured with the FWPM_FILTER_FLAG_CLEAR_ACTION_RIGHT flag. This flag ensures that Secure Workload filters are not overridden by Microsoft Firewall rules. For each Secure Workload Network policy rule, one or more WFP filters are configured based on the direction (inbound or outbound) and protocol.

For TCP inbound policy,

```
id: 14 , TCP Allow 10.195.210.184 Dir=In localport=3389
```

The WFP filters configured are:

```
Filter Name:                  Secure Workload Rule 14
-----------------------------------------------------
EffectiveWeight:                        18446744073709551589
LayerKey:                               FWPM_LAYER_ALE_AUTH_LISTEN_V4
Action:                                 Permit
Local Port:                             3389
Filter Name:                            Secure Workload Rule 14
-----------------------------------------------------
EffectiveWeight:                        18446744073709551589
LayerKey:                               FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V4
Action:                                 Permit
RemoteIP:                               10.195.210.184-10.195.210.184
```

Secure Workload agent configures **Secure Workload Default Inbound** and **Secure Workload Default Outbound** filters for inbound and outbound CATCH-ALL policy respectively.

Agent WFP support and Windows Firewall

- The agent **does not monitor** WAF rules or WAF profiles.
- The agent **does not monitor** firewall states.
- The agent **does not require** firewall state to be enabled.
- The agent **does not conflict** with GPO policies.

Effective Setting and Mixed-List Policies

Agent enforcement in WFP mode supports mixed-list or grey list policies.

Consider the mixed-list (both allow and deny) policy example from the Enforcement Agent section:

```
1. ALLOW 1.2.3.30 tcp port 80-           wt1000
2. BLOCK 1.2.3.0/24 ip-                  wt998
3. ALLOW 1.2.0.0/16 ip-                  wt997
4. Catch-all: DROP ingress, ALLOW egress - wt996
```

When a packet headed for the host 1.2.3.30 tcp port 80 reaches the firewall, it matches filter 1 and is allowed. However, a packet that is headed for the host 1.2.3.10 is blocked because of filter 2. A packet that is headed for host 1.2.2.10 is allowed by filter 3.

Stateful Enforcement

Secure Workload WFP filters are configured at the ALE layer. Network traffic is filtered for socket connect(), listen(), and accept() operations. Network packets related to a L4 connection are not filtered after the connection is established.

Visibility of Configured WFP Filters

You can view the configured Secure Workload WFP filters using `c:\program files\tetration\tetenf.exe`. Supported options:

- With administrative privileges, run `cmd.exe`.
- Run `c:\program files\tetration\tetenf.exe -l -f <-verbose> <-output=outfile.txt>`.

OR

- With administrative privileges, run `cmd.exe`.
- Run `netsh wfp show filters`.
- View the **filters.xml** file for configured Secure Workload filters.

Disable Stealth Mode Filters in WFP Mode

To disable stealth mode filters (port scanning filters):

Procedure

Step 1 Edit `\conf\enforcer.cfg`.

Step 2 Add **disable_wfp_stealth_mode**: 1

Step 3 Save the file.

Step 4 With administrative privileges, restart the CswAgent service by:

a) Run the command: `sc stop cswagent` to stop the CswAgent Service.

b) Run the command: `sc start cswagent` to start the CswAgentService.

Step 5 To verify:

a) With administrative privileges, run `cmd.exe`.

b) Run the command: `c:\program files\tetration\tetenf.exe -l -f <-verbose> <-output=outfile.txt>`.

```
"Tetration Internal Rule block portscan" filters are not configured.
```

Delete Configured WFP Filters

You can delete the configured Secure Workload WFP filters using `c:\program files\tetration\tetenf.exe`. To avoid accidental deletions of filters, when you run the delete command, specify the token in <*yyyymm*> format, where *yyyy* is the current year and *mm* is the current month in the numerical form. For example, if today's date is 01/21/2021, then the token is **-token=202101**

Supported options are:

- With administrative privileges, run `cmd.exe`.
- To delete all configured Secure Workload filters, run `c:\program files\tetration\tetenf.exe -d -f -all - token=<yyyymm>`
- To delete all configured Secure Workload WFP objects, run `c:\program files\tetration\tetenf.exe -d -all -token=<yyyymm>`
- To delete a Secure Workload WFP filter by name, run `c:\program files\tetration\tetenf.exe -d -name=<WFP filter name> -token=<yyyymm>`

Known Limitations in WFP Mode

- The **Preserve Rules** setting in Agent Config Profile has no effect when you set Enforcement Mode to WFP.

Configure Policies for Windows Attributes

For more granularity when enforcing a policy on Windows-based workloads, you can filter network traffic by:

- Application Name
- Service Name
- User Names with or without User Groups

This option is supported in both WAF and WFP modes. Windows OS-based filters are categorized as *consumer filters* and *provider filters* in the generated network policy. The Consumer filters filter the network traffic that is initiated on the consumer workload and Provider filters filter the network traffic that is destined for the provider workload.

Before you begin

This procedure assumes you are modifying an existing policy. If you have not yet created the policy to which you want to add a Windows OS-based filter, create that policy first.

Important See Caveats, on page 44 and Known limitations, on page 43 for policies involving Windows attributes.

Procedure

Step 1 In the navigation pane, click **Defend** > **Segmentation**.

Step 2 Click the scope that contains the policy for which you want to configure Windows OS-based filters.

Step 3 Click the workspace in which you want to edit the policy.

Step 4 Click **Manage Policies**.

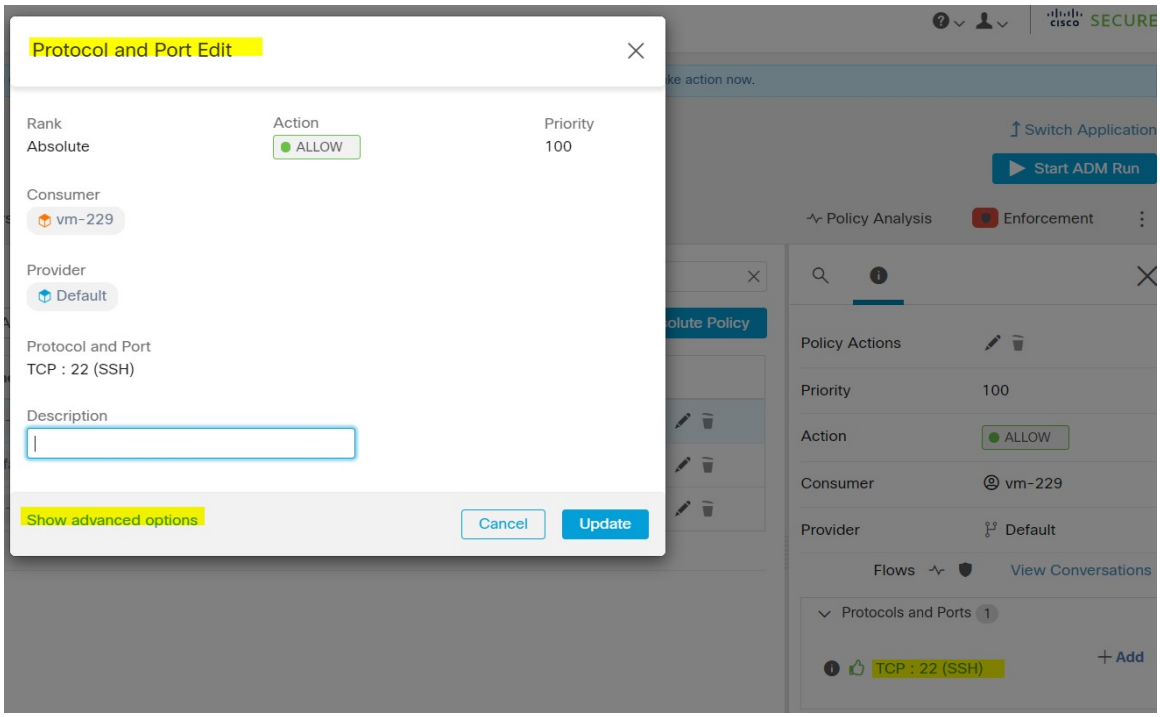
Step 5 Choose the policy to edit.

Important Consumer and Provider must include only Windows workloads.

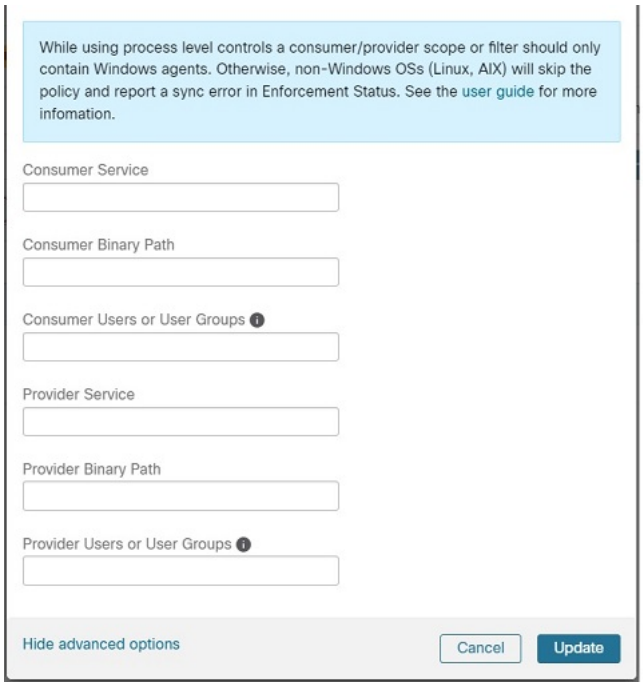
Step 6 In the table row for the policy to edit, click the existing value in the **Protocols and Ports** column.

Step 7 In the pane on the right, click the existing value under **Protocols and Ports**.

In the example, click **`TCP : 22 (SSH)`**.



Step 8 Click **Show advanced options**.



Step 9 Configure consumer filters based on Application name, Service name, or User name.

- The application name must be a full pathname.
- Service name must be a short service name.
- User name can be a local user name (For example, tetter) or domain user name (For example, sensor-dev@sensor-dev.com or sensor-dev\sensor-dev)
- User group can be local user group (For example, Administrators) or domain user group (For example, domain users\\sensor-dev)
- Multiple user names and/ or user group names can be specified, separated by ",".(For example, sensor-dev\@sensor-dev.com,domain users\\sensor-dev)
- Service name and User name cannot be configured together.

Step 10 Configure provider filters based on Application name, Service name, or User name.

Follow the same guidelines as given for consumer filters in the previous step.

Step 11 Enter the paths to the binary, as applicable.

For example, enter **`c:\test\putty.exe`**

Step 12 Click **Update**.

Recommended Windows OS-Based Policy Configuration

Always specify ports and protocols in policies when possible; we recommend not to allow ANY port, ANY protocol.

For example, a generated policy with port and protocol restrictions might look like this:

```
 dst_ports {
  start_port: 22
  end_port: 22
  consumer_filters {
    application_name: "c:\\test\\putty.exe"
  }
 }}
 ip_protocol: TCP
```

In contrast, if you allow network connections that are initiated by iperf.exe with ANY protocol and ANY port, the generated policy looks like this:

```
match_set {
  dst_ports {
    end_port: 65535
    consumer_filters {
      application_name: "c:\\test\\iperf.exe"
    }
  }
  address_family: IPv4
  inspection_point: EGRESS
  match_comment: "PolicyId=61008290755f027a92291b9d:61005f90497d4f47cedacb86:"
}
```

For the above filter, Secure Workload creates a policy rule to allow the network traffic on the provider as follows:

```
match_set {
    dst_ports {
    end_port: 65535
  }
  address_family: IPv4
  inspection_point: INGRESS
  match_comment: "PolicyId=61008290755f027a92291b9d:61005f90497d4f47cedacb86:"
}
```

This network rule opens all the ports on the Provider. We strongly recommend not to create OS-based filters with *Any* protocol.

Known limitations

- Windows 2008 R2 does not support Windows OS based filtering policies.
- Network policy can be configured with a single user name whereas MS Firewall UI supports multiple users.

Caveats

- While using the Windows OS-based policies, a consumer/ provider scope or filter should only contain Windows agents. Otherwise, non-Windows OSs (Linux, AIX) skip the policy and report a sync error in Enforcement Status.
- Avoid creating Windows OS filters with *loose* filtering criteria. Such criteria may open unwanted network ports.
- If OS filters are configured for consumer, then the policies are applicable only to consumer, similarly if it is configured for provider then it is applicable only to provider.
- Due to limited or no knowledge of the process context, user context or service context of the network flows, there will be discrepancy in the policy analysis if the policies have Windows OS-based filters.

Verify and Troubleshoot Policies with Windows OS-Based Filtering Attributes

If you use Windows OS-based filtering attributes, the following topics provide you with verification and troubleshooting information.

Cisco TAC can use this information as needed to troubleshoot such policies.

Policies Based on Application Name

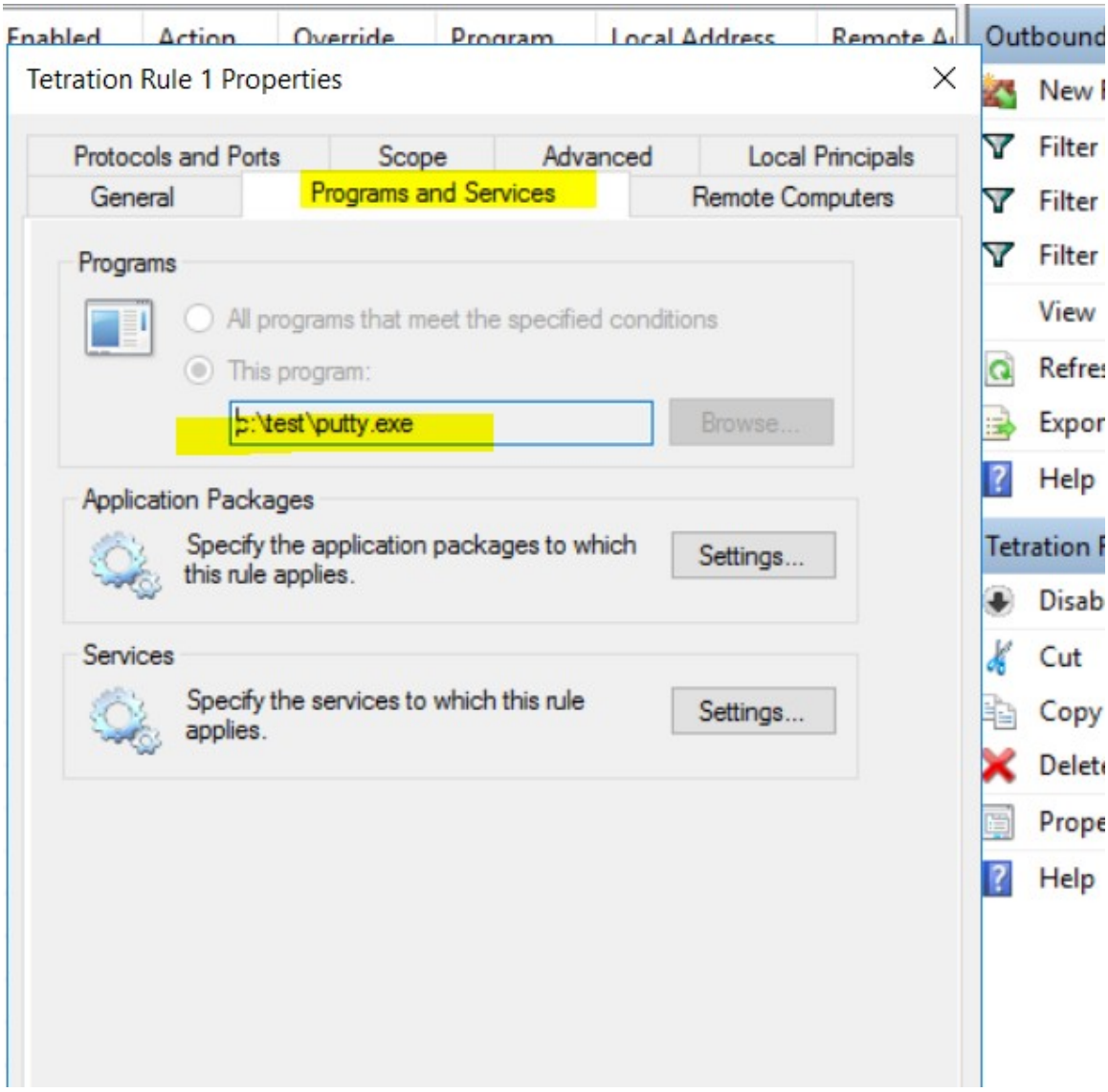
Use the following information to verify and troubleshoot policies based on application name on Windows OS workloads.

The following sections describe the way policies should appear on the workload for an application binary entered as **`c:\test\putty.exe`**.

Sample Policy Based on Application Name

```
dst_ports {
start_port: 22
end_port: 22
consumer_filters {
application_name: "c:\test\putty.exe"
}
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

Generated Firewall Rule



Generated Filter Using netsh

To verify, using native Windows tools, that a filter has been added to an advanced policy:

- With administrative privileges, run `cmd.exe`.
- Run `netsh wfp show filters`.
- The output file, **filters.xml**, is generated in the current directory.
- Check FWPM_CONDITION_ALE_APP_ID for the application name in the output file: filters.xml.

```
<fieldKey>FWPM_CONDITION_ALE_APP_ID</fieldKey>
                    <matchType>FWP_MATCH_EQUAL</matchType>
                    <conditionValue>
                           <type>FWP_BYTE_BLOB_TYPE</type>
```

```
                              <byteBlob>
                                     <data>
‹→5c00640065007600690063006500 5c006800610072006400640069007 3006b0076006f006
‹→</data>
                                   <asString>\device\harddiskvolume2\temp\putty.exe</
‹→asString>
                 </byteBlob>
       </conditionValue>
```

Generated WFP Filter Using tetenf.exe -l -f

```
Filter Name:                    Secure Workload Rule 1
-----------------------------------------------------
EffectiveWeight:                18446744073709551592
LayerKey:                       FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                         Permit
RemoteIP:                       10.195.210.15-10.195.210.15
Remote Port:                    22
Protocol:                       6
AppID:                          \device\harddiskvolume2\test\putty.exe
```

Invalid Application Name

- In WAF mode, Firewall rule is created for an invalid application name.
- In WFP mode, the WFP filter is not created for an invalid application name but the NPC is not rejected. The agent logs a warning message and configures the rest of the policy rules.

Policies Based on Service Name

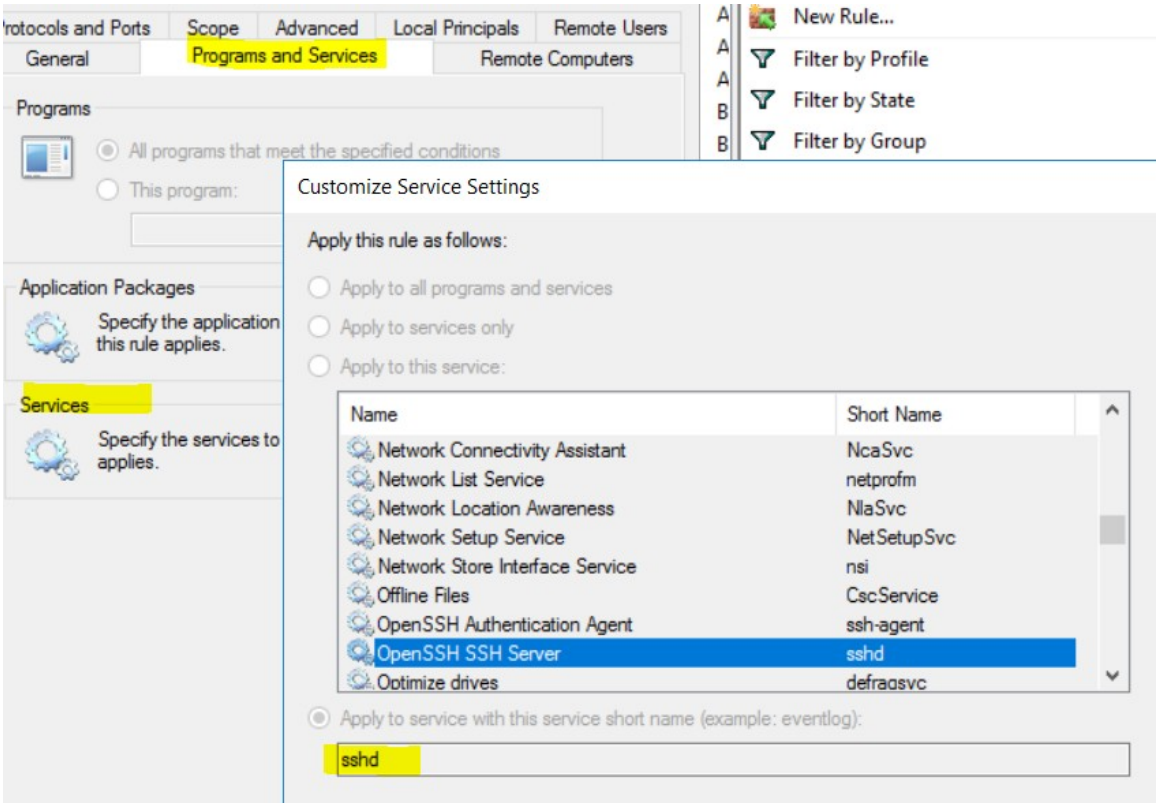
Use the following information to verify and troubleshoot policies based on Service name on Windows OS workloads.

The following sections describe the way that the policies should appear on the workload.

Sample Policy Based on Service Name

```
    dst_ports {
             start_port: 22
             end_port: 22
             provider_filters {
                    service_name: “sshd”
             }
         }}
         ip_protocol: TCP
         address_family: IPv4
         inspection_point: INGRESS
```

Generated Firewall Rule



Generated Filter Using netsh

To verify using native Windows tools, that a filter has been added for an advanced policy:

- With administrative privileges, run `cmd.exe`.
- Run `netsh wfp show filters`.
- The output file, **filters.xml**, is generated in the current directory.
- Check FWPM_CONDITION_ALE_USER_ID for user name in the output file: filters.xml.

```
<item>
                    <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
                    <matchType>FWP_MATCH_EQUAL</matchType>
                    <conditionValue>
                              <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>

<sd>O:SYG:SYD:(A;;CCRC;;;S-1-5-80-3847866527-469524349-687026318-
→516638107)</sd>
                    </conditionValue>
</item>
```

Generated WFP Filter Using tetenf.exe -l -f

```
Filter Name:       Secure Workload Rule 3
----------------------------------------------------------
EffectiveWeight:            18446744073709551590
```

```
LayerKey:                   FWPM_LAYER_ALE_AUTH_RECV_ACCEPT_V4
Action:                     Permit
Local Port:                 22
Protocol:                   6
User or Service:            NT SERVICE\sshd
```

Invalid Service Name

- In WAF mode, the Firewall rule is created for a nonexistent service name.
- In WFP mode, the WFP filter is not created for a nonexistent service name.
- Service SID type must be *Unrestricted* or *Restricted*. If the service type is *None*, the Firewall Rule and WFP filter can be added but they have no effect.

  To verify the SID type, run the following command:

  ```
  sc qsidtype <service name>
  ```

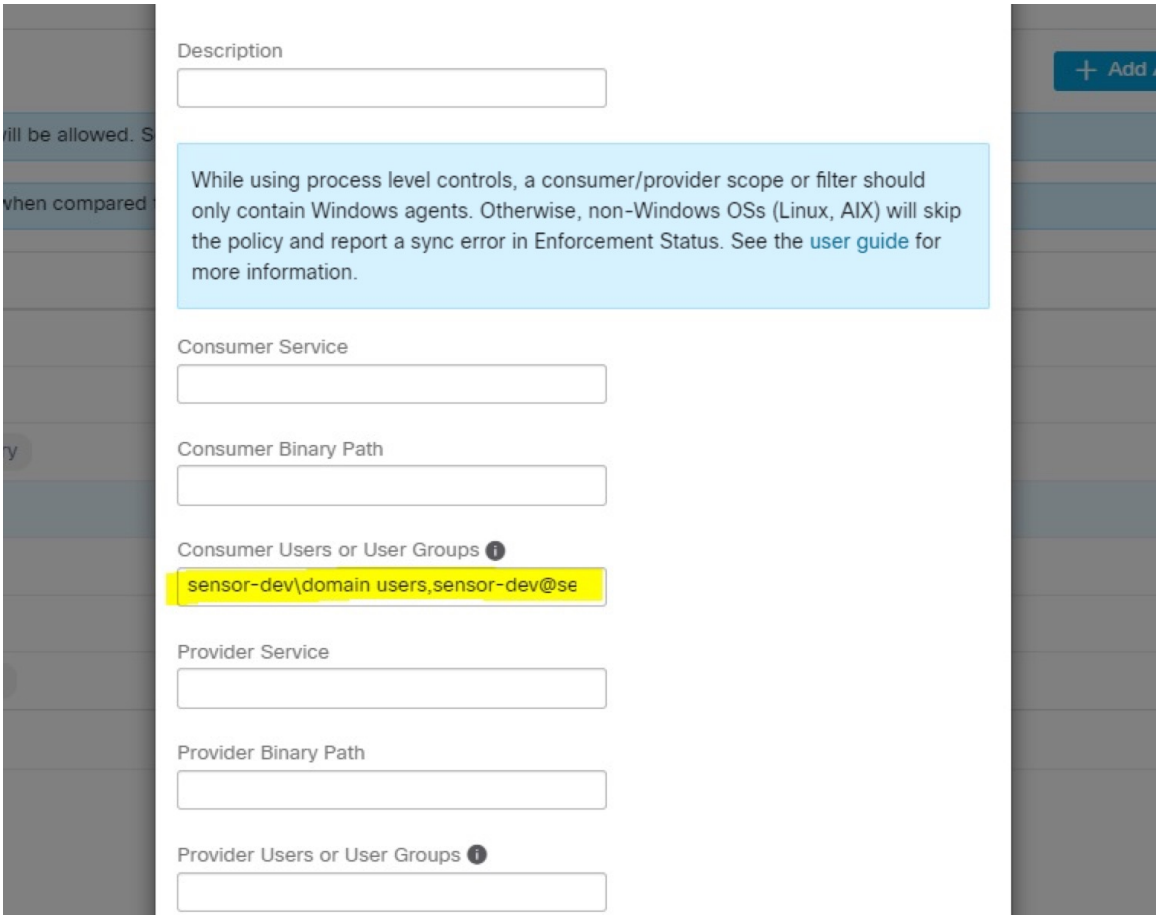
Policies Based on User Group or User Name

Use the following information to verify and troubleshoot policies based on user name (with and without user group name) on Windows OS workloads.

Sections in this topic describe the way that the policies should appear on the workload.

Examples in this topic are based on policies that are configured with the following information:

Figure 4: Policies Based on User Group or User Name



Sample Policy Based on User Name

```
dst_ports {
          start_port: 30000
          end_port: 30000
          provider_filters {
               user_name: “sensor-dev\sensor-dev”
           }
}}
ip_protocol: TCP
address_family: IPv4
inspection_point: EGRESS
```

Sample Policy Based on User Group and User Name

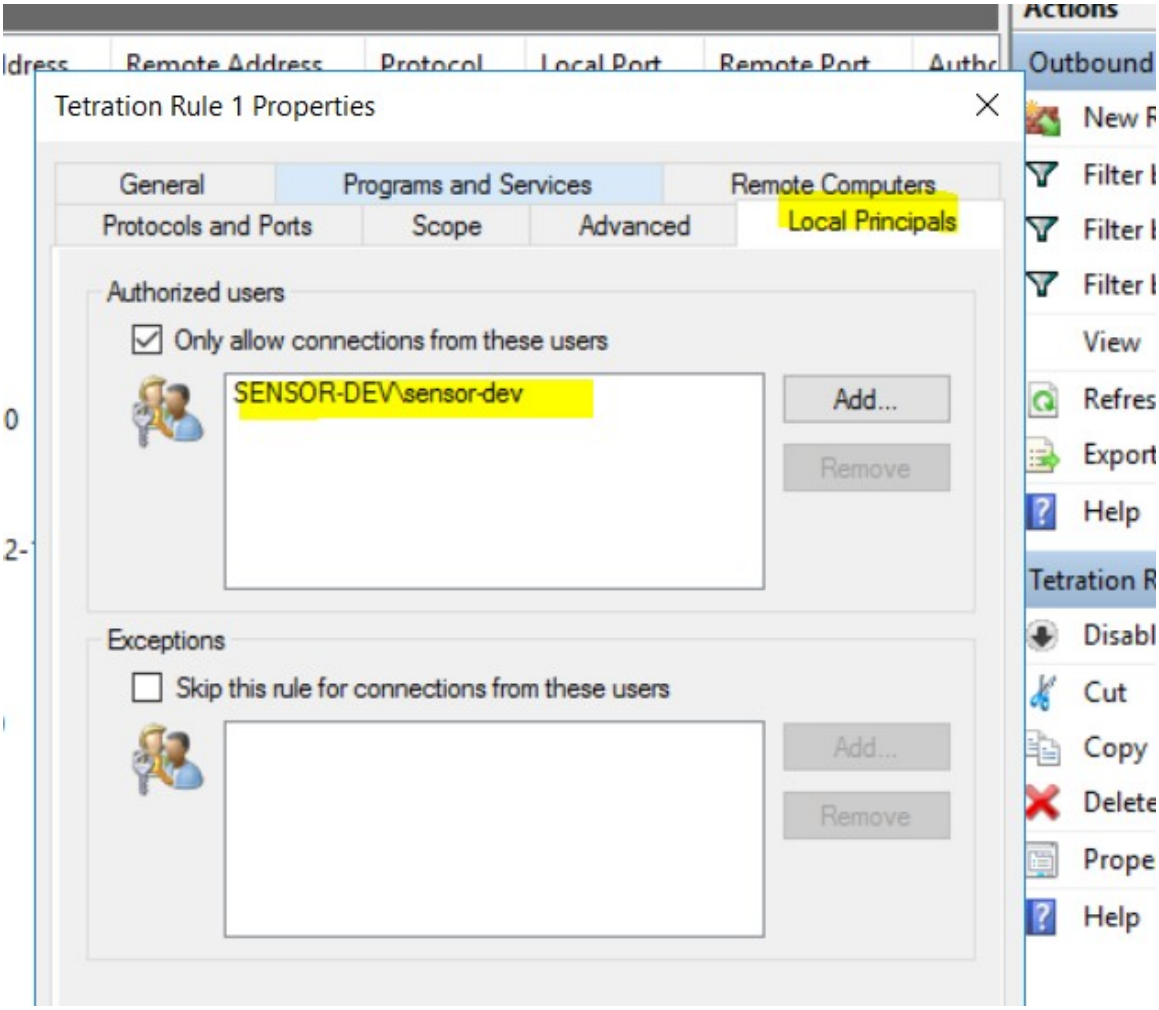
```
dst_ports {
start_port: 30000
end_port: 30000
provider_filters {
user_name: “sensor-dev\domain users,sensor-dev\sensor-dev”
}
}}
ip_protocol: TCP
```

```
address_family: IPv4
inspection_point: EGRESS
```

Generated Firewall Rule

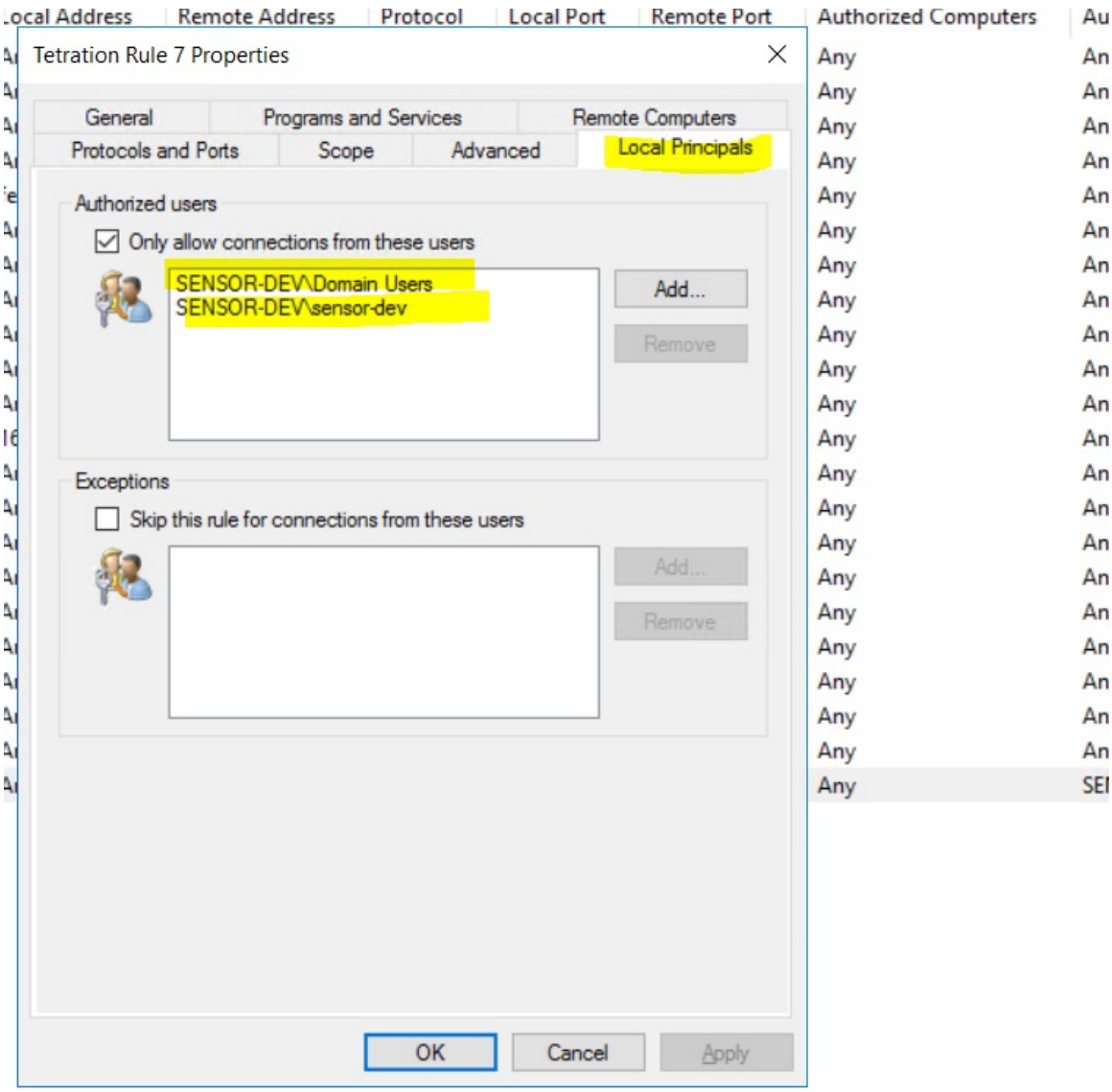
Firewall Rule Based on User Name

Example: Firewall rule based on User Name, sensor-dev\\sensor-dev



Firewall Rule Based on User Group and User Name

Example: Firewall rule based on User Name, sensor-dev\\sensor-dev and user group, domain users\\sensor-dev



Generated Filter Using netsh

To verify using native Windows tools that a filter has been added for an advanced policy:

- With administrative privileges, run `cmd.exe`.
- Run `netsh wfp show filters`.
- The output file, **filters.xml**, is generated in the current directory.
- Check FWPM_CONDITION_ALE_USER_ID for user name in the output file: filters.xml.

```
<item>
            <fieldKey>FWPM_CONDITION_ALE_USER_ID</fieldKey>
            <matchType>FWP_MATCH_EQUAL</matchType>
            <conditionValue>
                  <type>FWP_SECURITY_DESCRIPTOR_TYPE</type>
```

```
                    <sd>O:LSD:(A;;CC;;;S-1-5-21-4172447896-825920244-2358685150)</sd>
                </conditionValue>
    </item>
```

Generated WFP Filters Using tetenf.exe -l -f

Filter based on User Name

Example: WFP Rule based on User Name, SENSOR-DEV\sensor-dev

```
Filter Name:                    Secure Workload Rule 1
-----------------------------------------------------
EffectiveWeight:                18446744073709551590
LayerKey:                       FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                         Permit
RemoteIP:                       10.195.210.15-10.195.210.15
Remote Port:                    30000
Protocol:                       6
User or Service:                SENSOR-DEV\sensor-dev
```

Filter based on User Group and User Name

Example: WFP Rule based on User Name, SENSOR-DEV\\sensor-dev and User Group name, SENSOR-DEV\\Domain Users

```
Filter Name:          Secure Workload Rule 1
-----------------------------------------------------
EffectiveWeight:             18446744073709551590
LayerKey:                    FWPM_LAYER_ALE_AUTH_CONNECT_V4
Action:                      Permit
RemoteIP:                    10.195.210.15-10.195.210.15
Remote Port:                 30000
Protocol:                    6
User or Service:             SENSOR-DEV\Domain Users, SENSOR-DEV\sensor-dev
```

Service name and user name cannot be configured for a Network policy rule.

Note The network policy is rejected by the Windows agent if the user name or the user group is invalid.

Enforcement of Kubernetes Pods on Windows Nodes

After you install the Kubernetes DaemonSet agent on the Windows worker nodes, it captures the network flows from the Windows worker nodes and the Kubernetes pods in an AKS environment.

Requirements

- Enforcement of Kubernetes pods is supported in an AKS environment with Windows nodes.
- Enforcement mode MUST be WFP with **Preserve Rules** set to Off.
- Supported on Microsoft Windows Server 2019 and Windows Server 2022.

The policies are enforced on vSwitch for ports that are connected to pods using VFP. The Virtual Filtering Platform (VFP) is a component of vSwitch used to configure filters for processing network traffic. While enforcing the policies, the Preserve Mode is Off.

Each filter has the following attributes:

- Id: Filter Name
- Direction: In or Out
- RuleType: Switch or Host.
  - Configure the filter on vSwitch when the type is Switch.
  - Create a WFP filter when the type is Host.
- Action: Allow or Block
- LocalPorts: This can be a port or range. For example, 80 or 100-200.
- RemotePorts: Same as LocalPorts.
- LocalAddresses: It is an address or range. For example, 10.224.0.5, 10.224.1.0/24 (10.224.1.1-10.224.1.10 is not allowed).
- RemoteAddress: Same as LocalAddresses
- Protocol: ICMP/TCP/UDP/IGMP Protocol 255 is IPPROTO_RAW and 256 – PROTO_MAX

The ports can only be specified for UDP and TCP, and ports are not allowed in the policy unless a protocol is specified.

Configuring a policy on a virtual port is a transaction-based operation. If one of the filters is invalid, enforcing the entire policy is rendered unsuccessful.

This is the stateful enforcement. Application, user, or service-based policies are currently not supported.

Compatibility with Calico

Pods enforcement works in "preserve rules" off mode. When the Windows agent enforces the rules on pods, it deletes the already configured policies. If the Calico plug-in enforces the network policies after the agent, the agent identifies it as **deviation** and network policies that are configured by Calico are deleted and agent policies are re-enforced.

Note The enforced policies are deleted when the Windows agent is uninstalled on the Windows nodes.

Visibility of Configured VFP Filters

An option to list the pod filters using Secure Workload is not available. In an AKS environment, you can use the built-in PowerShell script. Run the following PowerShell script: `c:\k\debug\collectlogs.ps1`. View the output files **vfpoutput.txt** and **hnsdiag.txt** for the configured filters.

Delete VFP Filters Configured by Windows Agent

1. Run **cmd.exe** with administrative privileges.
2. Run the command: `<installation folder>\tetenf.exe -d -f -pods -token=<yyyymm>`.

Note The command deletes VFP filters for all the pods.

Troubleshoot Enforced Policies and Network Flows

1. Run the command: `netsh wfp start capture keywords=19`.
2. Run network traffic.
3. Stop capturing the flows: `netsh wfp stop capture`.
4. Extract **wfpdiag.xml** from the **wfpdiag.cab** file. View the dropped flows.

To map the allowed or dropped network flows to Pod policies:

1. Start ETW session: `logman start <session name> -p Microsoft-Windows-Hyper-V-VfpExt -o <output file.etl> -ets`
2. Run network traffic.
3. Stop capturing flows: `logman stop <session name>`.
4. In the command prompt, run: `tracerpt <output file.etl>`. The command creates the **dumpfile.xml** file. View the network flows.

Agent Enforcement on AIX Platform

On the AIX platform, the Secure Workload agent uses IPFilter utilities to enforce network policies. By default, after the agent is enabled on the host, the agent controls and programs the IPv4 filter table. IPv6 enforcement is not supported.

IPFilter

The IPFilter package on AIX is used to provide firewall services and is available on AIX as a kernel expansion pack. It loads as a kernel extension module, */usr/lib/drivers/ipf*. It includes ipf, ippool, ipfstat, ipmon, ipfs, and ipnat utilities that are used to program ipfilter rules and each of these rules specifies the match criteria for a packet. For more information, see the IPFilter pages in the AIX manual.

When enforcement is enabled, the agent uses IPFilter to program the IPv4 filter table that contains rules for allowing or dropping of IPv4 packets. The agent groups these rules to categorize and manage the policies using the controller. These rules include Secure Workload rules that are derived from the policies and rules that are generated by the agent.

When an agent receives platform-independent rules, it parses and converts them into ipfilter or ippool rules and inserts these rules into the filter table. After programming the firewall, the enforcement agent monitors the firewall for any rule or policy deviation and if so, reprograms the firewall. The agent keeps track of the policies that are programmed in the firewall and reports their status periodically to the controller.

A typical policy in a platform-independent network policy message consists of:

```
source set id: "test-set-1"
destination set id: "test-set-2"
source ports: 20-30
```

```
    destination ports: 40-50
    ip protocol: TCP
    action: ALLOW
    ...
    set_id: "test-set-1"
       ip_addr: 1.2.0.0
       prefix_length: 16
       address_family: IPv4
    set_id: "test-set-2"
       ip_addr: 5.6.0.0
       prefix_length: 16
       address_family: IPv4
```

Along with other information, the agent processes the policy and converts it into platform-specific ippool and ipfilter rule:

```
    table role = ipf type = tree number = 51400
    { 1.2.0.0/16; };

    table role = ipf type = tree number = 75966
    { 5.6.0.0/16; };

pass in quick proto tcp from pool/51400 port 20:30 to pool/75966 port 40:50 flags S/SA group
 TA_INPUT
pass out quick proto tcp from pool/75966 port 40:50 to pool/51400 port 20:30 flags A/A group
 TA_OUTPUT
```

Caveats

Host Firewall Backup

When enforcement is enabled for the first time in an Agent Config Profile, the agents running on AIX hosts, before taking control of the host firewall, store the current content of ippool and ipfilter into */opt/cisco/tetration/backup*. Successive disable or enable transitions of enforcement configuration do not generate backups. The directory is not removed upon agent uninstallation.

Known Limitations

IPv6 enforcement is not supported.

Check Agent Status and Statistics

Procedure

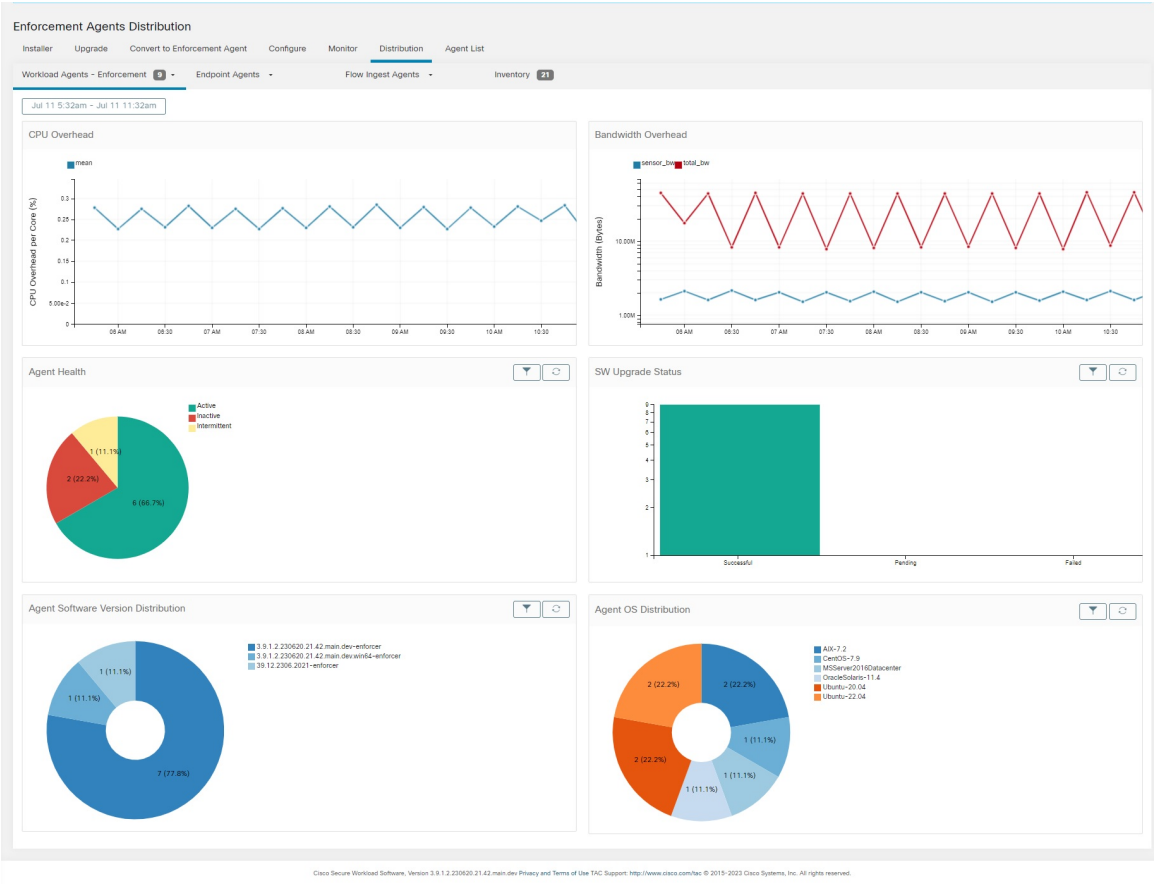
Step 1 In the navigation pane, click **Manage** > **Workloads** > **Agents**.

Step 2 Click the **Distribution** tab.

Step 3 Click an agent type from the top of the page.

Step 4 On this page, you can check CPU Overhead, Bandwidth Overhead, Agent Health, Software Update Status, Agent Software Version Distribution, and Agent OS Distribution.

Figure 5: Agent Distribution Page



Note **Agent Health**: The agent periodically checks in every 10–30 minutes. If there is no check-in for more than 1 hour 30 minutes, then the agent is inactive. To reduce false alarms, the agent health status is set to intermittent instead of inactive if the check-in gap is between 1 hour and 1 hour 30 minutes.

For more information on the enforcement status, see the Enforcement Status section.

View Agent Details

The following steps provide one of the available options to navigate to the Workload Profile page, which displays details about the workload and its installed agent.

Procedure

Step 1 In the navigation pane, click **Organize** > > **Scopes and Inventory**.

Step 2 Search for a workload for which you want to view details.

Step 3 Click the IP address to view the details such as agent health, IP address, Scopes, Inventory Type, Enforcement Groups, Experimental Groups, User Labels, and Traffic Volume (Total Bytes/Total Packets).

For more information, see Workload Profile.

Software Agent Config

Requirements and Prerequisites for Configuring Software Agents

- Ensure that you have the required Secure Workload user role credentials:
  - Site Administrator
  - Customer Support

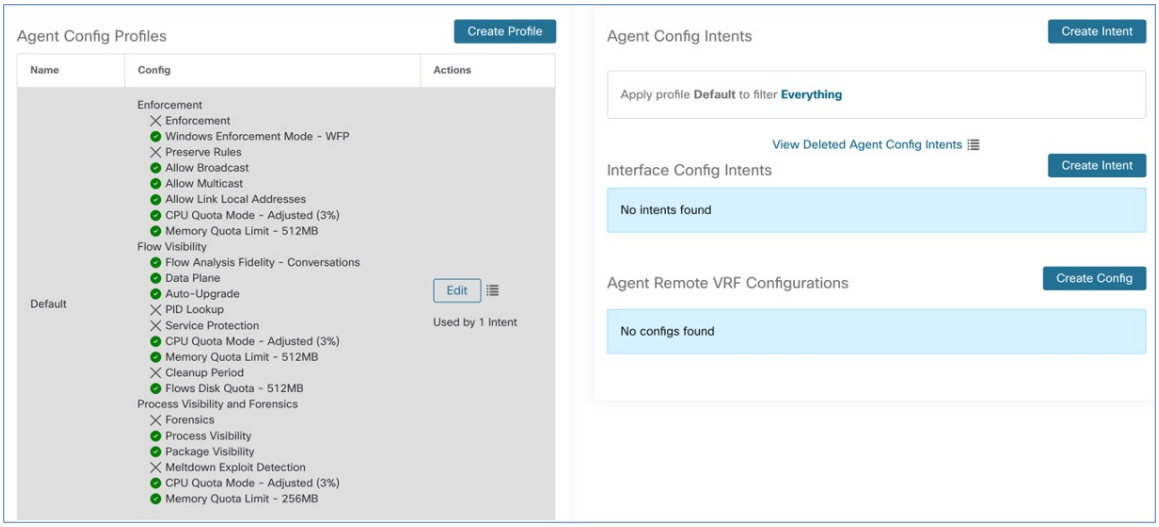
  For more information, see User Roles and Access to Agent Configuration, on page 57.
- Ensure that you have privileges on the host to run the agent service on each workload. For more information, see Service Management of Agents.
- Verify the supported platforms, requirements, and installation instructions for agents. For more information, see Deploy Software Agents.

User Roles and Access to Agent Configuration

1. Root scope owners have access only to create a Configuration Profile and Configuration Intent specification.
2. As a Root Scope owner, you can create configuration profiles that are associated with only owned scopes and impose these configuration profiles on agents.

Note Under the Agent Configuration Profile, you can now view the number of intents using the configuration profile before you edit the profile.

Figure 6: Software Agent Configuration for Scope Owners



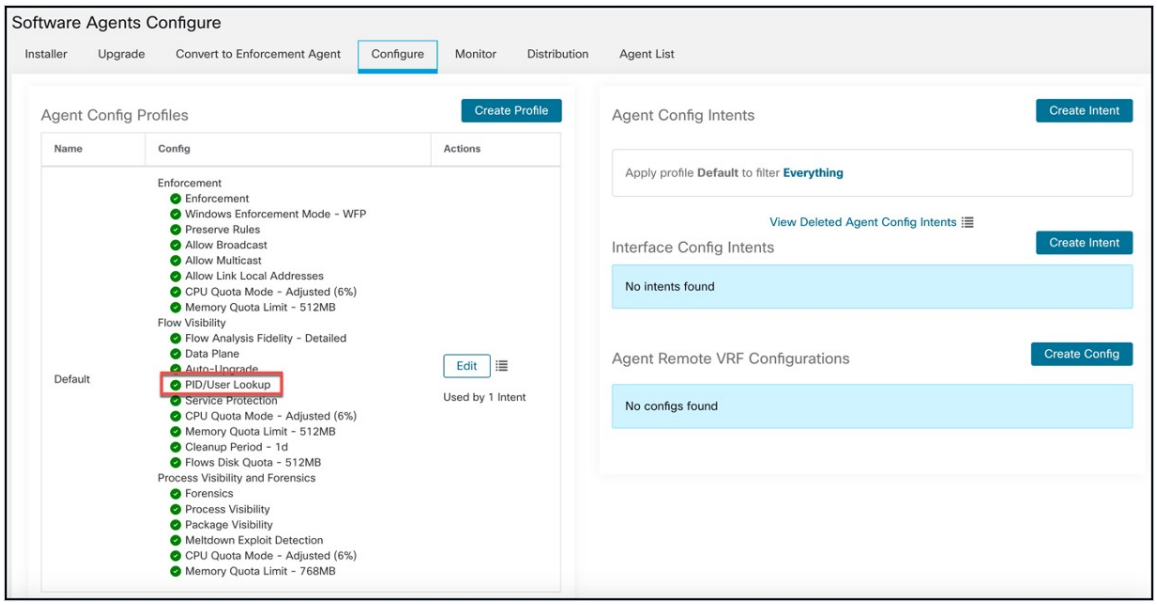
3. Site administrators have access to all the components in the Agent Configuration page that includes specifying interface configuration intents, remote virtual routing and forwarding configurations.

Configure Software Agents

On the Software Agent Configuration page, configure the software agents to create intents that are associated with either an **Inventory Filter** or a **Scope**. For each agent, apply the first matching intent. For more information, see Manage Inventory for Secure Workload.

Note For any Secure Workload deployment, use the default agent configuration on all agents that are not associated with any specific configuration profile.

Figure 7: Software Agent Configuration



Create an Agent Configuration Profile

Before you begin

See Requirements and Prerequisites for Configuring Software Agents, on page 57.

Procedure

Step 1 In the navigation pane, choose **Manage** > **Workloads** > **Agents**.

Step 2 Click the **Configure** tab.

Step 3 Click the **Create Profile** button.

Step 4 Enter a name for the profile and choose the scope where the profile is available.

Step 5 Enter the appropriate values in the fields listed in the following table.

Table 7: Creating Software Agent Configuration Profile Field Descriptions

Field	Description
Enforcement	
Enforcement	**Enable** - Enable policy enforcement on the agent. After you enable enforcement, the agent enforces the most recently received policy set. **Disable** (Default) - The agent does not enforce a policy. **Note** If you enable, disable and re-enable policy enforcement on the agent, it clears the firewall state and sets the catch-all default action to ALLOW.

Field	Description
Windows Enforcement Mode	On Windows workloads, agents can enforce network policies using: • **WFP** - Windows Filtering Platform (by directly programming WFP filters in the Windows Filter Engine). See Agent Enforcement on the Windows Platform in WFP Mode, on page 37. • **WAF** (Default) - Windows Advanced Firewall. See Agent Enforcement on the Windows Platform in WAF mode, on page 35.
Preserve Rules	**Enable** - Preserves existing firewall rules on the agent. **Disable** (Default) - Clears existing firewall rules before applying enforcement policy rules from Secure Workload. Behaviour of the Preserve Rules attribute is platform-specific. You can view the details of the attributes in the Preserve Rules section in each platform.
Allow Broadcast	**Enable** (Default) - Adds rules to the firewall to allow ingress and egress broadcast traffic on workload. **Disable** - Does not add any rules. The broadcast traffic drops if the default policy on the agent is DENY.
Allow Multicast	**Enable** (Default) - Adds rules to the firewall to allow ingress and egress multicast traffic on workload. **Disable** - Does not add any rules. The Multicast traffic drops if the default policy on the agent is DENY.
Allow Link Local Addresses	**Enable** (Default) - Adds rules to the firewall to allow link local addresses traffic on workload. **Disable** - Does not add any rules. The Multicast traffic drops if the default policy on the agent is DENY.
CPU Quota Mode	**Adjusted** (Default) - The CPU limit adjusts according to the number of CPUs on the system. For example, if there are 10 CPUs, set the CPU limit to 3%, the agents use only a total of 30% (measured by top). **Top** - The CPU limit value matches the top view on average. For example, if you set the CPU limit to 3% and there are 10 CPUs in the system, the CPU usage is 3%. It is a fairly restrictive mode, use it only when necessary. **Disable** - Disable the CPU limit feature. The agent uses CPU resources that used in the operating system. For more information, see Secure Workload Data Sheet.
CPU Quota Limit (%)	Specify the actual limit in percentage of the system processing power.
Memory Quota Limit (MB)	Specify the memory limit (in MB) for processes. If the process hits this limit, it restarts.

Field	Description
	Flow Visibility
Flow Analysis Fidelity	**Conversations** (Default) - Enable conversation mode on all agents. **Detailed** - Enable detailed mode on all agents.
Data Plane	**Enable** (Default) - Enable the agent to send reports to the cluster. **Disable** - Disable the agent's reports.
Auto-Upgrade	**Enable** (Default) - Automatically upgrade the agent when a new package is available. **Disable** - Do not automatically upgrade the agent.
PID/User Lookup	**Enable** - Process ID (PID) and User Lookup in agents. Set the **Flow Analysis Fidelity** option to detailed mode for PID and User Lookup. When you enable this feature, the agent associates network flows with running processes and users in the workload. During the process, note that some flows that might not be associated with any process even after you enable the configuration. **Disable** (Default) - Do not enable process ID and User Lookup in agents. **Note** User Lookup is not supported on Windows Server 2008 R2.
Service Protection	**Enable** - Enable service protection on the agent. When enabled, the agent ensures it prevents users from disabling the service, from uninstalling the agent, and from restarting the service. However, after disabling the service protection, you can continue to stop or can uninstall the agent. **Note** • Do not disable service protection for normal auto upgrade of an agent. • Do not enable service protection for manual upgrade of an agent. • Service protection blocks any forced upgrades, such as using the installer script - forceUpgrade option. • Any system-initiated upgrade works when you enable the service protection. **Disable(*)**-By default, disable the service protection on the agent. **Detailed** (Default) - Enable detailed mode on all agents. **Note** This feature is available only for Windows agent.

Field	Description
CPU Quota Mode	**Adjusted** (Default) - Adjust the CPU limit according to the number of CPUs on the system. For example, if there are 10 CPUs in the system, set the CPU limit to 3%. Choose this mode to allow the agent to use a total of 30% (measured by top). **Top**- The CPU limit value matches the top view on average. For example, set the CPU limit to 3% for the 10 CPUs in the system, the CPU usage is only 3%. It is a fairly restrictive mode and uses it only when necessary. **Disable** - Disable the CPU limit feature. The agent uses CPU resources that are used in the operating system.
CPU Quota Limit (%)	Specify the actual limit in percentage of the system processing power that the agent can use.
Memory Quota Limit (MB)	Specify the memory limit in MB that the process allows to use. If the process hits this limit, the process restarts.
Cleanup period (days)	**Enable** - Enable automated cleanup on the agent. Enter the number of days after which remove the inactive agent. **Disable** (Default) - Do not enable automated cleanup on the agent.
Flows Disk Quota (MB)	Enter the maximum size limit (in MB) for storing the flow data. If the Flows Disk Quota field is: • 0: The agents do not store offline flows locally. • Blank: Enable the Flows Time Window field. After you enter the duration in the Flows Time Window, the Flows Disk Quota field automatically sets the value to16 GB. You can either choose the Flows Disk Quota or the Flows Time Window option for flow log buffering in case of connectivity break between the agent and the cluster. For example, if you have set the Flows Time Window as one hour and the agent is unable to communicate with the cluster, the agent stores flow data for the last hour. Any flow data locally stored on the workload beyond the last hour is overridden by newer logs. Specify in MB the total size limit of stored flow data.

Field	Description
Flows Time Window (Hours)	Specify in hours how long the agent must capture and store flows locally. Choose either **Flows Disk Quota** or **Flows Time Window**; it's either size-based or time-based rotation. On choosing Flows Time Window, set the Flows Disk Quota to 16 GB. Setting Flows Disk Quota to 0 disables this feature. The flow data is rotated when it reaches either size limit or time limit. This field is displayed only when there is no value that is entered in the Flows Disk Quota field. Enter the duration, in hours, for the agents to capture the flows and store them locally. • After the connectivity to the agents is restored, the agents send the live flow data. • While sending the live flow data, the agents also initiates to upload the buffered telemetry data. The telemetry data is sent in small packets at regular intervals. • Depending on the size of the buffered telemetry data and transmission transfer speed, it takes multiple intervals to send all the buffered data. • The agents progressively deletes the locally stored flow data. Remove the outdated flow data that is stored locally after it reaches the configured size or time limit.
Process Visibility and Forensics	
Forensics	**Enable** - Enable forensics on the agent. This feature consumes extra CPU cycles that are specified in the CPU limit below. For example, if the CPU limit is 3% and you enable this feature, the agent uses up to 6% in total. **Disable** (Default) - Disable forensics on the agent.
Meltdown Exploit Detection	**Enable** - Enable Forensics and Meltdown exploit detection on the agent. For more information, see Side Channel in the Compatibility. **Disable** (Default) - Disable Meltdown exploit detection on the agent.
CPU Quota Mode	**Adjusted** (Default) - Adjust the CPU limit according to the number of CPUs on the system. For example, set the CPU limit to 3% with 10 CPUs in the system. Choose this mode to use a total of 30% (measured by top). **Top** - The CPU limit value matches the top view on average. For example, set the CPU limit to 3% with 10 CPUs in the system, the CPU usage remains at 3%. Use this restrictive mode only if necessary. **Disable** - Disable the CPU limit feature, the agent uses CPU resources permissible by the operating system.
CPU Quota Limit (%)	Specify the actual limit, in percentage, of the system processing power the agent can use.
Memory Quota Limit (MB)	Specify the memory limit (in MB). If the storage limit goes beyond the specified limit, the process restarts.

Step 6 Click **Save**

What to do next

Associate the created profile with an agent configuration intent. For more information, see Creating an Agent Config Intent, on page 64.

Creating an Agent Config Intent

Before you begin

- See Requirements and Prerequisites for Configuring Software Agents, on page 57.
- Create an agent config profile. See Create an Agent Configuration Profile, on page 59.

Procedure

Step 1 In the navigation bar on the left, click **Manage** > **Agents**.

Step 2 Click the **Configure** tab.

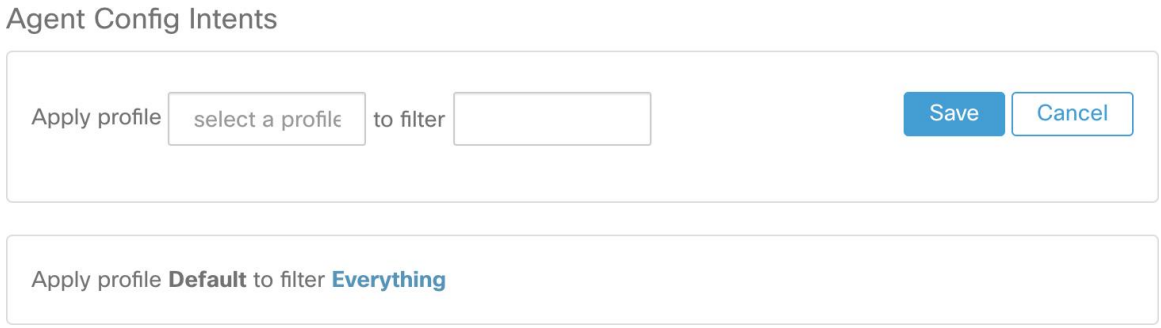
Step 3 Click the **Create Intent** button next to the **Agent Config Intent** heading.

Step 4 Enter the appropriate values in the fields listed in the table below:

Field	Description
Profile (required)	Enter the name of an existing profile and select it from the dropdown menu.
Filter (required)	Enter the name of an existing filter or scope or select *Create new filter* from the dropdown menu. See Filters for more information on creating filters.

Step 5 Click **Save**.

Figure 8: Agent Config Intents



Creating a Remote VRF configuration for agents

This is the recommended way to assign VRFs for Secure Workload software agents. Using this configuration, Secure Workload appliance assigns VRFs to software sensors based on the source IP address and source port seen for those agent on connections to Secure Workload appliance.

Procedure

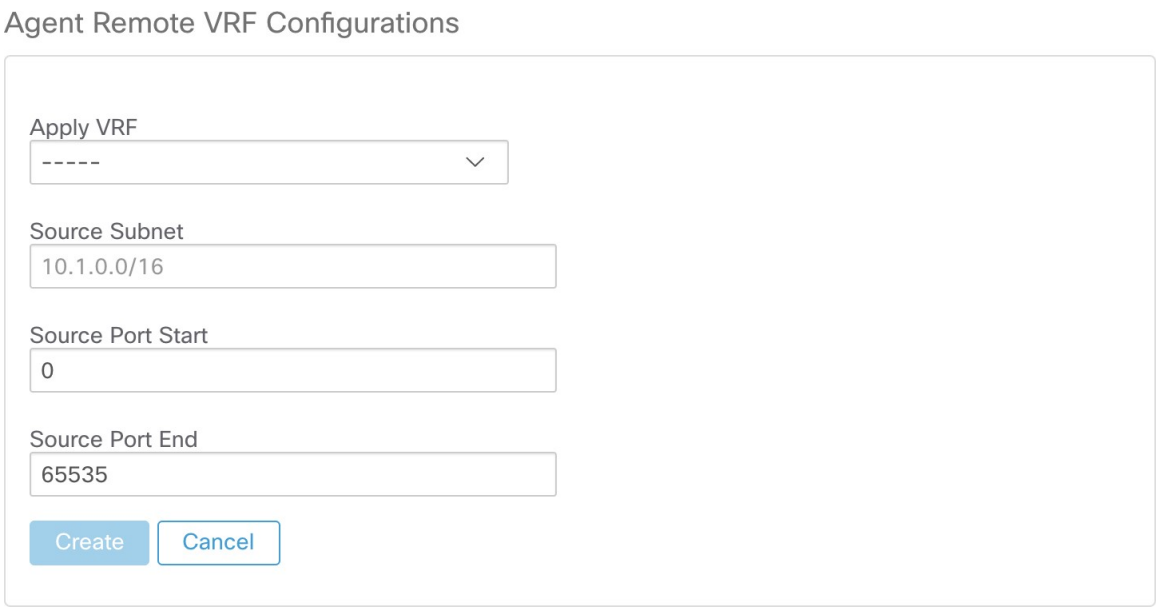
Step 1 In the navigation bar on the left, click **Manage** > **Agents**.

Step 2 Click the **Configure** tab.

Step 3 Click the **Create Config** button next to the **Agent Remote VRF Configurations** heading.

Step 4 Enter the appropriate values in the fields and click **Save**.

Figure 9: Remote VRF configuration



Create an Interface Configuration Intent

We recommend assigning virtual routing and forwarding (VRFs) to agents in using Remote VRF configuration settings. In rare cases, when agent hosts have multiple interfaces that must be assigned to different VRFs, you can choose to assign them VRFs using Interface Configuration Intents.

Procedure

Step 1 Navigate to **Manage** > **Agents**.

Step 2 Click the **Configure** tab.

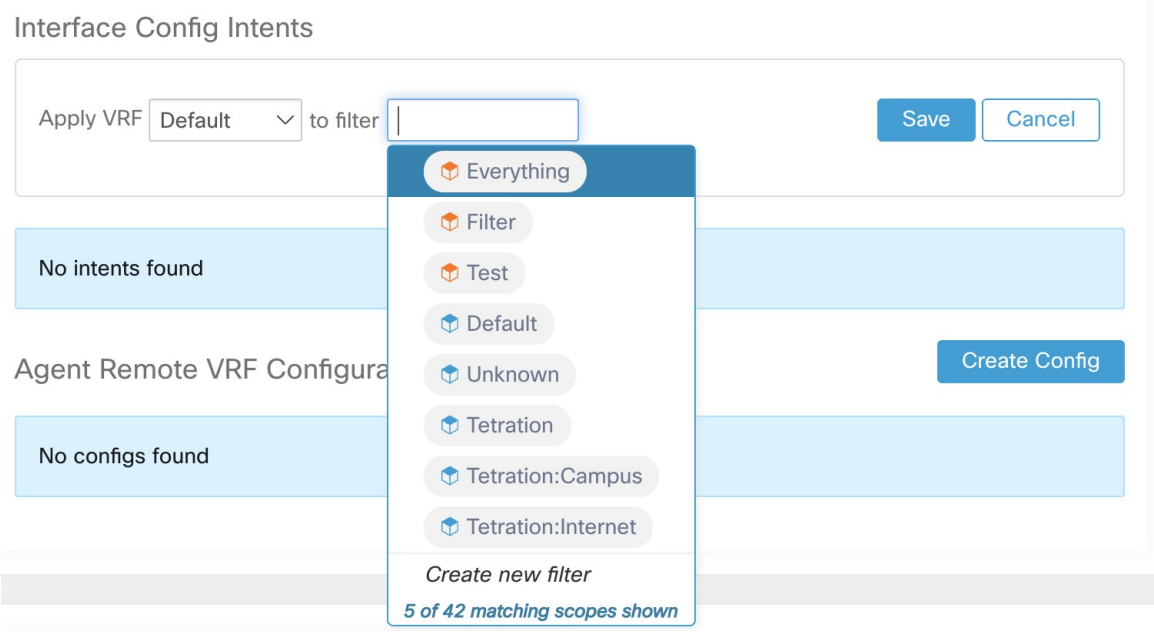
Step 3 Click the **Create Intent** button next to the **Interface Config Intent** heading.

Step 4 Enter the appropriate values in the fields listed in the table:

Field	Description
VRF	Choose a VRF from the drop-down list (required).
Filter	Enter the name of an existing filter or scope or choose *Create a new filter* from the drop-down list (required). For more information, see Filters .

Step 5 Click **Save**.

Figure 10: Interface Configuration Intents



Note When you delete an interface with a higher priority config intent, the agents do not fall back to the default catch all intent.

View Detailed Agent Status in the Workload Profile

Procedure

Step 1 Follow the steps above to check Agent status.

Step 2 On the Enforcement Agents page, click **Agent OS Distribution**. Select an operating system and click filter image on the top-right corner of the box.

Step 3 On the Software Agent List page, agents with selected operating system Distribution is listed.

Step 4 Click on **Agent** for the agent details, and click IP address. On the Workload Profile page, you can view details of the Host Profile, Agent Profile and agent specific details, such as Bandwidth, Long- lived Processes, Packages, Process Snapshot, Configuration, Interfaces, Stats, Policies, Container Policies and so on.

Step 5 Click **Config tab** to see the configuration on the end-host.

Step 6 Click **Policies tab** to see the enforced policies on the end-host.

Figure 11: Workload Profile - Config

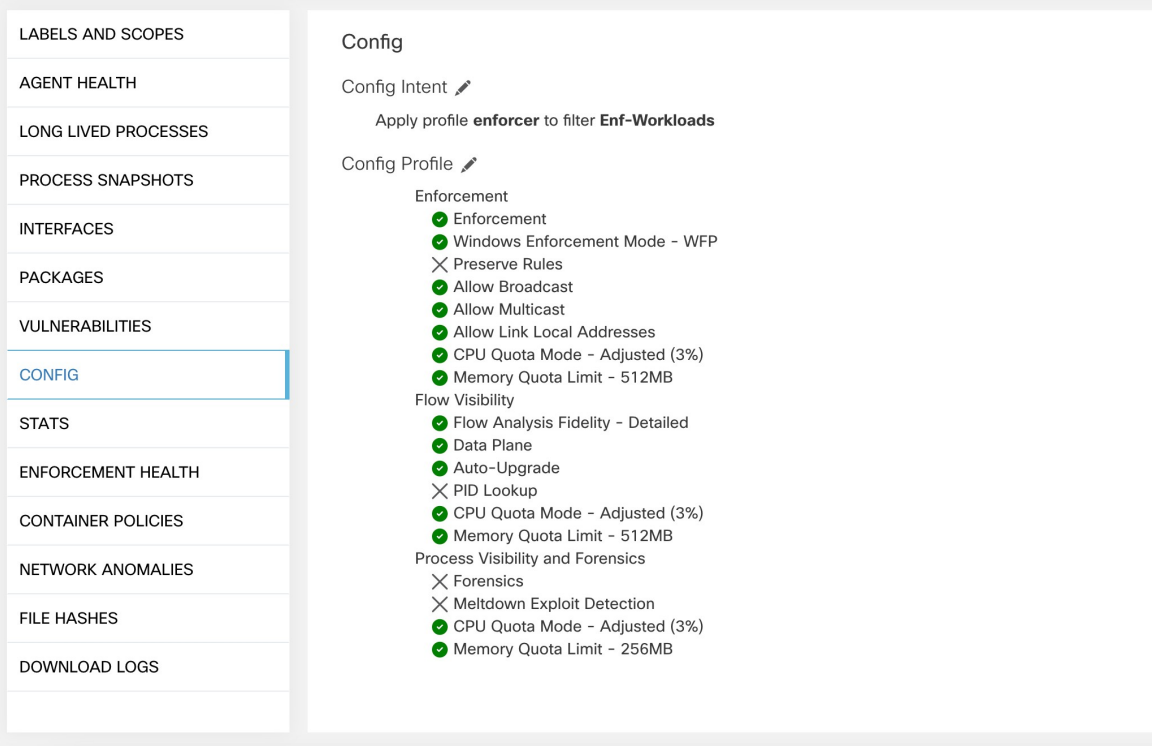
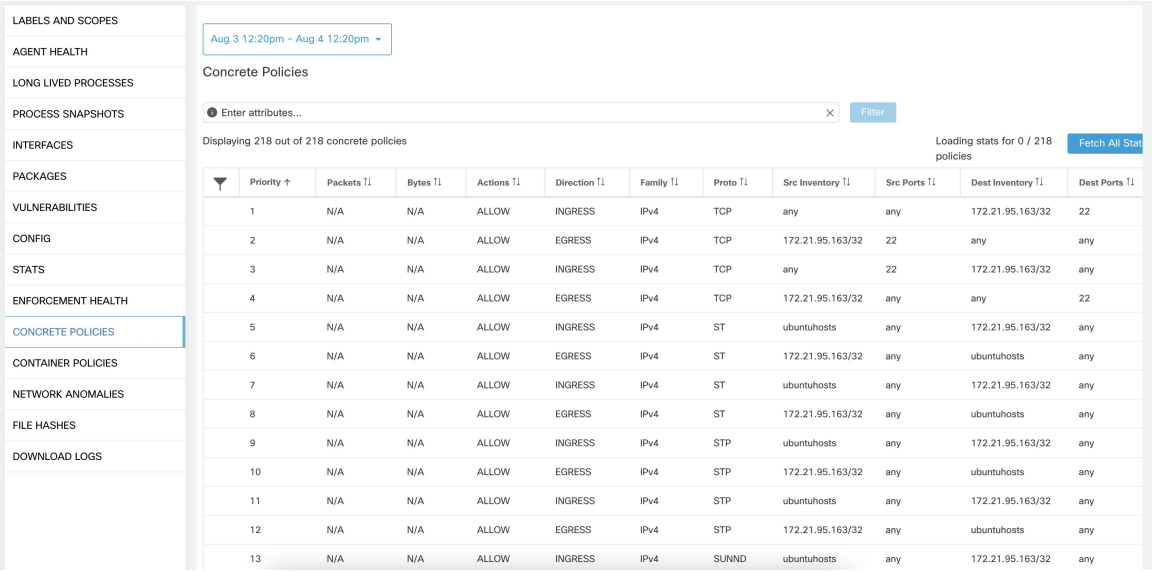


Figure 12: Workload Profile - Policies



Note **Fetch All Stats** is not supported on Windows agent hosts, which is used to provide statistics for individual policies.

Rehoming of Agents

Rehoming of agents is the method to move users from On-premises to SaaS or SaaS to On-premises.

User Roles

- Site Administrator
- Customer Support Representative

You can migrate to or from a SaaS environment, especially, when you move from SaaS to On-premises, you must work with an internal support team.

Workflow

- Enter the Activation Key, Sensor virtual IP, and Sensor certificate authority (CA) and Enable Rehoming, on page 68.
- Select Agents to Rehome, on page 70.
- Disable Rehoming, on page 70.

Note At any given time, you can move an agent to only one destination. We recommend that you Disable Agent Rehoming after you move the agent.

Enable Rehoming

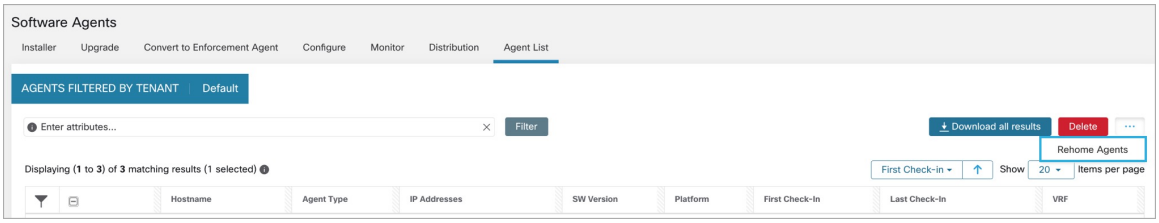
Procedure

Step 1 In the left navigation menu, click **Manage** > **Workloads** > **Agents**.

Step 2 Click the **Agent List** tab.

Step 3 Click the menu icon and select **Rehome Agents**.

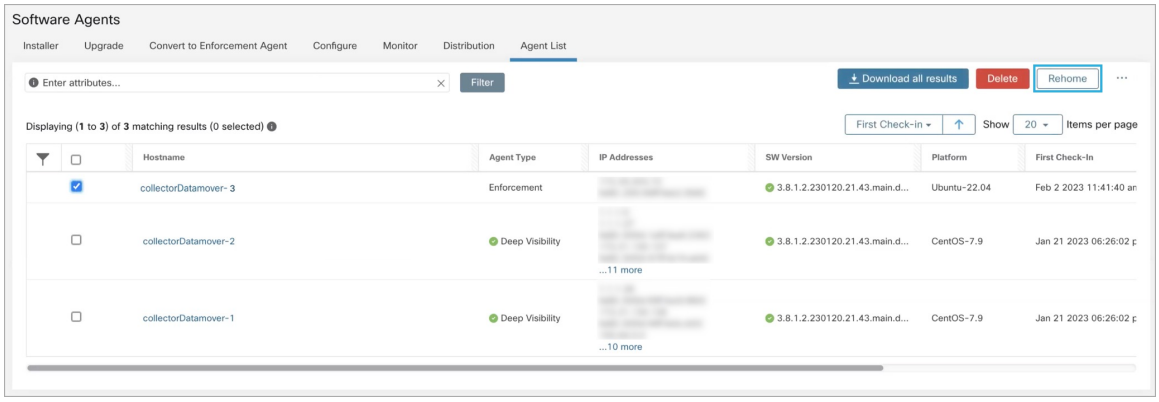
Figure 13: Rehome Agents



Step 4 On the **Agent Rehoming** window, fill in the following details:

Field	Description
Destination Scope Activation Key	**a.** Navigate to **Manage** > **Workloads** > **Agents**. **b.** Click the **Installer** tab. **c.** Select **Manual install using classic packaged installers**. **d.** Click **Next**. **e.** Click **Agent Activation Key**. **f.** Copy the **Key** value and paste it into the **Destination Scope Activation Key** field.
Destination Sensor VIP	**a.** Navigate to **Platforms** > **Cluster Configuration**. **b.** Copy the **Sensor VIP** and paste it into the **Destination Sensor VIP field**.
HTTPS proxy	Enter a proxy domain or address if the agent needs to use a proxy for outbound communication.
Destination Sensor CA Cert	**a.** Navigate to **Platforms** > **Cluster Configuration**. **b.** Click **Download Sensor CA Cert**.

Figure 14: Enable Agent Rehoming



Step 5 Click **Enable Agent Rehoming**.

The configuration is saved. The Rehome button appears at the top right.

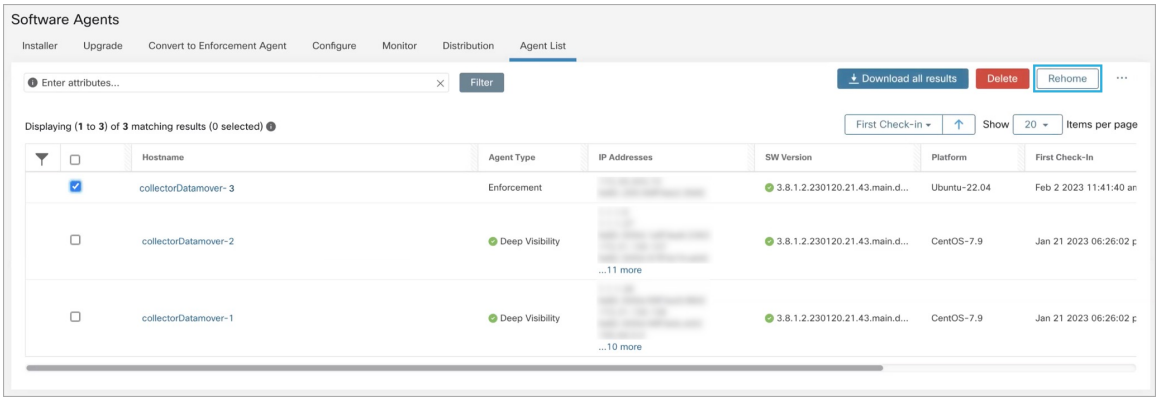
Select Agents to Rehome

Procedure

Step 1 Select an agent.

Step 2 Click **Rehome**.

Figure 15: Select Agents to Rehome



Step 3 Click **Yes** to confirm.

Disable Rehoming

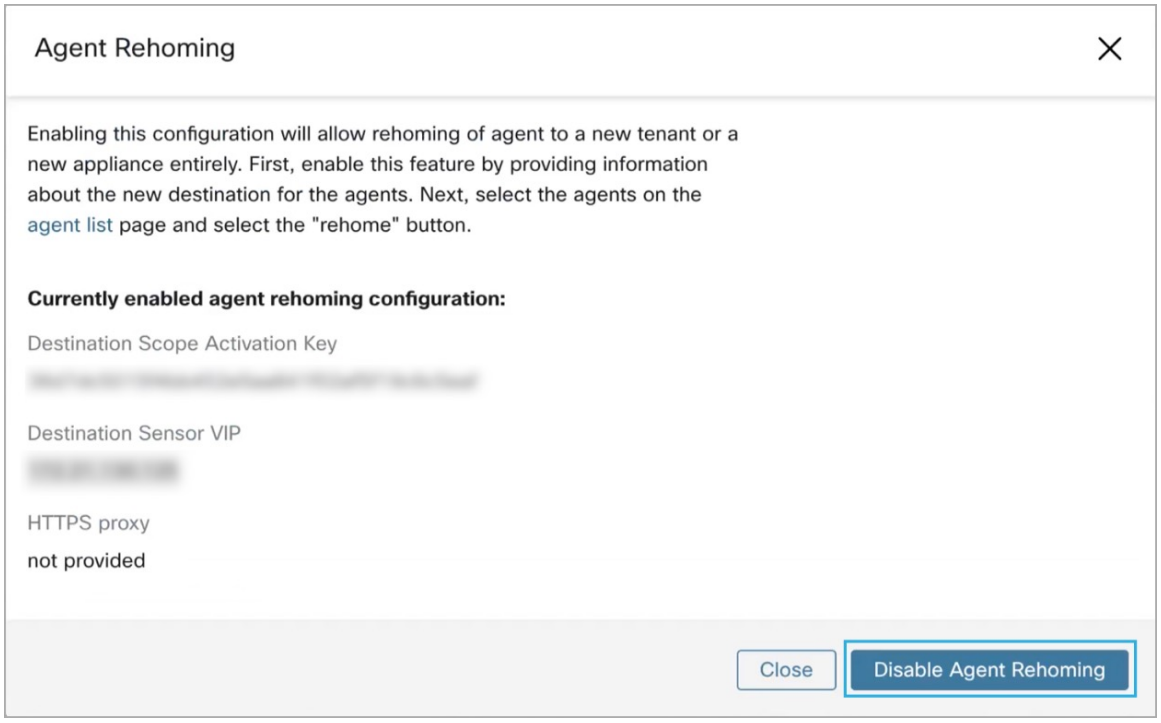
Note If there are multiple users rehoming to or from SaaS, the site administrator has to move each tenant or an appliance separately. To do this, disable Rehoming to clear the settings, and then enable Rehoming for the new user.

Procedure

Step 1 Click the menu icon and choose **Rehome Agents**.

Step 2 On the **Agent Rehoming window**, click **Disable Agent Rehoming**.

Figure 16: Disable Agent Rehoming



Generate Agent Token

In the agent configuration profile, you can enable service protection to prevent uninstallation, disabling, and stopping Windows agent services. To perform any changes to the agents, you can disable this protection on the agent configuration profile. However, if you are unable to disable the protection because of connectivity issues, you can generate an agent token to disable the service protection on workload. The token is valid for 15 minutes.

Supported roles to generate and retrieve agent tokens:

- **Site administrators**: For clusters or tenants.
- **Customer support**: For tenants.
- **Agent installer**: For agent-specific tokens.

Note You can generate time-based agent tokens only for Windows OS-based software agents.

To generate and download agent tokens, perform these steps:

Procedure

Step 1 In the navigation pane, click **Manage** > **Workloads** > **Agents** > **Agent List**.

Based on your requirement, you can choose one of the agent token types—Cluster, tenant, or agent-specific. For the agent-specific token, go to Step 5.

Step 2 Click the menu icon and choose **Agent Token**.

Note The **Agent Token** option is only visible for site administrators or customer support user roles.

Step 3 Select a token type:

- Token For Cluster—This option is visible only to site administrators and the token is applicable for all the agents.
- Token For Tenant—Applicable for the agents under a selected tenant.

Step 4 To download the token key, click **Download Token**.

Step 5 To view and download token key details of a specific agent:

a) Go to the **Agent List** tab and click the required agent. Under **Agent Details** > **Agent Token**, you can view the token key and expiry details of the token.

b) To download the agent-specific token, click **Download Token**.

What to do next

After downloading the agent token file, run the following command on the agent to disable service protection: `"C:\Program Files\Cisco Tetration\TetSen.exe" -unprotect <token>`, where `token` is the downloaded agent token.

After the service protection is disabled using a token, it may be automatically re-enabled when the service restarts and connects to the Secure Workload cluster.

Host IP Address Change when Enforcement is Enabled

Changing the IP address on hosts when enforcement is enabled may have an impact if the host IP is seen in the host firewall rules and catch all is set to deny. In this scenario, the following steps are recommended to change the host IP address:

Procedure

Step 1 On the Secure Workload UI, create a new Agent Config Profile with enforcement disabled.

Step 2 Create Intent with list of hosts that need IP address change with their old and new IP address.

Step 3 Apply the newly created Agent Config Profile to the Intent and save the Intent.

Step 4 These selected hosts should have enforcement disabled.

Step 5 Change the IP address on these hosts.

Step 6 On the Secure Workload UI, update the filters in the scope with the new IP address of these hosts.

Step 7 Verify the IP address change from Agent Workload Profile page “Interfaces” tab. In the “Policies” tab, make sure policies are generated with new IP address.

Step 8 Remove the Intent/Profile created above.

Step 9 If the original Agent Config Profile for the scope had enforcement disabled, then enable enforcement.

Upgrading Software Agents

Upgrade Agents from UI

Agents can be upgraded using Agent Config Intent workflow as described here - Software Agent Config. While configuring an agent config profile, there is an **Auto Upgrade** option which can be enabled or disabled. If the option is enabled, the agents matching inventory filter criteria are automatically upgraded to the latest available version.

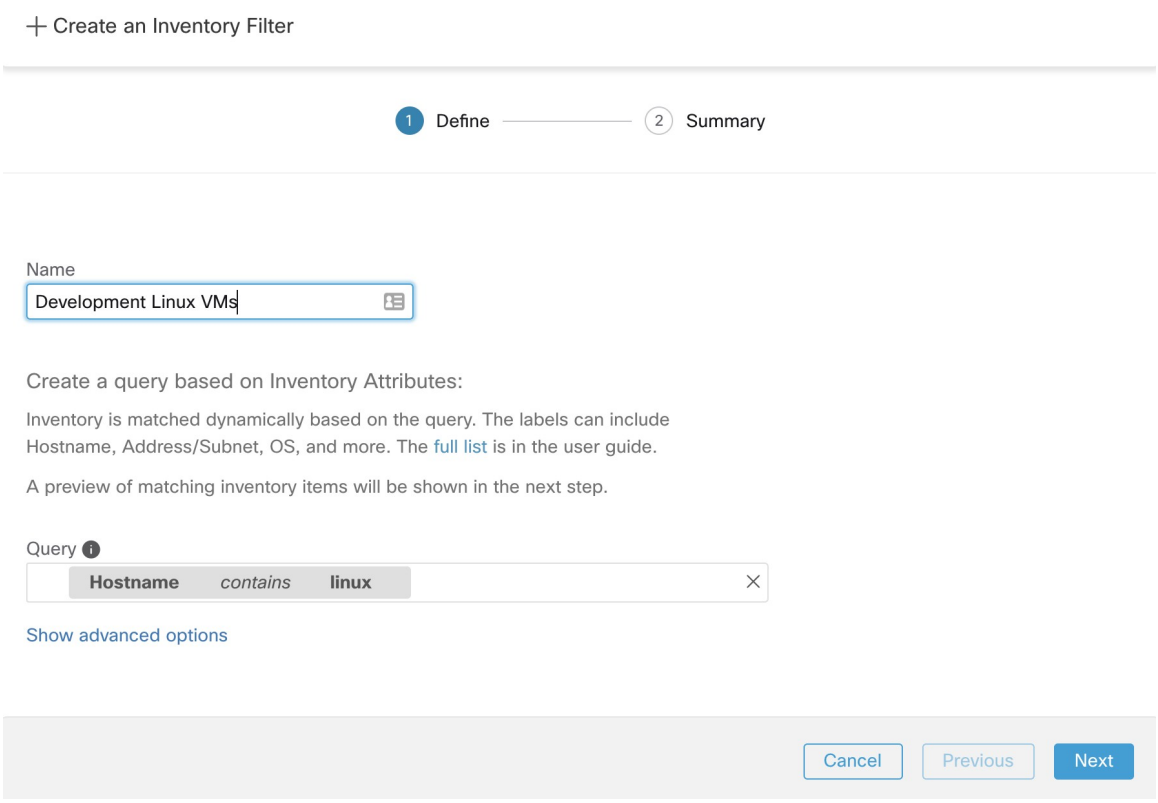
On the **Software Agents** > **Agent List** page, software agents with outdated versions are highlighted with a warning sign under the **SW Version** column. It is important to upgrade these agents to the latest available version on the cluster.

To use software agent config intent workflow to configure software agent upgrade:

Procedure

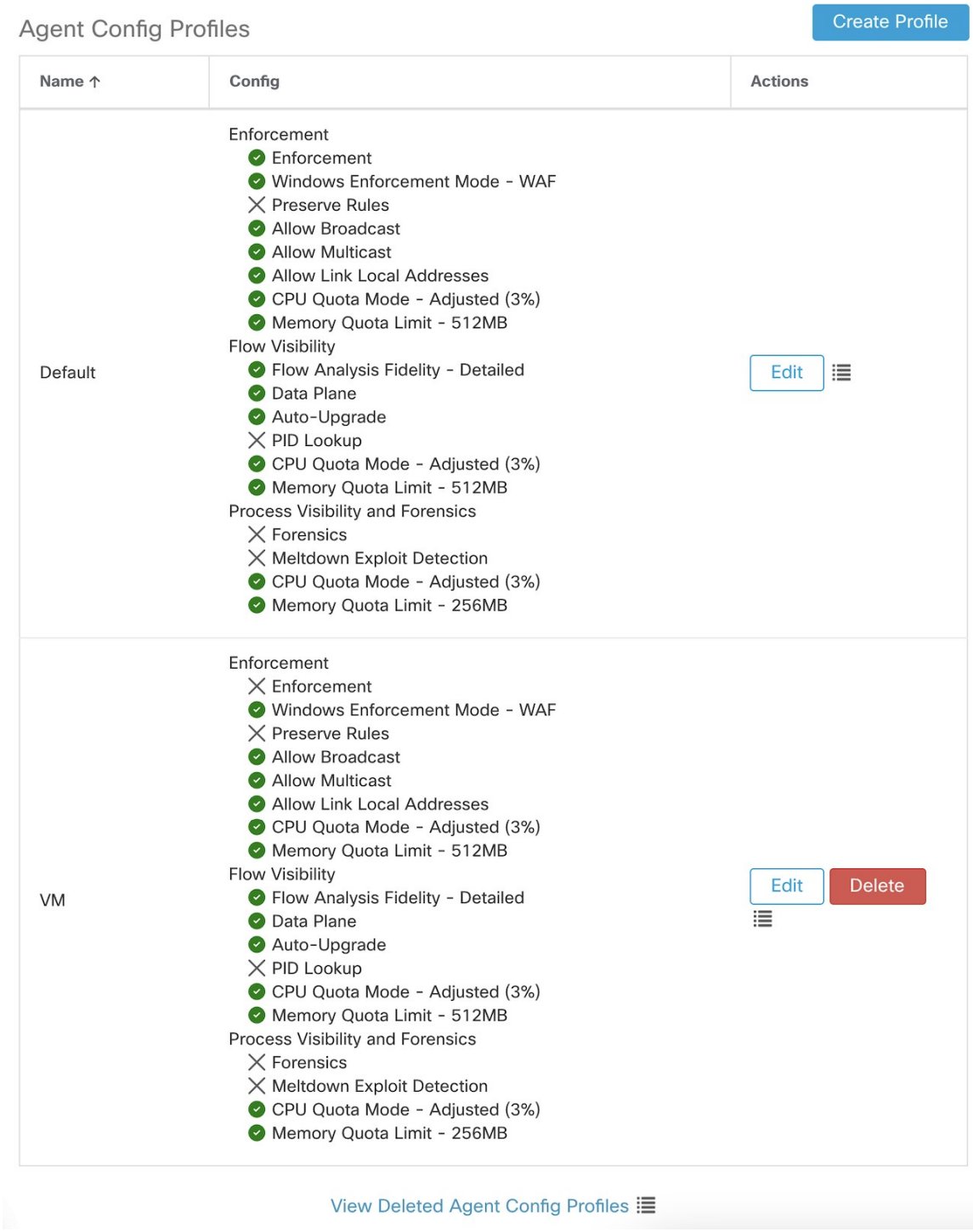
Step 1 Create an inventory filter on the **Inventory Filters** page. For more information, see Filters.

Figure 17: Inventory Filter



Step 2 Create an Agent Config profile for the agents selected by the inventory filter. Optionally, you can enable the **Auto Upgrade** option to automatically upgrade the selected agents.

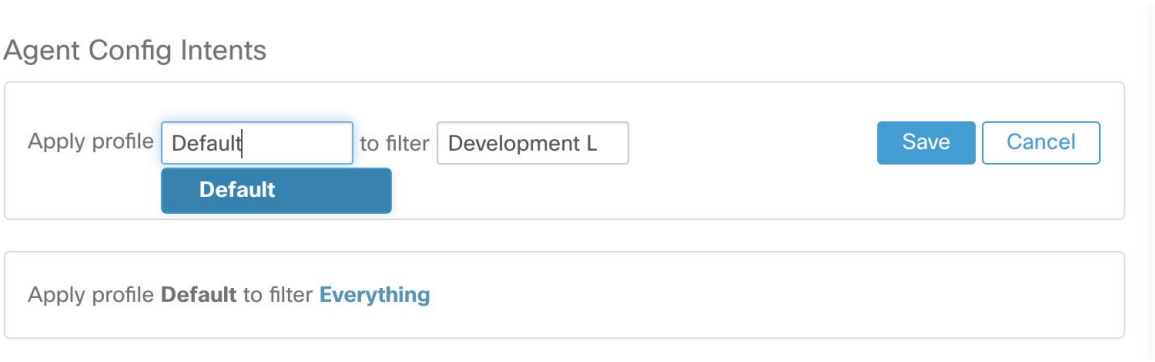
Figure 18: Agent Config



Step 3 Create an agent config intent to apply the config profile to the agents selected using inventory filter. If the auto upgrade option is enabled, the selected agents are automatically upgraded.

It normally takes up to 30 minutes to upgrade an agent after an agent profile is applied to them.

Figure 19: Agent Config Intent



Note Auto Upgrade setting in the default agent profile applies to ERSPAN.

Manual Agent Upgrade

The following section explains how to manually upgrade agents without using the Sensor Config intent workflow.

Procedure

Step 1 In the left navigation pane, click **Manage** > **Workloads** > **Agents**.

Step 2 Click the **Upgrade** tab.
Deep visibility and enforcement agents are displayed and for each agent only newer versions to which it is upgradable are listed. By default, the latest version is selected.

Step 3 To filter specific agents, enter your search query in the filter box. For example, enter Platform = CentOS-7.6.

Step 4 Select the agents to be upgraded to the selected version and click **Upgrade**.

Note Under normal circumstances, allowing the agent to automatically upgrade is strongly recommended and is the only supported upgrade method. If you want to control the upgrade by manually downloading the latest version and directly deploying it to the agents which are running on workloads, ensure that you follow the safety precautions.

Upgrade Behaviour of Kubernetes/Openshift Agent

Agents installed on Kubernetes/Openshift nodes using the daemonset installer script are capable of self-upgrade. The upgrade process is controlled by either the auto-upgrade option or by manually triggering an upgrade for any node in the Kubernetes/Openshift cluster. The mechanism of the upgrade in this environment is to upgrade the Docker image in the daemonset specification, which means that an upgrade of one agent affects all agents covered by the daemonset, as explained in the next paragraph.

When a Daemonset Pod specification changes, Kubernetes/Openshift will trigger a graceful shutdown, fetch the new docker image(s) and start the Secure Workload agent pods on ALL nodes in the Kubernetes/Openshift

cluster. This will cause agents to be upgraded on other nodes, even if the policy to allow upgrades is applicable only to a subset of the nodes in the cluster.

If auto-upgrade is disabled for all nodes, manual upgrade is possible by downloading a new installer script and re-running the install. The installation script auto-detects the case of new installation vs upgrading an existing installation and will work to manually upgrade the daemonset pods when it detects an installation is already in place.

Removing Software Agents

Remove a Deep Visibility or Enforcement Linux Agent

RPM based installation:

1. Run command: **rpm -e tet-sensor**

Agent uninstallation event is communicated to the cluster and the agent will be marked as uninstalled on **Software Agent** page.

Manually delete the agent from UI on the **Software Agent** page or the user can enable automated cleanup or removal of the agent by turning on the **cleanup period** from agent config profiles.

Note By default, the **cleanup period** is turned off.

Ubuntu .deb based installation:

Fresh installation of Ubuntu agents now uses the native .deb format.

1. Run command: **dpkg –purge tet-sensor**

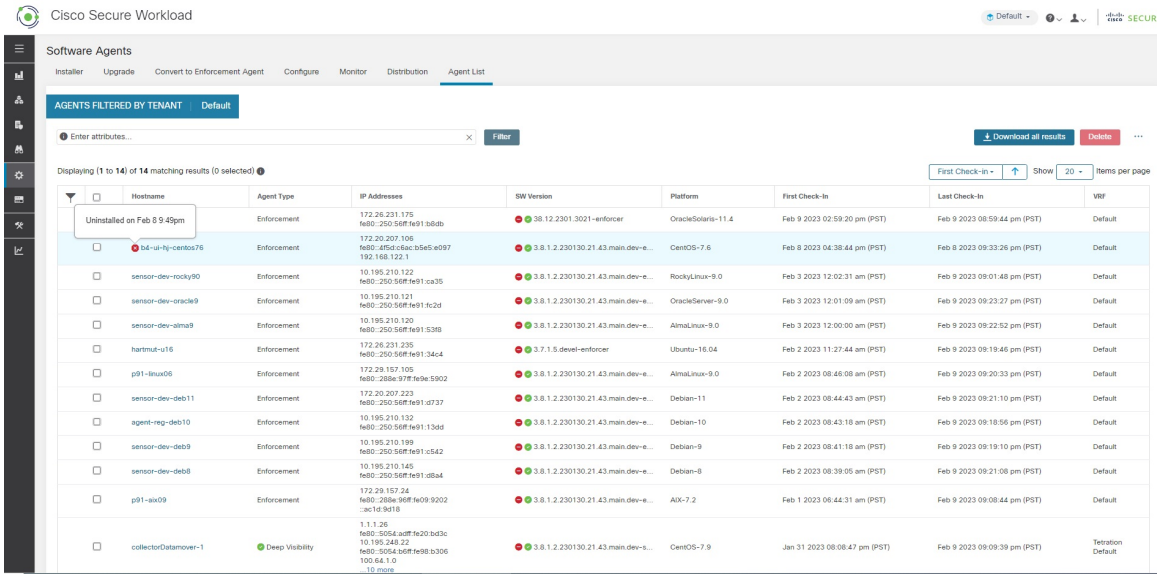
Agent uninstallation event is communicated to the cluster and the agent will be marked as uninstalled on **Software Agent** page.

Manually delete the agent from UI on **Software Agent** page or the user can enable automated cleanup or removal of the agent by turning on the **cleanup period** from agent config profiles.

Note

- By default, the **cleanup period** is turned off.
- During the agent operations, it is possible that some kernel modules will be loaded automatically by the kernel. For example, if enforcement is enabled in Linux, Netfilter modules might be loaded. Agents do not have a list of modules loaded by kernel. Therefore, during agent uninstallation, it cannot possibly unloaded the kernel modules.
- If enforcement agent applied a policy to the system firewall, uninstalling agent clears the applied policy and opens the system firewall.

Figure 20: Agent Uninstallation Alert



Removing a Deep Visibility/Enforcement Windows Agent

There are two options to uninstall Secure Workload agents:

Procedure

Step 1 Go to Control Panel / Programs / Programs And Features, and uninstall **Cisco Secure Workload Agent** (Cisco Tetration Agent).

Step 2 Alternatively, run the shortcut **Uninstall.lnk** within **'C:\Program Files\Cisco Tetration'**

Step 3 If enforcement agent applied a policy to the system firewall, uninstalling agent clears the applied policy, and opens the system firewall.

The Agent uninstallation event will be communicated to the cluster and the agent will be marked as uninstalled on **Software Agent** page.

Manually delete the agent from UI on the **Software Agent** page or the user can enable automated cleanup or removal of the agent by turning on the **cleanup period** from agent config profiles.

Note By default, the **cleanup period** is turned off.

Note

- If Npcap has been installed during agent installation, it will also get uninstalled.
- By default log files, config files and certs will not get removed during uninstall. If you'd like to remove them, run the shortcut **UninstallAll.lnk** in same folder.

Remove a Deep Visibility or Enforcement AIX Agent

Procedure

Run command: **'installp -u tet-sensor'**.

The Agent uninstallation event will be communicated to the cluster and the agent will be marked as uninstalled on the **Software Agent** page.

Manually delete the agent from UI on the **Software Agent** page or the user can enable automated cleanup or removal of the agent by turning on the **cleanup period** from agent config profiles.

Note

- By default, the **cleanup period** is turned off.
- The Deep Visibility Agent is controlled by System Resource Controller as tet-sensor. It is possible to start, stop, restart, and remove it. The service is made persistent with inittab as tet-sen-engine.
- The Enforcement Agent is controlled by System Resource Controller as tet-enforcer. It is possible to start, stop, restart, and remove it. The service is made persistent with inittab as tet-enf-engine.
- During the agent operations, it is possible that some kernel modules will be loaded automatically by the kernel. For example, if enforcement is enabled in AIX, ipfilter modules are loaded. Agents do not have a list of modules loaded by kernel. Therefore, during agent uninstallation, it cannot possibly unloaded the kernel modules.
- If enforcement agent applied a policy to the system firewall, uninstalling agent clears the applied policy and opens the system firewall.

Remove Universal Linux Agent

Procedure

Step 1 Run the uninstall script: **'/usr/local/tet-light/uninstall.sh'**

Step 2 Delete the agent from UI on the **Software Agent** page

Remove Universal Windows Agent

Procedure

Step 1 Run the uninstall script: **'C:\Program Files\Cisco Tetration\Lightweight Sensor\uninstall.cmd'**

Step 2 Delete the agent from UI on the **Software Agent** page

Remove an Enforcement Kubernetes or OpenShift Agent

Procedure

Step 1 Locate the original installer script or download a new script from the Secure Workload UI.

Step 2 Run the uninstall option: **install.sh –uninstall**. The same considerations apply as during the install.

- Only supported on Linux x86_64 architectures.
- Either ~/.kube/config contains an admin credentials user or use the –kubeconfig option to point to the kubectl admin credentials file.

Step 3 Delete the agents for all the Kubernetes nodes from UI on the **Software Agent** page

Remove a Deep Visibility Solaris Agent

Procedure

Step 1 Run command: `pkg uninstall tet-sensor`

Step 2 Delete the agent on the **Software Agent** page.

Data collected and exported by workload agents

This section describes the main components of a software agent, how it is registered with backend services, what data are collected and exported to the cluster for analytical purposes.

Registration

After the agent has been successfully installed onto the system, it needs to register with the backend services to obtain a valid unique identifier. The following information is sent in the registration request:

- Hostname
- BIOS-UUID
- Platform information (such as CentOS-6.5)
- Self-generated client certificate (generated with openssl command)
- Agent type (visibility or enforcement.)

If the agent fails to obtain a valid id from the server, it will keep retrying until it gets one. It is very important that the agent is registered, otherwise all the subsequent communication with other services (such as collectors) will be rejected.

Agent upgrade

Periodically (around 30 minutes), the agent sends a message to backend service to report its current version. The backend service uses the agent's id and its current version to decide whether a new software package is available for the agent. The following information is sent:

- Agent's id (obtained after successful registration)
- Current agent's version

Config server

Agents export the following information to the configured config server:

- Hostname
- Agent's id (obtained after successful registration)
- List of interfaces, each includes:
  1. Interface's name
  2. IP family (IPv4 or IPv6)
  3. IP addresses
  4. Netmask
  5. Mac addresses
  6. Interface's index

As soon as any interface property changes (such as an IP address of an existing interface changes, or a new interface comes up), this list is refreshed and reported to the config server.

Network Flow Information

Network flow information is the summarization of all packets flowing through the system. There are two modes of capturing flow information: Detailed and Conversation. By default, the **Conversation** mode is used to capture the flow information. The captured flows are exported to a collector and the exported information includes:

- Flow identifier: Uniquely identify the network flow. It includes the general information such as: IP protocol, source and destination IP, and layer 4 ports.
- IP Information: Contains information that is seen in the IP header, such as: TTL, IP flags, Packet ID, IP options, and Fragmentation flags.
- TCP Information: Contains information that is seen in the TCP header, such as: sequence number, Ack number, TCP options, Rcvd windows size.

- Flow Information: Statistics of the flow (such as total packets, total bytes, TCP flags statistics, packet length statistics, and socket statistics), interface index from which the flow was observed, start time and end time of flow.
- In a K8s environment, the agent captures network flows from pods and hosts, and then correlates the flows and reports as related flows. This is qualified with the following CNIs:
  - Calico
  - Flannel
  - Weave
  - AKS/GKE/AWS VPC CNI
  - Openshift CNI
  - Cilium CNI

Note Network flows are captured from pods and hosts, however, the correlation of flows is not possible when Cilium CNI is used.

In Conversation mode, the agent exports only TCP flows that are bidirectional in nature along with other connectionless flows. Conversation mode is supported for Windows, AIX, and Linux platforms. For more information on Conversation mode, see Conversation Mode.

Note
- In K8s environment, correlation of Pod or Host flows are not done in Conversation mode.
- In either of the modes, agents do not export the following flows:
  - ARP/RARP conversations
  - Agent's flows to collectors

Machine information

Machine info describes all the processes running on the host. In addition, it contains network information that is associated with the processes and the command used to launch the processes. Machine info is exported every minute and includes the following information:

- Process ID
- User ID: owner of the process
- Parent Process ID
- Command string used to launch the process
- Socket information: protocol (such as UDP or TCP), address type: IPv4 or IPv6, source and destination IP, source and destination port, TCP state, process's start and end time, path to process binary
- Forensic information: for more information, see the section Compatibility.

Agent statistics

Agent keeps track of various statistics, including system's statistics and its own, such as:

- Agent's start time and uptime
- Agent's run time in user mode and kernel mode
- Number of packets received and dropped
- Number of successful and failed SSL connections
- Total flow packets and bytes
- Total exported flows and packets to collectors
- Agent's memory and CPU usage

Enforcement Alerts

There are three types of enforcement alerts:

- Agent Reachability

  This alert detects when the agent is not reachable. This alert triggers if the agent has not communicated with the Secure Workload cluster for more than the configured number of seconds.

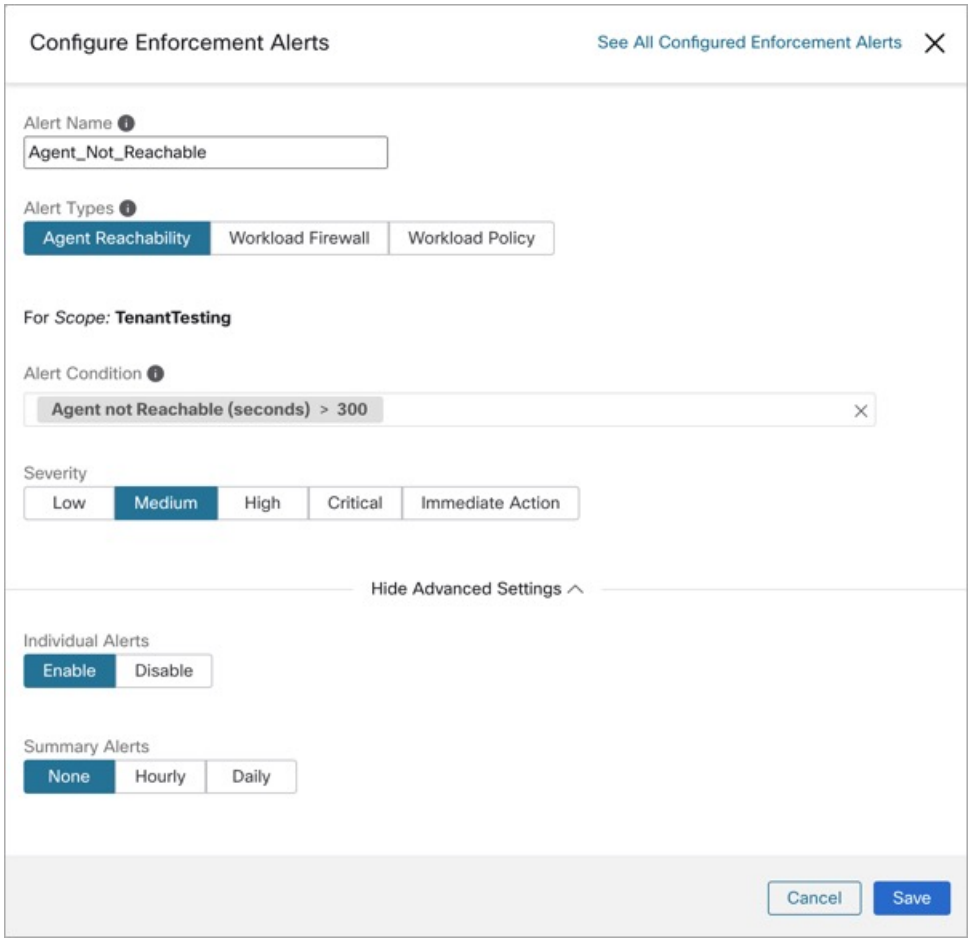
- Workload Firewall

  This alert triggers if enforcement is configured on a workload but the workload Firewall is detected to be off, since this condition will prevent Secure Workload Agent from enforcing traffic policies.

- Workload Policy

  This alert triggers if the workload firewall rules are different from the Secure Workload policies applicable to this workload (the workload's "concrete policies".)

Figure 21: Enforcement Alerts Types



You can set the Severity of the alert as well as other per-type configuration parameters.

To configure enforcement alerts, see Configure Alerts.

Figure 22: View Configured Enforcement Alerts

Enforcement UI Alerts Details

Figure 23: Enforcement alert details

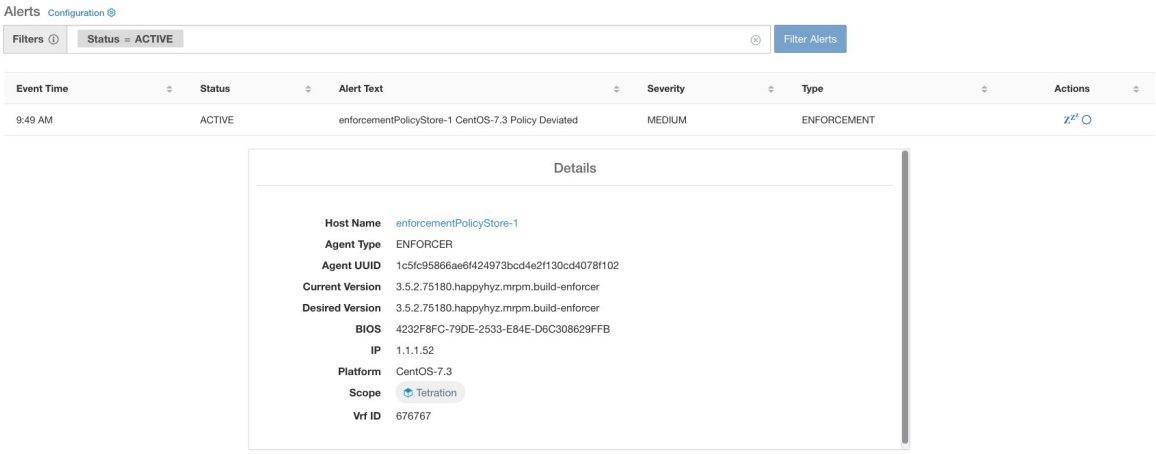
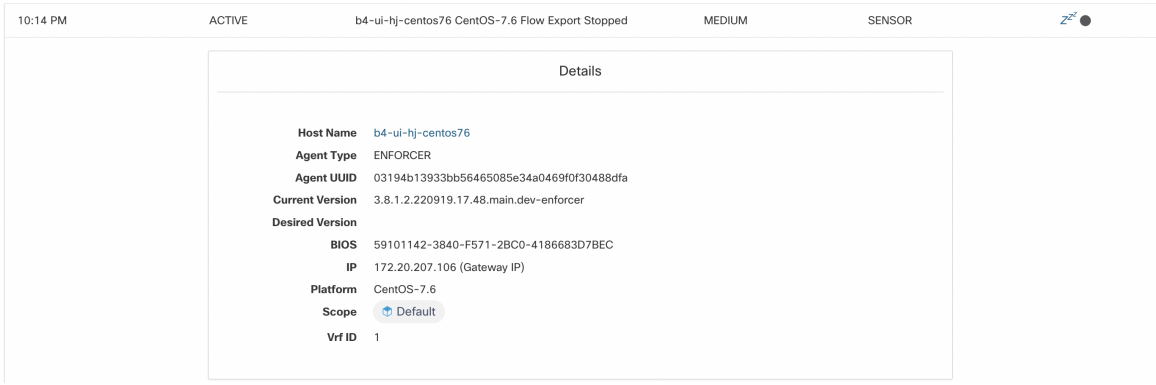


Figure 24: Enforcement alert details when proxy is enabled on the host



Enforcement Alert Details

See Common Alert Structure for general alert structure and information about fields. The *alert_details* field is structured and contains the following subfields for enforcement alerts

Field	Alert Type	Format	Explanation
AgentType	*all*	string	"ENFORCER" or "SENSOR" depending on the installed type
HostName	*all*	string	Host name on which the agent is deployed
IP	*all*	string	IP address of the node/gateway

Field	Alert Type	Format	Explanation
Bios	*all*	string	BIOS UUID of the node
Platform	*all*	string	Platform/OS information of the node
CurrentVersion	*all*	string	Software version of the agent on the node
DesiredVersion	*all*	string	Software version desired for the agent
LastConfigFetchAt	*all*	integer	Unix timestamp of when the agent last sent https request

Example of alert_details for an enforcement alert

```
{
  "AgentType":"ENFORCER",
  "Bios":"72EF1142-03A2-03BC-C2F8-F600567BA320",
  "CurrentVersion":"3.5.1.1.mrpm.build.win64-enforcer",
  "DesiredVersion":"",
  "HostName":"win2k12-production-db",
  "IP":"172.26.231.193",
  "Platform":"MSServer2012R2Standard"
}
```

Sensor Alerts

Sensor Alert Configuration provides the ability to configure different types of alerts, you can set the severity of the alert and types of configuration parameters.

For more information, see Alert Configuration Modal.

Note Starting Secure Workload 3.5, you can configure Sensor Alerts, using the *Alert Configuration Model*.

Configure Sensor alerts to report when an agent failes to upgrade. This alert triggers if the agent failed to upgrade to the needed version.

Configure Sensor alerts to detect when agent flow export must stop. This alert triggers if connectivity is blocked between the agent and the cluster, therefore preventing flows and other system information from sent or delivered.

Configure sensor alerts to detect when agent check_in times out. This alert triggers if the cluster does not received a check-in request from an agent after more than 90 minutes.

Figure 25: Configure Sensor Alerts

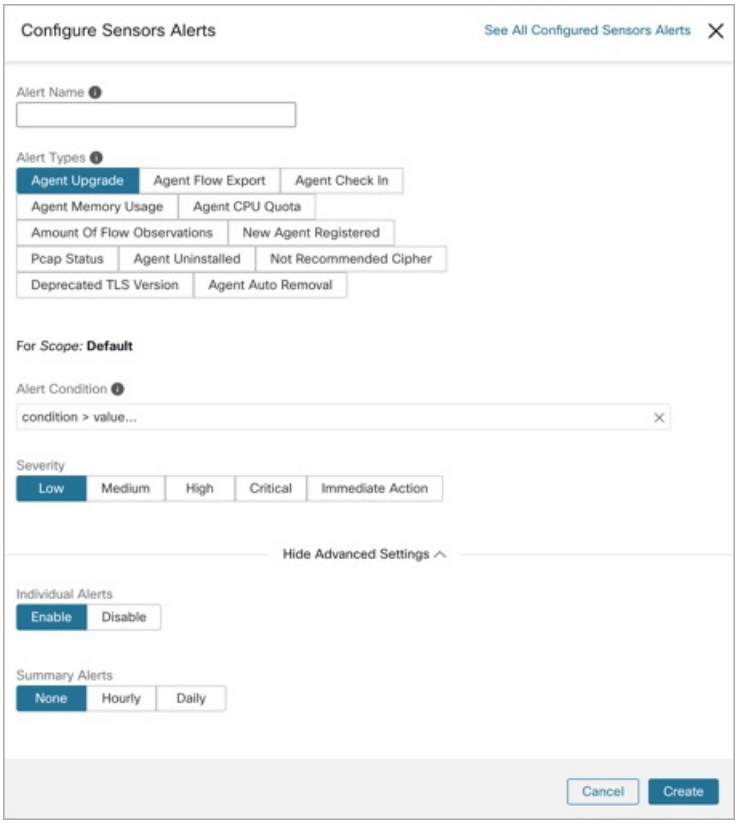
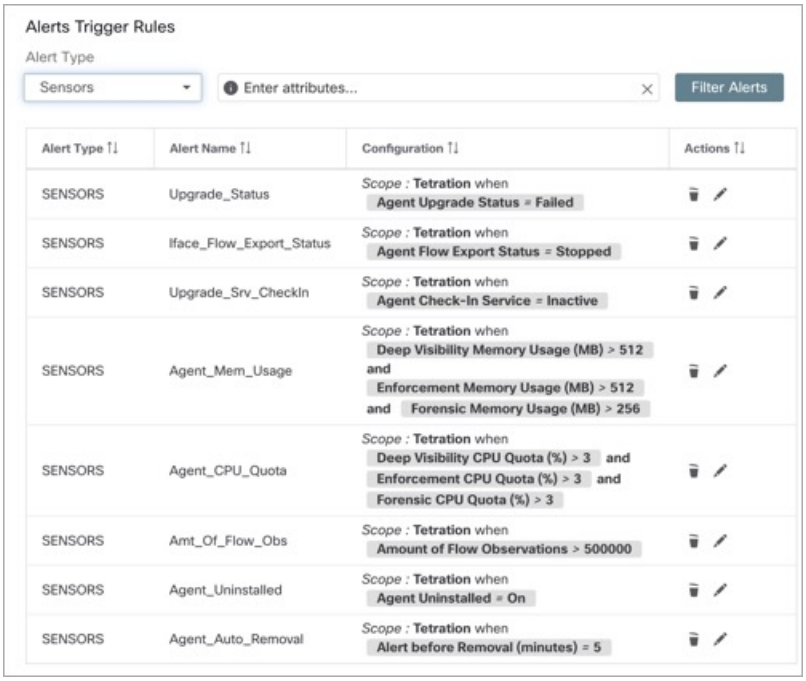
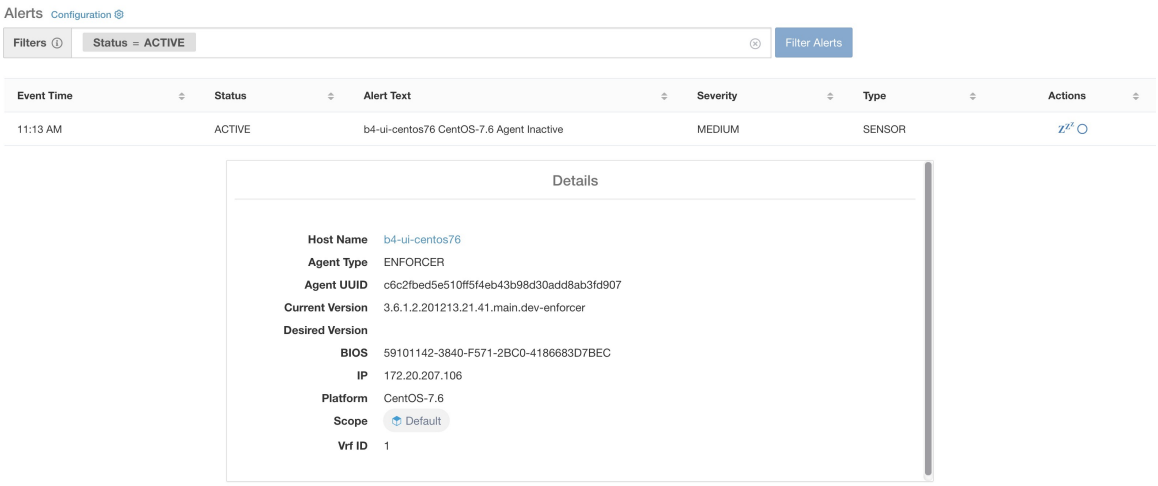


Figure 26: View Sensor Alerts



Sensor UI Alerts Details

Figure 27: Sensor Alerts



Sensor Alert Details

For the general structure of alerts and for information about fields, see Common Alert Structure. The *alert_details* field is structured and contains the following subfields for sensor alerts

Field	Alert Type	Format	Explanation
AgentType	*all*	string	ENFORCER or SENSOR depending on the installed type
HostName	*all*	string	Host name on which the agent is deployed
IP	*all*	string	IP address of the node/gateway
Bios	*all*	string	BIOS UUID of the node
Platform	*all*	string	Platform/OS information of the node
CurrentVersion	*all*	string	Software version of the agent on the node
DesiredVersion	*all*	string	Software version desired for the agent
LastConfigFetchAt	*all*	integer	Unix timestamp of when the agent last sent HTTPS request

Example of alert_details for a sensor alert


```
{
  "AgentType":"SENSOR",
  "Bios":"72EF1142-03A2-03BC-C2F8-F600567BA320",
  "CurrentVersion":"3.5.1.1.mrpm.build.win64-sensor",
  "DesiredVersion":"",
  "HostName":"win2k12-production-db",
  "IP":"172.26.231.193",
  "Platform":"MSServer2012R2Standard"
}
```


Frequently Asked Questions

This section lists some potential issues that you could possibly face during deployment and operating the software agents.

General

Log files: Log files get stored inside the <install-location>/logs or <install-location>/log folder. The log files get monitored and rotated through the Secure Workload services.

Agent deployment

Linux

Q: What do I do when the command

```
rpm -Uvh tet-sensor-1.101.2-1.el6-dev.x86_64.rpm
```

fails to install agents and displays the following error:

```
error: cannot create transaction lock on /var/lib/rpm/.rpm.lock (Permission denied).
```

A: If you do not have the right privileges to install the agents, either switch to root or use sudo to install the agents.

Q: What happens when you run “sudo rpm -Uvh tet-sensor-1.0.0-121.1b1bb546.el6-dev.x86_64.rpm” and encounter the following error:

```
Preparing...                ########################################### [100%]
which: no lsb_release in (/sbin:/bin:/usr/sbin:/usr/bin:/usr/X11R6/bin)
error: %pre(tet-sensor-site-1.0.0-121.1b1bb546.x86_64) scriptlet failed, exit status 1
error:   install: %pre scriptlet failed (2), skipping tet-sensor-site-1.0.0-121.1b1bb546
```

A: The system does not satisfy the requirements to install the agents. In this particular case, lsb_release tool is not installed.

For more information, see the Software Agents Deployment Label section and install the required dependencies.

Q: What happens whn you run “sudo rpm -Uvh tet-sensor-1.0.0-121.1b1bb546.el6-dev.x86_64.rpm” and encounter the following error:

```
Unsupported OS openSUSE project
error: %pre(tet-sensor-1.101.1-1.x86_64) scriptlet failed, exit status 1
error: tet-sensor-1.101.1-1.x86_64: install failed
warning: %post(tet-sensor-site-1.101.1-1.x86_64) scriptlet failed, exit status 1
```

A: Your OS is not supported to run software agents (in this particular case, “openSUSE project” is a non-supported platform).

For more information, see the Software Agents Deployment Label section.

Q: After I have installed all the dependencies and run installation with proper privileges with no errors. How do I know the agents installation was successful?

A: After you have installed the agents, to verify if the installation, run the following command:

```
$ ps -ef | grep -e csw-agent -e tet-
root     14158     1  0 Apr03 ?        00:00:00 csw-agent
root     14160 14158  0 Apr03 ?        00:00:00 csw-agent watch_files
root     14161 14158  0 Apr03 ?        00:00:03 csw-agent check_conf
root     14162 14158  0 Apr03 ?        00:01:03 tet-sensor -f conf/.sensor_config
root     14163 14158  0 Apr03 ?        00:02:38 tet-main --sensoridfile=./sensor_id
root     14164 14158  0 Apr03 ?        00:00:22 tet-enforcer --logtostderr
tet-sen+ 14173 14164  0 Apr03 ?        00:00:21 tet-enforcer --logtostderr
tet-sen+ 14192 14162  0 Apr03 ?        00:07:23 tet-sensor -f conf/.sensor_config
```

You must see three entries of *csw-agent* and at least two entries of *tet-sensor*. If the services are not running, ensure that the following directories are available, else the installation has failed.

- `/usr/local/tet` for most Linux distributions
- `/opt/cisco/tetration` for AIX, Ubuntu
- `/opt/cisco/secure-workload` for Solaris, Debian

Windows

Q: When I run the PowerShell agent installer script, I get one of the following errors:

1. The underlying connection was closed: An unexpected error occurred on a receive.
2. The client and server cannot communicate, because they do not possess a common algorithm

A: It is most likely because host and the server has mismatched SSL/TLS protocols configured. One can check the SSL/TLS version using the following command:

```
[Net.ServicePointManager]::SecurityProtocol
```

To set the SSL/TLS to be matching with server one can use the following command (note, this is not a permanent change, only temporary with the current PowerShell session):

```
[Net.ServicePointManager]::SecurityProtocol =
[System.Net.SecurityProtocolType]’Ssl3,Tls,Tls11,Tls12’
```

Q: When I run the MSI installer from the downloaded bundle, I get the following error:

```
This installation package could not be opened. Verify that the package exists and that you
 can access it, or contact the application vendor to verify that this is a valid Windows
Installer package.
```

A: Make sure *C:\Windows\Installer* path exists. If running the MSI installer from the command line, make sure to not include the relative path when pointing to the msi file. Example of correct syntax:

```
msiexec /i "TetrationAgentInstaller.msi" /l*v "msi_install.log" /norestart
```

Q: I have observed that Windows Sensor software fails to upgrade if underlying NIC is Nutanix VirtIO Network Driver.

A: There is an incompatibility issue between Npcap 0.9990 and Nutanix VirtIO Network Driver version earlier than 1.1.3 and Receive Segment Coalescing is enabled.

The resolution for this is to upgrade Nutanix VirtIO Network Driver to version 1.1.3 or later.

Q: I have installed windows sensor. The sensor doesnt seem to register and the sensor_id file contains the following: uuid-invalid-platform

A: You may not have system32 in PATH variable for Windows. Check if system32 is in PATH, if not run the following:

```
set PATH=%PATH%;C:\Windows\System32\
```

Q: I am not receiving the network flows from Kubernetes Pods on Windows Nodes.

A: To verify if the required sessions are running to capture the flows from Kubernetes pods on Windows nodes, perform the following:

1. Run **cmd.exe** with administrative privileges.
2. Run the following command: `logman query -ets`

   Ensure that the following sessions are running:

   - CSW_MonNet: Captures network flows
   - CSW_MonHCS: Monitors creation of pods
   - CSW_MonNat: Monitors NATed flows

Kubernetes

If the installer script fails during Kubernetes Daemonset Installation, there are a large number of possible reasons.

Q: Is the Docker Registry serving images reachable from nodes ?

A: Debug Direct or HTTPS Proxy issues with the cluster pulling images from Cisco Secure Workload cluster

Q: Is the container runtime complaining about SSL/TLS insecure errors ?

A: Verify that the Secure Workload HTTPS CA certificates are installed on all Kubernetes nodes in the appropriate location for the container runtime.

Q: Docker Registry authentication and authorization of image downloads failures ?

A: From each node, attempt to manually docker pull the images from the registry urls in the Daemonset spec using the Docker pull secrets from the secret created by the Helm Chart. If the manually image pull also fails, need to pull logs from the Secure Workload Cluster registryauth service to debug the issue further.

Q: Is the Kubernetes cluster hosted inside the Secure Workload appliance heathy ?

A: Check the service status page for the cluster to ensure all related services are healthy. Run the dstool snapshot from the explore page and retrieve the logs generated.

Q: Are the Docker Image Builder daemons running ?

A: Verify from the dstool logs that the build daemons are running.

Q: Are the jobs that build Docker images failing ?

A: Verify from the dstool logs that the images have not been built. Docker build pod logs can be used to debug errors during the buildkit builds. Enforcement Coordinator logs can also be used to debug the build failures further.

Q: Are the jobs creating Helm Charts failing ?

A: Verify from the dstool logs that the Helm Charts have not been built. Enforcement Coordinator logs will contain the output of the helm build jobs and can be used to debug the exact reason for the Helm Chart build job failures.

Q: Installation bash script was corrupt ?

A: Attempt to download the installation bash script again. The bash script contains binary data appended to it. If the bash script is edited in any way with a text editor or saved as a text file, special characters in the binary data may be mangled/modified by the text editor.

Q: Kubernetes cluster configuration – too many variants and flavors, we support classic K8s.

A: If the customer is running a variant of Kubernetes, there can be many failure modes at different stages of the deployment. Classify the failure stage - kubectl command run failure, helm command run failures, pod image download failures, pod privileged mode options rejected, pod image trust content signature failures, pod image security scan failures, pod binaries fail to run (architecture mismatch), pods run but the Secure Workload services fail to start, Secure Workload services start but have runtime errors due to unusual operating environment.

Q: Are the Kubernetes RBAC credentials failing ?

A: In order to run privileged daemonsets, we need admin privileges to the K8s cluster. Verify the the kubectl config file has its default context pointing towards the target cluster and admin-equivalent user for that cluster.

Q: Busybox image available or downloadable from all cluster nodes ?

A: Fix the connectivity issues and manually test that the busybox image can be downloaded. The exact version of busybox that is used in the pod spec must be available (pre-seeded) or downloadable on all cluster nodes.

Q: API Server and etcd errors or a general timeout during the install ?

A: Due to the instantiation of daemonset pods on all nodes in the Kubernetes cluster, the CPU/Disk/Network load on the cluster can spike suddenly. This is highly dependent on the customer specific installation details. Due to the overload, the installation process (images pulled on all nodes and written to disks) might take too long or overload the Kubernetes API server or the Secure Workload Docker Registry endpoint or, if configured, the proxy server temporarily. After a brief wait for image pulls on all nodes to complete and a reduction in CPU/Disk/Network load on the Kubernetes cluster nodes, retry the installation script again. API Server and etcd errors from the Kubernetes control plane indicate that the Kubernetes control plane nodes may be underprovisioned or affected by the sudden spike in activity.

Q: Secure Workload Agent experiencing runtime issues with its operations ?

A: Refer to the Linux Agent troubleshooting section if the pods are correctly deployed and the agent has started running but is experiencing runtime issues. The troubleshooting steps are the same once the Kubernetes deployment has successfully installed and started the pods.

Anomaly Types

These are the most common issues encountered on the workflow when using and managing Secure Workload Agents.

Agent Inactivity

Agent has stopped checking to the cluster services. This can happen due to several reasons:

- The host might have been down
- The network connectivity has been broken or blocked by firewall rules
- The agent service has been stopped

All platforms

- Verify the host is active and healthy
- Verify the agent service is up and running
- Verify the network connectivity to the cluster is working

Upgrade Failure

Agent upgrade has failed. This can be triggered by few cases such as:

- Not finding the package when the check in script attempts to download it - the upgrade package cannot be unpacked or the installer from the package cannot be verified.
- Installation process failing from an OS issue or dependency.

Windows

- Missing CA root certificate: Certificate Issues
- If agent was originally installed manually with a MSI install package, check if the Windows edition matches list of supported platforms in user guide: Check If Platform Is Currently Supported
- Check to make sure OS is configured correctly for Windows Installer operation: Windows Installer Issues
- Make sure there is enough free disk space on host

Linux

- If the host OS has been upgraded since the last agent installation, verify the current release matches list of supported platforms in user guide: Check If Platform Is Currently Supported
- Make sure there have been no changes to the required dependencies since the last installation. You can run the agent installer script with *–no-install* option to re-verify these dependencies.
- Make sure there is enough free disk space on host

AIX

- Make sure there have been no changes to the required dependencies since the last installation. You can run the agent installer script with *–no-install* option to re-verify these dependencies.
- Make sure there is enough free disk space on host

Convert Failed

The current agent type mismatches desired agent type and the convert attempt has timed out. This issue can be caused by a communication issue when an agent does check_in to download the package, or wss service failed to push convert_commnad to the agent.

All Platforms

- Verify the current release and agent type matches list of supported platforms in user guide: Check If Platform Is Currently Supported

Convert Capability

The ability to convert the agent from one type (such as deep visibility) to another type (such as enforcement) is not available by all agents. If an agent that is not capable to do the conversion is required to convert, the anomaly will be reported.

Policy Out of Sync

The current policy (NPC) version last reported by the agent does not match the current version generated on the cluster. This can be caused by a communications error between the agent and the cluster, the agent failing to enforce the policy with the local firewall, or the agent enforcement service not running.

Windows

- If enforcement mode is WAF, verify there are no GPOs present on the host that would prevent the Firewall from being enabled, adding rules (with Preserve Rules Off) or setting default actions: GPO Configurations
- Verify there is connectivity between the host and the cluster: SSL Troubleshooting
- Verify the generated rule count is less than **2000**
- Verify the WindowsAgentEngine service is running: *sc query windowsagentengine*
- Verify there are available system resources

Linux

- Verify iptables and ipset is present with the *iptables* and *ipset* command
- Verify there is connectivity between the host and the cluster: SSL Troubleshooting
- Verify the tet-enforcer process is running: *ps -ef | grep tet-enforcer*

AIX

- Verify ipfilter is installed and running with the *ipf -V* command

- Verify there is connectivity between the host and the cluster: SSL Troubleshooting
- Verify the tet-enforcer process is running: *ps -ef | grep tet-enforcer*

Flow Export: Pcap Open

If the Secure Workload Agent cannot open the pcap device to capture flows, you see errors in the Agent logs. A successfully opened Pcap device will report as follows:

Windows Log: *C:\Program Files\Cisco Tetration\Logs\TetSen.exe.log*

```
I0609 15:25:52.354 24248 Started capture thread for device <device_name>
I0609 15:25:52.354 71912 Opening device {<device_id>}
```

Linux Log: */usr/local/tet/logs/tet-sensor.log*

```
I0610 03:24:22.354 16614 Opening device <device_name>
[2020/06/10 03:24:23:3524] NOTICE: lws_client_connect_2: <device_id>: address 172.29.
↪136.139
```

Flow Export: HTTPS Connectivity

Connectivity between the agent and the cluster is externally blocked therefore preventing flows and other system information from being delivered. This is caused by one or more configuration issues with network firewalls, SSL decryption services, or third party security agents on the host.

- If there are known firewalls or SSL decryption security devices between the agent and the cluster, make sure that communications to all Secure Workload collector and VIPs IP addresses are being permitted. For on-prem clusters, the list of collectors will be listed under **Troubleshoot** > **Virtual Machines** in the navigation bar at the left side of the Secure Workload web interface. Look for collectorDatamover-*. For Secure Workload cloud, all the IP addresses that need to be permitted will be listed in your Portal.
- To help identify if there is SSL decryption, openssl s_client can be used to make a connection and display the returned certificate. Any additional certificate added to the chain will be rejected by the Agent's local CA. SSL Troubleshooting

Certificate Issues

Windows

Certificate Issues for MSI installer

MSI installer is signed using code signing certificate:

For MSI Installer, version 3.6.x onwards and 3.5.1.31 onwards

- Leaf Certificate: Cisco Systems, Inc
- Intermediate Certificate: DigiCert Trusted G4 Code Signing RSA4096 SHA384 2021 CA1
- Root Certificate: DigiCert Trusted Root G4

For MSI Installer, earlier versions

- Leaf Certificate: Cisco Systems, Inc

- Intermediate Certificate: Symantec Class 3 SHA256 Code Signing CA
- Root Certificate: VeriSign Class 3 Public Primary Certification Authority - G5

It uses timestamp certificate:

For MSI Installer, version 3.6.x onwards and 3.5.1.31 onwards

- Leaf Certificate: Symantec SHA256 TimeStamping Signer - G3
- Intermediate Certificate: Symantec SHA256 TimeStamping CA
- Root Certificate: VeriSign Universal Root Certification Authority

For MSI Installer, earlier versions

- Leaf Certificate: Symantec SHA256 Timestamping Signer - G2
- Intermediate Certificate: Symantec SHA256 Timestamping CA
- Root Certificate: VeriSign Universal Root Certification Authority

Windows Sensor Installation or upgrade will fail if digital signature of MSI installer is invalid. Digital signature is invalid if

- MSI Installer Signing Root Certificate or MSI Installer timestamp Root Certificate is not in a “Trusted Root Certification Authority” store
- MSI Installer Signing Root Certificate or MSI Installer timestamp Root Certificate is expired or revoked.

Issue 1

Installation of agent might fail with below error in the TetUpdate.exe.log “Msi signature is not trusted. 0x800b0109"

Resolution

- Run the command *certmgr* from command prompt
- Check if MSI Installer Signing Root Certificate or MSI Installer timestamp Root Certificate is in *Untrusted Certificates* store.
- Move it to *Trusted Root Certification Authority* store.

Issue 2

Windows Sensor upgrade fails with the following error in TetUpdate.exe.log “Msi signature is not trusted. 0x800B010C"

A certificate was explicitly revoked by its issuer.

Resolution

- Run the command *certmgr* from command prompt
- Check if MSI Installer Signing Root Certificate or MSI Installer timestamp Root Certificate is in *Untrusted Certificates* store.

- Copy it to *Trusted Root Certification Authority* store.

Issue 3

Windows Sensor upgrade fails with the following in TetUpdate.exe.log “Msi signature is not trusted. 0x80096005"

Resolution

- Run the command *certmgr* from command prompt
- Check if MSI Installer Signing Root Certificate and MSI Installer timestamp Root Certificate is in “Trusted Root Certification Authority” store

If it the certificate is missing, import it from other machine.

To import the certificate, follow below steps:

First export the certificate VeriSign Universal Root Certification Authority from one of Working server. Follow below steps:

- Run the command *certmgr* from command prompt
- Right click on the certificate “VeriSign Universal Root Certification Authority” under “Trusted Root Certification Authorities” and go to All tasksExport.
- Copy the exported certificate to the Non-working server and then import the certificate.

To import the certificate, follow below steps:

First export the certificate VeriSign Universal Root Certification Authority from one of Working server. Follow below steps:

- Run the command *certmgr* from command prompt
- Right click on the certificates tab under Trusted Root Certification Authorities and go to All tasksImport.
- Select the Root certificate that you copied and add it in the store.

Certificate Issues for NPCAP installer

Applicable to Windows 2012 , Windows 2012 R2, Windows 8, Windows 8.1

NPCAP version: 1.55

NPCAP Signing Certificate:

- Leaf Certificate: Insecure.Com LLC
- Intermediate Certificate: DigiCert EV Code Signing CA (SHA2)
- Root Certificate: DigiCert High Assurance EV Root CA

NPCAP Timestamp certificate:

- Leaf Certificate: DigiCert Timestamp 2021

- Intermediate Certificate: DigiCert SHA2 Assured ID Timestamping CA
- Root Certificate: DigiCert Assured ID Root CA

Issue 1

Windows Agent Installation might fail with below error in msi_installer.log

```
CheckServiceStatus : Exception System.InvalidOperationException: Service npcap was not found
 on computer
'.'. —> System.ComponentModel.Win32Exception: The specified service does not exist as an
installed service
```

Resolution

- Run the command *certmgr* from command prompt
- Check “DigiCert High Assurance EV Root CA” in “Trusted Root Certification Authority” store.
- If it the certificate is missing, import it from other machine.

To import the certificate, follow below steps:

First export the certificate “DigiCert High Assurance EV Root CA” from one of Working server. Follow below steps:

- Run the command *certmgr* from command prompt
- Right click on the certificate “DigiCert High Assurance EV Root CA” under “Trusted Root Certification Authorities” and go to All tasksExport.
- Copy the exported certificate to the Non-working server and then import the certificate.

To import the certificate, follow below steps:

- Run the command *certmgr* from command prompt
- Right click on the certificates tab under Trusted Root Certification Authorities and go to All tasksImport.
- Select the Root certificate that you copied and add it in the store.

Applicable to Windows 2008 R2

NPCAP version: 0.991

NPCAP Signing Certificate:

- Leaf Certificate: Insecure.Com LLC
- Intermediate Certificate: DigiCert EV Code Signing CA
- Root Certificate: DigiCert High Assurance EV Root CA

NPCAP Timestamp certificate:

- Leaf Certificate: DigiCert Timestamp Responder
- Intermediate Certificate: DigiCert Assured ID CA-1

- Root Certificate: VeriSign DigiCert Assured ID Root CA

Issue 1

Windows Agent Installation might fail with below error in msi_installer.log

```
CheckServiceStatus : Exception System.InvalidOperationException: Service npcap was not found
 on
computer '.'. —> System.ComponentModel.Win32Exception: The specified service does not exist
 as an
installed service
```

Resolution

- Run the command *certmgr* from command prompt
- Check *DigiCert High Assurance EV Root CA* in *Trusted Root Certification Authority* store.
- If it the certificate is missing, import it from other machine.

To import the certificate, follow below steps:

First export the certificate “DigiCert High Assurance EV Root CA” from one of Working server. Follow below steps:

- Run the command *certmgr* from command prompt
- Right click on the certificate “DigiCert High Assurance EV Root CA” under “Trusted Root Certification Authorities” and go to All tasksExport.
- Copy the exported certificate to the Non-working server and then import the certificate.

To import the certificate, follow below steps:

- Run the command *certmgr* from command prompt
- Right click on the certificates tab under Trusted Root Certification Authorities and go to All tasksImport.
- Select the Root certificate that you copied and add it in the store.

Windows Host Rename

Scenario 1: Not able to see IP Addresses and VRF info after renaming the Windows Host Steps to fix the issue:

- Remove the entry(with new Hostname that is missing IP Addresses and VRF info) from the TaaS UI.
- Uninstall ‘Cisco Secure Workload Agent’ from the Windows Host and delete the ‘Cisco Tetration’ directory (typically the path for this will be : ‘C:Program FilesCisco Tetration’).
- Install ‘Cisco Secure Workload Agent’ on the Windows Host.

Following the above steps should register the Agent on the TaaS UI successfully with the IP Addresses and VRF info.

Scenario 2: Planned Windows Host rename (in advance) Steps to follow:

- Uninstall ‘Cisco Secure Workload Agent’ from the Windows Host and delete the ‘Cisco Tetration’ directory (typically the path for this will be : ‘C:Program FilesCisco Tetration’).
- Rename the Windows Host and Reboot.
- Install ‘Cisco Secure Workload Agent’ on the Windows Host(with new Hostname).

Following the above steps for planned Host rename should register the Agent on the TaaS UI with new Hostname.

Check If Platform Is Currently Supported

Windows

- Run the command *winver.exe*
- Compare this release to what is listed here: Supported Platforms and Requirements

Linux

- Run *cat /etc/os-release*
- Compare this release to what is listed here: Supported Platforms and Requirements

AIX

- Run the command *uname -a*
- Note: The major and minor versions are reversed

```
p7-ops2> # uname -a
AIX p7-ops2 1 7 00F8AF944C00
```

- In this example, the first number after the host name is the minor and the second number is the major version, so AIX version 7.1. Compare this release to what is listed here: Supported Platforms and Requirements

Windows Installer Issues

- Make sure there is a *C:\Windows\Installer* directory. This is not visible in File Explorer, easiest way to verify is in a CMD session and running: *dir C:\Windows\Installer*
- Check if the *Windows Installer* service is not disabled. It must be set to *Manual*
- Check to see if there are no other errors being reported by Windows Installer. Check Windows System Event logs under **Windows Logs** > **Application** > **Source** > **MsiInstaller**

Required Windows Services

Below is a list of services, that when disabled, have been linked to installation issues of the agent. It is recommended these services are running during the initial installation and any upgrade of the Deep Visibility and Enforcement agents.

Table 8: Required Windows Services

Service	Purpose for installation
Device Setup Manager	Device driver management for the installation of the Npcap filter driver.
Device Install Service	Also used for the installation of the Npcap filter driver.
Windows Installer	Required for the installation of agent MSI package.
Windows Firewall	Required for WAF enforcement mode.
Application Experience	Used to determine capatibility executables on the system.

Note Application Experience service only applies to Windows Server 2008, 2008R2, 2012, 2012R2 and Win- dows 7. If disabled, a file lock may occur during Npcap installation causing it to fail.

Npcap Issues

Npcap is a pcap tool used for Windows Agent only. Ten seconds after the agent service starts, it will attempt to install or upgrade Npcap to the supported version. If Npcap service fails to install or upgrade, the agent will retry the installation within the next 30 minutes. After 3 failed attempts, the agent will attempt to rollback Npcap to a previous supported version if available. After, the agent will no longer try to install Npcap. You can check *C:\Program Files\Cisco Tetration\Logs\TetUpdate.exe.log* and *C:\Program Files\Cisco Tetration\Logs\npcap_install.log* to identify the error.

Npcap will not upgrade (manually or via agent)

- Npcap will sometimes not uninstall correctly if a process is currently using the Npcap libraries. To check for this run the following command:

```
PS C:\Program Files\Npcap> .\NPFInstall.exe -check_dll
WindowsSensor.exe, Wireshark.exe, dumpcap.exe
```

If you see processes listed, they must be stopped before the Npcap upgrade can continue. If no processes are using Npcap the above command will simply show *<NULL>*

Npcap will not install

- Check CA certificates installed on the system: Certificate Issues for NPCAP installer
- Check Windows Installer issues: Windows Installer Issues

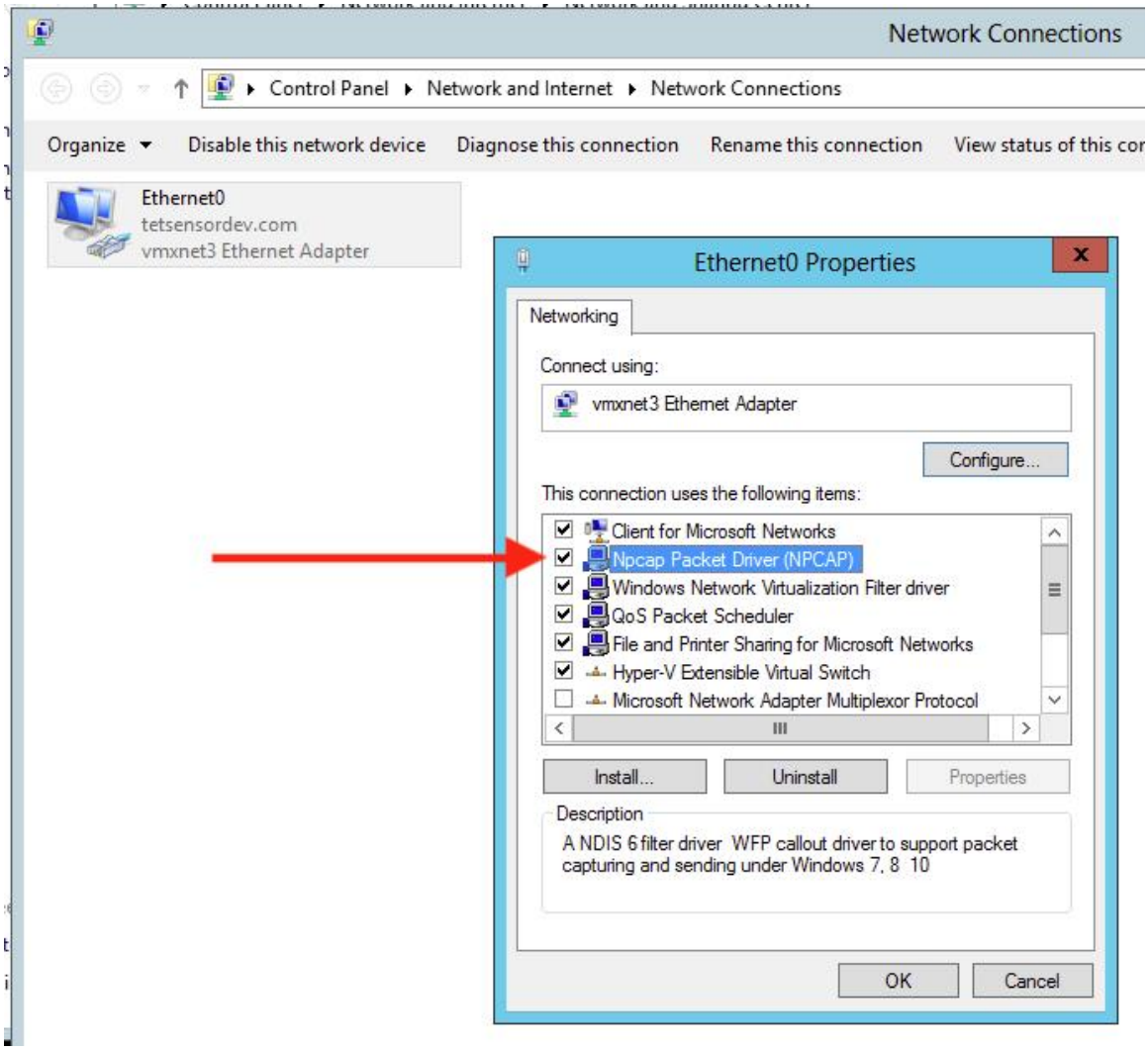
- Verify no other user on the system is making changes to the network interfaces. This can cause a COM lock preventing NDIS driver binding.

Verify if Npcap is fully installed

Procedure

Step 1 Check **Control Panel** > **Programs** and **Features** to see if Npcap is listed as an installed application

Step 2 Make sure the Npcap Packet Driver has a binding to the NIC in question (checkmark is present)



Step 3 Check if the network driver is installed

```
C:\Windows\system32>pnputil -e | findstr Nmap
Driver package provider : Nmap Project
```

Step 4 Check if the driver service is installed and RUNNING

```
C:\Windows\system32>sc query npcap
SERVICE_NAME: npcap
         TYPE : 1 KERNEL_DRIVER
         STATE : 4 RUNNING
```

Step 5 Check if the registry entry is there (used by the agent to verify Npcap exists already)

```
C:\Windows\system32>reg query HKLM\software\wow6432node\npcap
HKEY_LOCAL_MACHINE\software\wow6432node\npcap
          AdminOnly REG_DWORD 0x1
          WinPcapCompatible REG_DWORD 0x0
         (Default) REG_SZ C:\Program Files\Npcap
```

Step 6 Check if the installed Npcap program files are all there

```
C:\Windows\system32>dir "c:\program files\npcap"
 Directory of c:\program files\npcap
04/29/2020 02:42 PM <DIR> .
04/29/2020 02:42 PM <DIR> ..
01/22/2019 08:16 AM 868 CheckStatus.bat
11/29/2016 03:43 PM 1,034 DiagReport.bat
12/04/2018 11:12 PM 8,908 DiagReport.ps1
01/09/2019 09:22 PM 2,959 FixInstall.bat
04/29/2020 02:42 PM 134,240 install.log
01/11/2019 08:52 AM 9,920 LICENSE
03/14/2019 08:59 PM 10,434 npcap.cat
03/14/2019 08:57 PM 8,657 npcap.inf
03/14/2019 09:00 PM 74,040 npcap.sys
03/14/2019 08:57 PM 2,404 npcap_wfp.inf
03/14/2019 09:00 PM 270,648 NPFInstall.exe
04/29/2020 02:42 PM 107,783 NPFInstall.log
03/14/2019 09:01 PM 175,024 Uninstall.exe
         13 File(s) 806,919 bytes
          2 Dir(s) 264,417,628,160 bytes free
```

Step 7 Check to see if the .sys driver file is in the Windows driver folder

```
C:\Windows\system32>dir "C:\Windows\System32\Drivers\npcap.sys"
Directory of C:\Windows\System32\Drivers
03/14/2019 09:00 PM 74,040 npcap.sys
                  1 File(s) 74,040 bytes
```

Network Connectivity issues during NPCAP installation or upgrade

Applicable to Windows 2016 Only

If you have a 3rd party LWF (Light Weight Filter) driver (e.g. netmon) or a teaming adapter is configured in your setup, and NPCAP is installed during agent deployment, you might experience

RDP is reconnected

NetBios service is restarted

Similar network connectivity issues

This is due to a BUG in Windows 2016 OS

NIC teaming compatibility issues with NPCAP

Teaming NIC functionality is based on underneath Physical NICs (Intel, Broadcom, Realtek, MS virtual adapter etc) and Teaming driver configuration (switch based, loadbalancing or failover, algorithm to distribute the packets across multiple NICs).

Some NPCAP versions have compatibility issues with Teaming NICs, especially during binding to the underneath Teaming NICs.

The current Secure Workload Sensor software is tested using Microsoft supported NIC teaming.

```
NIC type : Intel(R) 82574L Gigabit Network Connection
Teaming Mode : Switch Independent
Load Balancing Mode: Address Hash
OS : Windows 2012 , Windows 2012 R2, Windows 2016, Windows 2019
NPCAP version: 1.55
```

.

Note Windows 2008R2 does not support Microsoft supported NIC teaming.

VDI instance VM does not report network flows

The TetSensor service occasionally does not capture the network flows on cloned VMs when NPCAP service is running. This can happen when the agent is installed without the **nostart** flag using MSI installer or without **goldenImage** flag using PowerShell Installer on a VM template or golden image.

In this case, Secure Workload agent services start running on the VM template. NPCAP is installed and bound to the Network stack on the VM template. When a new VM is cloned from the VM template, NPCAP is not properly bound to the Network stack on the new cloned VM. As a result, NPCAP fails to capture the network flows.

Network Performance with NPCAP

It is observed that Network performance will be affected when Windows TetSensor service is running. Windows Tet- Sensor service (tetsen.exe) captures the network flows using NPCAP. NPCAP implementation to capture the network flows and the network flows to the tetsen.exe affects the network performance.

Compare the Network Performance after installing tetsensor, Client : Windows 2016

NPCAP 1.55

TetSensor Config : Conversation Mode with Enforcement mode WFP

Server : Windows 2016

NPCAP 1.55

TetSensor Config : Conversation Mode with Enforcement mode WFP

Run cmd : iperf3.exe -c <server_ip> -t 40

Table 9: 121071: Network Performance with NPCAP 155

Setup	Network Performance
No TetSensor Installed NO NPCAP	[ID] Interval Transfer Bandwidth [4] 0.00-40.00 sec 18.2 GBytes 3.90 Gbits/sec sender [4] 0.00-40.00 sec 18.2 GBytes 3.90 Gbits/sec receiver
TetSensor Installed NPCAP Installed	[ID] Interval Transfer Bandwidth [4] 0.00-40.00 sec 17.3 GBytes 3.72 Gbits/sec sender [4] 0.00-40.00 sec 17.3 GBytes 3.72 Gbits/sec receiver

Network Performance with NPCAP 0.9990

Compare the Network Performance after installing tetsensor, Client : Windows 2016

NPCAP 0.9990

TetSensor Config : Conversation Mode with Enforcement mode WFP

Server : Windows 2016

NPCAP 0.9990

TetSensor Config : Conversation Mode with Enforcement mode WFP

Run cmd : iperf3.exe -c <server_ip> -t 40 .. table:: Network Performance with NPCAP 0.9990

class longtable

Setup	Network Performance
TetSensor Installed NPCAP Installed	[ID] Interval Transfer Bandwidth [4] 0.00-40.00 sec 16.3 GBytes 3.50 Gbits/sec sender [4] 0.00-40.00 sec 16.3 GBytes 3.50 Gbits/sec receiver

Note Performance may vary based on Windows NPCAP version installed, Windows OS, and network Configuration.

OS Performance and/or stability Issues

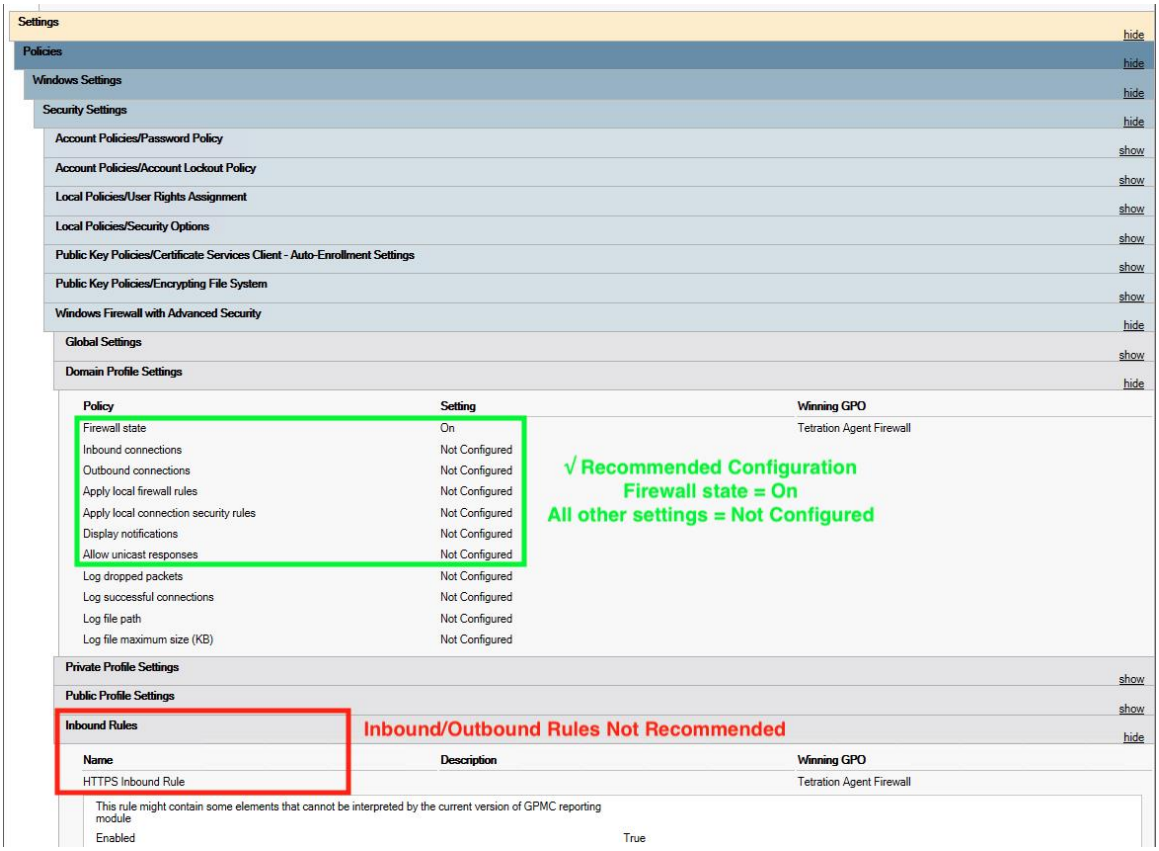
OS may experience unknown performance or stability issues if the installed NPCAP version or NPCAP configuration is not supported by the Secure Workload Software.

Supported NPCAP Version: : 0.991 and 1.55

GPO Configurations

Agents that enforce policy require only the Firewall to be enabled with either a local setting or GPO. All other GPO settings should not be set and left as "Not Configured."

- To check if a GPO setting is blocking enforcement you can check the *C:\Program Files\Cisco Tetration\Logs\TetEnf.exe.log* log and search for the following error examples:
- **Rules conflicting with "Preserve Rules=No" setting:** "There are firewall rules set in the Group Policy. Secure Workload agent does not have permission to remove these"
- **Firewall set to off:** "GPO has disabled firewall for DomainProfile"
- **Default Action is set:** "Group Policy has conflicting default inbound action for DomainProfile"
- To check what GPO policies are being applied to the host, run *gpresult.exe /H gpreport.html* and open the generated HTML report. In the example below *Secure Workload Agent Firewall* is applying a Inbound rule which will conflict with Enforcement if "Preserve Rules" is set to "No."



Agent To Cluster Communications

The Secure Workload Agent maintains connections to the cluster over multiple channels. Depending on the type of Agent, the number of connections varies.

Types of connections

- **WSS:** Persistent socket connection over port 443 to the cluster
- **Check in:** A HTTPS call to the cluster every 15-20 minutes to check for current configurations, check for updates and to update the active state of the agent to the cluster. This also reports upgrade failures.
- **Flow export:** Persistent SSL connection over port 443 (TaaS) or 5640 (On-premise) to send flow metadata to the cluster
- **Enforcement:** Persistent SSL connection over port 443 (Taas) or 5660 (On-premise) to pull in enforcement policies and report enforcement state

Checking the connection state

The Teration UI will report either an inactive agent (no longer checking-in), no exported flows (on Agent Workload Profile page under Stats), or failed enforcement. Depending on the error, you can check different logs on the workload to help determine the source of the issue.

Inactive Agent

Windows Log: *C:\Program Files\Cisco Tetration\Logs\TetUpdate.exe.log*

Linux Log: */usr/local/tet/logs/check_conf_update.log*

An HTTP response code of 304 is expected and means there is no configuration change. Error code = 2 is expected as well. Any other HTTP response code will indicate a issue talking to the WSS service on the Secure Workload cluster.

```
Tue 06/09/2020 17:25:25.08 check_conf_update: "curl did not return 200 code, it's 304,
‹→ exiting"
Tue 06/09/2020 17:25:25.08 check_conf_update: "error code after running check_conf_
‹→update = 2"
```

- **304** Expected, no config change. Successful check-in
- **401** Registration is not successful, missing Activation Key (TaaS)
- **403** Agent already registered to the cluster with same UUID
- **000** Indicates connection issue with SSL. Either curl could not reach the WSS server or there is a issue with the certificate. See SSL troubleshooting: SSL Troubleshooting

No exported flows

Windows Log: *C:\Program Files\Cisco Tetration\Logs\TetSen.exe.log*

Linux Log: */usr/local/tet/logs/tet-sensor.log*

The following indicates a successful connection to WSS

```
cfgserver.go:261] config server: StateConnected, wss://<config_server_ip>:443/wss/
‹→<sensor_id>/forensic, proxy:
```

The following indicates a successful connection to the Collectors

```
collector.go:258] next collector: StateConnected, ssl://<collector_ip>>:5640
```

If there are errors connecting to either WSS or the Collectors, check your firewall configuration or verify if any SSL decryption is occurring between the agent and Secure Workload. See: SSL Troubleshooting

Failed to enforce policy

Windows Log: *C:\Program Files\Cisco Tetration\Logs\TetEnf.exe.log*

Linux Log: */usr/local/tet/logs/tet-enforcer.log*

```
ssl_client.cpp:341] Successfully connected to EFE server
```

If there are errors connecting to the EFE server, check your firewall configuration or verify if any SSL decryption is occurring between the agent and Secure Workload. See: SSL Troubleshooting

SSL Troubleshooting

Agent Communications Overview

Secure Workload agents use TLS to secure the TCP connections to the Secure Workload Cloud SaaS servers. These connections are broken down into three distinctive channels.

- Agent -> Cisco Secure Workload SaaS control channel over port TCP/443 (TLS) (sensorVIP)

  This is a low volume control channel that allows the agent to register with Secure Workload and also handles configuration pushes and software upgrade notifications.

- Agent -> Cisco Secure Workload SaaS flow data over TCP/443 (TLS) (collector)

  Flow data is the extracted flow metadata information; the data will be sent to 1 set of 16 IP addresses at a time. The second set of IP addresses is for standby. This is around 1 – 5% of actual server traffic.

- Agent ->Cisco Secure Workload SaaS enforcement data over TCP/443 (TLS) (efe)

  The enforcement data channel is a low volume control channel that is used to push the policies to the sensors and also gather enforcement statistics.

The sensor validates the TLS certificate from the Secure Workload Cloud control, data and enforcement servers against a local CA that is installed with the agent. No other CAs are used, so any other certificate sent to the agent will result in a verification failure and the agent will not connect. This will result in the agent not registering, checking-in, sending flows or receiving enforcement policies.

Configuring IP traffic for Agent Communications

A typical configuration for most will be to have a perimeter firewall and possibly a proxy between the agents (workflows) and Secure Workload TaaS.

Note Secure Workload gathers your gateway/NAT IP information during the on-boarding and automatically adds the information at the time of tenant creation. If you add new IP addresses or change IP addresses in the portal, the changes require review and approval by Secure Workload staff.

In addition to adding your gateway/NAT IP addresses in the TaaS portal, there might be more changes required to your network to allow the traffic outbound and unmodified:

Allow outbound port 443 over TLS/HTTPS on the perimeter firewall

Configure proxy bypass and SSL/TLS bypass on the web proxy, if a decrypting web proxy is being used.

Note If you are using a transparent web proxy at the data center, you must route the specific SaaS IP address and configure the bypass rules. Sensors are connections that cannot do automatic HTTPS redirection.

The list of IPs the agents communicates with is available on the TaaS portal. The IPs to add to your firewall outbound configuration and proxy bypass are labeled collector-n, efe-n (only if enforcement is being deployed), and sensorVIP. There are typically 17 to 33 IPs to add for agent communication, but there could more or less depending on your TaaS configuration.

Troubleshooting SSL/TLS Connections

As discussed in the previous section, it is important to configure your explicit or transparent web proxy to bypass SSL/TLS decryption for agent communications. If the bypass is not configured, these proxies might attempt to decrypt

SSL/TLS traffic by sending its own certificate to the agent. Because the agent only uses its local CA to validate the certificate, these proxy certificates will cause connection failures.

Symptoms include agent failing to register to the cluster, agent not checking-in, agent not sending flows, and/or agent not receiving enforcement configuration (if enforcement is enabled).

Note Troubleshooting steps below are assuming default installation paths were used. Windows: C:Program FilesCisco Tetration Linux: /usr/local/tet. If you installed your agents in a different location, substitute that location in the instructions.

SSL/TLS Connection issues are reported in the agent logs. To verify if there are SSL errors in the logs, run the following commands for the associated issue being observed.

Registration, check-in

Linux

```
grep "NSS error" /usr/local/tet/log/check_conf_update.log
```

Windows (PowerShell)

```
get-content "C:\Program Files\Cisco Tetration\logs\TetUpdate.exe.log" | select-
‹→string -pattern "curl failed SSL peer certificate"
```

Flows

Most of the SSL/TLS connection issues seen are during the initial connection and registration of the agent. Sending flows relies on the registration to be complete before attempting to connect. SSL/TLS errors seen here would be the result of the sensorVIP IPs being allowed but not the collector IPs.

Linux

```
grep "SSL connect error" /usr/local/tet/log/tet-sensor.log
```

Windows (PowerShell)

```
get-content "C:\Program Files\Cisco Tetration\logs\WindowsSensor*.log" | select-
‹→string -pattern "Certificate verification error"
```

Enforcement

Linux

```
grep "Unable to validate the signing cert" /usr/local/tet/log/tet-enforcer.log
```

Windows (PowerShell)

```
get-content "C:\Program Files\Cisco Tetration\logs\WindowsSensor*.log" | select-
‹→string -pattern "Handshake failed"
```

If an SSL error is seen in the log checks above you can verify what certificate is being sent to the Agents with the following commands.

Explicit Proxy - where a proxy is configured in user.cfg

Linux

```
curl -v -x http://<proxy_address>:<port> https://<sensorVIP>:443
```

Windows (PowerShell)

```
cd "C:\Program Files\Cisco Tetration"
.\curl.exe -kv -x http://<proxy_address>:<port> https://<sensorVIP>:443
```

Transparent Proxy - No user.cfg proxy configuration required. It's a proxy configured between all HTTP(S) traffic from agent to the internet.

Linux

```
openssl s_client -connect <sensorVIP from TaaS Portal>:443 -CAfile /usr/local/tet/
‹→cert/ca.cert
```

Windows (PowerShell)

```
cd C:\Program Files\Cisco Tetration
.\openssl.exe s_client -connect <sensorVIP from TaaS Portal>:443 -CAfile cert\ca.cert
```

You are looking for the following in the openssl s_client respose

```
Verify return code: 0 (ok)
```

If you see an error, examine the certificate. An example certificate (chain) should include only the following cert (CN IP is an example):

Certificate chain

```
0 s:/C=US/ST=CA/L=San Jose/O=Cisco Systems, Inc./OU=Tetration, Insieme BU/CN=129.146.
‹→155.109
i:/C=US/ST=CA/L=San Jose/O=Cisco Systems, Inc./OU=Tetration Analytics/CN=Customer CA
```

If you see additional certificates, then there is possibly a Web decrypting proxy between the agent and Secure Workload. Contact your security or network group and verify if the proxy bypass is configured using the listed IPs from the Configuring IP traffic for Agent Communications section.

Windows sensor installation script fails on Windows 2016 servers: Error message that might appear "The underlying connection was closed: An unexpected error occurred on a receive." Possible reason might be the SSL/TLS versions set in PowerShell.

To check the SSL/TLS versions running, run the following command:

```
[Net.ServicePointManager]::SecurityProtocol
```

If the output from the above command is:

```
Ssl3, Tls
```

Then use the below command to change the allowed protocols and retry the installation:

```
[Net.ServicePointManager]::SecurityProtocol = [System.Net.SecurityProtocolType]'Ssl3,
‹→Tls,Tls11,Tls12'
```

Agent operations

Q: I have installed the agents successfully, but I didn't see it on UI Sensor Monitoring page.

A: An agent is required to register with backend server running within cluster before it could start operating. When an agent is not shown on UI page, most likely it's because the registration has failed. There are a few things we could check to see why a registration failed:

- Check if the connection between the agent and the backend server is working properly
- Check if the curl request could be sent to backend server properly
- Check HAProxy access and backend server logs to see if the registration request made it to the server
- Check the error return from curl request in the log file

Q: The agent is installed and I could find in on UI page. However, the "SW Ver" column shows "initializing" instead of a version string.

A: After the initial agent is installed and registered with the backend server, it would take another 30 minutes for the agent to report its version.

Q: The agent is upgraded properly, but the "SW Ver" fields still show the old version after a long time (like several hours).

A: After the agent is upgraded successfully, it will try to send a curl request to report its current running version and check for new version in the same request. It is possible that the request couldn't make it to the backend, due to several reason:

- The request is timed out, couldn't get the response in time
- The network is facing problem, agent couldn't connect to backend servers

Q: I have an agent running on RHEL/CentOS-6.x and it is working properly. I am planning to upgrade the OS to RHEL/CentOS-7.x. Would the agent still work after the upgrade?

A: currently we do not support the scenario in which the OS has been upgraded, especially upgrading the major releases. In order to have the agent work after OS upgrade, do the following steps:

- Uninstall the existing agent software
- Clean up all files, including certs
- Go to UI, delete the agent entry
- Upgrade the OS to the desired version
- Install the agent software on the new OS

Q: I have an agent running on RHEL/CentOS-6.x and it is working properly. I am planning to rename the host. Would the agent still work after rename/reboot?

A: An agent identity is calculated based on the host's uniqueness, including hostname and bios-uuid. Changing hostname changes the host's indentify. It is recommended to do the following:

- Uninstall the existing agent software

- Clean up all files, including certs
- Go to UI, delete the old agent entry
- Rename the host and reboot
- Install the agent software again

Q: On Windows host, firewall deviation was caused by adding/deleting/modifying a rule. How do I find the rule?

A: On deviation detection, agent logs the last 15 seconds of firewall events to "C:\Windows\System32\config\systemprofile\AppData\Roaming\tet\firewall_events". Rule that caused deviation will be found in the latest file created as policy_dev_<policy id>_<timestamp>.txt

Q: I have installed the agent on a Windows host successfully. Why do I not see any reported flows from the sensor?

A: Npcap is required to collect flows on a Windows host. Ten seconds after the agent is installed successfully, it will install Npcap. If the sensor does not report flows after several minutes, check if the agent and the backend server is connected and if Npcap is installed properly on the Npcap Issues.

Q: I have installed the agent on Windows host, 2008 R2, successfully. Why does the system clock drift when tetsensor service is running?

A: This is a known problem with Go and Windows 2008 R2. For more information, see Golang and Win2008 R2.

The process, tet-main.exe, running as a part of tetsensor service, is built using Go Version 1.15. That is why the system clock drifts when the tetsensor service is running.

This issue occurs when Windows 2008 R2 workload is configured to use the external NTP server or Domain Controller as NTP server.

The possible work around :

1. Periodially force NTP to sync the clock: w32tm /resync /force
2. Disable tet-main.exe manually.
   - Run cmd.exe with "administrator" privilege.
   - Run regedit.exe
   - Go to "HKEY_LOCAL_MACHINE\SYSTEM\ControlSet001\Services\TetSensor"
   - Double click on "ImagePath"
   - Edit value, remove tet-main.exe

     before "C:\Program Files\Cisco Tetration\TetSenEngine.exe" TetSensor TetSen.exe "-f sensor_config" tet-main.exe " " TetUpdate.exe

     after "C:\Program Files\Cisco Tetration\TetSenEngine.exe" TetSensor TetSen.exe "-f sensor_config" TetUpdate.exe
   - Restart tetsensor service

Note Disable tet-main.exe after every time agent is upgraded.

3. Remove external NTP server configuration:
   - Run command : w32tm /config /update /manualpeerlist: /syncfromflags:manual /reliable:yes
   - Restart Windows Time Service, W32Time