

Network Visibility Module

- About Network Visibility Module, on page 1
- Provide Full Disk Access Manually for macOS, on page 4
- How to Use Network Visibility Module, on page 6
- Collection Parameters for Network Visibility Module, on page 7
- Transport Securely to Connector, on page 13
- Network Visibility Module Profile Editor, on page 14
- Extend Linux Kernel Capabilities, on page 19
- About Flow Filters, on page 20
- Osquery Data Configuration, on page 21
- Customer Feedback Module Gives NVM Status, on page 21

About Network Visibility Module

Because users are increasingly operating on unmanaged devices, enterprise administrators have less visibility into what is going on inside and outside of the network. The Network Visibility Module (NVM) collects rich flow context from an endpoint on or off premise and provides visibility into network connected devices and user behaviors when coupled with a Cisco solution such as Cisco Secure Cloud Analytics, Splunk, or a third-party solution. The enterprise administrator can then do capacity and service planning, auditing, compliance, and security analytics. Network Visibility Module provides the following services:

- Monitors application use to enable better informed improvements (expanded IPFIX collector elements in nvzFlow protocol specification: https://developer.cisco.com/site/network-visibility-module/) in network design.
- Classifies logical groups of applications, users, or endpoints.
- Finds potential anomalies to help track enterprise assets and plan migration activities.

This feature allows you to choose whether you want the telemetry targeted as opposed to whole infrastructure deployment. The Network Visibility Module collects the endpoint telemetry for better visibility into the following:

- The device—the endpoint, irrespective of its location
- The user—the one logged into the endpoint
- The application—what generates the traffic

- The location—the network location the traffic was generated on
- The destination—the actual FQDN to which this traffic was intended

When on a trusted network, Cisco Secure Client Network Visibility Module exports the flow records to a collector such as Cisco Secure Cloud Analytics, Splunk, or a third-party vendor, which performs the file analysis and provides a UI interface and reports. The flow records provide information about the capabilities of the user, and the values are exported with ids (such as LoggedInUserAccountType as 12361, ProcessUserAccountType as 12362, and ParentProcessUserAccountType as 12363). For more information about Cisco Endpoint Security Analytics (CESA) built on Splunk, refer to http://www.cisco.com/go/cesa. Since most enterprise IT administrators want to build their own visualization templates with the data, we provide some sample base templates through a Splunk app plugin.

NVM on Desktop Cisco Secure Client

Historically, a flow collector provided the ability to collect IP network traffic as it enters or exits an interface of a switch or a router. It could determine the source of congestion in the network, the path of flow, but not much else. With Network Visibility Module on the endpoint, the flow is augmented by rich endpoint context such as type of device, the user, the application, and so on. This makes the flow records more actionable depending on the capabilities of the collection platform. The exported data provided with Network Visibility Module which is sent via IPFIX is compatible with Cisco NetFlow collectors and Splunk, as well as other 3rd party flow collection platforms. See platform-specific integration documentation for additional information, For example, Splunk integration is available via https://www.cisco.com/c/en/us/support/docs/security/anyconnect-secure-mobility-client/200600-Install-and-Configure-Cisco-Network-Visi.html.

When using Network Visibility Module Collector in releases 4.9 or later, you must use Splunk app 3.x to view the additional parameters.

The Cisco Secure Client profile for Network Visibility Module gets pushed from the ISE or Secure Firewall ASA headend if this feature is enabled. On the ISE headend, you can use the standalone profile editor, generate the Network Visibility Module service profile XML, upload it to ISE, and map it against the new Network Visibility Module module, just as you do with Network Access Manager. On the Secure Firewall ASA headend, you can use either the standalone or ASDM profile editor.

Network Visibility Module gets notified when the VPN state changes to connected and when the endpoint is in a trusted network.

Note If you are using Network Visibility Module with Linux, make sure that you have completed the preliminary steps in Using Network Visibility Module on Linux.

Standalone NVM

For those who do not have Cisco Secure Client deployment or are using another VPN solution, you can install the Network Visibility Module standalone package for your Network Visibility Module needs. This package works independently but provides the same level of flow collection from an an endpoint as the existing Cisco Secure Client Network Visibility Module solution. If you install the standalone Network Visibility Module, the active processes (such as the Activity Monitor on macOS) indicate the use.

Standalone Network Visibility Module is configured with the Network Visibility Module Profile Editor, and Trusted Network Detection (TND) configuration is mandatory. Using the TND configuration, Network

Visibility Module determines if the endpoint is on the corporate network and then applies the appropriate policies.

If you are using NVM standalone and do not have other Cisco Secure Client deployments, the following guidelines must be considered:

- You cannot use the VPN uninstaller for NVM standalone uninstallation.
- You must manually remove NVM standalone if you want to install AnyConnect VPN.
- The following cannot be installed at the same time on the system:
  - cisco-secure-client-vpn and cisco-secure-client-vpn-cli
  - cisco-secure-client-vpn and cisco-secure-client-nvm-standalone
  - cisco-secure-client-vpn-cli and cisco-secure-client-nvm-standalone

Troubleshooting and logging is still done by Cisco Secure Client DART, which can be installed from the Cisco Secure Client package.

Deployment Modes

You can deploy Network Visibility Module 1) with the Cisco Secure Client package or 2) with the Standalone Network Visibility Module package (on Cisco Secure Client desktop only). Refer to the *Deploy Cisco Secure Client* chapter for steps to deploy as part of the Cisco Secure Client package. Otherwise, you can initially install the Standalone Network Visibility Module without the complete Cisco Secure Client package by downloading the following packages:

- cisco-secure-client-win-[version]-nvm-standalone-k9.msi (for Windows)
- cisco-secure-client-macos-[version]-nvm-standalone.dmg (for macOS)
- cisco-secure-client-linux64-[version]-nvm-standalone.tar.gz (for Linux)

Additionally Network Visibility Module is a core part of Cisco XDR. You can send telemetry directly to Cisco XDR without needing an on-premise collector by installing the XDR Default Deployment on your endpoints. Cisco XDR uses this data to create new detections, correlate multiple events into a single incident, and fill invisibility gaps in your network. Within XDR, you can navigate to Client Management > Deployments to see a list of all Secure Client deployments in your Cisco XDR organization and allows users to define a list of all packages and related profiles that must be installed on all computers in a specific deployment within an organization. Refer to XDR documentation for further details.

The Standalone Network Visibility Module does not depend on VPN for its functionining; therefore, you can deploy it on the endpoint without having to install VPN.

If standalone Network Visibility Module is already installed, you can seamlessly migrate to a full Cisco Secure Client installation of the same or higher version, and all Network Visibility Module data files and profiles will be retained.

To upgrade to a Network Visibility Module standalone configuration, you must use an out-of-band method (such as SMS) with an Network Visibility Module profile. If you need both VPN and Network Visibility Module functionality on the endpoint, we recommend that you deploy the Cisco Secure Client package to install both VPN and Network Visibility Module, as a separate installation is not recommended. Installation fails in the following scenarios:

- downgrading standalone Network Visibility Module

- installing an older version of Cisco Secure Client Network Visibility Module where a newer version of standalone Network Visibility Module already existed. This scenario would result in uninstallation of standalone Network Visibility Module.
- installing any version of standalone Network Visibility Module where Cisco Secure Client Network Visibility Module already existed

NVM on Mobile Cisco Secure Client

The Network Visibility Module (NVM) is included in the latest version of the Cisco Secure Client for Android available in the Google Play Store. Network Visibility Module is supported on Samsung devices running Samsung Knox version 2.8 or later. No other mobile devices are currently supported.

Network Visibility Module on Android is part of the service profile configurations. To configure Network Visibility Module on Android, the Cisco Secure Client Network Visibility Module profile is generated by the Cisco Secure Client Network Visibility Module Profile Editor, and then pushed to the Samsung mobile device using Mobile Device Management (MDM).

Guidelines

- Network Visibility Module is supported on Samsung devices running Samsung Knox version 3.0 or later. No other mobile devices are currently supported.
- On mobile devices, connectivity to the Network Visibility Module Collector is supported over IPv4 or IPv6.
- Data collection traffic on Java based apps is supported.

Provide Full Disk Access Manually for macOS

To gather a process tree hierarchy where a separate record is sent for each process in the flow, you must provide full disk access. On macOS 14 (and later), you must perform the following operation to gain the approval for Cisco Secure Client - AnyConnect VPN Service and Network Visibility Module. To gain the approval for the Network Visibility Module Standalone App, go to Full Disk Access to Network Visibility Module Standalone App, on page 5. To provide Full Disk Access through an MDM Profile, go to Full Disk Access Through MDM Profile, on page 5. Additionally, Secure Client release 5.1.7.x (and later) requires full disk access.

Before you begin

Before an application such as Network Visibility Module can access parts of the file system that contain personal user data (such as Contacts, Photos, Calendar, and other applications), you must configure macOS 14 (and later) to gain the approval.

Procedure

Step 1 Enable **Full Disk Access** on the macOS Security & Privacy Preferences pane.
You receive a popup to restart the Cisco Secure Client - AnyConnect VPN Service.

Step 2 Choose the **Quit & Reopen** option.
Upon restart, you should see that Cisco Secure Client - AnyConnect VPN Service now has *Full Disk Access*.

Step 3 Run the following command to restart the vpnagent:

sudo launchctl stop com.cisco.secureclient.vpn.service.agent

What to do next

Use the Network Visibility Module Profile Editor to enable the process tree hierarchy.

Full Disk Access to Network Visibility Module Standalone App

Network Visibility Module Standalone versions 5.1.7.x (and later) require Full Disk Access for acnvmagent.

Before you begin

Procedure

Step 1 Enable **Full Disk Access** on the macOS Security & Privacy Preferences pane.

Step 2 Enable **acnvmagent** in the Full Disk Access window.
When Full Disk Access is enabled for acnvmagent, you receive a popup to allow modification of the privacy settings.

Step 3 Click **Use Password** and enter your credentials in the Privacy & Security popup.
After you authenticate, you should see that acnvmagent now has Full Disk Access.

Step 4 Run the following command on the terminal to restart the nvmagent service:

sudo launchctl stop com.cisco.secureclient.nvmagent

Full Disk Access Through MDM Profile

You must configure a Privacy Preference Policy Control (PPPC) for macOS to provide Full Disk Access on MDM.

Procedure

Step 1 Open a new or existing PPPC configuration.

Step 2 Add a new entry under the 10.14+ System Policy (All files) privacy preference.

Step 3 Provide the following values:

- **Cisco Secure Client - AnyConnect VPN Service and Network Visibility Module**

Identifier—com.cisco.secureclient.vpn.service

Identifier Type—Choose **Bundle ID** from the drop-down.

Code Requirement—

```
anchor apple generic and identifier
"com.cisco.secureclient.vpn.service" and (certificate leaf[field.1.2.840.113635.100.6.1.9] /* exists
 */ or certificate 1[field.1.2.840.113635.100.6.2.6] /* exists */ and certificate
leaf[field.1.2.840.113635.100.6.1.13] /* exists */ and certificate leaf[subject.OU] =
DE8Y96K9QP)
```

- **Network Visibility Module Standalone Apps**

Identifier—com.cisco.secureclient.nvmagent

Identifier Type—Choose **Bundle ID** from the drop-down.

Code requirement—

```
anchor apple generic and certificate
 1[field.1.2.840.113635.100.6.2.6] /* exists / and certificate
leaf[field.1.2.840.113635.100.6.1.13] / exists */ and certificate leaf[subject.OU] =
DE8Y96K9QP
```

- **Cisco Secure Client - ISE Posture Module**

Identifier—/opt/cisco/secureclient/bin/csc_iseposture

Identifier Type—Choose **path** from the drop-down.

Code requirement—

```
anchor apple generic and certificate
      1[field.1.2.840.113635.100.6.2.6] /* exists / and certificate
      leaf[field.1.2.840.113635.100.6.1.13] / exists */ and certificate leaf[subject.OU] =
      DE8Y96K9QP
```

Step 4 Enable **Static Code**.

Step 5 Enable **Allowed**.

Step 6 Save the configuration and distribute it to the required endpoints.

What to do next

You must restart the vpnagent or nvmagent service (or perform a system restart) for this PPPC configuration to take effect after the distribution and installation on the endpoint.

How to Use Network Visibility Module

You can use Network Visibility Module for the following scenarios:

- To audit a user's network history for potential exfiltration after a security incident occurred.
- To see how system or adminstrative rights impact what network connected processes are running on a user's machine.
- To get a list of all devices running a legacy OS.
- To determine what application in your network is running the highest network bandwidth.
- To determine how many versions of Firefox are being used in your network.
- To determine what percentage of Chrome.exe connections are IPv6 in your network.

Collection Parameters for Network Visibility Module

Of the five syslog data sources: per flow, endpoint identity, interface info, process info, and osquery data, the the Unique Identifier (UDID) field is used as a way to correlate records between these sources. You can use the InterfaceInfoUDID field to correlate a per flow record with an interface info record for collecting details on that specific interface. The following parameters are collected at the endpoint and exported to the collector:

Table 1: Endpoint Identity

Parameter	Description / Notes
Endpoint Tag	The tag to identiy an endpoint as part of a sub-group
Virtual Station Name	Device name configured on the endpoint (for example, Boris-Macbook) Domain joined machines will be in the form <machinename>.<domainname>.<com> (for example, CESA-WIN10-1.mydomain.com) Empty for Android; not provided by Samsung.
UDID	Universally Unique Identifier. Uniquely identifies the endpoint corresponding to each flow. This UDID value is also reported by Secure Firewall Posture in Desktop, and ACIDex in Mobile.
OS Name	Name of the operating system on the endpoint (for example, WinNT)
OS Version	Version of the operating system on the endpoint (for example, 6.1.7601)
OS Edition	The OS edition, such as Windows 8.1 Enterprise Edition
SystemManufacturer	Endpoint manufacturer (for example, Lenovo, Apple, and so on)
System Type	Set to `arm` for Android. `x86` or `x64` for other platforms.
Agent Version	Version of Network Visibility Module client software running on the endpoint. Typically of the form major_v.minor_v.build_no
Timestamp	Absolute timestamp of the endpoint data in milliseconds.
AMP GUID	Unique endpoint ID of Secure Endpoint (AMP)

Table 2: Interface Information

Parameter	Description / Notes
Endpoint UDID	Same as UDID.

Parameter	Description / Notes
InterfaceInfoUID	Unique ID for an interface metadata. Used to look up the interface metadata from the InterfaceInfo records.
Interface Index	The index of the network interface as reported by the OS.
Interface Type	Interface type, such as wired, wireless, cellular, VPN, and so on.
Interface Name	Network interface/adapter name as reported by the OS.
Interface Details List	State and SSID, attributes of InterfaceDetailsList. Indicate the network state of the interface (trusted or untrusted), and the SSID of the connection.
Interface MAC address	MAC address of the interface. Desktop only. Empty for Android (not supported).
Timestamp	Absolute of the interface record in milliseconds.

Table 3: Osquery Data

Parameter	Description / Notes
UDID	Same as UDID
Osquery Version	Version of Osquery software running on the endpoint. Typically of the form major_v.minor_v.build_no
Query ID	8 byte unique ID for each query. First 4 bytes are reserved for Enterprise Number; the last 4 bytes are for query Index.
Timestamp	Absolute timestamp of the interface record in seconds
Current Page	Since JSON response for a query can be too large to send, Network Visibility Module paginates the JSON output. This field indicates which page is present in the current record.
Total Pages	Total number of pages into which the JSON response is divided.
Osquery JSON Response	Paginated JSON response output.

Table 4: Flow Information

Parameter	Description / Notes
Source IPv4 Address	IPv4 address of the interface from where the flow was generated on the endpoint.
Destination IPv4 Address	IPv4 address of the destination to where the flow was generated from the endpoint.
Source Transport Port	Source port number from where the flow was generated on the endpoint.

Parameter	Description / Notes
Destination Transport Port	Destination port number to where the flow was generated from the endpoint.
Flow Direction	The direction of the flow observed at the endpoint. it is a mandatory parameter collected from the endpoint. The two values are 0:ingress flow or 1:egress flow.
Source IPv6 Address	IPv6 address of the interface from where the flow was generated on the endpoint. Empty for Android (not supported).
Destination IPv6 Address	IPv6 address of the destination to where the flow was generated from the endpoint. Empty for Android (not supported).
Start Sec End Sec	The absolute timestamp of the start or end of the flow in seconds.
Start Msec End Msec	The absolute timestamp of the start or end of the flow in milliseconds.
Flow UDID	Same as UDID.
Logged In User	The logged in username on the physical device, in the form Authority\Principal Empty for Android (not supported).
Logged In User Account Type	Account type of the logged in user. Empty for Android (not supported).
Process ID	Process ID of the process that intitiated the network flow.
Process UID (PUID)	Unique identifier for the process
Process Name	Name of the executable generating the network flow on the endpoint.
Process Hash	Unique SHA256 hash for the executable generating the network flow on the endpoint.
Process Account	The fully qualified account, in the form Authority\Principle, under whose context the application generating the network flow on the endpoint was executed. Empty for Android (not supported).
Process Account Type	Account type of the process account. Empty for Android (not supported).

Parameter	Description / Notes
Process Path	Filesystem path of the process that intiated the network flow Empty for Android (not supported).
Process args	Command line arguments of the process that intiated the network flow, excluding the process path. Empty for Android (not supported).
Parent Process ID	Process ID of the parent of the process that initated the network flow.
Parent Process Name	Name of the parent process of the application generating the network flow on the endpoint.
Parent Process Hash	Unique SHA256 hash for the executable of the parent process of the application generating the network flow on the endpoint. Set to 0 for Android.
Parent Process Account	The fully qualified account, in the form Authority\Principle, under whose context the parent process of the application generating the network flow on the endpoint was executed. Empty for Android (not supported).
Parent Process Account Type	Account type of the parent process account. Empty for Android (not supported).
Parent Process Path	Filesystem path of the parent of the process that initiated the network flow. Empty for Android (not supported).
Parent Process Args	Command line arguments of the parent of the process that intiated the network flow, excluding the parent process path. Empty for Android (not supported).
DNS Suffix	Configured on the interface associated with the flow on the endpoint.
L4ByteCountIn	The total number of bytes downloaded during a given flow on the endpoint at layer 4, not including L4 headers.
L4ByteCountOut	The total number of bytes uploaded during a given flow on the endpoint at layer 4, not including L4 headers.
Destination Hostname	Actual FQDN that resolved to the destination IP on the endpoint
HTTP Host	Contents of HTTP Host header for HTTP/1.1 traffic.
Interface UID	Same as interface UID in interface information table. Used to identify the interface information for this flow from the interface records sent along with UDID.

Parameter	Description / Notes
Module Name List	List of 0 or more names of the modules hosted by the process that generated the flow. This can include the main DLLs in common containers, such as dllhost, svchost, rundll32, and so on. It can also contain other hosted components, such as the name of the jar file in a JVM. Empty for Android (not supported). For browser flows (on Chrome, Edge, Firefox, Safari), this field has a list of Browser Plugins installed for the corresponding browsers in "<name>-<version>" format. If there are more than 20 plugins, the module name list only contains 19 plugins' details, and the last entry will be the count of plugins (such as 19 of 23). **Note** In this case, module hash list is empty.
Module Hash List	List of 0 or more SHA256 hashes of the modules associated with the Module Name List. Empty for Android (not supported).
Additional Logged In Users List	(Windows Only) The list of logged in users on the device (other than nzFlowLoggedInUser), each in the form SessionType:AccountType:Authority\Principal. For example, rdp:8001:ACME\JSmith console:0002:<machine>\Administrator During upgrades, by default, this parameter is excluded from being reported: 1) if the profile in the older version of NVM had no Data Collection Policy or an include Data Collection Policy 2) if the profile in the older version of NVM had an exclude Data Collection Policy, and the profile was opened and saved with the 4.10 profile editor. **Note** For non-system processes, this field is empty.
Process Integrity Level	Integrity level defines the trust between process and another object (files, processes, or threads)
Parent Process Integrity Level	Integrity level defines the trust between parent process and another object (files, processes, or threads)
Flow Report Stage	Stage of the flow record. 0: End flow record, 1: Start flow record, and 2: Periodic/Intermediate flow record. EVE UUID : 16 byte Unique ID that ties EVE records with flow records.
EVE Data	EVE UUID : 16 byte Unique ID that ties EVE records with flow records EVE FDC : EVE blob returned by mercury library by fingerprinting client hello sent during TLS session.

Table 5: Process Information

Field	Description
UDID	Universally Unique Identifier. Uniquely identifies the endpoint corresponding to each flow. This UDID value is also reported by Secure Firewall Posture in Desktop, and ACIDex in Mobile.
Process UID (PUID)	Process UID (PUID)
Parent Process UID (PPUID)	PUID of the Parent process which has created the said process
Creation Parent Process UID (CPPUID)	PUID of the Parent process which has created the said process
Process Creation Timestampt (msec)	Creation Time of the process since Epoch in milliseconds
Process ID	Process ID of the process that intitiated the network flow.
Process Name	Name of the executable generating the network flow on the endpoint
Process Hash	Unique SHA256 hash for the executable generating the network flow on the endpoint.
Process Path	Filesystem path of the process that intiated the network flow Empty for Android (not supported).
Process args	Command line arguments of the process that intiated the network flow, excluding the process path. Empty for Android (not supported).
Process Account	The fully qualified account, in the form Authority\Principle, under whose context the application generating the network flow on the endpoint was executed. Empty for Android (not supported).
Process Account Type	Account type of the process account. Empty for Android (not supported).
Process Integrity Level	Integrity level defines the trust between process and another object (files, processes, or threads)

Field	Description
Module Name List	List of 0 or more names of the modules hosted by the process that generated the flow. This can include the main DLLs in common containers, such as dllhost, svchost, rundll32, and so on. It can also contain other hosted components, such as the name of the jar file in a JVM. Empty for Android (not supported).
Module Hash List	List of 0 or more SHA256 hashes of the modules associated with the Module Name List. Empty for Android (not supported)

Transport Securely to Connector

The Network Visibility Module allows clients to identify themselves using DTLS or mDTLS. You configure Network Visibility Module to determine how to send data over plain UDP (IPFIX) or DTLS (IPFIX over DTLS). DTLS requires that Network Visibility Module clients verify the Collector's identity before sending data. With mDTLS, the Collector can also verify the Network Visibility Module Client's identity before the connection is established and data is received.

You enable the transport method in the Network Visibility Module Profile Editor.

Network Visibility Module supports the following:

- RSA with PKCS1
- PSS padding
- ECDSA keys
- SHA hashing algorithms
- Client certificates with OCSP configuration

You must install the certificates in the appropriate designated location, as noted below. Only the client certificates with EKU ClientAuth and private keys (identity certificates) are used for client authentication. The Root or Intermediate client authentication certificates must also be present in the client certificate store.

- Windows client certificates—CERT_SYSTEM_STORE_LOCAL_MACHINE
- macOS Network Visibility Module client certificates—system keychain
- (Ubuntu) Linux client certificates—either /etc/ssl/certs or /etc/ssl/private
- (RedHat) Linux client certificates—either etc/pki/tls/certs or /etc/pki/tls/private

Network Visibility Module Profile Editor

In the profile editor, configure the IP address or FQDN of the collection server. You can also customize the data collection policy choosing what type of data to send, and whether data is anonymized or not.

Network Visibility Module can establish connection with a single stack IPv4 with an IPv4 address, a single stack IPv6 with an IPv6 address, or a dual stack IPv4/IPv6 to the IP address as preferred by the OS.

The mobile Network Visibility Module can establish a connection using IPv4 only. IPv6 connectivity is not supported.



Note The Network Visibility Module sends flow information only when it is on the trusted network. By default, no data is collected. Data is collected only when configured as such in the profile, and the data continues to be collected when the endpoint is connected. If collection is done on an untrusted network, it is cached and sent when the endpoint is on a trusted network. If you are sending collection data to Cisco Secure Cloud Analytics 7.3.1 and prior releases (or something other than Splunk or similar SIEM tool), cache data is sent once on a trusted network but not processed. For Cisco Secure Cloud Analytics applications, refer to the Cisco Secure Cloud Analytics Enterprise Endpoint License and NVM Configuration Guide.

If TND is configured in the Network Visibility Module profile, then the trusted network detection is done by Network Visibility Module and does not depend on VPN to determine if the endpoint is in a trusted network. Also, if VPN is in a connected state, then the endpoint is considered to be on the trusted network, and the flow information is sent. The NVM-specific system logs show Trusted Network Detection use.

When configuring TND directly in the Network Visibility Module profile, an administrator-defined trusted server and certificate hash determine whether the user is on a trusted or untrusted network. Administrators configuring Trusted Network Detection for the core VPN profile would alternatively configure the Trusted DNS Domains and Trusted DNS Servers in the core VPN profile: Cisco Secure Client Profile Editor, Preferences (Part 2).

- **Desktop** or **Mobile**—Determines whether you are setting up Network Visibility Module on a desktop or mobile device. **Desktop** is the default.
- **Collector Configuration**
  - **IP Address/FQDN**—Specifies the IPv4 or IPv6 IP address/FQDN of the collector.
  - **Port**—Specifies at which port number the Collector is listening.
  - You cannot set DTLS and mDTLS to operate at the same time.
- **Client Authentication**—Determines if you want Network Visibility Module to securely send data to the collector over mDTLS. You can choose this checkbox only if the *Secure* checkbox in Collector Configuration is also selected. When enabled, Network Visibility Module uses mDTLS for transport. With mDTLS, the Collector (server) can also verify the Network Visibility Module Client's identity before the connection is established and data can be received.

  For mDTLS, you must install client certificates in the System certificate store, such as:
  - Windows—CERT_SYSTEM_STORE_LOCAL_MACHINE
  - macOS—System keychain

- Ubuntu—"/etc/ssl/certs" or "/etc/ssl/private"
- Red Hat—"etc/pki/tls/certs" or "/etc/pki/tls/private

Only the client certificates with EKU ClientAuth and private keys (identity certificates) are used for client authentication. Root/Intermediate CA certificates of client certificates must also be present in the client Certstore. Currently Network Visibility Module supports RSA with PKCS1 and PSS padding as well as ECDSA keys and SHA hashing algorithms.

- **Configure**—Allows you to specify Client Match Criteria for Certificates used by client authentication (mDTLS). You can add up to four Distinguished Name Definitions as part of the Certificate Match Criteria. Network Visibility Module currently allows matching by subject name fields CN and OU, and Issuer subject name fields ISSUER-CN and ISSUER-OU.
- **Cache Configuration**
  - **Max Size**—Specify the maximum size the database can reach. The cache size previously had a pre-set limit, but you can now configure it within the profile. The data in the cache is stored in an encrypted format, and only processes with root privileges are able to decrypt the data.

    Once a size limit is reached, the oldest data is dropped from the space for the most recent data.
  - **Max Duration**—Specify how many days of data you want to store. If you also set a max size, the limit which reaches first takes precedence.

    Once the day limit is reached, the oldest day's data is dropped from the space for the most recent day. If only Max Duration is configured, there is no size cap; if both are disabled, the size is capped at 50MB.
- **Periodic Template**—Specify the period interval at which templates are sent out from the endpoint. The default value is 1440 minutes.
- **Periodic Flow Reporting** (Optional, applies to desktop only)—Click to enable periodic flow reporting. By default, Network Visibility Module sends information about the flow at the end of connection (when this option is disabled). If you need periodic information on the flows even before they are closed, set an interval in seconds here. The value of 0 means the flow information is sent at the beginning and at the end of each flow. If the value is *n*, the flow information will be sent at the beginning, every *n* seconds, and at the end of each flow. Use this setting for tracking long-running connections, even before they are closed.
- **Aggregation Interval**—Specify at which interval the data flows should be exported from the endpoint. When the default value of 5 seconds is used, more than one data flow is captured in a single packet. If the interval value is 0 seconds, each packet has a single data flow. The valid range is 0 to 600 seconds.
- **Throttle Rate**—Throttling controls at what rate to send data from the cache to the collector so that the end user is minimally impacted. You can apply throttling on both real time and cached data, as long as there is cached data. Enter the throttle rate in Kbps. The default is 500 Kbps.

  The cached data is exported after this fixed period of time. Enter 0 to disable this feature.
- **Collection Mode**—Specify when data from the endpoint should be collected by choosing: collection mode is off, trusted network only, untrusted network only, or all networks.
- **Collection Criteria**— You can reduce unnecessary broadcasts during data collection so that you have only relevant data to analyze. Control collection of data with the following options:
  - **Broadcast packets** and **Multicast packets** (Applies to desktop only)—By default, and for efficiency, broadcast and multicast packet collection are turned off so that less time is spent on backend

resources. Click the checkbox to enable collection for broadcast and multicast packets and to filter the data.

- **KNOX only** (Optional and mobile specific)—When checked, data is collected from the KNOX workspace only. By default, this field is not checked, and data from inside and outside the workspace is collected.

- **Data Collection Policy**—You can add data collection policies and associate them with a network type or connectivity scenario. You can apply one policy to VPN and another to non-VPN traffic since multiple interfaces can be active at the same time.

  When you click Add, the Data Collection Policy window appears. Keep these guidelines in mind when creating policies:

  - By default, HTTP headers are excluded from the Network Visibility Module Data Collection Policy.
  - By default, all fields are reported and collected if no policy is created or associated with a network type.
  - Each data collection policy must be associated with at least one network type, but you cannot have two policies for the same network type.
  - The policy with the more specific network type takes precedence. For example, since VPN is part of the trusted network, a policy containing VPN as a network type takes precedence over a policy which has trusted as the network specified.
  - You can only create a data collection policy for the network that applies based on the collection mode chosen. For example, if the **Collection Mode** is set to **Trusted Network Only**, you cannot create a **Data Collection Policy** for an **Untrusted Network Type**.
  - If a profile from an earlier Cisco Secure Client release is opened in a later Cisco Secure Client release profile editor, it automatically converts the profile to the newer release. Conversion adds a data collection policy for all networks that exclude the same fields as were anonymized previously.
  - **Name**—Specify a name for the policy you are creating.
  - **Network Type**—Determine the collection mode, or the network to which a data collection policy applies, by choosing VPN, trusted, or untrusted. If you choose trusted, the policy applies to the VPN case as well.
  - **Flow Filter Rule**—Defines a set of conditions and an action that can be taken to either Collect or Ignore the flow when all conditions are satisfied. You can configure up to 25 rules, and each rule can define up to 25 conditions. Use the up and down buttons to the right of the Flow Filter Rules list to adjust the priority of rules and give them higher consideration over subsequent rules. Click **Add** to set up the component of a flow filter rule.
    - Name—The unique name of the flow filter rule.
    - Type—Each filter rule has a Collect or Ignore type. Determine the action (Collect or Ignore) to apply if the filter rule is satisfied. If collect, the flow is allowed when conditions are met. If ignore, the flow is dropped.
    - Conditions—Add an entry for each field that is to be matched and an operation to decide if the field value should be equal or unequal for a match. Each operation has a field identifier and a corresponding value for that field. The field matches are case sensitive unless you apply case-insensitive operations (EqualsIgnoreCase) to the rule set when you are setting up the filter

engine rules. After it has been enabled, the input in the Value field set under the rule is case insensitive.

- **Include/Exclude**
  - **Type**—Determine which fields you want to **Include** or **Exclude** in the data collection policy. The default is **Exclude**. All fields not checked are collected. When no fields are checked, all fields are collected.
  - **Fields**—Determine what information to receive from the endpoint and which fields will be part of your data collection to meet policy requirements. Based on the network type and what fields are included or excluded, Network Visibility Module collects the appropriate data on the endpoint.



Note During an upgrade, the ProcessPath, ParentProcessPath, ProcessArgs, and ParentProcessArgs are excluded by default from being reported in the flow information, if one of these scenarios exist:

- If the profile in the older version of Network Visibility Module had no Data Collection Policy or had an include Data Collection Policy.
- If the profile in the older version of Network Visibility Module had an exclude Data Collection Policy, and the profile was opened and saved with a newer version profile editor. If the profile in the older version of Network Visibility Module had an exclude Data Collection Policy but the profile was *not* opened and saved with the newer 4.9 (or later) version profile editor, then these four fields are included.

If Network Visibility Module is unable to compute the parent process id, the value defaults to 4294967295.

FlowStartMsec and FlowStopMsec determine the Epoch timestamp of the flow in milliseconds.

You can choose Interface State and SSID, which specifies whether the network state of the interface is trusted or untrusted.

- **Optional Anonymization Fields**—If you want to correlate records from the same endpoint while still preserving privacy, choose the desired fields as anonymized. They are then sent as the hash of the value rather than actual values. A subset of the fields is available for anonymization.

  Fields marked for include or exclude are not available for anonymization; likewise, fields marked for anonymization are not available for include or exclude.

- **Data Collection Policy for Knox (Mobile Specific)**—Option to specify data collection policy when mobile profile is selected. To create Data Collection Policy for Knox Container, choose the **Knox-Only** checkbox under Scope. Data Collection policies applied under Device Scope apply for Knox Container traffic also, unless a separate Knox Container Data Collection policy is specified. To add or remove Data Collection Policies, see the Data Collection Policy description above. You can set a maximum of 6 different Data Collection Policies for mobile profile: 3 for Device, and 3 for Knox.

- **Acceptable Use Policy** (Optional and mobile specific)—Click **Edit** to define an Acceptable Use Policy for mobile devices in the dialog box. Once complete, click **OK**. A maximum of 4000 characters is allowed.

  This message is shown to the user once after Network Visibility Module is configured. The remote user does not have a choice to decline Network Visibility Module activities. The network administrator controls Network Visibility Module using MDM facilities.

- **Export on Mobile Network (Optional and Mobile Specific)**—Specifies whether the exporting of Network Visibility Module flows is allowed when a device is using a mobile network. If enabled (the default value), an end user can override an administrator when an Acceptable User Policy window is displayed or later by enabling the **Settings** > **NVM-Settings** > **> Use mobile data for NVM** checkbox in the Cisco Secure Client Android application. If you uncheck the **Export on Mobile Network** checkbox, Network Visibility Module flows are not exported when the device is using a mobile network, and an end user cannot change that.

- **Process Tree Hierarchy**—Until release 5.1.6, the flows reported by the Network Visibility Module only included information about the process and its parent process, and not information about the further process lineage. By clicking this checkbox, the Network Visibility Module will report the process lineage. Then a separate record (Process Record) is sent for each process in the process hierarchy. The process records will have a unique identifier (PUID) used to link between process records and flow records. You can receive a report of the process hierarchy for on-premises.



Note Currently, this feature is only supported in desktop platforms (Windows--only x86 and x64, macOS, and Linux), although macOS requires approval before an application can access parts of the file system. Refer to Provide Full Disk Access Manually for macOS, on page 4.

- **Enable EVE Data**— Allows Network Visibility Module to report EVE records.

- **Endpoint Tag**—Use this text box to enter a tag for the profile, if you choose to have one. This tag helps identify endpoints across overlapping subnet spaces or when inconsistencies with the logged-in user account exist.You can create virtual sub-groups of endpoints within the organization to be sent as an endpoint template that can be searched. Follow these rules for the tag:

  - Can contain only UTF-8 encoding compliant characters
  - No larger than 40 bytes. Any string that is larger than 40 bytes will not be accepted and will be reverted to the last acceptable string entered.
  - Cannot contain characters that are non-compliant with the XML format (such as &, <, or > and so on). We recommend that you use the profile editor instead of directly altering the profile with these certain characters, as it avoids runtime profile parsing errors.

- **Trusted Network Detection**—This feature detects if an endpoint is physically on the corporate network. The network state is used by the Network Visibility Module to determine when to export data and to apply the appropriate Data Collection Policy. Click **Configure** to set the configuration for Trusted Network Detection. An SSL probe is sent to the configured trusted headend, which responds with a certificate, if reachable. The thumbprint (SHA-256 hash) is then extracted and matched against the hash set in the profile editor. A successful match signifies that the endpoint is in a trusted network; however, if the headend is unreachable, or if the certificate hash does not match, then the endpoint is considered to be in an untrusted network.

Note When operating from outside your internal network, Trusted Network Detection makes DNS requests and attempts to establish an SSL connection to the configured server. Cisco strongly recommends the use of an alias to ensure that the name and internal structure of your organization are not revealed through these requests by a machine being used outside your internal network.

1. **https://**—Enter the URL (IP address, FQDN, or port address) of each trusted server and click **Add**.

Note Trusted servers behind proxies are not supported.

2. **Certificate Hash (SHA-256)**—If the SSL connection to the trusted server is successful, this field is populated automatically. Otherwise, you can set it manually by entering the SHA-256 hash of the server certificate and clicking **Set**.
3. **List of Trusted Servers**—You can define multiple trusted servers with this process. (The maximum is 10.) Because the servers are attempted for trusted network detection in the order in which they are configured, you can use the **Move Up** and **Move |Down** buttons to adjust the order. If the endpoint fails to connect to the first server, it tries the second server and so on. After trying all of the servers in the list, the endpoint waits for ten seconds before making another final attempt. When a server authenticates, the endpoint is considered within a trusted network.

- **Osquery Data Configuration**—Click **Configure** to allow endpoints to set options related to the Osquery Data Template. Network Visibility Module uses Osquery to run the queries and send output in JSON format with the Osquery Data Template.

  Currently, Network Visibility Module only allows the sending of Browser Plugins Data using Osquery.You can enable the reporting of browser plugins and also configure the browser plugin report intervals. The Browser Plugins Report Interval is from 1 to 720 hours and is 24 hours by default.

Note Currently Network Visibility Module is sending Chrome-Based Browsers' Browser Plugins and Firefox Browser Plugins.

Save the profile as `NVM_ServiceProfile.xml`. You must save the profile with this exact name or Network Visibility Module fails to collect and send data.

Extend Linux Kernel Capabilities

NVM supports two capture mechanisms: eBPF and a kernel module (kmod). Using a Berkeley Packet Filter (eBPF), you can extend Linux Kernel capabilities within Network Visibility Module for networking, security, and observability. If eBPF loads successfully but the kernel module build/insert fails, installation should still succeed, with NVM operating in eBPF-only mode. If both eBPF and the kernel module cannot initialize, the installation will fail. To enable and use eBPF for network monitoring, enter **<EBPF>true</EBPF>** anywhere within the <NVMProfile> root element. The default behavior is to enable it even if the tag is omitted. To

disable eBPF and force fallback to the common Linux Kernel Module usage (regardless of whether the Kernel supports eBPF), enter **<EBPF>false</EBPF>**.

About Flow Filters

The addition of flow filters extends the current data collection policy from being just field centric, where the action is configured for a given field in each flow. With flow filter, you can create and apply rules to collect or ignore entire flows (as opposed to only particular fields), thus monitoring only the traffic of interest and potentially reducing storage requirements.

Rule Conditions

- A rule is a match only if *all* of the conditions specified in the rule are satisfied when matched against the flow data.
- The first rule that is satisfied is applied on the flow.
- The rest of the data collection policy (include/exclude fields, anonymized fields) is also applied on the flow if it is allowed by the filter policy.
- With instances of multiple rules,
  - No action is taken on the flow if no rule matches the flow data. The default behavior is followed, which is to collect the flow.
  - If a rule matches the flow data, the action specified in that rule of the flow is applied. Subsequent rules are not checked. The order of rules, as designated in the Network Visibility Module Profile Editor Flow Filter Rule parameter, indicates the priority if multiple matches could occur.

Use of Wildcard, CIDR, and Escape Sequence Support

When entering rule conditions, you can define a wider range of field values using wildcard characters or CIDR notations, in the case of IP addresses. Also, you can use certain escape sequences in the field value. For IP fields, the CIDR/slash notation can specify the IP address that the rule should match. For example, "192.30.250.00/16" would match all addresses that have the routing prefix "192.30.0.0" derived by applying the subnet mask of "255.255.0.0." For text fields, you can use wildcards (* and ?) and escape sequences (*, \?, and \\) to catch a wider range of inputs. For example, logged in user "Jane*" would match all user names that start with "Jane."

Sample Configurations to Achieve Flow Filtering Scenarios

To drop all UDP traffic on a particular port (such as port 53), configure a flow filter rule with type *Ignore* and two conditions:

- Condition 1: Specify that the Flow Protocol *Equals* UDP.
- Condition 2: Specify that the port number *Equals* 53.

To only collect traffic originating from only one particular process (such as Tor Browser), configure a filter rule with type *Ignore* to drop all other flows by adding one condition:

- Condition 1: Specify that the process name *Not Equals* Tor Browser.

To only collect traffic originating from only one particular IP in a subnet, configure two rules:

- Rule 1: Set a rule of type *Collect* with the condition that the IPv4 source address *Equals* 192.168.30.14.
- Rule 2: Set a second rule of type *Ignore* with the condition that the IPv4 source *Equals* 192.168.30.0/24.

Osquery Data Configuration

This feature allows the endpoint to configure options related to Osquery Data Template. Network Visibility Module uses Osquery to run the queries and send output in JSON format with the Osquery Data Template. Click **Configure** to set the configuration for Osquery Data.

Currently, Network Visibility Module only allows the sending of Browser Plugins Data using Osquery.You can enable the reporting of browser plugins and also configure the browser plugin report intervals. The Browser Plugins Report Interval is from 1 to 720 hours and is 24 hours by default.



Note Currently Network Visibility Module is sending Chrome-Based Browsers' Browser Plugins and Firefox Browser Plugins.

Customer Feedback Module Gives NVM Status

Part of the Customer Feedback Module collection provides data about whether Network Visibility Module is installed or not, the number of flows per day, and the DB size.