

Understand and Configure macOS Service ISE Posture Condition

Contents

[Introduction](#)

[Prerequisites](#)

[Requirements](#)

[Components Used](#)

[Background Information](#)

[Configure](#)

[Identify the Service Name to be Checked](#)

[\(Optional\) Check the Details of the Service to Define if its Either an Agent or a Deamon](#)

[Select the Service Operator to be Evaluated](#)

[Loaded Services](#)

[Not Loaded Services](#)

[Loaded & Running](#)

[Loaded with Exit Code](#)

[Loaded & Running or with Exit Code](#)

[Configure the Requirement and Posture Policy for Such Condition](#)

[Verify](#)

[Troubleshoot](#)

[Certificate not Trusted](#)

[Bypassing Cisco Secure Client Scan](#)

[Other Issues](#)

Introduction

This document describes the process of configuring macOS service condition in Cisco ISE.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Basic knowledge of macOS.
- Knowledge about ISE Posture flow.



Note: This document covers the configuration for the macOS service condition. Initial posture configuration is not covered in this document.

Components Used

The information in this document is based on these software and hardware versions:

- Cisco ISE 3.3 patch 1
- macOS device running Sonoma 14.3.1
- Cisco Secure Client 5.1.2.42
- Compliance Module version 4.3.3432.64000

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, ensure that you understand the potential impact of any command.

Background Information

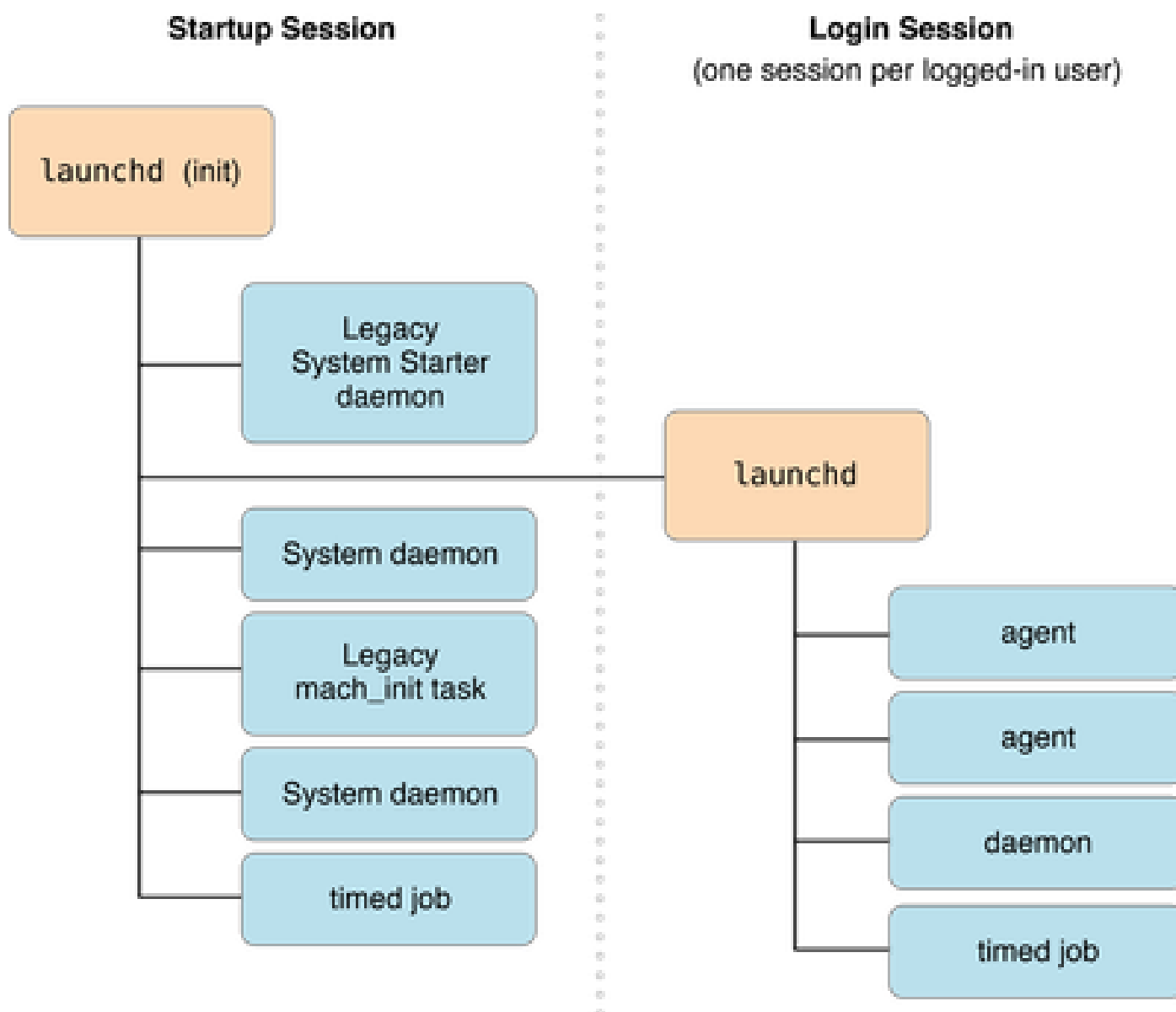
macOS service condition is useful when you have to use case to check if a service is loaded in the macOS

device, and also allows you to check if it is either running or not running. The macOS service condition can check two different service types: daemons and agents.

A daemon is a program that runs in the background as part of the overall system (that is, it is not tied to a particular user). A daemon cannot display any GUI; more specifically, it is not allowed to connect to the window server. A web server is the perfect example of a daemon.

An agent is a process that runs in the background on behalf of a particular user. Agents are useful because they can do things that daemons cannot, like reliably access the user's home directory or connect to the window server. A calendar monitoring program is a good example of an agent.

In the below diagram you can see how each is loaded based on device startup and user log in:



More information about daemons and agents can be found here in [Apple documentation](#)

Deamons and Agents available on your macOS device are found in the following locations:

Location	Description
----------	-------------

~/Library/LaunchAgents	Per-user agents provided by the user.
/Library/LaunchAgents	Per-user agents provided by the administrator.
/Library/LaunchDaemons	System wide daemons provided by the administrator.
/System/Library/LaunchAgents	OS X Per-user agents
/System/Library/LaunchDaemons	OS X System wide daemons

You can check the list of each category from macOS terminal using these commands:

```
ls -ltr ~/Library/LaunchAgents
ls -ltr /Library/LaunchAgents
ls -ltr /Library/LaunchDaemons
ls -ltr /System/Library/LaunchAgents
ls -ltr /System/Library/LaunchDaemons
```

The previous locations can show you all daemons and agents that are available on the macOS device, however not all of them are either loaded or running.

Configure

The configuration for macOS service condition can be done using these steps:

1. Identify the service name to be checked.
2. (Optional) Check the details of the service to define if its either an Agent or a Deamon.
3. Select the **service operator** to be evaluated.
4. Configure the **requirement** and **posture policy** for such condition.

Note: Service posture condition requires elevated privilege to work, hence, it is a MUST that ISE PSN is trusted by Cisco Secure Client (formerly AnyConnect) - [Reference Guide](#)

Identify the Service Name to be Checked

ISE Posture Compliance Module is able to check for the services that loaded, running and loaded, and running with exit code.

To check the services that are loaded use the command **sudo launchctl dumpstate**.

To check the services that are loaded and have an exit code use the command **sudo launchctl list**.

The previous commands can abruptly show a lot of information, instead, use these commands to just display the actual service name:

To check only the service names that are loaded, use this command:

```
sudo grep -B 10 -A 10 -E '^s*state = ' <<< "$(launchctl dumpstate)" | grep -aiE '\\.*= {' | sed 's|.*|/|;s| = {$|'
```

To check only the service names that are loaded and have an exit code use this command:

```
sudo launchctl list | awk '{if (NR>1) print $3}'
```

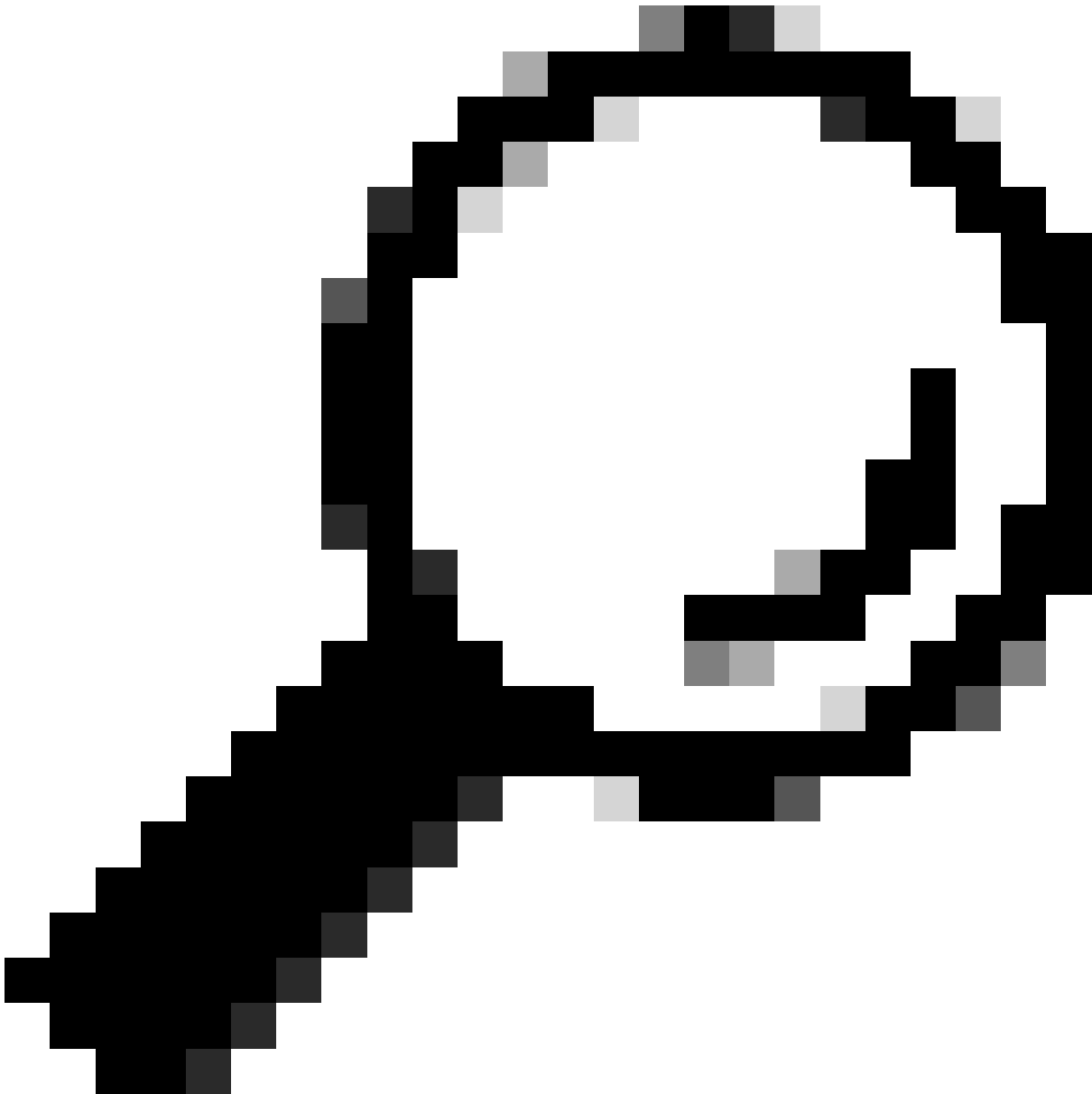
These commands show a lot of information, so at the end of each command it is recommended you use another grep filter to find the service you are looking for.
For example, if you are looking for a vendor specific service, you may use a keyword as a filter at the end.

For the case of Cisco services, the commands would be something like this:

```
sudo grep -B 10 -A 10 -E '^s*state = ' <<< "$(launchctl dumpstate)" | grep -aiE '\\.*= {' | sed  
's|.*//|;s| = {$||' | grep -i cisco  
sudo launchctl list | awk '{if (NR>1) print $3}' | grep -i cisco
```

(Optional) Check the Details of the Service to Define if its Either an Agent or a Deamon

In the second part of the configuration of this condition, you need to check if your service is daemon type or agent type.



Tip: This step is optional, since ISE allows you to select the option for **Daemon Or User Agent**, so you can just select that option and skip this part.

In case you want to be granular in this condition, you can check the type by doing this:

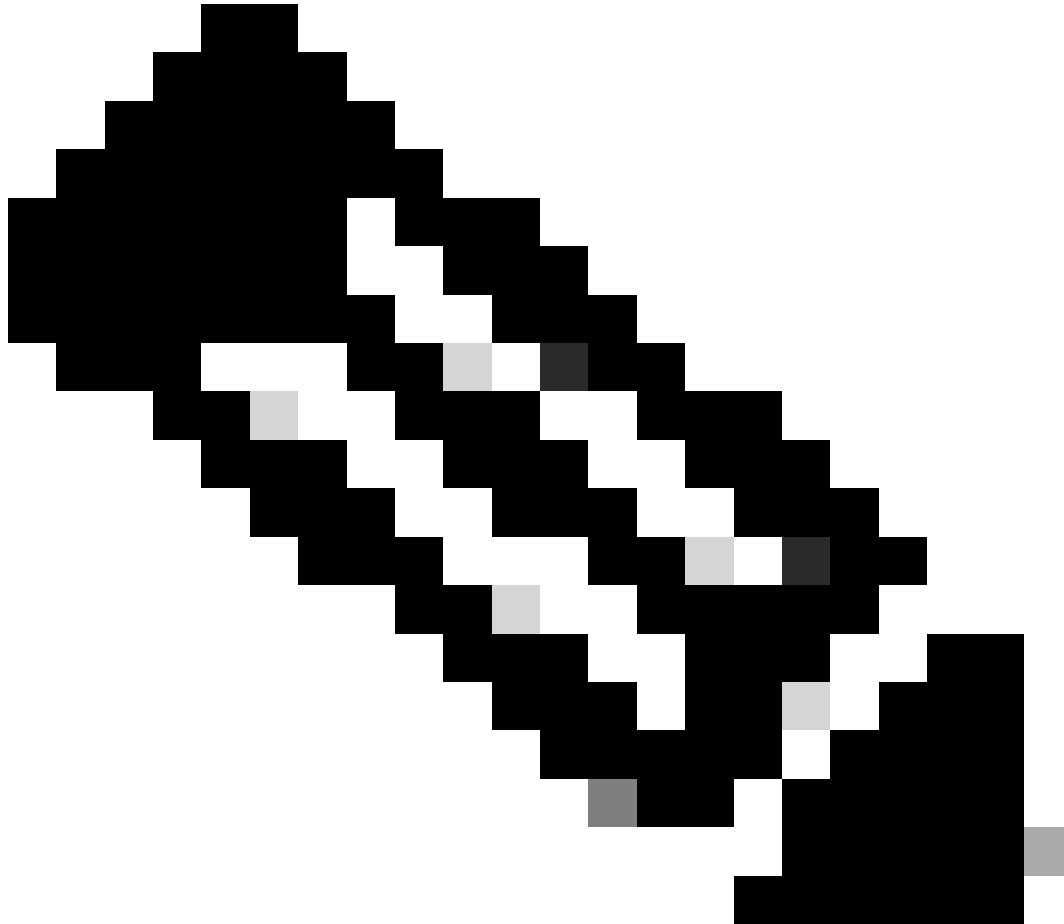
1. First, check the full launchctl name of the service with the command **sudo grep -B 10 -A 10 -E "^s*state = " <<< "\$(launchctl dumpstate)" | grep -aiE "\\.*= {" | sed 's|.|/|;s|= {\$|'| grep -i { Your service name }**

For example, for the command **sudo grep -B 10 -A 10 -E "^s*state = " <<< "\$(launchctl dumpstate)" | grep -aiE "\\.*= {" | sed 's|.|/|;s|= {\$|'| grep -i com.cisco.secureclient.iseposture**, the output is: gui/501/com.cisco.secureclient.iseposture.

2. Check the service type with the command **sudo launchctl print { Your launchctl service name } | grep -i 'type = Launch'**

Following the example, for the command: **sudo launchctl print gui/501/com.cisco.secureclient.iseposture | grep -i 'type = Launch'**, the output is: type = LaunchAgent.

This means that service type is Agent, otherwise it would show **type = LaunchDaemon**.



Note: In case the information is empty, select the option for **Daemon Or User Agent** in ISE for the service type setting.

Select the Service Operator to be Evaluated

ISE allows you to select 5 different service operators:

- Loaded
- Not Loaded
- Loaded & Running
- Loaded with exit code
- Loaded & Running or with exit code

Loaded Services

Are all those services which are listed when using these two commands:

```
sudo grep -B 10 -A 10 -E ""^s*state = " <<< "$(launchctl dumpstate)" | grep -aiE ""\.*= {" | sed 's|.*||;s|={ $||'
sudo launchctl list | awk '{if (NR>1) print $3}'
```

Not Loaded Services

Are all those services which have their property list (plist) defined, but have not been loaded, or services that do not even have a property list (plist) defined, so cannot be loaded at all.

These services are not easy to be identified, and it is most common for the use case when you want to check that an specific service should not exist in the macOS device.

For example, if you want to prevent the zoom service to be loaded on the macOS device, you may put here **us.zoom.ZoomDaemon** as the value for the service, this way you make sure zoom is not running or not installed at all.

There are services that cannot be uninstalled, and its property list is defined. For example, with this command, you can see that `dhcpcd` plist is defined:

```
ls -ltr /System/Library/LaunchDaemons | grep com.apple.dhcp6d.plist
```

Checking the services list, you can see that is not loaded:

```
sudo grep -B 10 -A 10 -E '^s*state = ' <<< "$(launchctl dumpstate)" | grep -aiE '\.*= {" | sed
's|\.*/||;s|={ $||' | grep -i com.apple.dhcp6d
sudo launchctl list | awk '{if (NR>1) print $3}' | grep -i com.apple.dhcp6d
```

If you set the value to **com.apple.dhcp6d**" the macOS device is compliant, because even though the service plist is defined, the service is not loaded.

Loaded & Running

Not all services are running, there are multiple states for each service, like running, not running, waiting, exited, uninitialized, and so on.

To check all those services that are running, use this command:

```
sudo grep -B 10 -A 10 -E "^\s*state = running" <<< "$(launchctl dumpstate)" | grep -aiE "\\.*= {" |
sed 's|.*/||;s|= {$|'
```

Services listed with above command hit **Loaded & Running** service operator condition.

Loaded with Exit Code

Some services may terminate with an expected or unexpected exit code, such services can be listed with command:

```
sudo grep -B 10 -A 10 "state = e" <<< "$(launchctl dumpstate)" | grep -aiE "'\.*= {" | sed 's/.\{3\}$//'
```

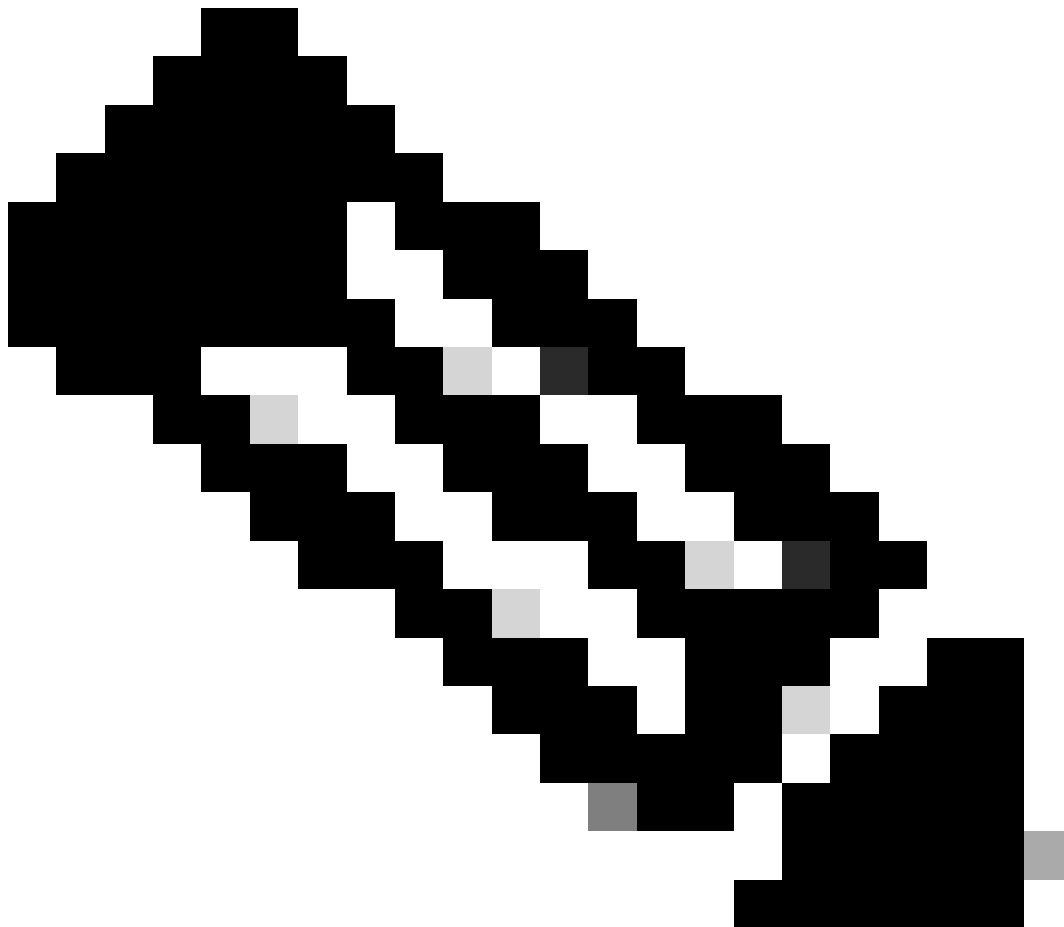
To know its exit code, you can pick any service and use the command:

```
sudo launchctl print { Your launchctl service name } | grep -i 'last exit code'
```

For example:

```
sudo launchctl print gui/501/com.apple.mdmclient.agent | grep -i 'last exit code'
```

which its output is: last exit code = 0



Note: Here the exit code 0 usually means that everything was done correctly by the service. If a computer does not match the 0 as the exit code, it means that the service did not perform the expected action.

Loaded & Running or with Exit Code

This last option works when the service is either Loaded & Running or Loaded with exit code.

This image shows an example of a macOS service condition.



Conditions

- Anti-Malware
- Anti-Spyware
- Anti-Virus
- Application
- Compound
- Dictionary Compound
- Dictionary Simple
- Disk Encryption
- External DataSource
- File
- Firewall
- Hardware Attributes
- Patch Management
- Registry
- Script

Service

USB

Remediations

- Requirements
- Allowed Protocols

[Service Conditions List](#) > macOS-Service-Condition

Service Condition

* Name

macOS-Service-Condition

Description

* Operating System

Mac OSX

Compliance Module

Any version

* Service Name

com.apple.sysmond

Service Type

Daemon Or User Agent

Service Operator

Loaded & Runnin

exit code

0



Note: Currently, only exact service name is supported. There is an enhancement request to support wildcard in the service names, Cisco bug ID [CSCwf01373](#)

Configure the Requirement and Posture Policy for Such Condition

Once the condition is configured, you need to create a requirement for such condition, use the **Message Test Only** option for this requirement.

Navigate to **ISE > Work Centers > Posture > Requirements** to create it.

Note: There are no remediation options for Service conditions.

Identity Services Engine

Work Centers / Posture

OverviewNetwork DevicesClient ProvisioningPolicy ElementsPosture PolicyPolicy SetsTroubleshootReportsSettings

Conditions

Anti-MalwareAnti-SpywareAnti-VirusApplicationCompoundDictionary CompoundDictionary SimpleDisk EncryptionExternal DataSourceFileFirewallHardware AttributesPatch ManagementRegistryScriptServiceUSB

Remediations

Requirements

Guide Me

service

Name	Operating System	Compliance Module	Posture Type	Condition
macOS-Service-Requireme	for Mac OSX	using 4.x or later	using Agent	met if

Remediation Action Details

Message Text

Only

Message

macOS Service is non compliant

Note:

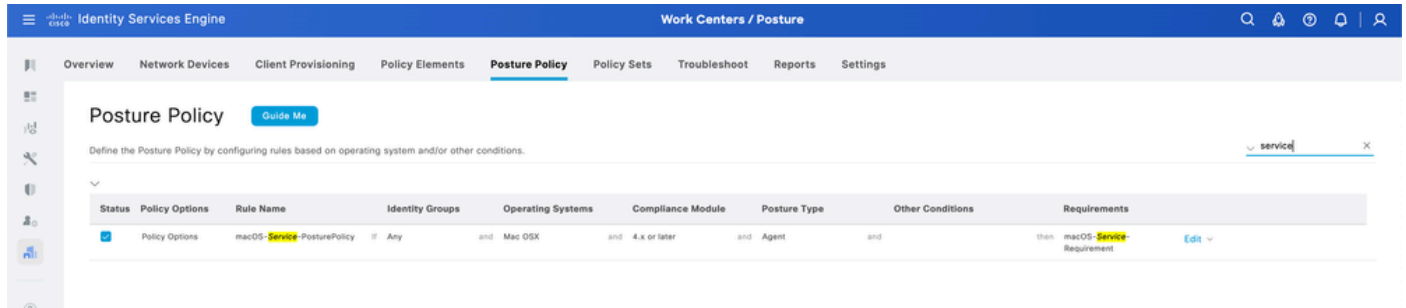
Remediation Action is filtered based on the operating system and stealth mode selection.
Remediation Actions are not applicable for Application Conditions (configured using the Provision By Category or Provision By Everything options), Hardware Conditions, and External Data source conditions.
Remediations Actions are not applicable for Agentless Posture type.

Save

Reset

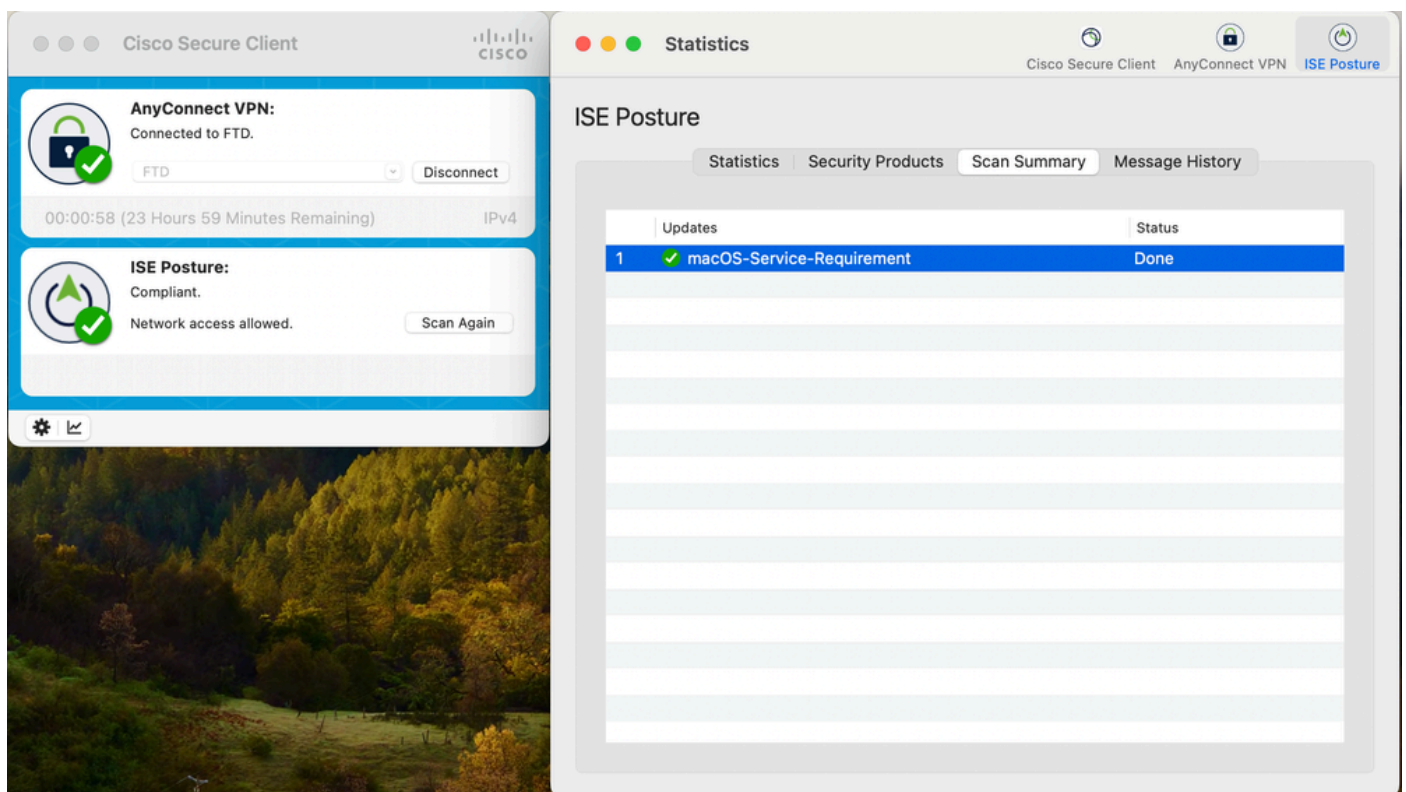
Once this is done, the last step is to configure the Posture Policy that uses the created requirement. Navigate to **ISE > Work Centers > Posture > Posture Policy** to create the policy.

Enable the **new policy**, name it as you wish and select the **requirement** you just created.



Verify

You can verify that macOS posture condition passed or failed, from the Cisco Secure Client GUI itself.



As well, you can check the ISE Posture report from **ISE > Operations > Reports > Reports > Endpoints and Users > Posture Assessment by Endpoint**.

Identity Services Engine

System User

ruben

User Domain

n/a

AV Installed

AS Installed

AM Installed

Gatekeeper;14.3.1::Xprotect;2186;

Posture Report

Posture Status

Compliant

Logged At

2024-02-28 09:44:28.926

Posture Policy Details

Policy	Name	Enforcement Type	Status	Passed Conditions	Failed Conditions	Skipped Conditions
macOS-Service-PosturePolicy	macOS-Service-Requirement	Mandatory	Passed	macOS-Service-Condition		

Rows/Page 1

Troubleshoot

Common issues you may encounter while configuring this macOS service posture condition are:

Certificate not Trusted

Cisco Secure Client

AnyConnect VPN:
Connected to FTD.

FTD Disconnect

00:02:42 (23 Hours 57 Minutes Remaining) IPv4

ISE Posture:
Scanning system ...

10% Scan Again

Security Warning: Untrusted Server Certificate!

Cisco Secure Client cannot verify server: ise-demo-6.ivillega.com

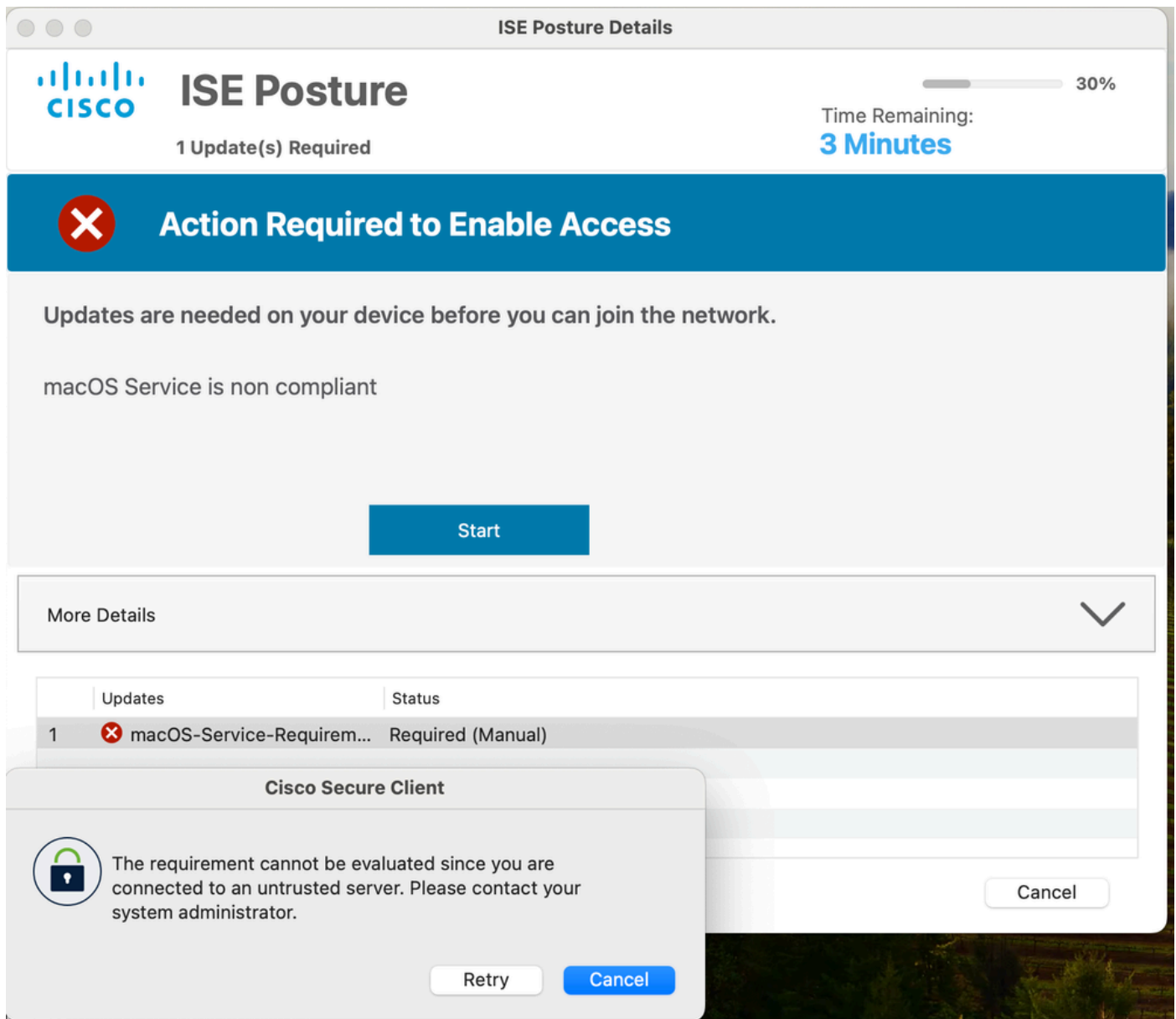
Certificate is not trusted.

Connecting to this server may result in a severe security compromise!
[Security Risks Explained](#)

Most users do not connect to untrusted servers unless the reason for the error condition is known.

Connect Anyway Cancel Connection

As indicated previously, service condition requires elevated permissions. It is imperative that certificate for posture scan process is trusted by the server. Otherwise you encounter this error:

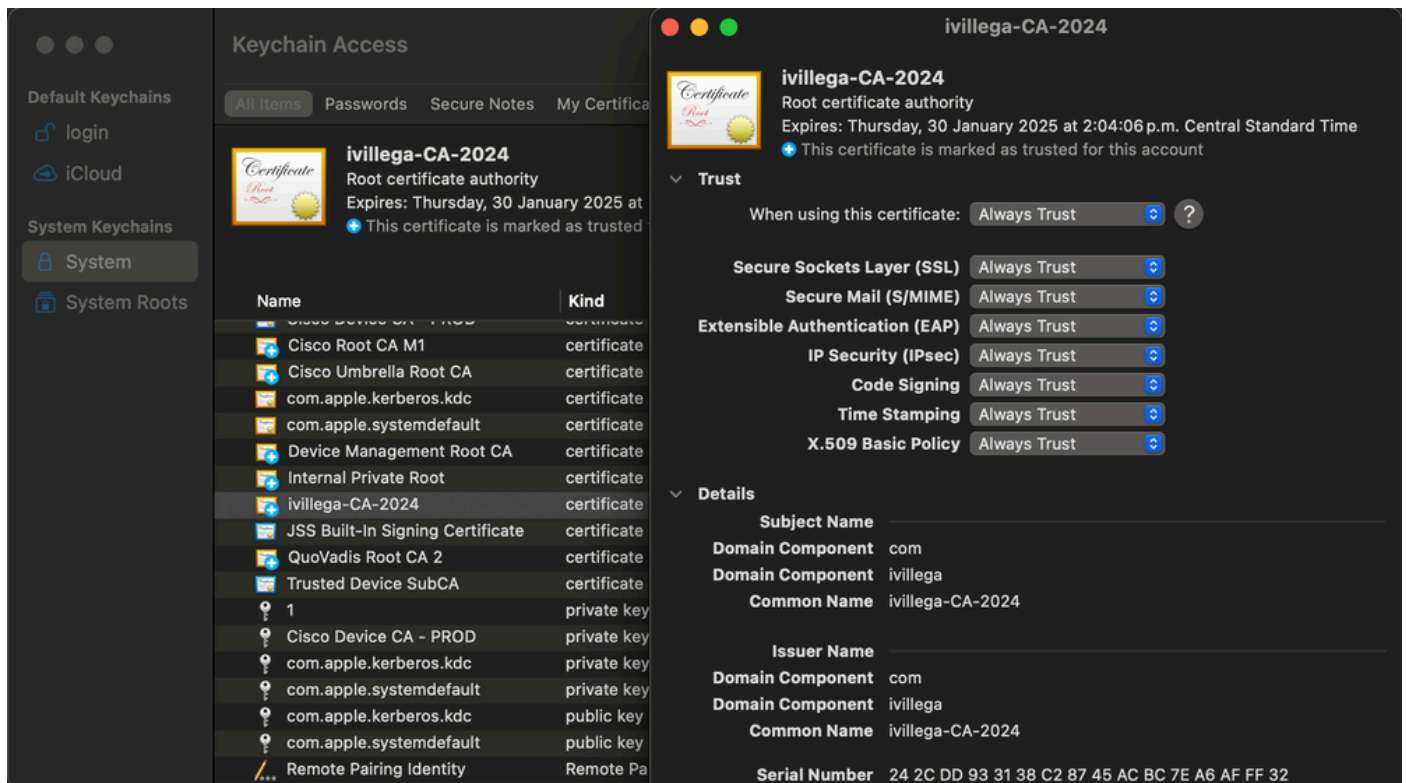


ISE Posture module discovers the PSN servers either by IP address or Fully Qualified Domain Name (FQDN). Best practice is to have the Posture configuration files to discover the ISE nodes through FQDN, so the Admin and Portal (Client Provisioning Portal) certificates should include the FQDN in the CN field or the SAN field. You can use wildcard certificate for this as well, wildcard certificates are supported for this flow.

Due to system securities, CN field cannot be trusted in the future. Include the wildcard entry or the FQDN in the SAN field as a best practice.

In case the ISE PSNs are discovered through IP address instead of FQDN, it is a requirement that IP address of the nodes are included either in the CN field or the SAN field of the certificate(s) linked to the Admin and Portal usage.

ISE Posture modules trust in the certificate presented by the ISE server. If its CA is in the System certificate store of the macOS **Keychain access**, this CA should have the setting **When using this certificate** set to **Always Trust**.



You may encounter the misbehavior that even when the certificate is loaded correctly and all the CN and SAN requirements are met, the macOS system still does not trust the certificate. In such cases, open **Keychain access** application, navigate to the **System certificate store** tab, and delete the **CA certificate** from there.

Then, navigate to macOS Terminal application and perform this command: **sudo /usr/bin/security add-trusted-cert -r trustRoot -d -k /Library/Keychains/System.keychain**

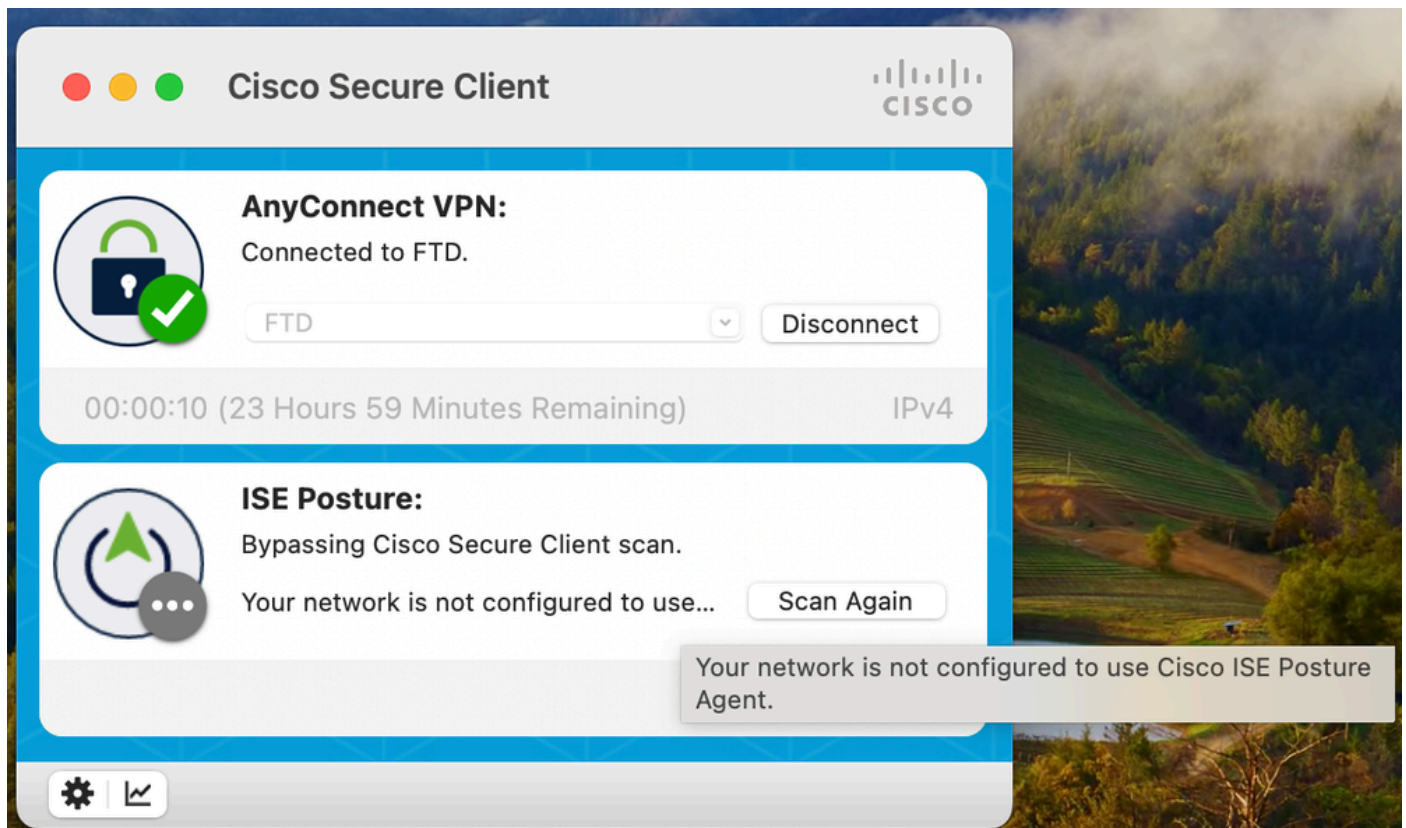
{Path to your CA certificate}

For example, if your certificate is in in your Desktop, the command is this: **sudo /usr/bin/security add-trusted-cert -r trustRoot -d -k /Library/Keychains/System.keychain /Users/JohnDoe/Downloads/CA_certificate.crt**

After doing the command, restart the computer and try again.

Bypassing Cisco Secure Client Scan

You may also encounter the error messages "Bypassing Cisco Secure Client Scan", and "Your network is not configured to use Cisco ISE Posture Agent":



This message appears because there are no profiles configured in the Client Provisioning in **ISE > Work Centers > Posture > Client Provisioning > Client Provisioning Policies**.

Even though you may see a condition for Mac OSX operating systems, that does not mean that you are covering all macOS versions.

By default, ISE does not include latest macOS versions, such as Sequoia (15.6.x), to avoid such message make sure the that posture is updated.

You must update the Posture feed from **ISE > Work Centers > Posture > Settings > Software Updates > Posture Updates**.

This can be updated online directly from ISE, or offline through a zip file that can be downloaded here from [Posture Offline site](#)

Other Issues

If you want to go into the details, you can collect a DART bundle from the postured macOS device. For this you must have the DART module installed, then, with Cisco Secure Client application active navigate to the **Menu** bar and click **Cisco Secure Client** and then, in **Generate Diagnostics Reports**.



Cisco Secure Client

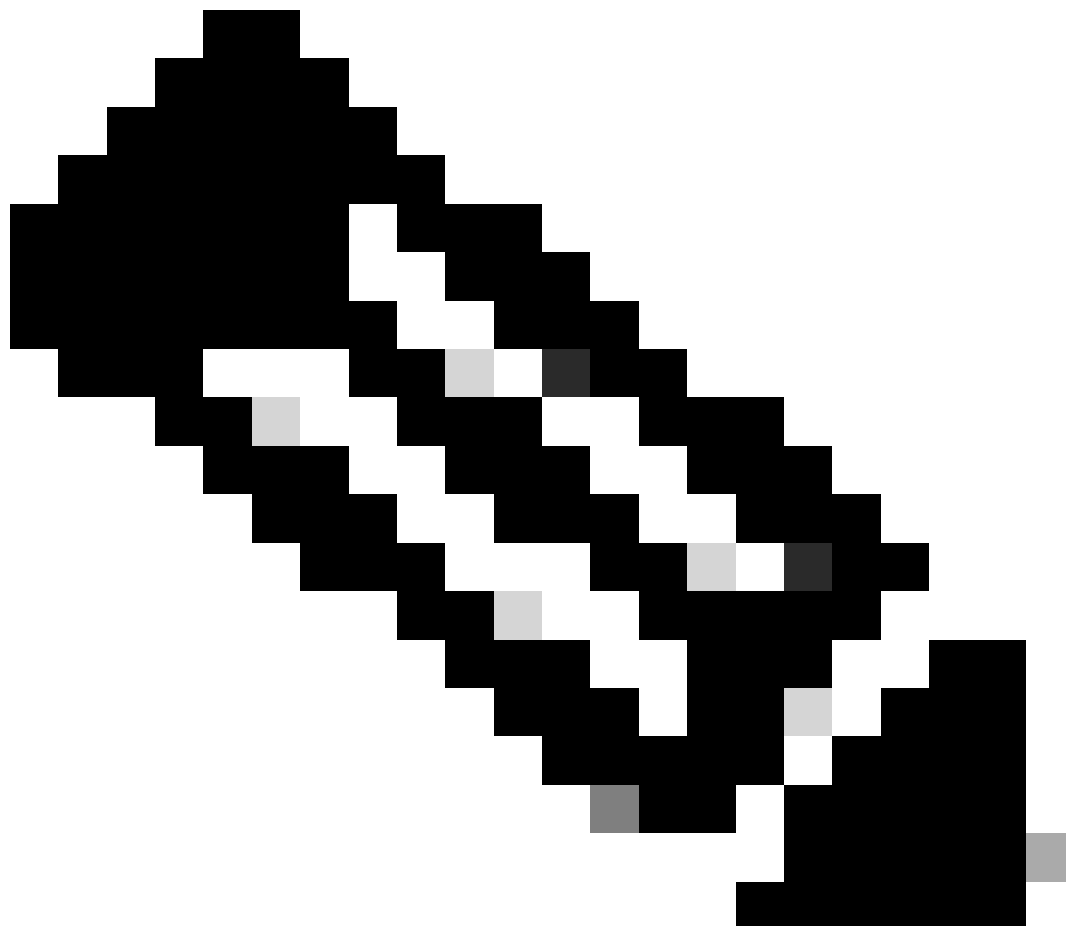
Edit

About Cisco Secure Client

Preferences...

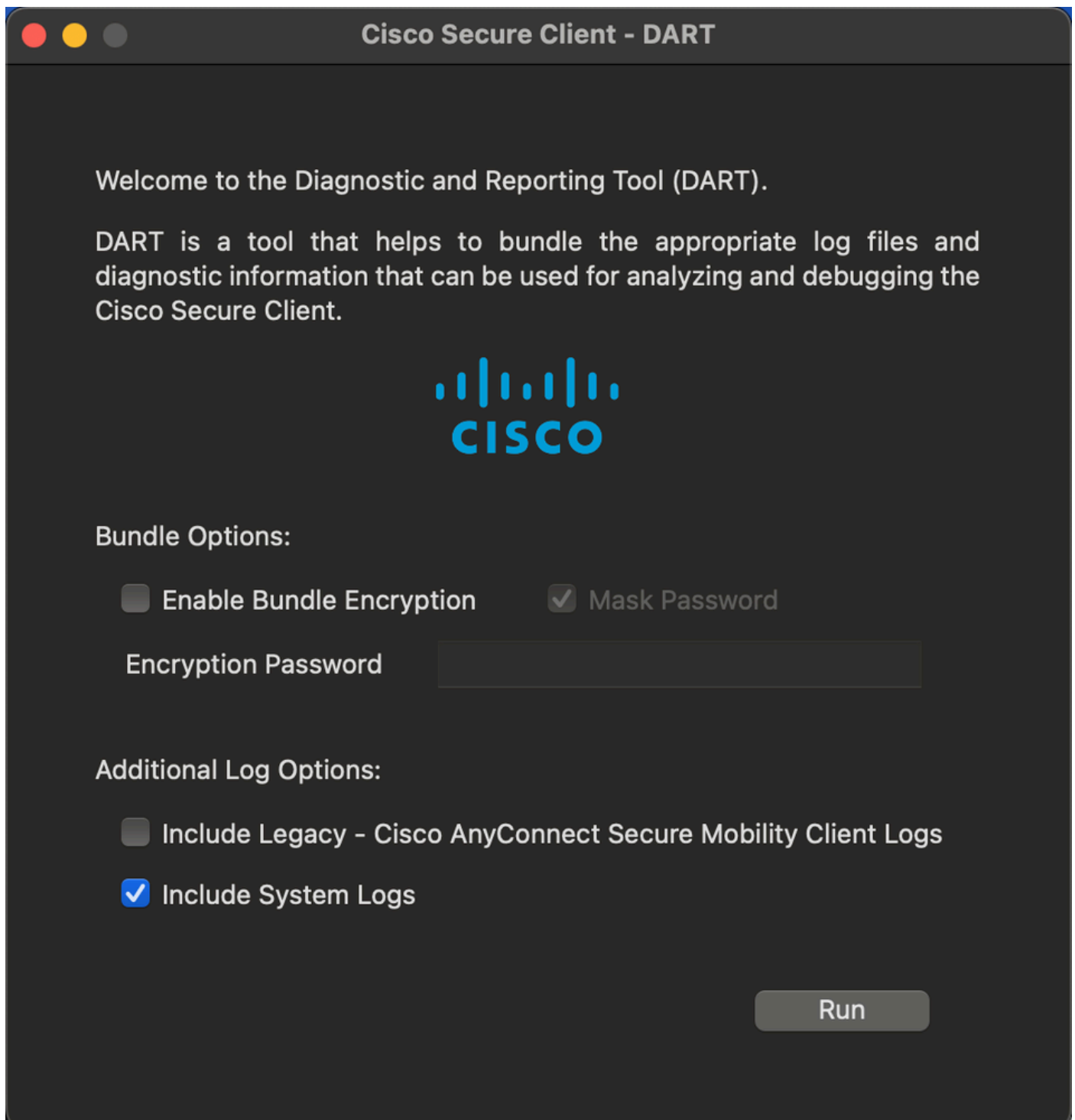


Generate Diagnostics Report



Note: It is important to have the **Include System Logs** option enabled when generating the DART

bundle, otherwise DART bundle is not going to include ISE posture module information.



Cisco Secure Client - DART

Welcome to the Diagnostic and Reporting Tool (DART).

DART is a tool that helps to bundle the appropriate log files and diagnostic information that can be used for analyzing and debugging the Cisco Secure Client.



Bundle Options:

☐ Enable Bundle Encryption ☒ Mask Password

Encryption Password

Additional Log Options:

☐ Include Legacy - Cisco AnyConnect Secure Mobility Client Logs

☒ Include System Logs

Run

Due to security reasons, some of the logs may be encrypted and not visible, but in the unified_log.log of the DART bundle you may see similar logs as shown:



Note: This log example is for the macOS service condition configured in this document.

```
[Tue Feb 27 10:30:58.576 2024][csc_ise posture]Function: createCheckXMLString Thread Id: 0x4A9FD7C0 File:
<check>
  <id>macOS-Service-Condition</id>
  <category>3</category>
  <type>303</type>
  <param>com.apple.sysmond</param>
  <operation>running</operation>
  <value>0</value>
</check>
[Tue Feb 27 10:30:58.576 2024][csc_ise agent]Function: processPostureData Thread Id: 0x4A9FD7C0 File: Au
<cleanmachines>
  <version2>ISE: 3.3.0.430</version2>
  <version>ISE: 2.x</version>
  <encryption>0</encryption>
  <package>
    <id>30</id>
    <name>macOS-Service-Requirement</name>
    <description>macOS Service is non compliant</description>
    <version/>
```

```

        <type>3</type>
        <optional>0</optional>
        <action>3</action>
        <check>
            <id>macOS-Service-Condition</id>
            <category>3</category>
            <type>303</type>
            <param>com.apple.sysmond</param>
            <operation>running</operation>
            <value>0</value>
        </check>
        <criteria>(macOS-Service-Condition)</criteria>
    </package>
</cleanmachines>
[Tue Feb 27 10:30:58.576 2024][csc_iseagent]Function: SMP_initCheck Thread Id: 0x4A9FD7C0 File: SMNavPo
<cleanmachines>
    <version2>ISE: 3.3.0.430</version2>
    <version>ISE: 2.x</version>
    <encryption>0</encryption>
    <package>
        <id>30</id>
        <name>macOS-Service-Requirement</name>
        <description>macOS Service is non compliant</description>
        <version/>
        <type>3</type>
        <optional>0</optional>
        <action>3</action>
        <check>
            <id>macOS-Service-Condition</id>
            <category>3</category>
            <type>303</type>
            <param>com.apple.sysmond</param>
            <operation>running</operation>
            <value>0</value>
        </check>
        <criteria>(macOS-Service-Condition)</criteria>
    </package>
</cleanmachines>
",isElevationAllowed:1,nRemediationTimeLeft:0}
[Tue Feb 27 10:30:58.646 2024][csc_eliseposture]Function: createCheckXMLString Thread Id: 0x4A9FD7C0 Fi
<check>
    <id>macOS-Service-Condition</id>
    <category>3</category>
    <type>303</type>
    <param>com.apple.sysmond</param>
    <operation>running</operation>
    <value>0</value>
</check>
)
[Tue Feb 27 10:30:58.646 2024][csc_eliseposture]Function: doCheck Thread Id: 0x4A9FD7C0 File: Rqmt.cpp
[Tue Feb 27 10:30:58.658 2024][csc_eliseposture]Function: doCheck Thread Id: 0x4A9FD7C0 File: CheckSvc.
[Tue Feb 27 10:30:58.658 2024][csc_eliseposture]Function: completeCheck Thread Id: 0x4A9FD7C0 File: Rqmt

```

As well, you can set the **posture** component at debug log level in the ISE PSN node that authenticates and postures the endpoint.

You can configure this log level from **ISE > Operations > Troubleshoot > Debug Wizard > Debug Log Configuration**. Click the **PSN Hostname** and change **Posture component log level** from INFO to DEBUG.

Using the same example for the macOS service condition, you can see similar logs within the ise-psc.log:

```
2024-02-27 10:30:58.658 DEBUG [https-jsse-nio-10.4.21.59-8443-exec-1][[]] cisco.cpm.posture.runtime.Pos
  <version2>ISE: 3.3.0.430</version2>
  <version>ISE: 2.x</version>
  <encryption>0</encryption>
  <package>
    <id>30</id>
    <name>macOS-Service-Requirement</name>
    <description>macOS Service is non compliant</description>
    <version/>
    <type>3</type>
    <optional>0</optional>
    <action>3</action>
    <check>
      <id>macOS-Service-Condition</id>
      <category>3</category>
      <type>303</type>
      <param>com.apple.sysmond</param>
      <operation>running</operation>
      <value>0</value>
    </check>
    <criteria>(macOS-Service-Condition)</criteria>
  </package>
</cleanmachines>
```

```
2024-02-27 10:30:58.659 DEBUG [https-jsse-nio-10.4.21.59-8443-exec-1][[]] cisco.cpm.posture.util.Status
<cleanmachines>
  <version2>ISE: 3.3.0.430</version2>
  <version>ISE: 2.x</version>
  <encryption>0</encryption>
  <package>
    <id>30</id>
    <name>macOS-Service-Requirement</name>
    <description>macOS Service is non compliant</description>
    <version/>
    <type>3</type>
    <optional>0</optional>
    <action>3</action>
    <check>
      <id>macOS-Service-Condition</id>
      <category>3</category>
      <type>303</type>
      <param>com.apple.sysmond</param>
      <operation>running</operation>
      <value>0</value>
    </check>
    <criteria>(macOS-Service-Condition)</criteria>
  </package>
</cleanmachines>
]
```

```
2024-02-27 10:31:06.044 DEBUG [https-jsse-nio-10.4.21.59-8443-exec-8][[]] cisco.cpm.posture.util.AgentU
```

If issues still persist, raise a TAC ticket with Cisco team.