# Troubleshoot ISE Session Management and Posture

## Contents

## Introduction

This document describes the common Identity Service Engine (ISE) posture services problem: **"AnyConnect ISE posture module shows compliant..."**

## Background Information

This document describes the common Identity Service Engine (ISE) posture services problem - **AnyConnect ISE posture module shows compliant while session status on ISE is pending.**

While symptoms are always the same, there are multiple root causes of this issue.

Often, troubleshooting of such an issue becomes extremely time-consuming which causes serious impact.

This document explains:

- Problem manifestation from end-user and ISE admin perspective.
- Common problematic scenarios.
- The theory behind ISE, AnyConnect, and network operations which trigger the problem.
- Algorithms of quick problem identification.
- Classical solutions to common problematic scenarios.
- Posture status sharing over the Radius session directory.

For a better explanation of the concepts described later, refer to:

ISE Posture Style Comparison for Pre and Post 2.2

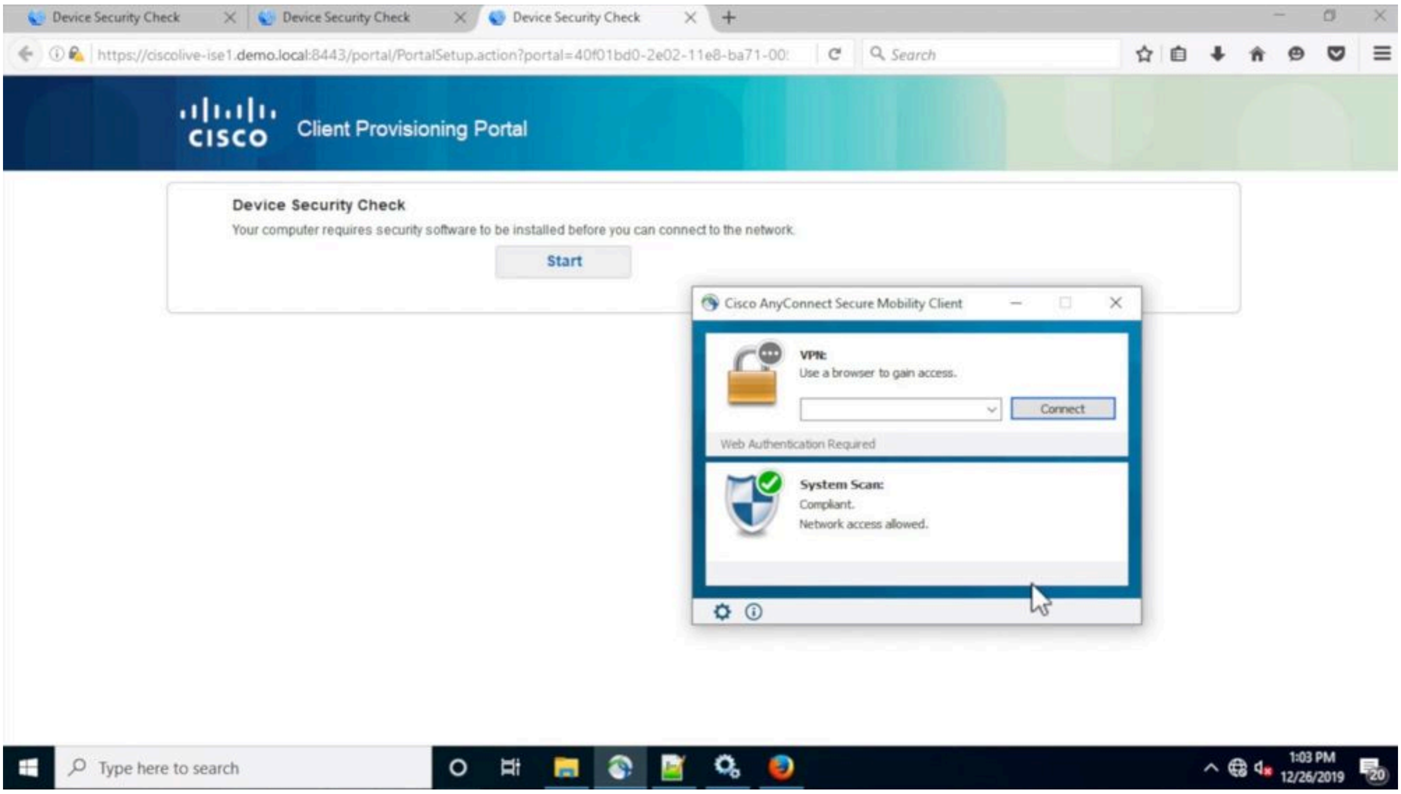
ISE under the magnifying glass. How to troubleshoot ISE - BRKSEC-3229

## Problem

## End-user Experience

This issue normally manifests in the absence of network access or constant redirection to the ISE client provision portal in the browser while, at the same time, AnyConect ISE posture module shows posture status as **Compliant.**

Typical end-user experience:


Device Security Check
Device Security Check
Device Security Check
https://ciscolive-ise1.demo.local:8443/portal/PortalSetup.action?portal=40f01bd0-2e02-11e8-ba71-00
Search
cisco
Client Provisioning Portal
Start
Cisco AnyConnect Secure Mobility Client
VPN:
Connect
System Scan:
Type here to search
1:03 PM
12/26/2019

Device Security Check

Your computer requires security software to be installed before you can connect to the network.

VPN: Use a browser to gain access.

Web Authentication Required

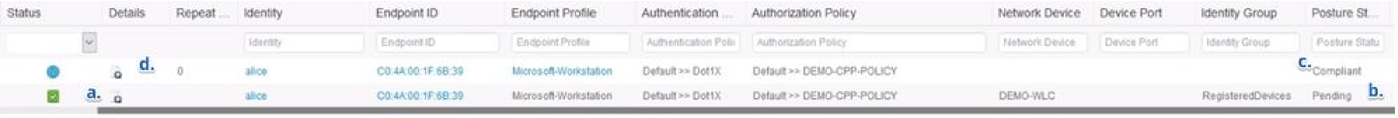
System Scan: Compliant. Network access allowed.

## ISE Admin Experience

Normally, in initial triage of this issue, ISE admin performs Radius Live logs investigation to ensure that there is an actual authentication that hits the ISE.

The first symptom discovered in this stage indicates a mismatch in a posture status between endpoint and ISE as in the live logs or Radius authentication reports last successful authentication for the endpoint shows **Pending** posture status.

Typical ISE admin experience:



| Status | Details | Repeat ... | Identity | Endpoint ID | Endpoint Profile | Authentication ... | Authorization Policy | Network Device | Device Port | Identity Group | Posture St... |
|---|---|---|---|---|---|---|---|---|---|---|---|
|  |  |  | Identity | Endpoint ID | Endpoint Profile | Authentication Poli | Authorization Policy | Network Device | Device Port | Identity Group | Posture Statu |
| ● | d. | 0 | alice | C0:4A:00:1F:6B:39 | Microsoft-Workstation | Default >> Dot1X | Default >> DEMO-CPP-POLICY |  |  |  | c. Compliant |
| ✓ a. |  |  | alice | C0:4A:00:1F:6B:39 | Microsoft-Workstation | Default >> Dot1X | Default >> DEMO-CPP-POLICY | DEMO-WLC |  | RegisteredDevices | Pending b. |

- Last successful authentication for Identity: alice.
- The posture status of the session is **Pending.**
- Last session event for Identity: alice.
- The session event shows posture status as **Compliant.**

**Note**: c. and d. are not always presented in the live logs when the described issue manifests. A session event with a posture status of **Compliant** is more common for scenarios caused by the stale or phantom sessions (described later in this document).

## Common Problematic Scenarios

This issue normally manifests in two problematic scenarios and each of them have multiple root causes. The scenarios:

- AnyConnect ISE posture module has been misinformed by the Policy Service Node (PSN) during the posture process which caused the wrong posture status to be displayed.

In this case, we normally deal with a stale or phantom session in the PSN session cache.

- AnyConnect ISE shows posture status from the previous discovery cycle because the current authentication did not trigger a discovery process.

ISE posture module in AnyConnect has a limited number of events that trigger the discovery process. It is possible that during authentication or re-authentication, none of those events were detected.

## Stale/Phantom Session Problem

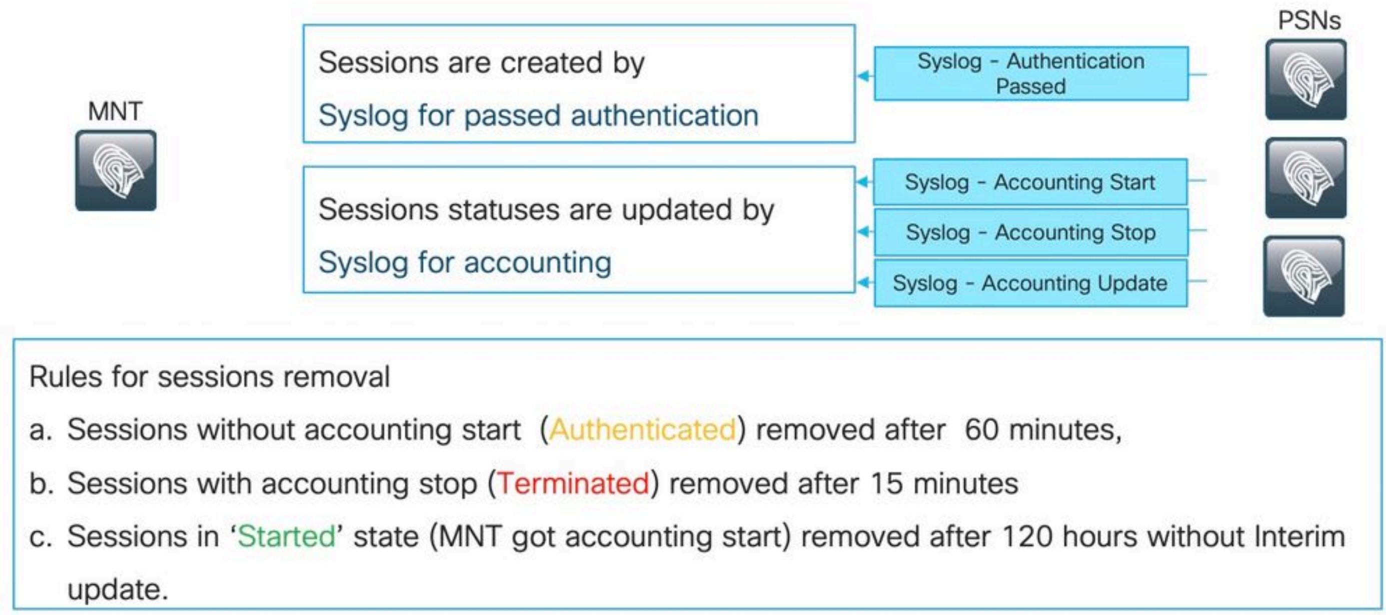
To better understand the issue, investigate the required ISE session management logic and AnyConnect discovery process.

### ISE Session Management Logic

In ISE deployment, there are two personas responsible for the session management process: PSN and Monitoring Node (MNT).

To properly troubleshoot and identify this problem, it is critical to understand the theory of session management on both personas.

### MNT and Session Management



Sessions are created by Syslog for passed authentication

Sessions statuses are updated by Syslog for accounting

Rules for sessions removal

a. Sessions without accounting start (Authenticated) removed after 60 minutes,

b. Sessions with accounting stop (Terminated) removed after 15 minutes

c. Sessions in 'Started' state (MNT got accounting start) removed after 120 hours without Interim update.

MNT
PSNs
Syslog - Authentication Passed
Syslog - Accounting Start
Syslog - Accounting Stop
Syslog - Accounting Update

As explained in this image, MNT node creates seasons based on the passed authentication Syslog messages which come from PSNs.

Session status can be updated later by the Syslog for accounting.

Session removal on MNT happens in 3 scenarios:

1. Sessions without **accounting start** removed approximately 60 minutes after they have been created: There is a cron job executed every 5 minutes to check session statuses and clean.

2. Terminated session removed approximately 15 minutes after the accounting stop has been processed by the same cron job.

3. The same cron on each execution removes as well sessions that have been in the 'Started' state for more than 5 days (120 hours). A started state means that the MNT node processed both authentication and accounting to start Syslog for the session.

Examples of Syslog messages from PSN:

Those messages are logged into prrt-server.log when the runtime-aaa component is enabled into DEBUG. Parts in bold can be used to construct search regular expressions.

Passed authentication :

<#root>

**AcsLogs**

,

**2020-04-07 10:07:29,202**

,DEBUG,0x7fa0ada91700,cntx=0000629480,sesn=skuchere-ise26-1/375283310/10872,CPMSessionID=0A3E946C0000007

**5200 NOTICE Passed-Authentication: Authentication succeeded**

, ConfigVersionId=87, Device IP Address=10.62.148.108, DestinationIPAddress=192.168.43.26, DestinationPo

**bob@example.com**

, NAS-IP-Address=10.62.148.108, NAS-Port=50105, Service-Type=Framed, Framed-IP-Address=192.168.255.205,

**0A3E946C00000073559C0123**

\;42SessionID=skuchere-ise26-1/375283310/10872\;, Calling-Station-ID=

**00-50-56-B6-0B-C6**

, NAS-Port-Type=Ethernet, NAS-Port-Id=GigabitEthernet1/0/5, EAP-Key-Name=, cisco-av-pair=service-type=F

Accounting Start :

<#root>

**AcsLogs**

,

**2020-04-07 10:07:30,202**

```
,DEBUG,0x7fa0ad68d700,cntx=0000561096,sesn=skuchere-ise26-1/375283310/10211,CPMSessionID=0A3E946C000000
3000 NOTICE Radius-Accounting: RADIUS Accounting start request
, ConfigVersionId=87, Device IP Address=10.62.148.108, UserName=
bob@example.com
, RequestLatency=7, NetworkDeviceName=3850-1-BB, User-Name=bob@example.com, NAS-IP-Address=10.62.148.10
0A3E946C00000073559C0123
:skuchere-ise26-1/375283310/10210, Called-Station-ID=00-E1-6D-D1-4F-05, Calling-Station-ID=
00-50-56-B6-0B-C6
, Acct-Status-Type=Start, Acct-Delay-Time=0, Acct-Session-Id=00000041, Acct-Authentic=Remote, Event-Tim
```

Interim Accounting Update :

<#root>

```
AcsLogs,2020-04-07 22:57:48,642,
DEBUG,0x7fa0adb92700,cntx=0000629843,sesn=skuchere-ise26-1/375283310/10877,CPMSessionID=0A3E946C0000007
3002 NOTICE Radius-Accounting: RADIUS Accounting watchdog update
, ConfigVersionId=87, Device IP Address=10.62.148.108, UserName=
bob@example.com
, RequestLatency=8, NetworkDeviceName=3850-1-BB, User-Name=bob@example.com, NAS-IP-Address=10.62.148.10
00-50-56-B6-0B-C6
, Acct-Status-Type=Interim-Update, Acct-Delay-Time=0, Acct-Input-Octets=2293926, Acct-Output-Octets=0,
0A3E946C00000073559C0123
, cisco-av-pair=method=dot1x, AcsSessionID=skuchere-ise26-1/375283310/10877, SelectedAccessService=Defa
```

Accounting Stop :

<#root>

```
AcsLogs,2020-04-08 11:43:22,356
,DEBUG,0x7fa0ad68d700,cntx=0000696242,sesn=skuchere-ise26-1/375283310/11515,CPMSessionID=0A3E946C000000
3001 NOTICE Radius-Accounting: RADIUS Accounting stop request
, ConfigVersionId=88, Device IP Address=10.62.148.108, UserName=
bob@example.com
, RequestLatency=12, NetworkDeviceName=3850-1-BB, User-Name=bob@example.com, NAS-IP-Address=10.62.148.1
00-50-56-B6-0B-C6
, Acct-Status-Type=Stop, Acct-Delay-Time=0, Acct-Input-Octets=4147916, Acct-Output-Octets=0, Acct-Sessi
0A3E946C00000073559C0123
```

```
, cisco-av-pair=method=dot1x, AcsSessionID=skuchere-ise26-1/375283310/11515, SelectedAccessService=Defa
```

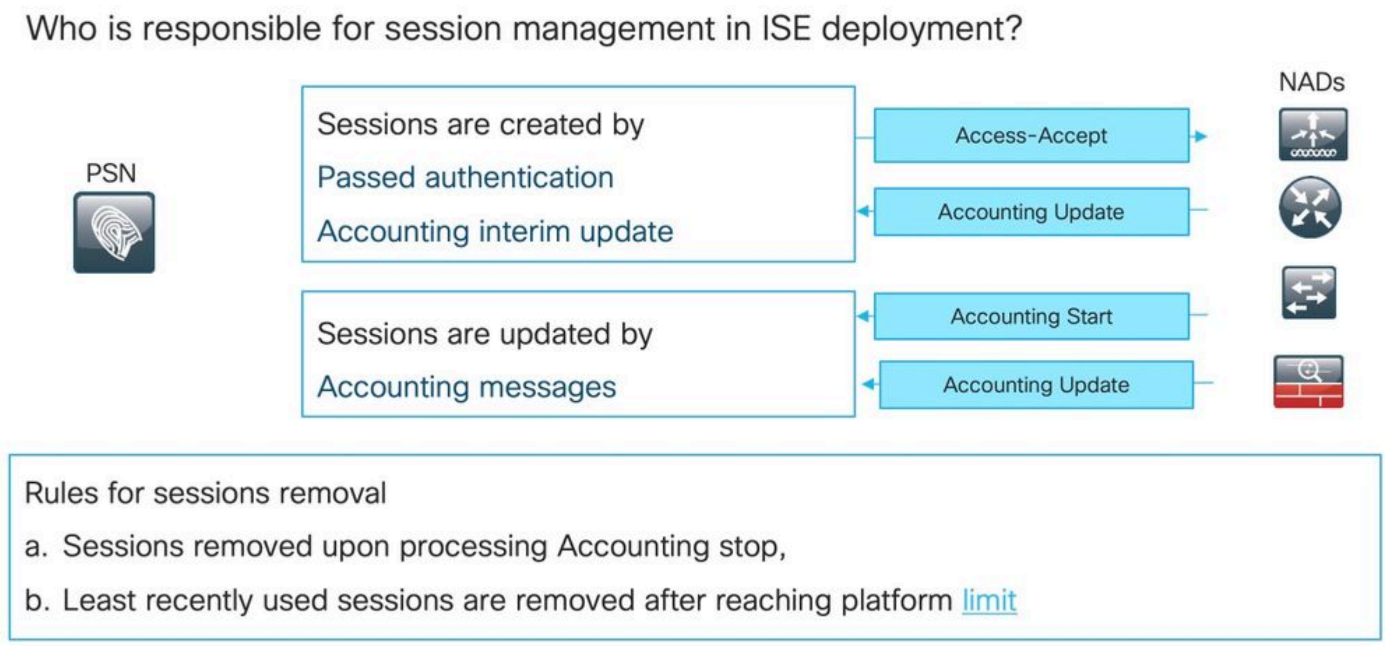
### PSN and Session Management

The PSN session cache is an in-memory database that stores all active sessions of specific PSN. Session cache is always local to the node.

There is no mechanism in ISE that can perform replication of FULL session state from one node to another.

For every active session ID, PSN stores all attributes that were collected during the authentication/authorization phase (for example Internal/External user groups, Network Access Device (NAD) attributes, certificate attributes, and so on). Those attributes are used by PSN to select different policy types (like Authentication, Authorization, Client Provisioning, and Posture).

Session cache is removed completely when the node (or services on the node) are restarted.



Who is responsible for session management in ISE deployment?

Sessions are created by

Passed authentication

Accounting interim update

Sessions are updated by

Accounting messages

Rules for sessions removal

a. Sessions removed upon processing Accounting stop,

b. Least recently used sessions are removed after reaching platform limit

Current session processing logic creates a new entry in the session cache in two scenarios. Later details of existing sessions can be updated from accounting messages which come from NADs.

- The session has been successfully authenticated on the PSN.
- PSN got an accounting interim update for the session which does not exist in the session cache.

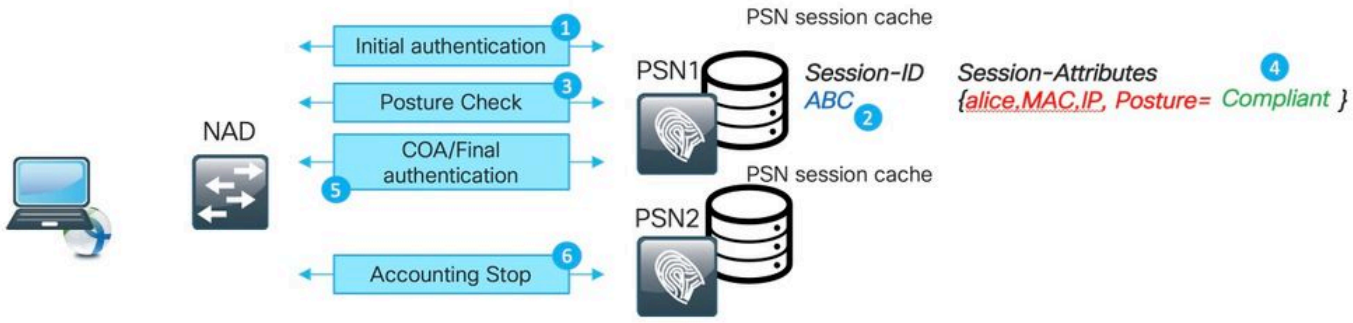
When it comes to session removal, PSN implements this logic:

- Session cache entry is removed immediately after processing the accounting stop message.
- PSN starts to remove the least recently used sessions when a node reaches the**limit** of active sessions.

### Stale Session on PSN

In ISE deployment, the accounting stop for an existing session has been processed by the PSN which did not perform the actual authentication:

Example of the stale session:



1. Successful authentication happens on PSN for session ABC.

2. PSN creates an entry in the session cache.

3. Posture assessment happens.

4. Session is marked as **Compliant**.

5. Change of Authorization (COA) (triggered by posture status change) leads to re-authentication of the endpoint to apply the next access level.

6. Accounting stop for session ABC comes to PSN2.

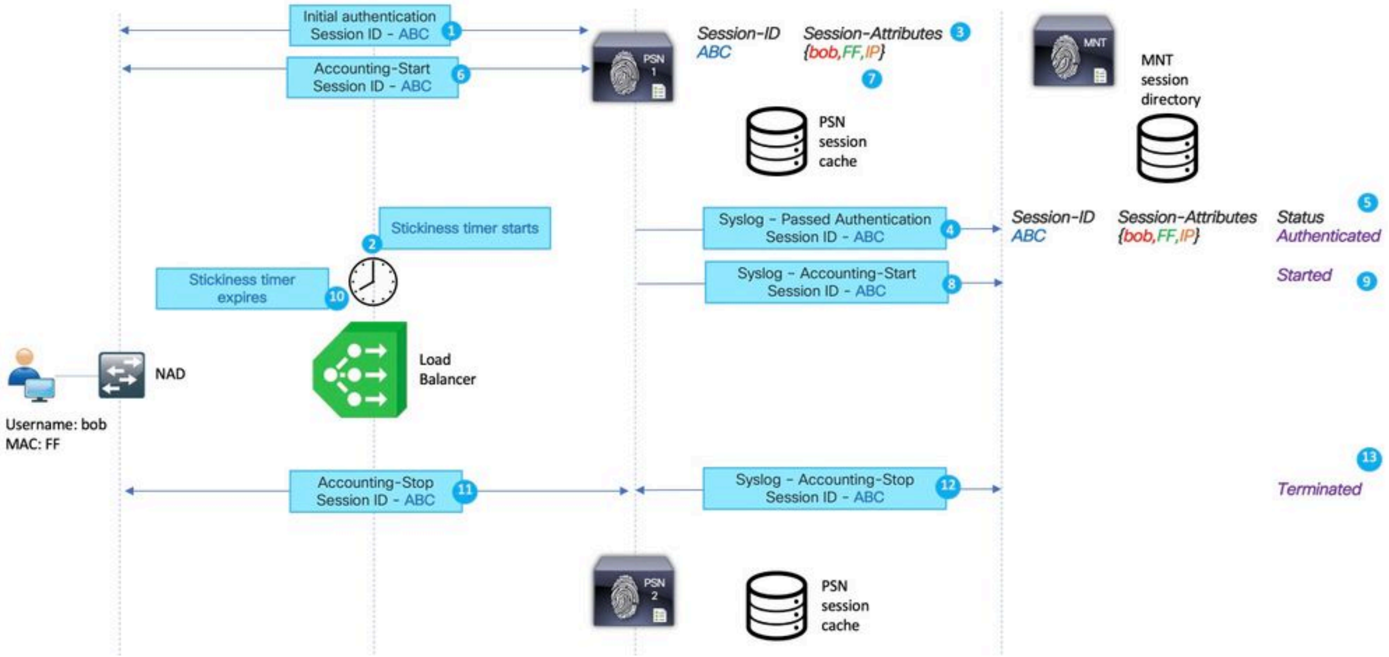
After that, ABC gets stuck in the stale state on the PSN1 because ther is no accounting stop message processed on this PSN to remove it.

The session is removed for a long time if deployment does not experience a high number of authentication attempts.

The stale session appears in PSN session cache in these scenarios:

- The accounting stop came to the wrong PSN due to stickiness timer expiration on the load balancer.
- The wrong configuration on the NAD is not the same PSN configured for authentication and accounting.
- Temporary connectivity issues on the network path that causes NAD failover to the next PSN.

Example of the stale session in Load Balancer (LB)  environment :



1. Initial authentication for the Session ABC is performed by PSN 1.

2. This authentication initiates a stickiness timer on the load balancer.

3. PSN 1 creates an entry for the session ABC in the local cache.

4. Syslog message for passed authentication transferred to MNT node.

5. Entry for session ABC is created into the MNT session directory with the state **Authenticated**.

6. Accounting start message for session ABC lands on PSN 1.

7. Session cache entry for session ABC is updated with information from Accounting-Start.

8. Syslog message for Accounting-Start is transferred to MNT node.

9. Session state is updated to **Started**.

10. Stickiness timer expires on the load balancer.

11. Accounting-Stop for session ABC is forwarded by the load balancer to PSN 2.

12. Syslog message for Accounting-Stop is forwarded by PSN 2 to MNT.

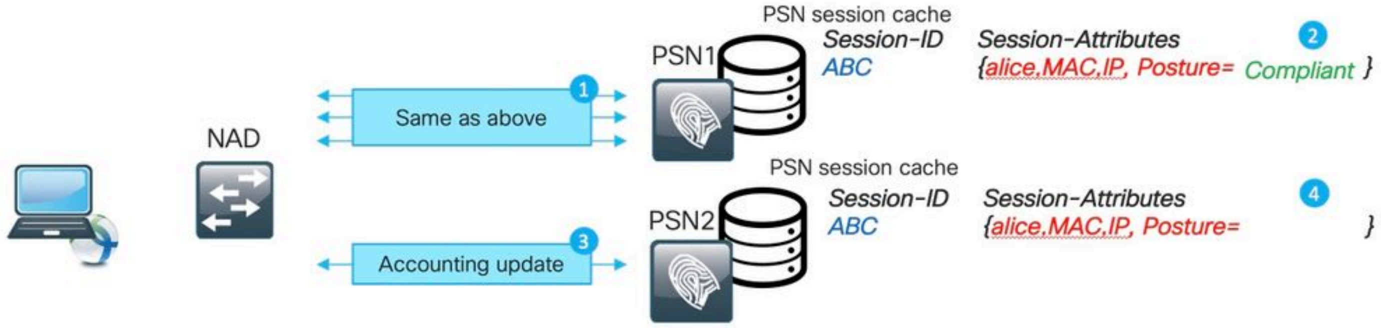
13. Session ABC is marked as **terminated** on MNT.

### Phantom Session on the PSN

The phantom session is a scenario when accounting interim update comes to the PSN which did not perform authentication for this specific session. In this scenario, a new entry is created in the PSN session cache.

If PSN does not get an accounting stop message for this session, the entry is not removed unless PSN reaches the limit of active sessions.

Example of the phantom session:



1. The same steps as described in the stale session example happens on PSN1 for the session ABC.

2. Session ABC has a status **Compliant** in the PSN1 session cache.

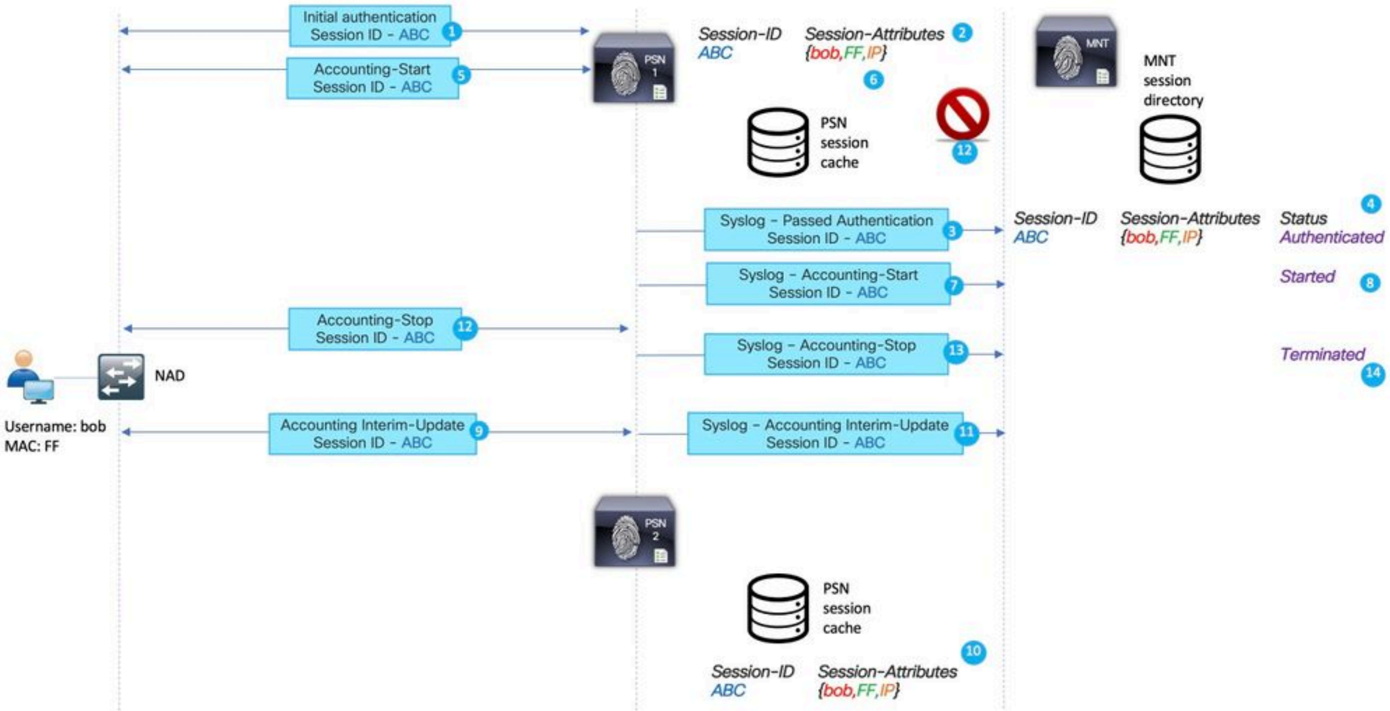
3. An accounting interim update for session ABC hits PSN2.

4. A session entry for session ABC is created on PSN2. Because the session entry is created from the accounting message, it has limited numbers of attributes. For example, posture status is not available for session ABC. Things like user groups and other authorization specific attributes are absent as well.

The phantom session appears in PSN session cache in these scenarios:

- Short-term outage on the network transit.
- Misbehavior of Network Access Device.
- Misbehavior or wrong configuration on Load Balancer.

Here is an example of a phantom session for the scenario with temporary issues on the network path towards PSN1:



1. Initial authentication for the Session ABC is performed by the PSN.

2. PSN1 creates an entry for the session ABC in the local cache.

3. The syslog message for passed authentication is transferred to MNT node.

4. An entry for session ABC is created into TimesTen DB with the state **Authenticated**.

5. The accounting start message for session ABC lands on PSN 1.

6. A session cache entry for session ABC is updated with information from Accounting-Start.

7. The Syslog message for Accounting-Start is transferred to MNT node.

8. The session state is updated to **Started**.

9. The Interim-Accounting update for session ABC is forwarded to PSN2.

10. PSN2 creates an entry for the session ABC in the local cache.

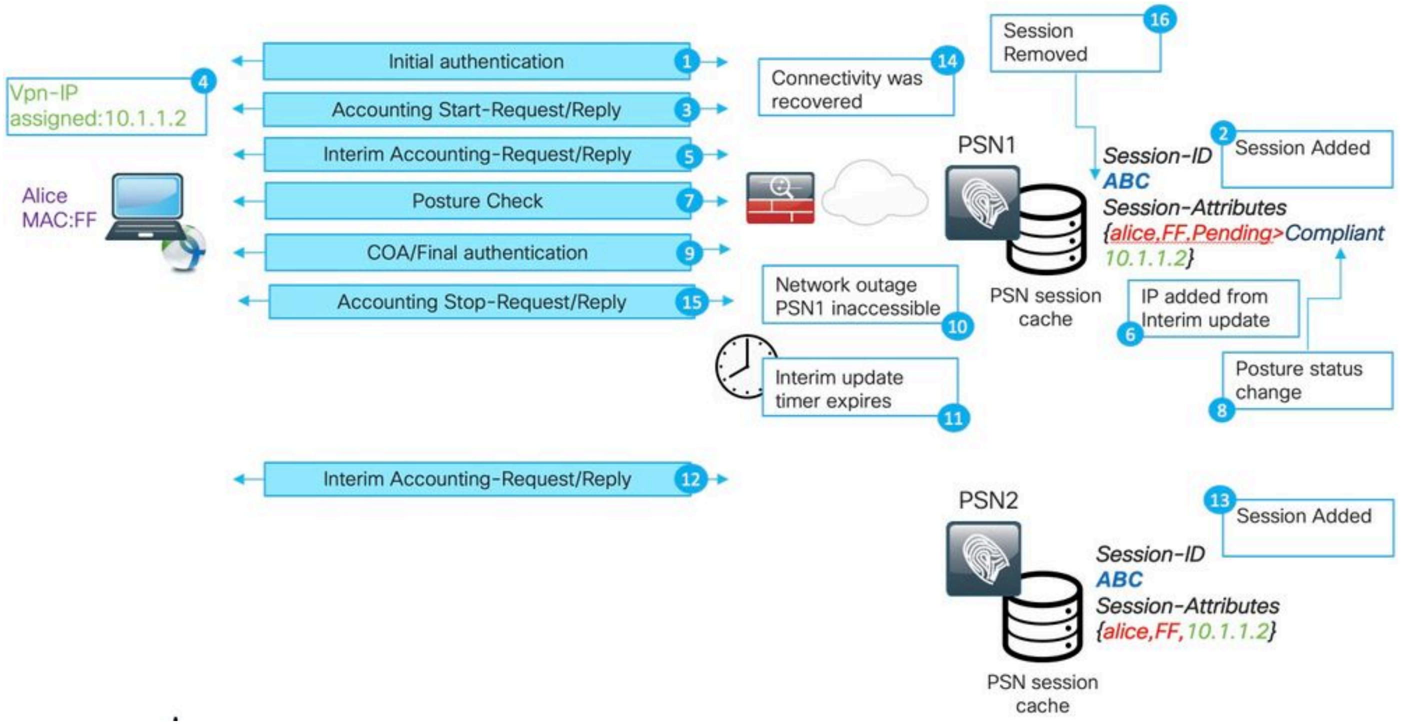
11. The Accounting-Stop for session ABC is forwarded to PSN1.

12. The entry for session ABC is removed from the session cache on PSN1.

13. A syslog message for Accounting-Stop is forwarded by PSN 1 to MNT.

14. The Session ABC is marked as terminated on MNT.

This depicts a scenario of the phantom session as created for the long-living VPN connection:



1. Initial authentication on PSN1.

2. Session ABC is created in the session cache.

3. Accounting starts the message processed by the PSN.

4. The new IP address is assigned to the Virtual Private Network (VPN) adapter.

5. An interim accounting update with IP address information lands on PSN.

6. IP address information is added to the session cache.

7. Posture assessment happens with PSN1.

8. Posture status is updated in the session.

9. A COA push is executed by ISE; this triggers a new access level to be assigned.

10. There is an outage on the network path which makes PSN1 inaccessible.

11. After an interim update interval expiration, ASA/FTD detects that PSN1 is inaccessible.

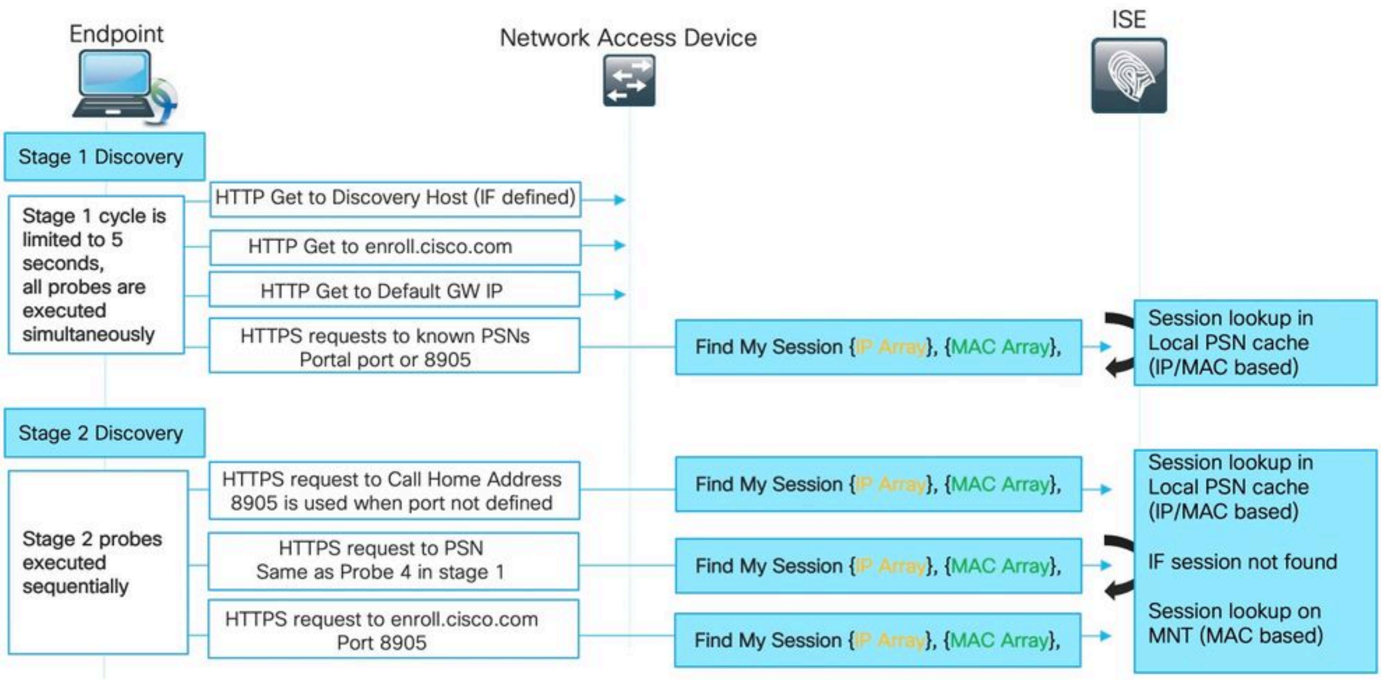
12. The interim accounting update comes to PSN2.

13. The phantom session is created in the PSN2 session cache.

If PSN1 becomes accessible later (14), all subsequent accounting messages are forwarded (15,16) there and this leaves session ABC in the PSN2 session cache for an undefined time.

### How Stale Session and Phantom Session Break the Posture Process

To understand how the stale session and the phantom session break the posture, you can review the AnyConnect ISE posture module discovery process:



Endpoint
Network Access Device
ISE
Stage 1 Discovery
Stage 1 cycle is limited to 5 seconds, all probes are executed simultaneously
HTTP Get to Discovery Host (IF defined)
HTTP Get to enroll.cisco.com
HTTP Get to Default GW IP
HTTPS requests to known PSNs Portal port or 8905
Find My Session {IP Array}, {MAC Array},
Session lookup in Local PSN cache (IP/MAC based)
Stage 2 Discovery
Stage 2 probes executed sequentially
HTTPS request to Call Home Address 8905 is used when port not defined
HTTPS request to PSN Same as Probe 4 in stage 1
HTTPS request to enroll.cisco.com Port 8905
Find My Session {IP Array}, {MAC Array},
Find My Session {IP Array}, {MAC Array},
Find My Session {IP Array}, {MAC Array},
Session lookup in Local PSN cache (IP/MAC based)
IF session not found
Session lookup on MNT (MAC based)

Stage 1 Discovery :

During this stage, the ISE posture module executes 4 simultaneous problems to locate the PSN which did an authentication for the endpoint.

First, 3 probes on the figure are redirect-based (Default GW IP. Discovery host IP (if defined) and enroll.cisco.com IP) ; Those probes always point the agent to the right PSN as redirected URL is taken from the NAD itself.

Probe number 4 is sent to all primary servers presented in the **ConnectionData.xml** file. This file is created after the first successful posture attempt. File content can be updated later in case the client migrates between PSNs.

On Windows systems, the file location is **C:\ProgramData\Cisco\Cisco AnyConnect Secure Mobility Client\ISE Posture\.**

Since all stage 1 probes are executed simultaneously, the result from probe 4 is used only if all other 3 probes failed or if the ISE posture module was unable to establish proper communication with the PSN returned in redirect URL within 5 seconds.

When probe 4 lands on the PSN, it contains a list of active IP and MAC addresses discovered on the endpoint. PSN uses this data to find a session for this endpoint in the local cache.

If PSN has a stale or phantom session for an endpoint, this can result in wrong posture status displayed later on the client-side.

When an agent gets multiple answers for probe 4 (**ConnectionData.xml** can contain more than one primary

PSN), the fastest reply is always used.

Stage 2 Discovery:

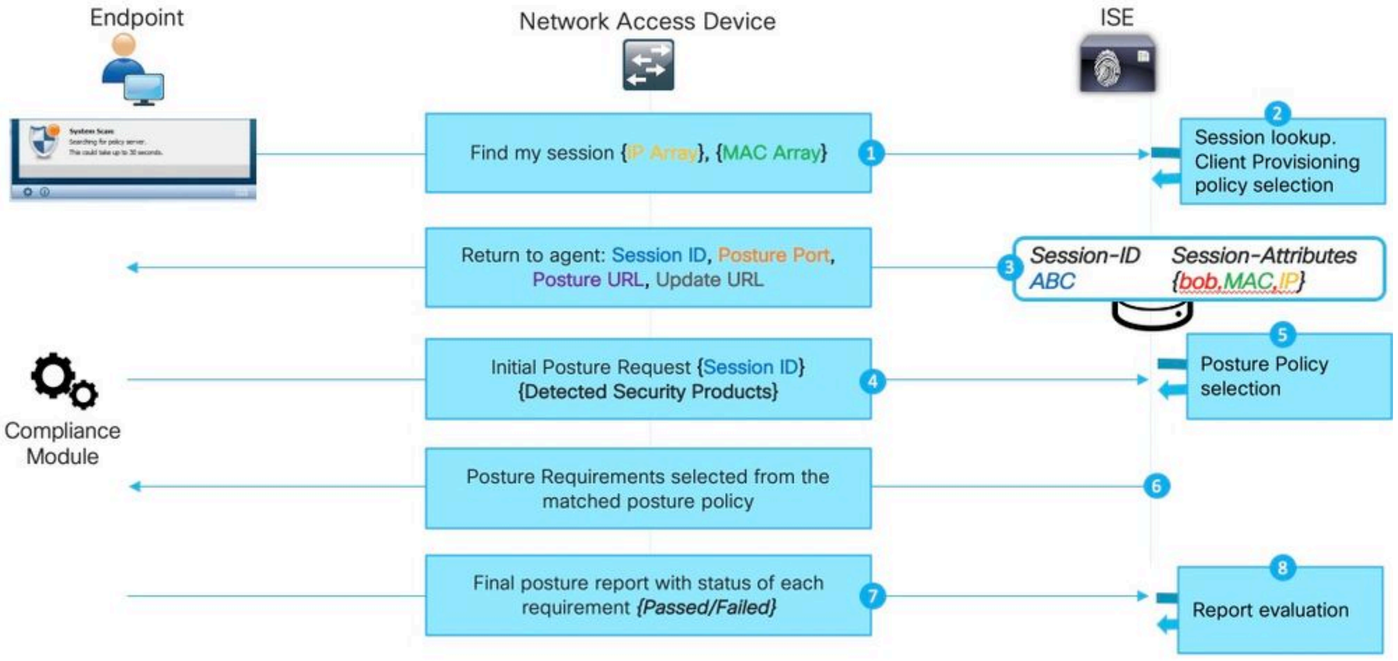
All stage 2 discovery probes are redirect-less, which means that every probe triggers a session lookup on the destination PSN.

If the PSN cannot locate the session in the local session cache, it has to perform a MNT lookup (MAC address-based only) to find a session owner and return owner name to the agent.

As all probes trigger session lookup, stage 2 discovery can be even more affected by issues as a result of from stale or phantom sessions.

If PSN gets to stage 2,  the discovery probe which exists in the session cache creates a stale or phantom entry for the same endpoint. It results in the wrong posture status returned to the end-user.

The example shows how posture happens when PSN holds a stale session or phantom session:



Endpoint
Network Access Device
ISE
System Scan
Find my session {IP Array}, {MAC Array}
1
2
Session lookup.
Client Provisioning policy selection
Return to agent: Session ID, Posture Port, Posture URL, Update URL
3
Session-ID
ABC
Session-Attributes
{bob,MAC,IP}
Compliance Module
Initial Posture Request {Session ID}
{Detected Security Products}
4
5
Posture Policy selection
Posture Requirements selected from the matched posture policy
6
Final posture report with status of each requirement {Passed/Failed}
7
8
Report evaluation

**Note**: This issue can manifest only when all redirect-based discovery probes fail or when non-redirect posture is implemented.

1. Any of **Find my session** probes issued by the ISE posture module.

2. PSN performs session lookup in the session cache. If the session is to be found, a stale or phantom session issue occurs.

3. PSN runs the Client provisioning policy selection. In a case where a phantom session that has a lack of authentication/authorization attributes and all policies configured by the customer are very specific (for example, policies are created for specific Active Directory groups), PSN is not able to assign a right client provisioning policy. This can manifest in the error message: "Bypassing AnyConnect scan your network is configured to use Cisco NAC Agent".

- In cases where client provisioning policies are generic (attributes available in the phantom session are enough to match policy with AnyConnect configuration) PSN replies with details needed for the continuation of the assessment process.

- At this step, when we can deal with stale sessions, PSN replies right away with a posture status of **Compliant** and all next steps are not performed. PSN does not send COA because it believes that the session is already compliant. In Radius Live logs, there is not a session event displayed with **Compliant** status.

4. For the phantom session scenario, the ISE posture module continues with the Initial posture request. This request contains information about all security and patch management products detected on the endpoint.

5. PSN uses information from the request and session attributes to match proper posture policy. Because the phantom session has a lack of attributes at this point, we have no policy to match. In such a case, PSN replies to the endpoint that it is compliant. This is a default ISE behavior in the case of posture policy not matching.

**Note**: When there is some generic policy that can be selected from phantom session attributes, we continue with step 6.

6. PSN returns selected posture policies back to the agent.

**Note**: When no policy can be selected, PSN returns Compliant status.

7. The agent returns statuses for each policy/requirement as either "passed" or "failed".

8. Report evaluation happens on ISE and session status changes to **Compliant.**

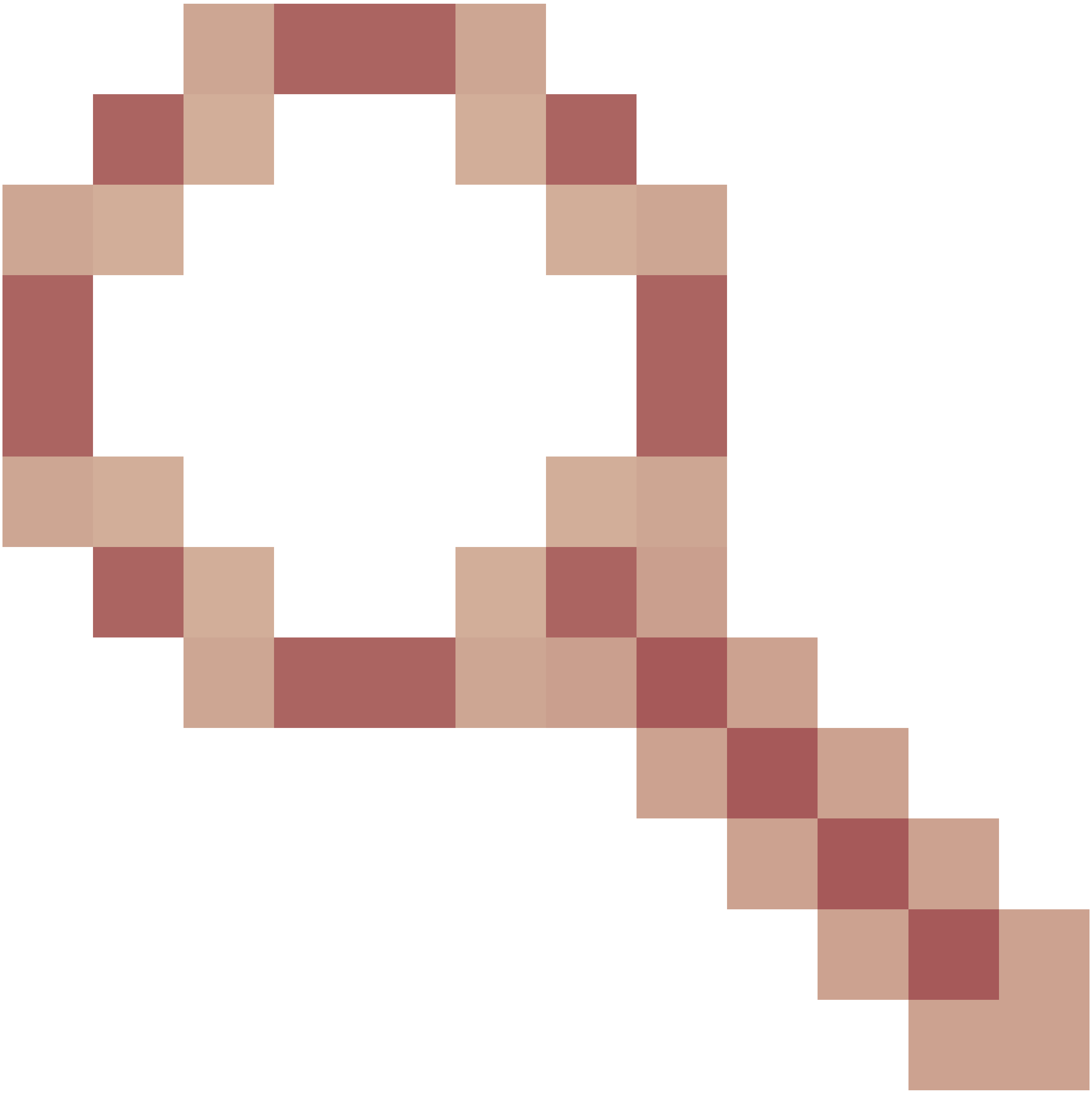
**Note**: In case of posture issues caused by the phantom session, the ISE administrator possibly notices some failed posture COAs. In such cases, COA requests are executed from the wrong PSNs and for wrong session IDs.

## Discovery Process does not Start on a New Authentication Attempt

The ISE posture module is designed to monitor a limited amount of events on the endpoint to trigger a discovery process.

Events which trigger discovery:

- Initial ISE posture module installation.
- User login.
- Power events.
- Interface status change.
- OS resume after sleep.
- Default Gateway (DG) change.
- Posture Reassessment (PRA) failure, see Cisco bug ID CSCvo69557



New dot1x authentication, PC unlock, and IP address change are not detected by the ISE posture module.

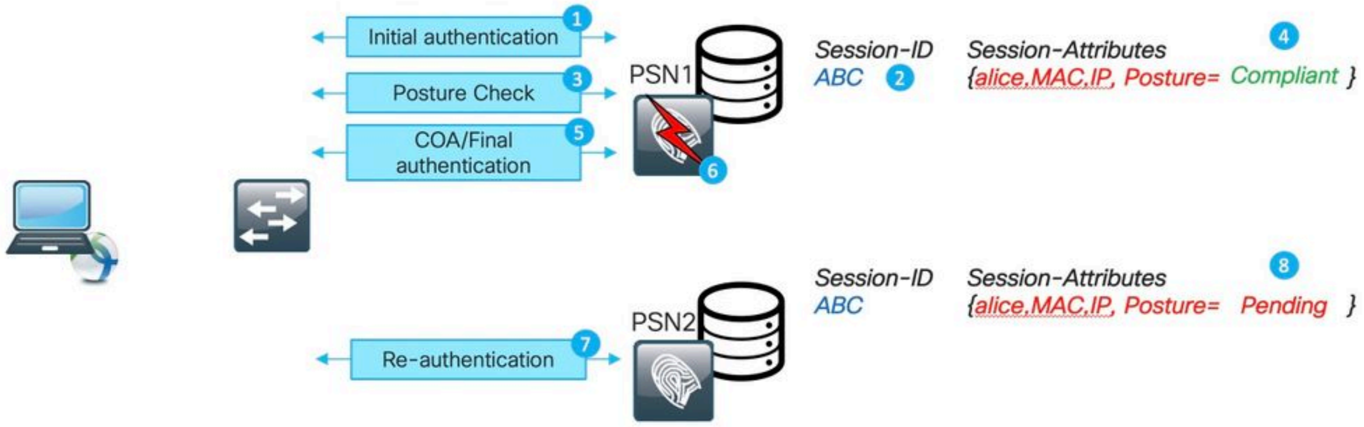
The ISE posture module is unable to detect a new authentication or re-authentication attempt in these scenarios :

- Re-authentication hits different PSN (either due to LB decisions or issues with original PSN).
- NAD generates new session-id on reauthentication.

**Reauthentication on Different PSN**

This diagram depicts an example of re-authentication on different PSN caused by the outage of the original PSN. A scenario with a load balancer looks very similar.

In the case of a load balancer, re-authentication is directed to the different PSN as a result of a stickiness timer expiration.



1. Initial authentication on PSN1

2. Session ABC is created in the PSN1 session cache.

3. Posture assessment is performed with PSN1.

4. Session ABS posture status moves to **Compliant.**

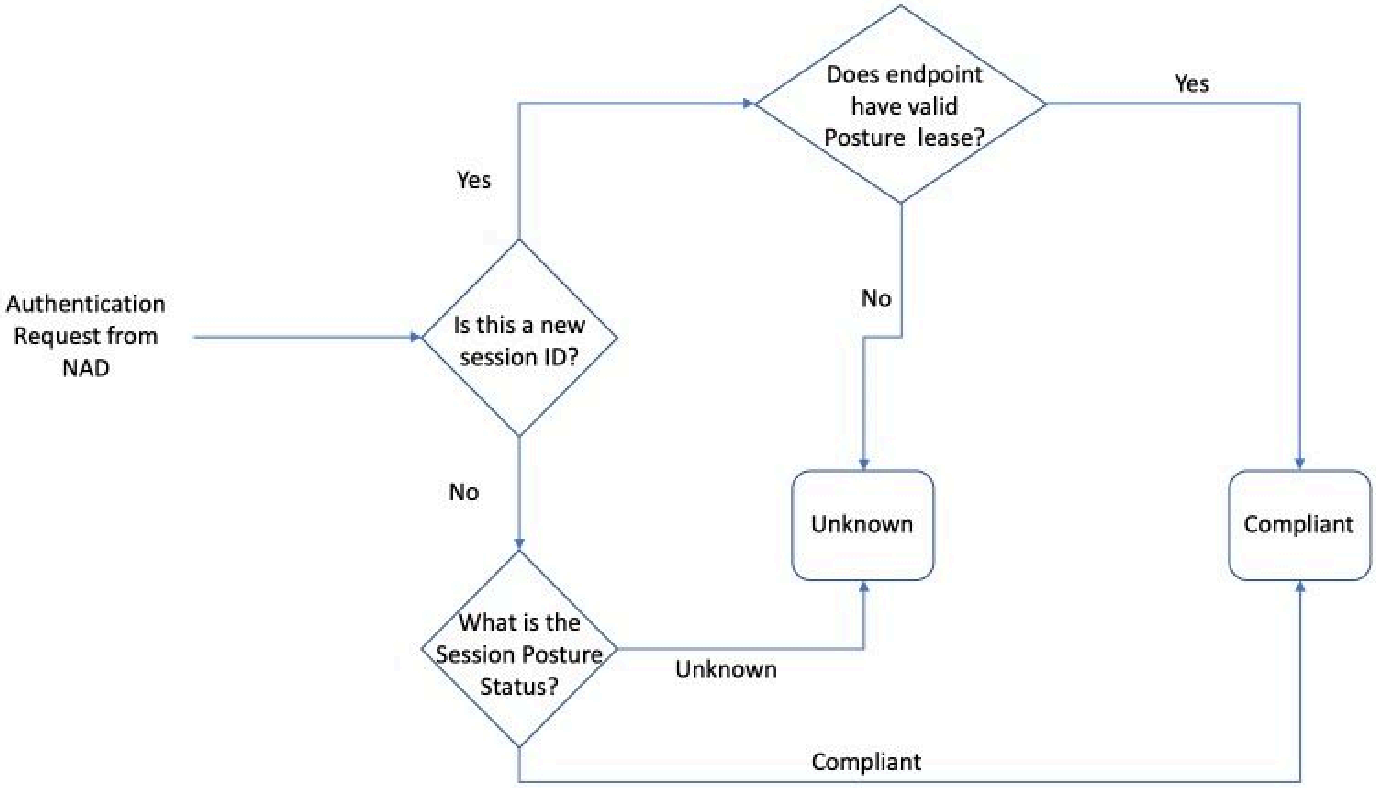
5. COA is triggered by posture status change and leads to re-authentication of the endpoint to apply the next access level.

6. PSN1 becomes unavailable.

7. Re-authentication for session ABC hits PSN2.

8. Because it is a new session for PSN2, posture status of the session becomes **Pending.**

Initial posture status is assigned by PSN to the session:


Authentication Request from NAD
Is this a new session ID?
Yes
No
Does endpoint have valid Posture lease?
Yes
No
Unknown
Compliant
What is the Session Posture Status?
Unknown
Compliant

**Note**: State-machine describes only an initial selection of the posture status. Each session which is initially marked as Unknown can later become Compliant or Non-Compliant based upon report evaluation received from the ISE posture module.

**NAD Generates New Session-ID on Reauthentication**

This could happen in the two most common scenarios:

- Re-authentication is improperly configured on the ISE side. The solution to this problem is covered later in this document.
- Misbehavior from the NAD side; normally NAD keeps the same session ID during the re-authentication attempt. This is a potentially buggy behavior that needs to be investigated on the NAD itself.

The new session ID can be generated in some other corner-case scenarios. For example, in some cases, wireless roaming can be a cause of it.

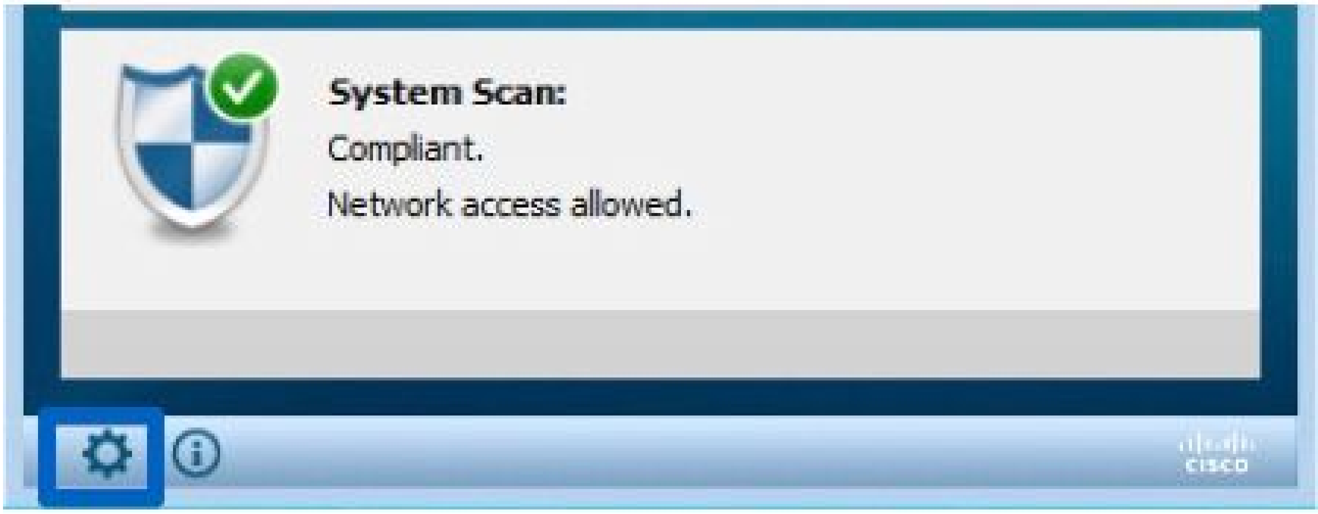
The main thing here is, ISE PSN always places a new session into posture **Pending** state unless the posture lease is configured. The posture lease is explained described later in this document.

## Quick Way to Identify When the Issue Was Caused by the Stale/Phantom Session

To identify whether AnyConnect shows compliance while it is in the redirect state is caused by the stale/phantom session. We need to get access to the endpoint while it is in the problematic state.

1. Investigate System Scan Details:

   1. Click the gear icon in AnyConnect UI

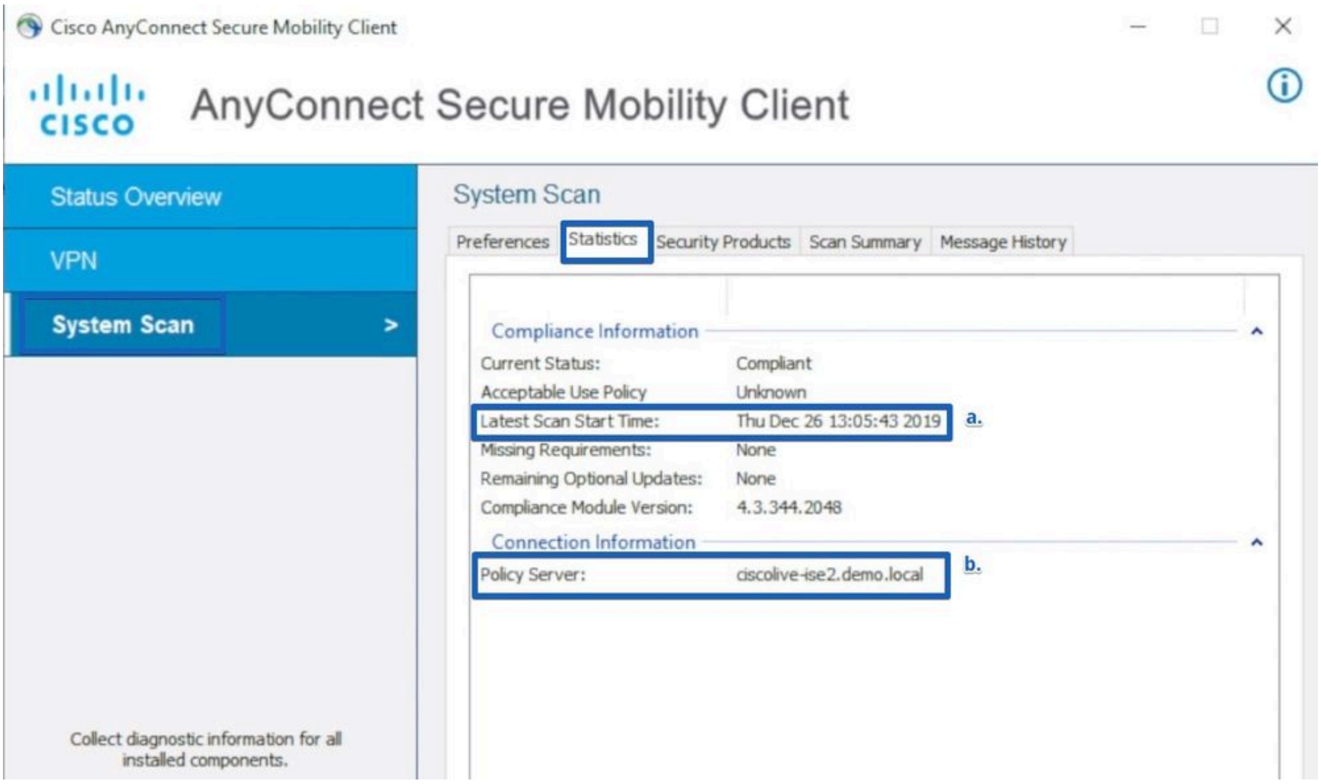


**System Scan:**

Compliant.

Network access allowed.

2. In the new Window navigate to **System Scan** > **Statistics**



Cisco AnyConnect Secure Mobility Client

AnyConnect Secure Mobility Client

Compliance Information

Current Status: Compliant

Acceptable Use Policy Unknown

Latest Scan Start Time: Thu Dec 26 13:05:43 2019

Missing Requirements: None

Remaining Optional Updates: None

Compliance Module Version: 4.3.344.2048

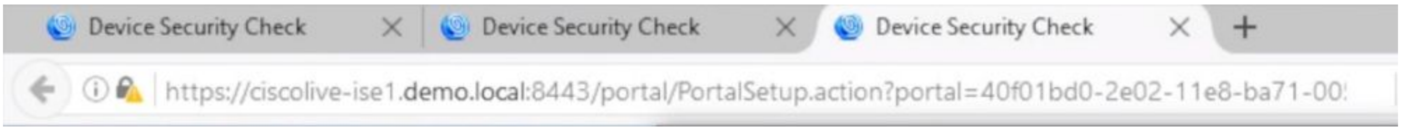
Connection Information

Policy Server: ciscolive-ise2.demo.local

Here, pay attention to two elements:

- Latest Scan Start Time - the timestamp here must be close the time when the problem was discovered.
- Policy Server - this filed indicated the name of the policy server which did a posture assessment for the endpoint. The FQDN from here needs to be compared with FQDN from Redirect-URL (for redirect base posture) or with the PSN name taken from the last authentication attempt (for redirect-less posture).

2. Compare Policy Server FQDN from System Scan Statistics with the node name which did authentication for endpoint:



https://ciscolive-ise1.demo.local:8443/portal/PortalSetup.action?portal=40f01bd0-2e02-11e8-ba71-00

The demo shows the recording of the steps needed for issue identification:

## Advance Troubleshoot of Stale/Phantom Session

The previous example serves to differentiate the issue of a stale or phantom session from the problem of the discovery process which did not start.

At the same time, we need to identify the actual session which triggered the problem to better understand how exactly it becomes a stale or phantom session issue.

While in some scenarios stale and phantom sessions cannot be avoided, we need to ensure that best practices are implemented so that no stale/phantom sessions are created in the environment.

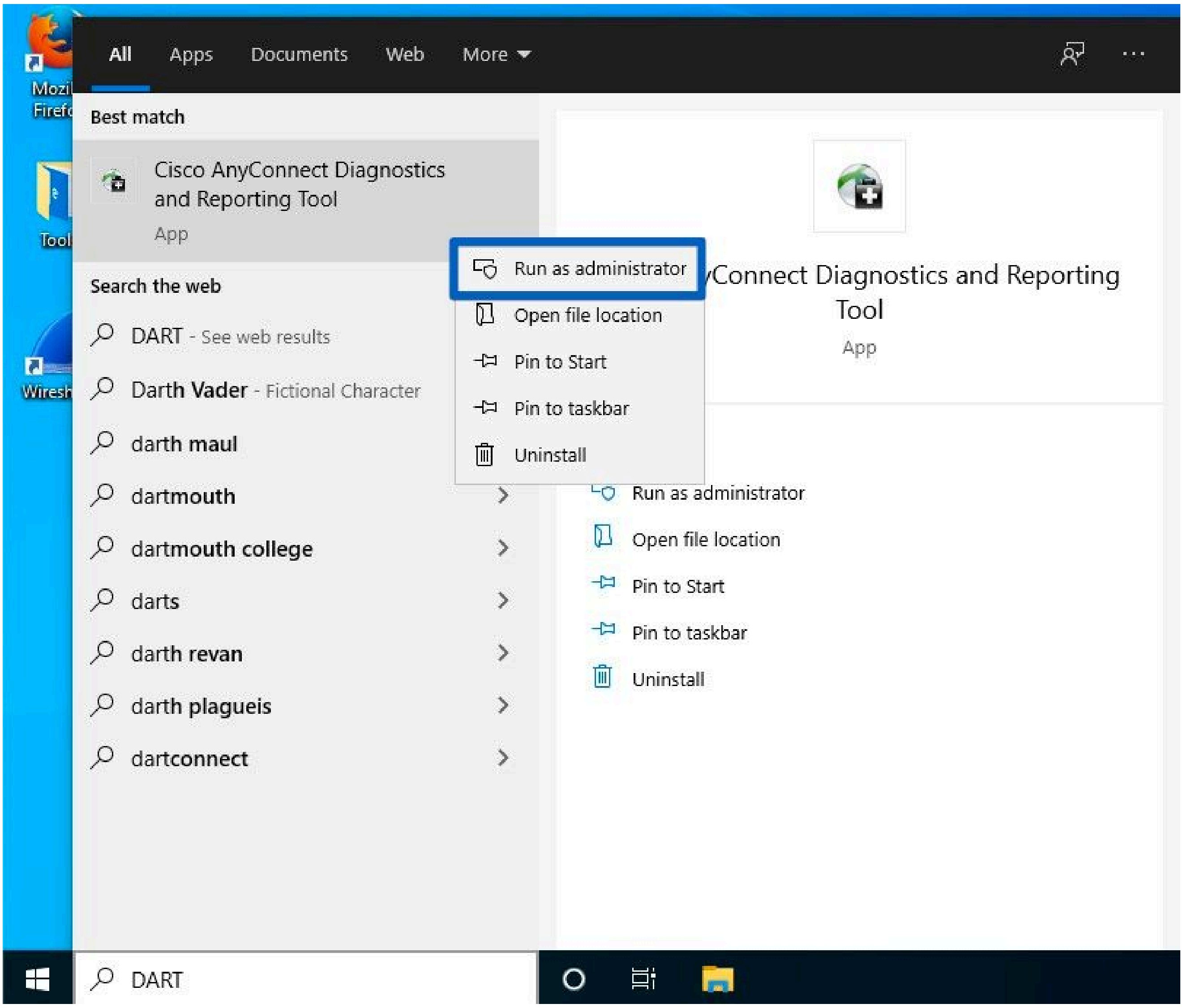
### DART Bundle Collection

Analyze a DART bundle taken from the endpoint that reproduces the problem.

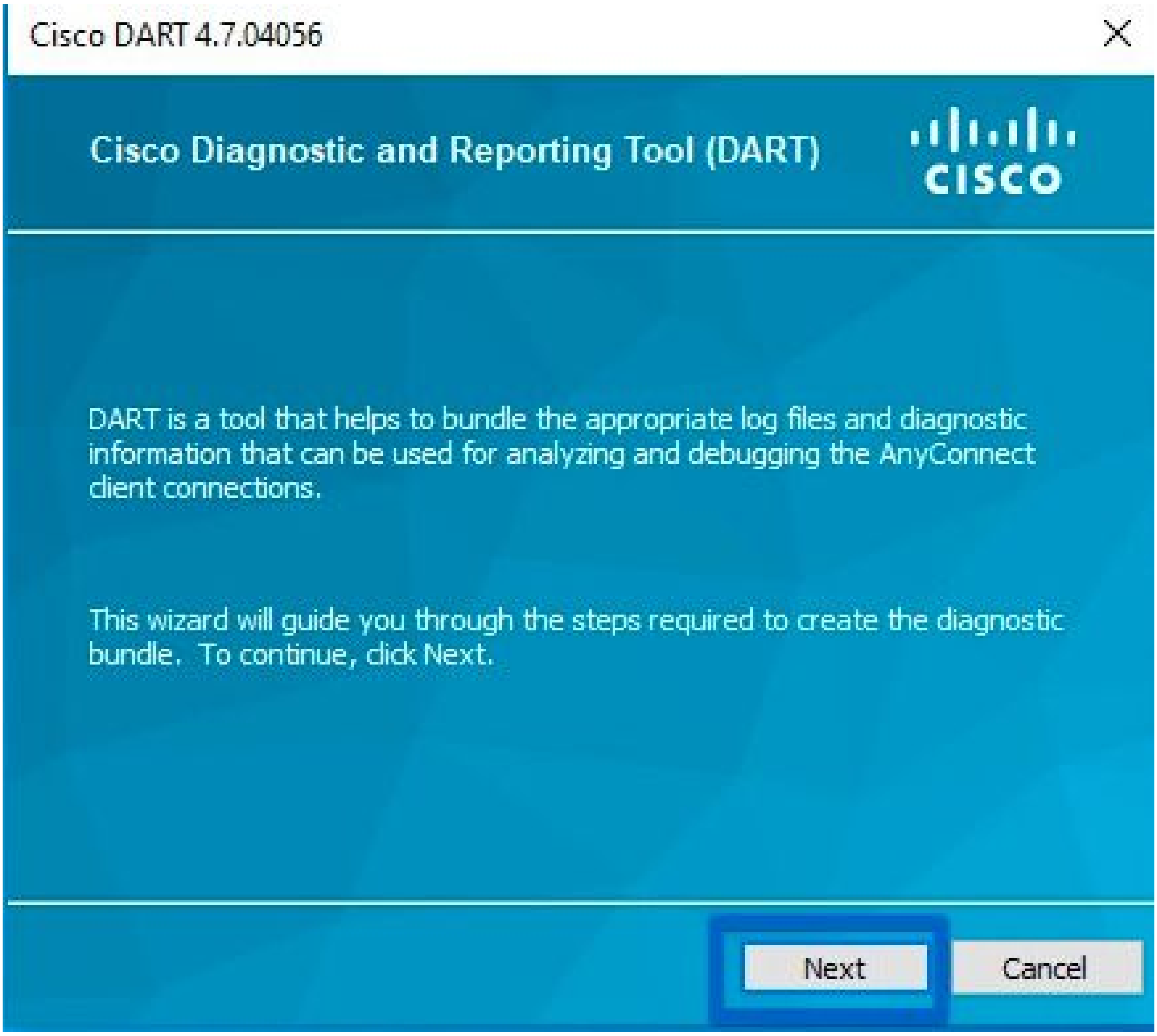
- Keep only important logs in the DART. It is recommended to clear the logs before the problem is reproduced.

To achieve this, the DART bundle utility needs to start as an Administrator and perform log clean up.

1. On Windows Navigate to Start and begin to type DART, right-click, and choose - **Run as administrator**



2. On the first wizard screen, click **Next.**



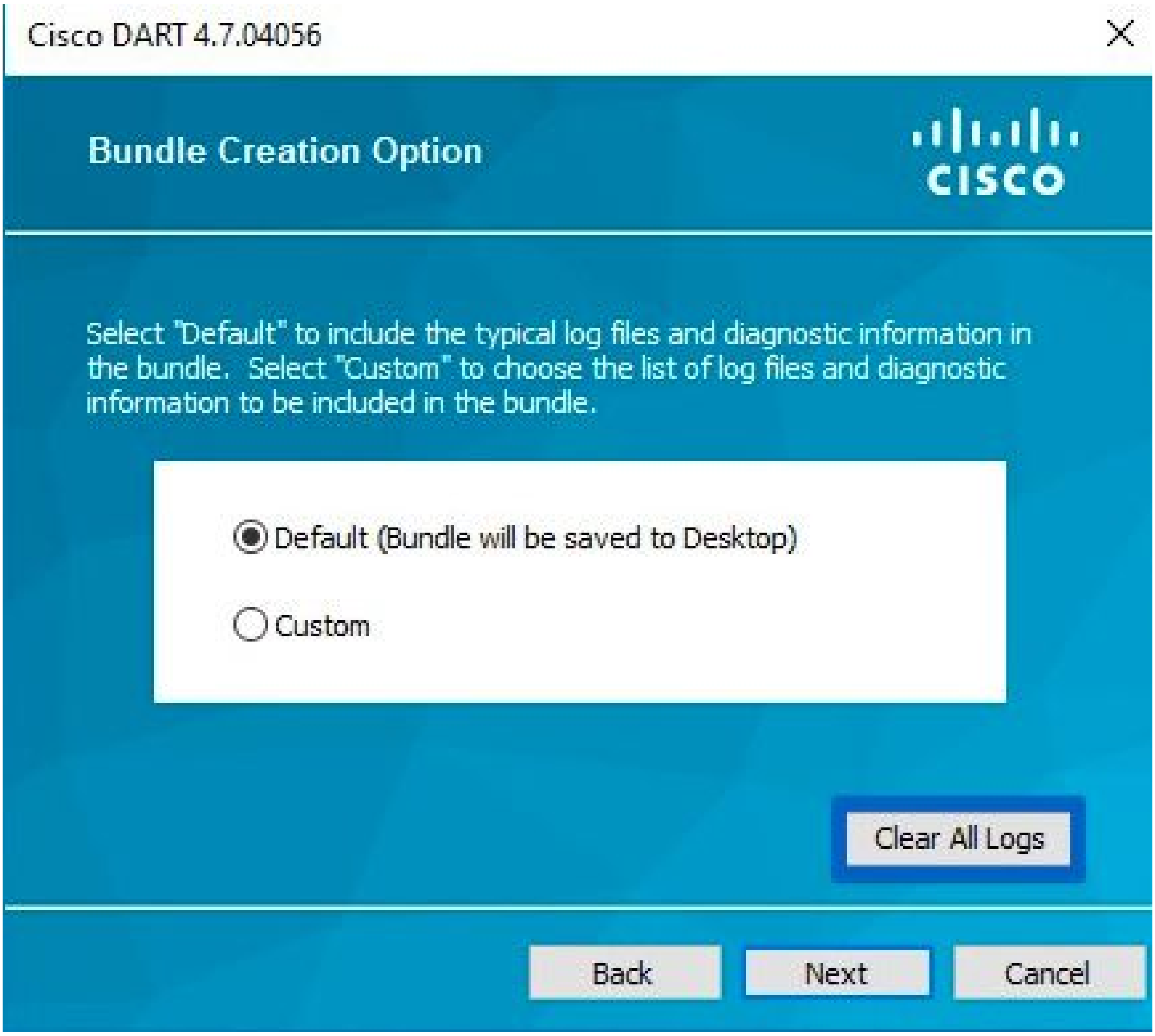
Cisco DART 4.7.04056

Cisco 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 AnyConnect client connections.

This wizard will guide you through the steps required to create the diagnostic bundle. To continue, click Next.

3. On the next wizard screen, click **Clear All Logs.**



Cisco DART 4.7.04056

Bundle Creation Option

Select "Default" to include the typical log files and diagnostic information in the bundle. Select "Custom" to choose the list of log files and diagnostic information to be included in the bundle.

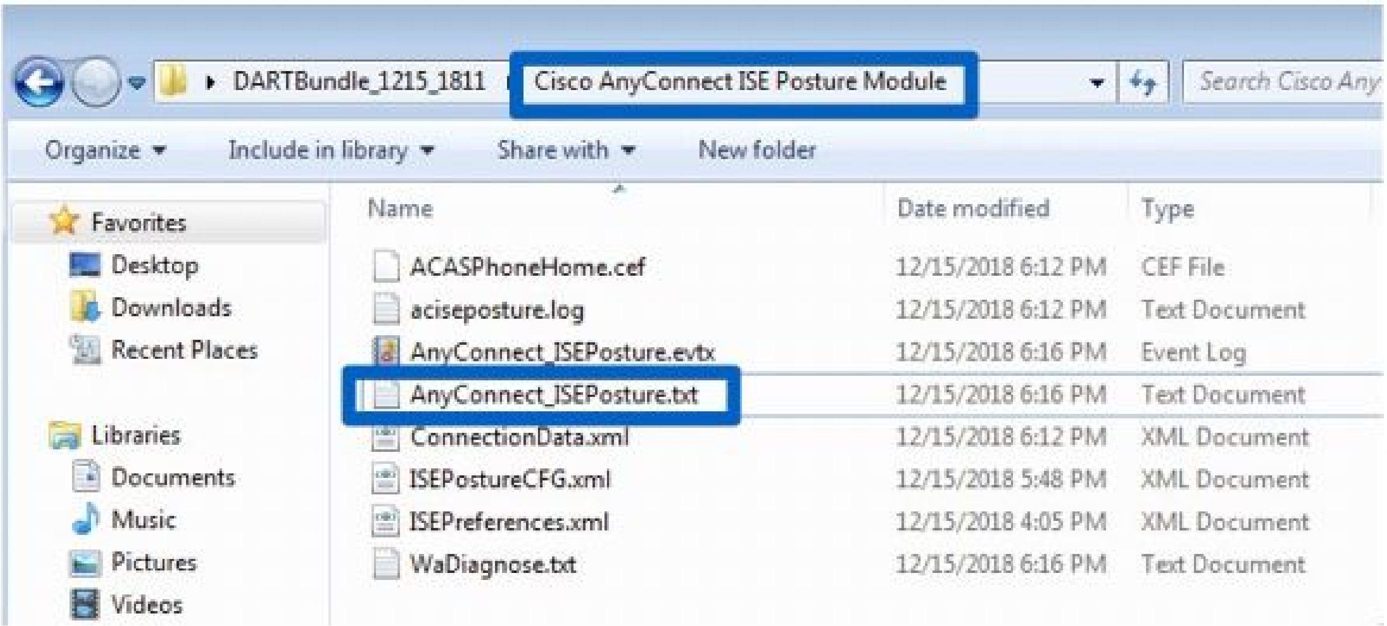
● Default (Bundle will be saved to Desktop)

○ Custom

4. After the problem is reproduced, DART can be collected from here; click **Next**.

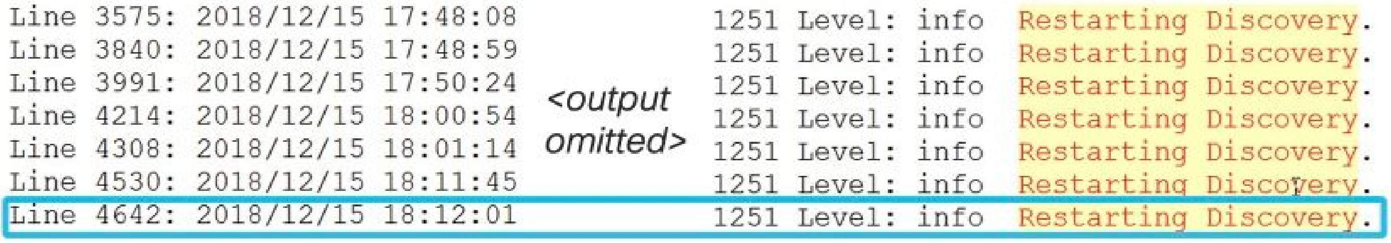
**DART bundle analysis**

After the DART bundle has been collected, un-archive it and focus on the file **AnyConnect_ISEPosture.txt** located in the folder **Cisco AnyConnect ISE Posture Module**. This file contains all discovery-related events.



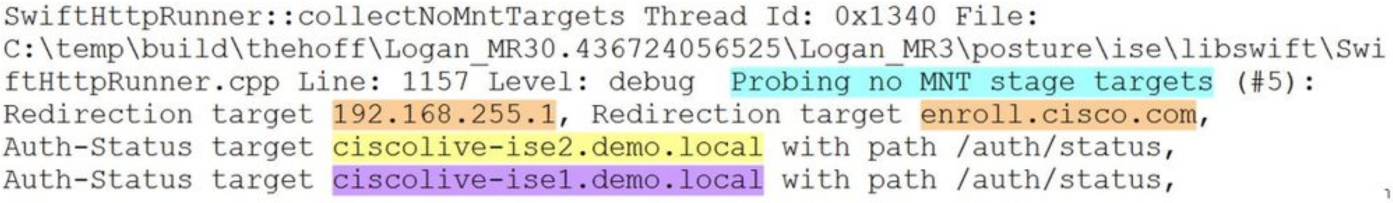
| Name | Date modified | Type |
|---|---|---|
| ACASPhoneHome.cef | 12/15/2018 6:12 PM | CEF File |
| aciseposture.log | 12/15/2018 6:12 PM | Text Document |
| AnyConnect_ISEPosture.evtx | 12/15/2018 6:16 PM | Event Log |
| AnyConnect_ISEPosture.txt | 12/15/2018 6:16 PM | Text Document |
| ConnectionData.xml | 12/15/2018 6:12 PM | XML Document |
| ISEPostureCFG.xml | 12/15/2018 5:48 PM | XML Document |
| ISEPreferences.xml | 12/15/2018 4:05 PM | XML Document |
| WaDiagnose.txt | 12/15/2018 6:16 PM | Text Document |

1. Start troubleshooting and identify all moments of discovery restart. Keywords to search are **Restarting Discovery** or HTTP Discovery. Here, navigate to the line with discovery restart that happened at the problematic moment:



```
Line 3575: 2018/12/15 17:48:08              1251 Level: info  Restarting Discovery.
Line 3840: 2018/12/15 17:48:59              1251 Level: info  Restarting Discovery.
Line 3991: 2018/12/15 17:50:24  <output     1251 Level: info  Restarting Discovery.
Line 4214: 2018/12/15 18:00:54  omitted>    1251 Level: info  Restarting Discovery.
Line 4308: 2018/12/15 18:01:14              1251 Level: info  Restarting Discovery.
Line 4530: 2018/12/15 18:11:45              1251 Level: info  Restarting Discovery.
Line 4642: 2018/12/15 18:12:01              1251 Level: info  Restarting Discovery.
```

2. Several lines after discovery restart, there is a line which contains - **Probing no MNT stage targets**. This is an indicator of Stage 1 discovery start:

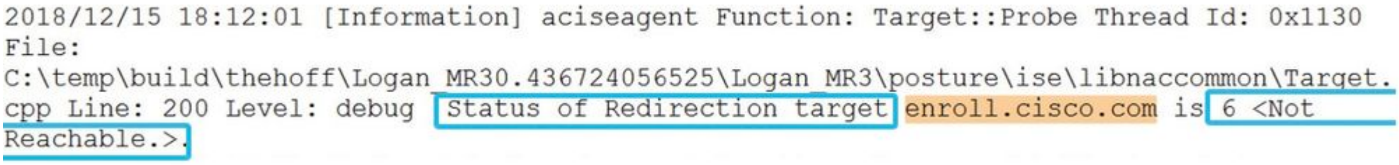


```
SwiftHttpRunner::collectNoMntTargets Thread Id: 0x1340 File:
C:\temp\build\thehoff\Logan_MR30.436724056525\Logan_MR3\posture\ise\libswift\Swi
ftHttpRunner.cpp Line: 1157 Level: debug  Probing no MNT stage targets (#5):
Redirection target 192.168.255.1, Redirection target enroll.cisco.com,
Auth-Status target ciscolive-ise2.demo.local with path /auth/status,
Auth-Status target ciscolive-ise1.demo.local with path /auth/status,
```

It is recommended to highlight all redirect-based probes with the same color and previously connected PSNs taken from **ConnectionData.xml** (Auth-Status targets) in a different color.

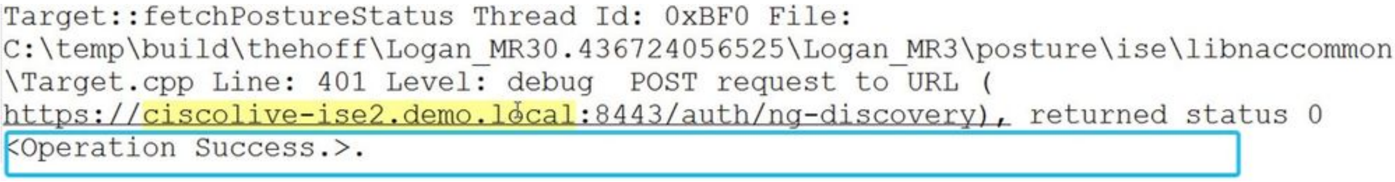
Normally PSN FQDNs are very similar and it is hard to spot the difference.

3.Read the log files to see a result for every single probe. This is an example of how the failed probe looks:



```
2018/12/15 18:12:01 [Information] aciseagent Function: Target::Probe Thread Id: 0x1130
File:
C:\temp\build\thehoff\Logan_MR30.436724056525\Logan_MR3\posture\ise\libnaccommon\Target.
cpp Line: 200 Level: debug  Status of Redirection target enroll.cisco.com is 6 <Not
Reachable.>.
```

4. Somewhere in the file after discovery restart for stage 1 or stage 2, you see a successful reply from one or more PSNs:

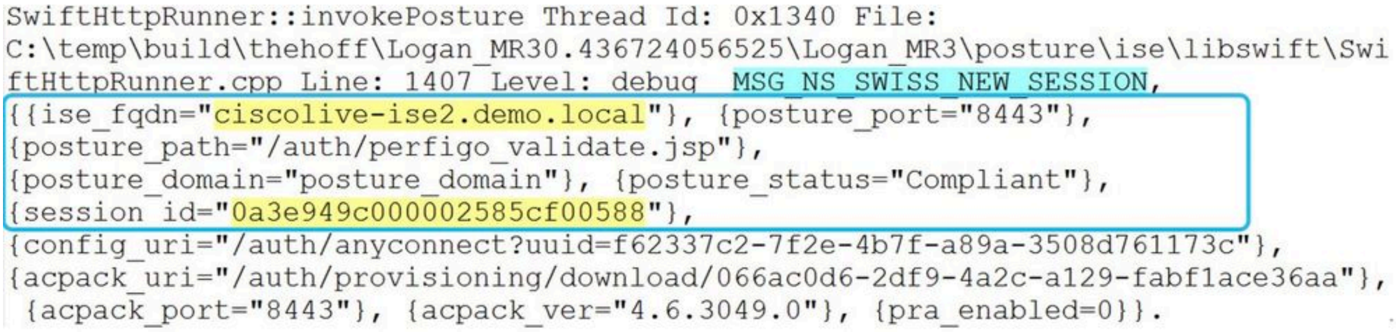


```
Target::fetchPostureStatus Thread Id: 0xBF0 File:
C:\temp\build\thehoff\Logan_MR30.436724056525\Logan_MR3\posture\ise\libnaccommon
\Target.cpp Line: 401 Level: debug  POST request to URL (
https://ciscolive-ise2.demo.local:8443/auth/ng-discovery), returned status 0
<Operation Success.>.
```

5. Several lines later, there is a line with the keyword **MSG_NS_SWISS_NEW_SESSION.**

This line contains an actual session ID that has been selected by PSN as a result of the session lookup.

Use this session ID for further investigation on ISE to determine how this session becomes stale/phantom:



```
SwiftHttpRunner::invokePosture Thread Id: 0x1340 File:
C:\temp\build\thehoff\Logan_MR30.436724056525\Logan_MR3\posture\ise\libswift\Swi
ftHttpRunner.cpp Line: 1407 Level: debug  MSG_NS_SWISS_NEW_SESSION,
{{ise_fqdn="ciscolive-ise2.demo.local"}, {posture_port="8443"},
{posture_path="/auth/perfigo_validate.jsp"},
{posture_domain="posture_domain"}, {posture_status="Compliant"},
{session_id="0a3e949c000002585cf00588"},
{config_uri="/auth/anyconnect?uuid=f62337c2-7f2e-4b7f-a89a-3508d761173c"},
{acpack_uri="/auth/provisioning/download/066ac0d6-2df9-4a2c-a129-fabf1ace36aa"},
 {acpack_port="8443"}, {acpack_ver="4.6.3049.0"}, {pra_enabled=0}}.
```

### Investigation on ISE Logs

In the guest.log with the **clientwebapp** component enabled into DEBUG, the PSN which replies with the Stale/Phantom session can be seen.

PSN gets a request from the ISE posture agent. This is a request from AnyConnect because of the User-Agent value:

<#root>

```
cisco.cpm.client.posture.PostureStatusServlet -::-

Got http request from 192.168.255.228 user agent is: Mozilla/4.0 (compatible; WINDOWS; 1.2.1.6.1.48; Any



cisco.cpm.client.posture.PostureStatusServlet -::-

mac_list

 from http request ==> C0:4A:00:1F:6B:39



cisco.cpm.client.posture.PostureStatusServlet -::-

iplist

 from http request ==> 192.168.255.228



cisco.cpm.client.posture.PostureStatusServlet -::-
```

```
Session id from http request -
req.getParameter
(
sessionId
) ==> null
```

The request contains arrays of IP addresses and MAC addresses. In this particular example, each array holds only one value.

The log shows that the session ID from the request is null which indicates that this a request from the non-redirect-based probe.

Later you can see how values from arrays are used to locate a session ID:

<#root>

```
cpm.client.provisioning.utils.ProvisioningUtil -::- the input ipAddress from the list currently processe

cpm.client.provisioning.utils.ProvisioningUtil -::- the ipAddress that matched the http request remote

cpm.client.provisioning.utils.ProvisioningUtil -::- the clientMac from the macarray list for the for lo

cisco.cpm.client.posture.PostureStatusServlet -::- Found Client IP matching the remote IP 192.168.255.2

cpm.client.provisioning.utils.ProvisioningUtil -::-
Session = 0a3e949c000000495c216240
```

After the line with keywords **Sent http response**, you can see content from the actual reply:

<#root>

```
cisco.cpm.client.posture.PostureStatusServlet -::- Sent an http response to  192.168.255.228 with X-ISE

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-PDP value is clemea19-ise1.demo.local

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-POSTURE value is /auth/perfigo_validat

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-POSTURE_PORT value is 8443

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-AC_PKG_PORT value is 8443

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-GUESTFLOW value is false

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-AC_CONFIG_URL value is https://clemea1

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-AC_CONFIG_URI value is /auth/anyconnec

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-AC_PKG_URL value is https://clemea19-i

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-AC_PKG_URI value is /auth/provisioning

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-AC_PKG_VER value is 4.6.3049.0
cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-STATUS_PATH value is /auth/status

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-BACKUP_SERVERS value is clemea19-ise2.

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-SessionId value is 0a3e949c000000495c2

cpm.client.provisioning.utils.ProvisioningUtil -::- header X-ISE-PostureDomain value is posture_domain

cpm.client.provisioning.utils.ProvisioningUtil -::-
```

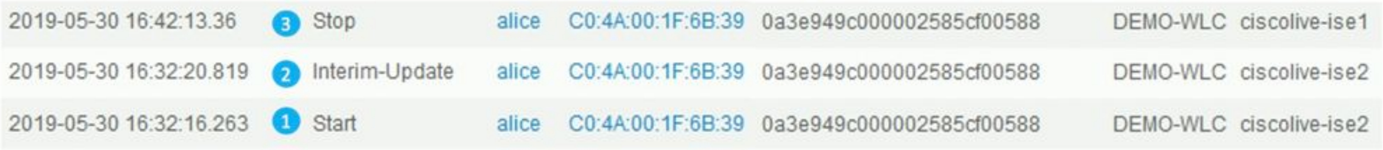
**header X-ISE-POSTURE_STATUS value is Unknown**

### Investigation on ISE Reports

After you know the ID of the stale/phantom session, you can investigate the Radius Accounting report to get a better understanding of what caused this session to become stale/phantom:

- Navigate to **Operations** > **Reports** > **Endpoints** and **Users** > **Radius Accounting report** and run this report for 7 days. Use an endpoint ID as a filter key.

Here is an example of a report which shows how stale session has been leftover on **ciscolive-ise2**:



| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 2019-05-30 16:42:13.36 | 3 | Stop | alice | C0:4A:00:1F:6B:39 | 0a3e949c000002585cf00588 | DEMO-WLC | ciscolive-ise1 |
| 2019-05-30 16:32:20.819 | 2 | Interim-Update | alice | C0:4A:00:1F:6B:39 | 0a3e949c000002585cf00588 | DEMO-WLC | ciscolive-ise2 |
| 2019-05-30 16:32:16.263 | 1 | Start | alice | C0:4A:00:1F:6B:39 | 0a3e949c000002585cf00588 | DEMO-WLC | ciscolive-ise2 |

1. The accounting start for the session came to the PSN **ciscolive-ise2**.
2. The interim update for the session was processed on the same PSN.
3. An accounting stop message for the problematic session ID came to a different PSN (ciscolive-ise1).

## A Quick Way to Identify When The Issue Was Caused by The Absence of Discovery Restart

Here, the same logic is applicable as for the previous issue. The only difference is that you need to focus on the Latest Scan Start Time. For this type of problem, the timestamp of the last scan is somewhere in the past.

Normally when the end-user discovers a problem, a scan which happened some time ago is seen. While in the ISE Radius Live logs, recent authentication attempts from the problematic endpoint are seen.

The demo shows the recording of the steps needed for issue identification:

## Advanced Troubleshoot The Absence of Discovery Restart

The approach here is very similar to **Advanced Troubleshoot Stale/Phantom Session** section. The main troubleshooting element is the DART bundle investigation.

Inside of the DART bundle, you can search for discovery restarts (as shown for the previous issue) and confirm that there was no discovery restart at the moment when the problem was reported.

On the ISE side, focus on the **Radius Live Logs/ Radius** authentication report to confirm that there was either failover between PSNs or new session ID has been generated by NAD.

# Solution

## Classical Approach - Issue Avoidance

Historically there was no feature on ISE which could solve issues described in this document so the only way was to rely on the set of best practices that are implemented on the network and ISE side the minimize risks.

### Best Practices That Can Minimize The Amount of Stale or Phantom Sessions in The ISE Deployment

### Always Implement Redirect Based Posture When Possible

A common counterargument to this recommendation is a bad user experience; pop-ups in the OS or Browsers are seen. This indicates redirection while AnyConnect ISE posture module in the background performs an assessment process.

As a solution to this, it is possible to redirect ONLY ISE Posture module discovery probes and selectively allow all other traffic.

This example shows redirect ACL designed to redirect only HTTP requests to Discovery Host (10.1.1.1 in this example) and enroll.cisco.com (172.16.1.80):

```
ip access-list extended REDIRECT-DH-ENROLL

 permit tcp any host 10.1.1.1 eq www

 permit tcp any host 172.16.1.80

 deny   ip any any
```

To keep an acceptable level of security, such redirect ACL can be combined with DACL assigned from ISE.

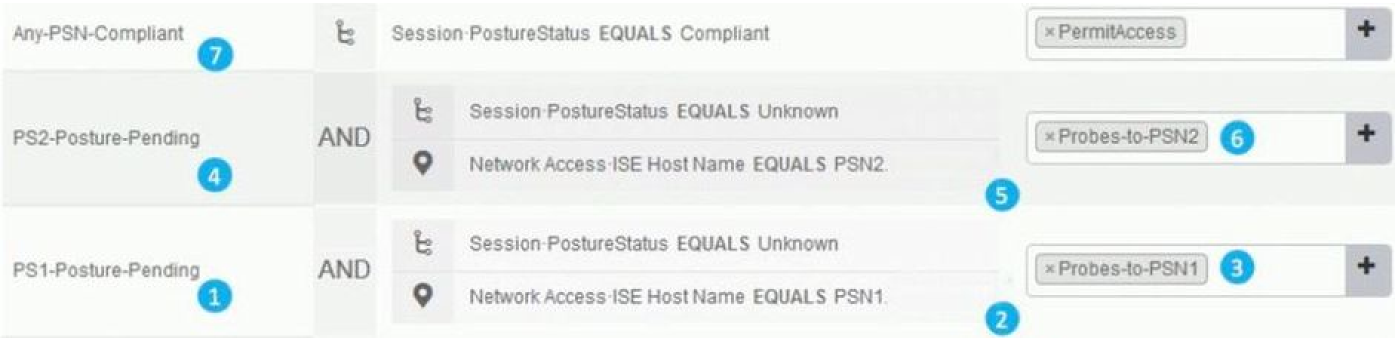
### Pending State Allows Connections Only to PSN Where Endpoint was Authenticated

This approach useful for the environments where url-redirection is not supported (for example implementations with the 3-rd party NADs).

As a solution, implement multiple **PosturePending** authorization policies (one per PSN). Each policy needs to contain as one of the conditions the name of PSN where authentication took place.

In the authorization profile assigned to each policy access to all PSNs must be blocked except the node where authentication happened.

Create authorization policies for 2 nodes deployment:



| | | | | |
|---|---|---|---|---|
| Any-PSN-Compliant 7 | | Session·PostureStatus EQUALS Compliant | ×PermitAccess | + |
| PS2-Posture-Pending 4 | AND | Session·PostureStatus EQUALS Unknown<br>Network Access·ISE Host Name EQUALS PSN2 5 | ×Probes-to-PSN2 6 | + |
| PS1-Posture-Pending 1 | AND | Session·PostureStatus EQUALS Unknown<br>Network Access·ISE Host Name EQUALS PSN1 2 | ×Probes-to-PSN1 3 | + |

1. Posture **Pending** policy for PSN1

2. PSN1 name used as a condition in the policy.

3. Authorization profile with ACL which blocks access to all PSNs except PSN1.

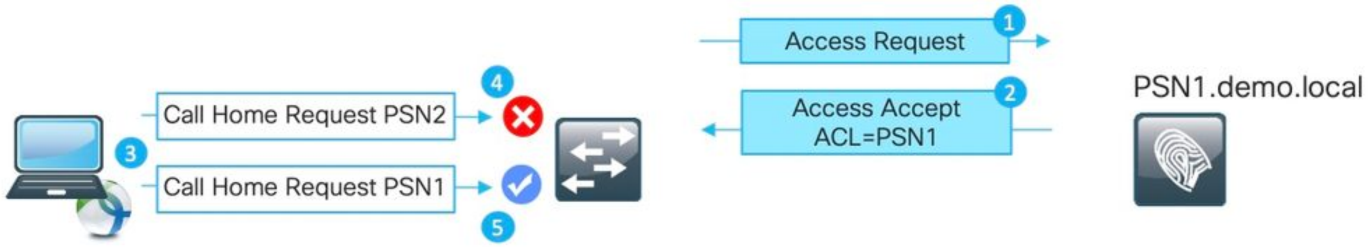
4. Posture **Pending** policy for PSN2.

5. PSN2 name used as a condition in the policy.

6. Authorization profile with ACL which blocks access to all PSNs except PSN2.

7. Posture 'Compliant' authorization policy.

The figure explains how this approach works:



1. Authentication hits PSN1.

2. As a result of configured authorization policies, PSN1 assigns authorization profile which blocks access to all other nodes except PSN1.

3. AnyConnect ISE posture module restarts the discovery process.

4. Probe to PSN2 is blocked by the NAD as by an ACL assigned earlier.

5. Probe to PSN1 is allowed by ACL assigned on NAD.

**Load Balancer Best Practices**

- Enable stickiness on the load balancer for authentication and accounting with Calling-Station-ID as a stickiness key. More details about LB best practices for ISE available here.
- Use stickiness timer longer than an average work-day to cover the moment when PC goes into sleep (for example 10 hours instead of 8 hours).
- In case re-authentication is implemented, use re-authentication timer slightly lower than stickiness

timer (for example 8 hours if stickiness configured for 10 hours). This ensures that the stickiness interval is prolonged by re-authentication.

### Posture Over VPN Use-case

- Ensure that the accounting-interim update interval is higher or equal to vpn-session-timeout. This minimizes accounting flapping between PSNs during long VPN sessions.

This example shows the interim accounting update interval configured for 20 hours. This does not prevent the initial interim update which carries IP address assigned to the endpoint.

```
aaa-server ISE protocol radius

 interim-accounting-update periodic 20

group-policy SSL-VPN attributes

 vpn-idle-timeout 1200

 vpn-session-timeout 1200
```

### Best Practices Can be Implemented to Minimize The Impact From The Absence of ISE Posture Module Discovery Restart

### Enable Posture Lease

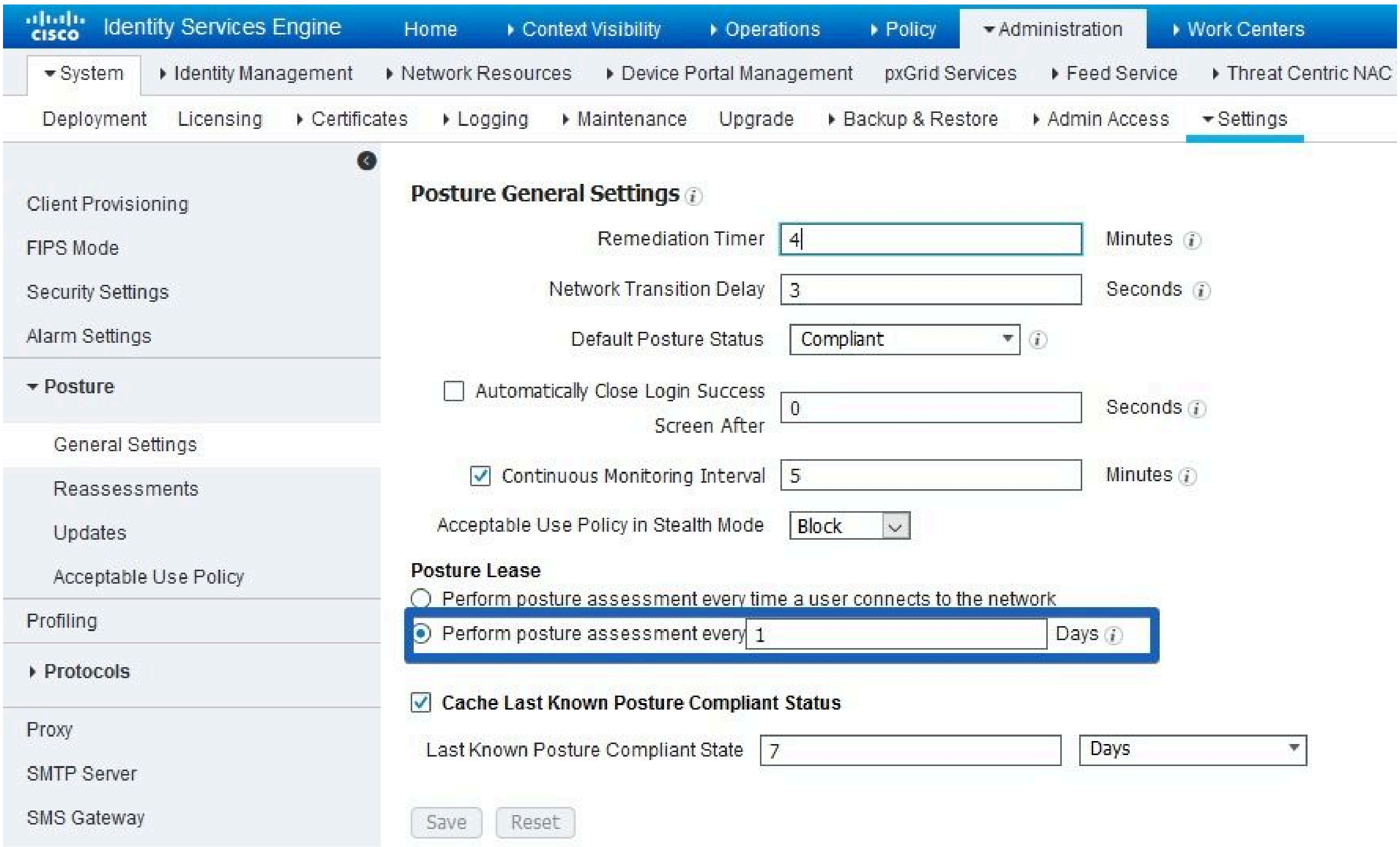
This is a feature on ISE which marks endpoint as **compliant** for a defined period (1-365 days). **Posture lease value** is an endpoint attribute which means that it is stored ISE DB.

All endpoint attributes which include posture lease are replicated across all the nodes in ISE deployment.

When PSN gets a new session for the endpoint posture, lease can be utilized to mark the session as **Compliant** right away.

To make this decision, PSN uses 3 values. Those values are:

1. Amount of days defined for posture lease in ISE settings: **Navigate to Administration > System > Posture > General Settings**:



**Posture General Settings**

Remediation Timer: 4 Minutes

Network Transition Delay: 3 Seconds

Default Posture Status: Compliant

□ Automatically Close Login Success Screen After: 0 Seconds

☑ Continuous Monitoring Interval: 5 Minutes

Acceptable Use Policy in Stealth Mode: Block

**Posture Lease**

○ Perform posture assessment every time a user connects to the network

● Perform posture assessment every 1 Days

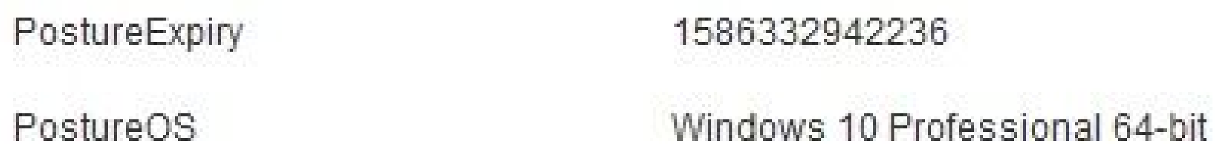
☑ **Cache Last Known Posture Compliant Status**

Last Known Posture Compliant State: 7 Days

2. **Value of PostureExpiry attribute** is an actual endpoint attribute which contains an Epoch timestamp. PostureExpiry value is initially populated upon the first successful posture attempt for endpoint after the ISE administrator enabled posture lease.

Later this value updated on the next successful posture attempt which happens after lease expiration.

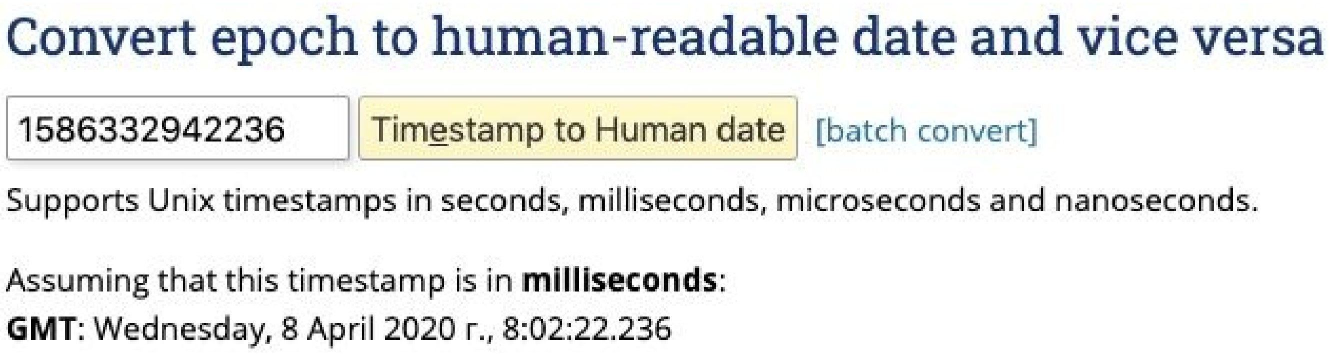
You can see a PostureExpiry in **Context Visibility > Endpoints** while one of the postured endpoints is opened:



PostureExpiry 1586332942236

PostureOS Windows 10 Professional 64-bit

This value can be converted into the human-readable timestamp, for example, here - https://www.epochconverter.com/



Convert epoch to human-readable date and vice versa

1586332942236

Supports Unix timestamps in seconds, milliseconds, microseconds and nanoseconds.

Assuming that this timestamp is in **milliseconds**:

**GMT**: Wednesday, 8 April 2020 г., 8:02:22.236

3. **PSN system time** at the moment when new authentication take place

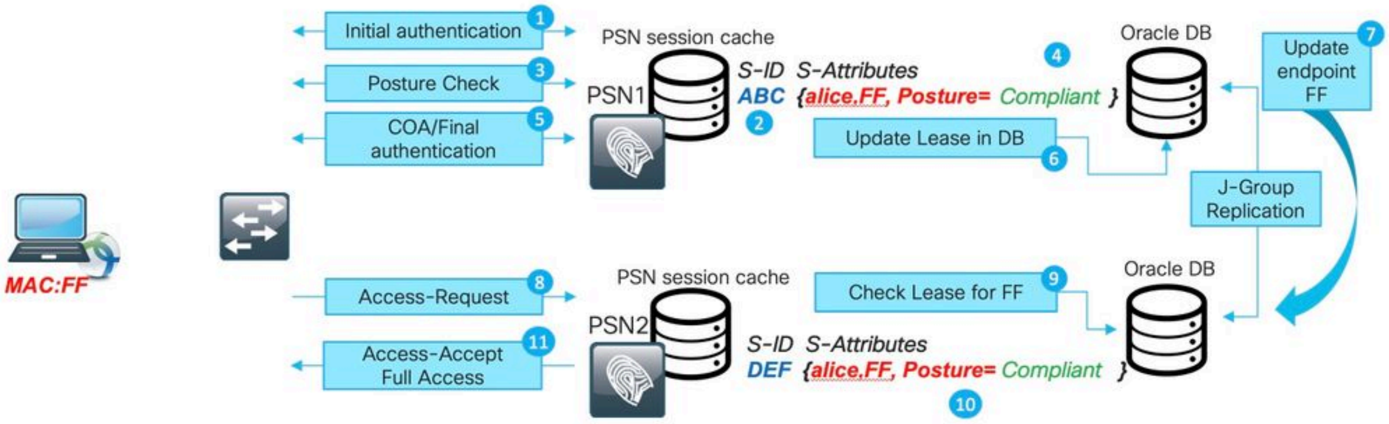
When authentication for an endpoint with posture lease hits PSN, it uses PostureExpiry and system date to get the number of days elapsed from the last successful posture check.

If the result value is within a posture lease interval defined in settings, the session gets a **Compliant** status.

If the result value is higher than the lease value, the session gets an **Unknown** status.

This triggers the posture to be executed again and the new PostureExpiry value can be saved.

This diagram depicts the process when failover happens:



1. Initial authentication happens with PSN1.

2. Session ABC is created in the session cache.

3. Posture assessment happens.

4. Session status changes to **Compliant**

5. COA is triggered by posture status change and leads to re-authentication of the endpoint to apply the next access level.

6. PostureExpiry value is saved in the endpoint.

7. Endpoint data is replicated across the deployment.

8. Next authentication hits PSN2.

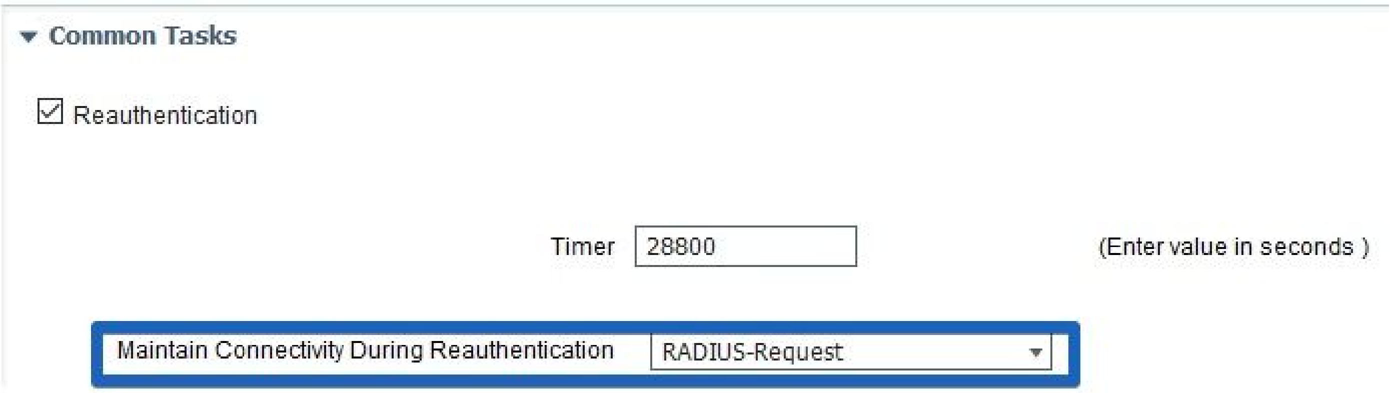
9. PSN2 checks if the endpoint is within a valid posture lease.

11. Session is added to the session cache as **Compliant**.

12. Due to the valid lease, the session is created with posture status **Compliant**.

**Re-authentication Implementation**

Always push re-authentication timer from ISE with **RADIUS-Request**, selected in **Maintain Connectivity During Reauthentication.** This setting ensures that NAD keeps the same session ID on re-authentication.



☑ Reauthentication

Timer 28800 (Enter value in seconds )

Maintain Connectivity During Reauthentication RADIUS-Request

.

### Environments With Load Balancers

The same set of best practices (explained in the stale/phantom session section) can be implemented.

### Different Subnets Can be Used for Pending and The Compliant States

When network design provides the opportunity to use different subnets **Pending** and **Compliant** states, this approach guarantees that every change in posture status results in the Default Gateway change.

### Posture Assessment Used in the Same interval as a Re-authentication Timer

Posture Assessment can be enabled with the interval equal to the reauthentication timer. In such a case, when the original PSN becomes not available, PRA failure restarts the discovery process.

## Modern Approach - Posture State Sharing

As part of an implemented enhancement (described in Cisco bug ID CSCvi35647 ) patch 6 for ISE 2.6 has a new feature that implements the sharing of session posture status across all the nodes in ISE deployment.

This enhancement is integrated into future releases: ISE 2.7 patches 2 and ISE 3.0.

This new feature is based on Light Session Directory (LSD) mechanism which has been introduced in ISE 2.6. In the newer versions, this functionality has been renamed to Light Data Distribution (LDD) Radius Session Directory. Light Data Distribution is enabled by default and allows the sharing of a limited session context between ISE nodes. There is no such thing as full session context replication between PSNs, only a limited amount of attributes shared for each session.

Light Session Directory removes the need to execute resource expensive API calls to MNT when one of the nodes in the deployment has to determine the current session owner.

Mostly, owner lookup is needed when COA flow starts. With LDD every PSN can find an actual owner of the session from the local Radius Session Directory cache.

### Light Data Distribution Architecture

This functionality contains these elements:

- Radius Session Directory (RSD) cache

This cache exists on every ISE node and stores all active sessions presented in ISE deployment. Every session has a limited amount of attributes in the cache.

Here are examples of the attributes stored in the Radius Session Directory for each session:

- Session ID
- Endpoint MAC
- CallingStationID
- Endpoint IP
- PSN IP - PSN where authentication happened
- PSN FQDN
- NAS-IP-Address
- NAS-IPv6-Address
- State - Authenticated, Started, Stopped

- RabbitMQ exchange

There is an exchange formed in which Publisher, related Queue, and Consumer are presented on every node in the ISE deployment. This ensures that the full-mesh topology formed between all the ISE nodes.

- Publisher

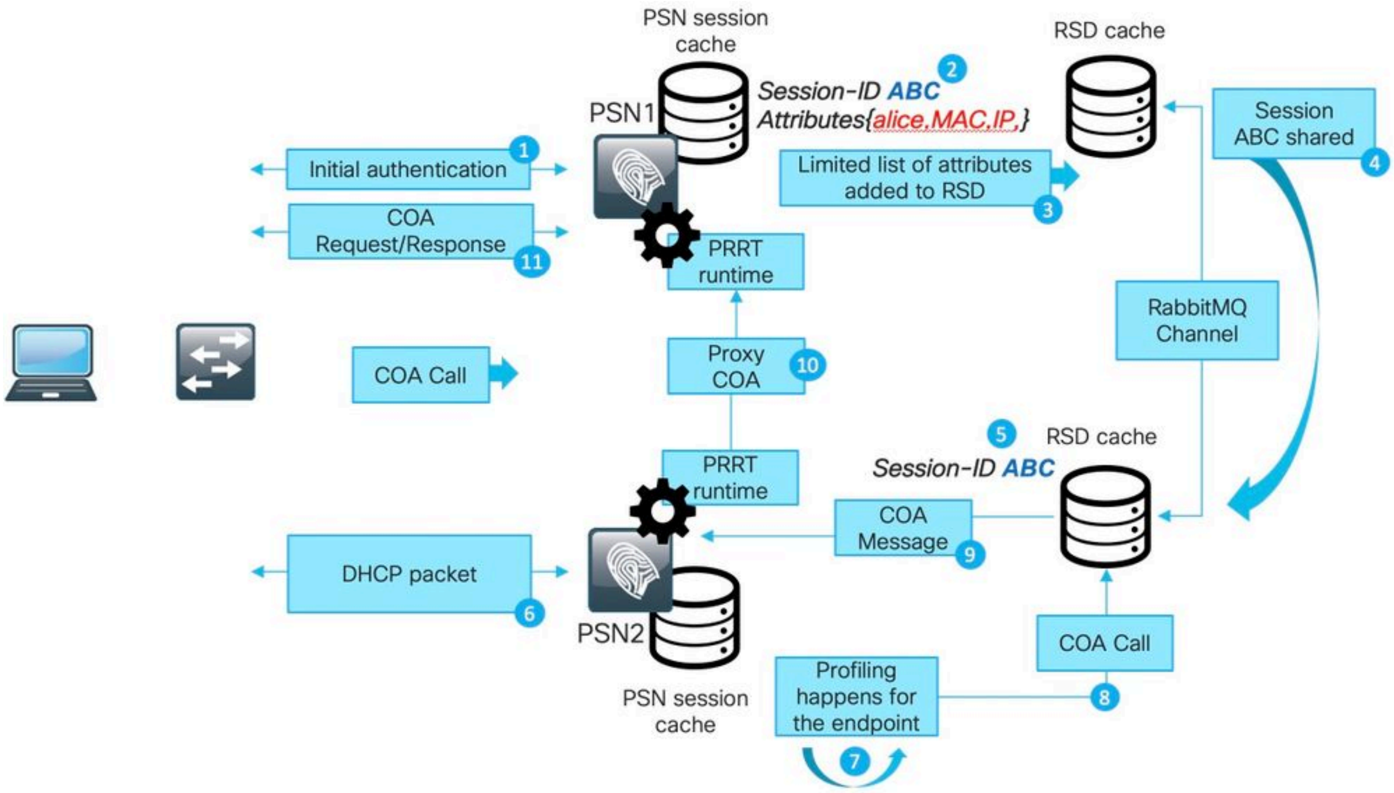
The Radius Session Directory represents a publisher here. When a new successful authentication is processed by a PSN, a new session gets created in the PSN session cache.

For this session, a limited set of attributes is placed into the Radius Session Directory.

- Consumer - on all other nodes; Radius Session Directory represents a consumer.

**Note**: General RabbitMQ terminology and architecture is outside of this document scope.

The figure explains how COA flow works with RSD cache:



1. Initial authentication happens with PSN1.

2. Session ABC is created in the session cache.

3. Required attributes are saved into RSD.

4. Session is shared over RabbitMQ with all other ISE nodes.

5. Session is created in RSD cache on all ISE nodes.

6. New profile data arrives on PSN2.

7. Endpoint is reprofiled and (in case of a change which requires COA execution PSN2) proceeds with the next step.

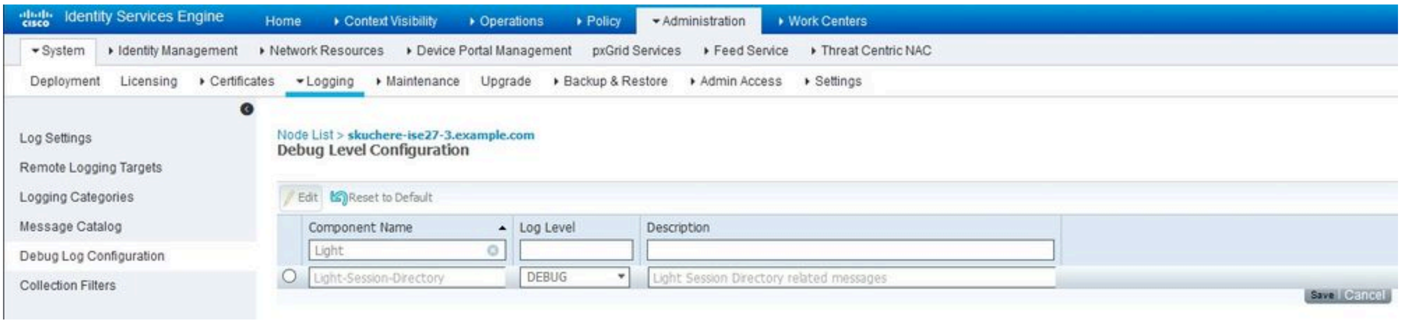
8. An internal API call is submitted to RSD cache to execute COA.

9. Data from the RSD cache is used to prepare a Proxy COA message. It goes from one ISE node to another and contains all details which the destination node can use to issue a CAO request back to NAD. The COA message is first transferred internally to PRRT Runtime (Actual AAA server inside of ISE).

10. PSN2 sends a COA message to PSN1.

11. PSN1 sends a COA message to NAD.

To troubleshoot communication over LDD on the ISE, enable **Light Session Director** component into DEBUG:



Node List > skuchere-ise27-3.example.com

Debug Level Configuration

| | Component Name | Log Level | Description |
|---|---|---|---|
| | Light | | |
| ○ | Light-Session-Directory | DEBUG | Light Session Directory related messages |

Here is an example of a debug message from **lsd.log** file for session creation and publication on the original PSN:

```
DEBUG  [pool-45-thread-6][] cisco.cpm.lsd.service.LSDRedisClient -::::- Mapping Session ID 0a3e9498000

DEBUG  [PrRTEvents-Executor-2][] cisco.cpm.lsd.service.LSDNetAccessEventListener -::::- Publishing sess

DEBUG  [PrRTEvents-Executor-2][] cisco.cpm.lsd.service.SessionPublisher -::::- Forwarding session 07a26
```

On all other ISE nodes, you see how a session was consumed:

```
[pool-35-thread-38][] cisco.cpm.lsd.service.SessionConsumer -::::- Consumer is processing : sessionID:[
```

### Posture Status Sharing Over RSD

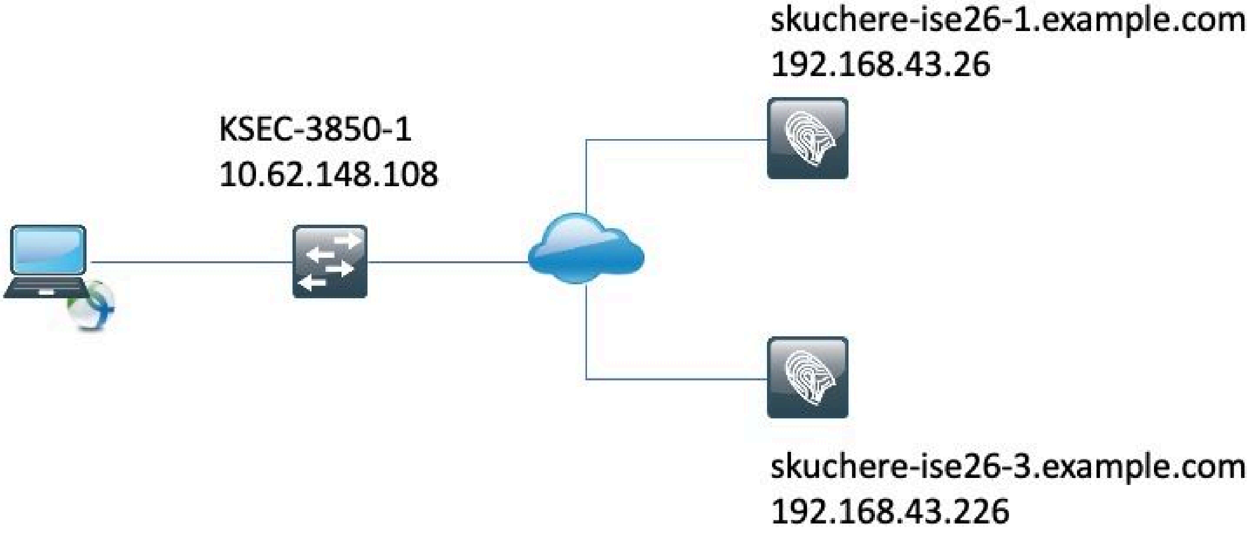
Posture status sharing solves problems when the root cause is either Stale/Phantom session or Re-authentication on different PSN which did not trigger discovery restart.

As soon as the session becomes Compliant, this information places into the session RSD, and later it can be used by every PSN in the deployment.

There are still some other corner cases that the described feature cannot solve. For example, a scenario when NAD runs re-authentication on the same PSN but with a different session ID.

Such scenarios can be handled with best practices described in this document.

This figure demonstrates the topology used for a test of posture status sharing:

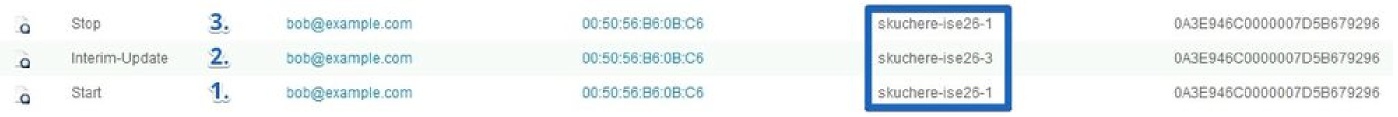


### Posture Status sharing Over RSD - Stale/Phantom Session

To create a stale session, authentication must be initially performed on the skuchere-ise26-1. Then, NAD must be reconfigured to send accounting to skuchere-ise26-3.

After one accounting message has been forwarded to the wrong PSN, NAD must be reconfigured (again) to send accounting back to skuchere-ise26-1.

The figure demonstrates an accounting report which proofs the presence of the phantom session on skuchere-ise26-3:



| | | | | | | |
|---|---|---|---|---|---|---|
| | Stop | 3. | bob@example.com | 00:50:56:B6:0B:C6 | skuchere-ise26-1 | 0A3E946C0000007D5B679296 |
| | Interim-Update | 2. | bob@example.com | 00:50:56:B6:0B:C6 | skuchere-ise26-3 | 0A3E946C0000007D5B679296 |
| | Start | 1. | bob@example.com | 00:50:56:B6:0B:C6 | skuchere-ise26-1 | 0A3E946C0000007D5B679296 |

1. Accounting-Start messages processed by skuchere-ise26-1.

2. Interim Accounting-Update for the same session processed by skuchere-ise26-3.

3. The session finishes later on  skuchere-ise26-1.

After some time endpoint again connects to the network but the redirection no longer works. In the guest.log of PSN - skuchere-ise26-3, you can see these log messages with **client-webapp** component enabled into DEBUG:

```
2020-04-08 13:30:48,217 DEBUG [https-jsse-nio-192.168.43.226-8443-exec-4][] cisco.cpm.client.posture.Ut
```

When PSN detects that it holds a stale/phantom session for the endpoint, it does not reply to the ISE posture module and this allows us to get the right answer from the PSN where the latest authentication happened.

As a solution to the stale/phantom session problem at the time of the session lookup, PSN checks the presence of any new session for the endpoint in the RSD.

In case RSD contains a different session ID from what PSN has in the local session cache, it assumes that the session (presented in the session cache) is stale.

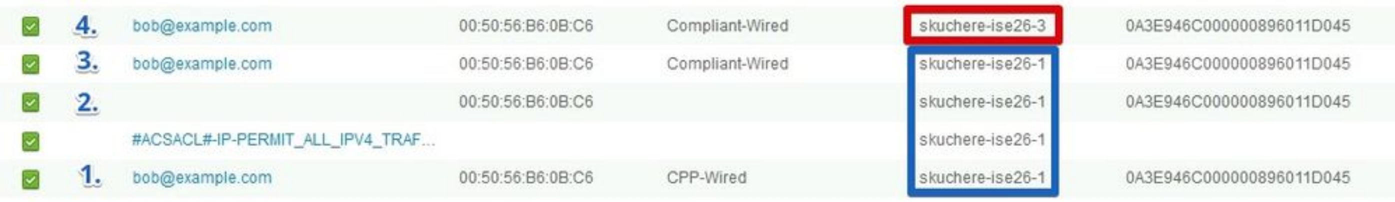
### Posture Status Sharing Over RSD - Failover Between PSNs

To reproduce this scenario, a short re-authentication timer is enabled in the authorization profile assigned to the compliant state endpoint.

Later, NAD is reconfigured to send authentication and accounting to another PSN (skuchere-ise26-3).

Upon re-authentication timer expiration, the same session is unauthenticated on the different PSN.

The figure demonstrates an authentication report which shows failover for the same session from skuchere-ise26-1 to skuchere-ise26-3:



| | | | | | | |
|---|---|---|---|---|---|---|
| ☑ | 4. | bob@example.com | 00:50:56:B6:0B:C6 | Compliant-Wired | skuchere-ise26-3 | 0A3E946C000000896011D045 |
| ☑ | 3. | bob@example.com | 00:50:56:B6:0B:C6 | Compliant-Wired | skuchere-ise26-1 | 0A3E946C000000896011D045 |
| ☑ | 2. | | 00:50:56:B6:0B:C6 | | skuchere-ise26-1 | 0A3E946C000000896011D045 |
| ☑ | | #ACSACL#-IP-PERMIT_ALL_IPV4_TRAF... | | | skuchere-ise26-1 | |
| ☑ | 1. | bob@example.com | 00:50:56:B6:0B:C6 | CPP-Wired | skuchere-ise26-1 | 0A3E946C000000896011D045 |

1. Authentication happens on skuchere-ise26-1, authorization profile with redirection is assigned.

2. COA after successful posture assessment.

3. Next authentication when authorization profile for the compliant state is assigned.

4. Authentication hits different PSN but it still gets authorization profile for the compliant state.

The session has compliant status on the new PSN after failover in ise-psc.log with **epm-pip** and **nsf-session** components enabled into DEBUG:

<#root>

```
2020-04-09 11:06:42,176 DEBUG [Thread-7979][] cpm.nsf.session.impl.SessionCache -::::-
```

```
Looking up session 0A3E946C000000896011D045 for attribute Session Session.PostureStatus
```

```
2020-04-09 11:06:42,176 DEBUG [Thread-7979][] cpm.nsf.session.api.ExecutionContext -::::- Execution con
2020-04-09 11:06:42,176 DEBUG [Thread-7979][] cpm.nsf.session.impl.PIPManager -::::- Returning a PIP co
2020-04-09 11:06:42,176 DEBUG [Thread-7979][] cpm.nsf.session.api.ExecutionContext -::::- Execution con
2020-04-09 11:06:42,176 DEBUG [Thread-7979][] cpm.nsf.session.impl.SessionCache -::::- Looking up sessi
2020-04-09 11:06:42,176 DEBUG [SessionLifecycleNotifier][] cpm.nsf.session.internal.LRUAgingAlogrithm -
2020-04-09 11:06:42,176 DEBUG [Thread-7979][] cpm.nsf.session.impl.SessionCache -::::- Returning for se
IndexValues: {}
```

```
2020-04-09 11:06:42,177 DEBUG [Thread-7979][] cisco.cpm.posture.pip.PostureStatusPIP -::::-
```

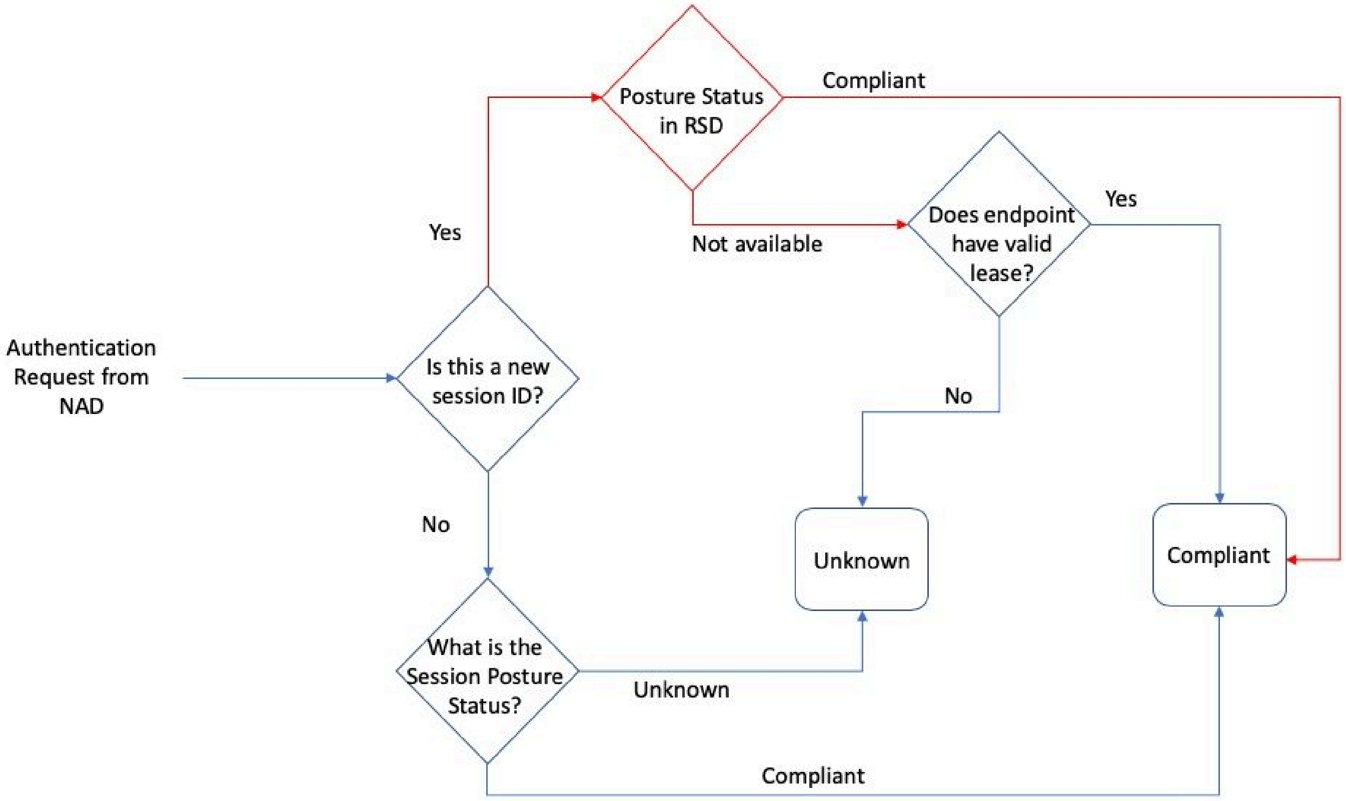
```
set postureStatus based on posture LSD dictionary: Compliant
```

```
2020-04-09 11:06:42,177 DEBUG [Thread-7979][] cisco.cpm.posture.pip.PostureStatusPIP -::::-
```

```
PostureStatusPIP for mac 00-50-56-B6-0B-C6 - Attribute Session.PostureStatus value is Compliant
```

The original issue is solved with the addition of extra logic into the posture status selection process.

This figure demonstrates what has been changed (changes highlighted in red):


Posture Status in RSD
Compliant
Yes
Not available
Does endpoint have valid lease?
Yes
Authentication Request from NAD
Is this a new session ID?
No
No
Unknown
Compliant
What is the Session Posture Status?
Unknown
Compliant