Troubleshoot ISE Session Management and Posture

Contents

Preview

Introduction

Background Information

Problem

- End-User Experience
- ISE Admin Experience
- Common Problematic Scenarios
- Stale/Phantom Session Problem
  - ISE Session Management Logic
  - MNT and Session Management
  - PSN and Session Management
  - Stale Session on PSN
  - Phantom Session on the PSN
  - How Stale Session and Phantom Session Break the Posture Process
- Discovery Process does not Start on a New Authentication Attempt
  - Reauthentication on Different PSN
  - NAD Generates New Session-ID on Reauthentication
- Quick Ways to Identify When an Issue was Caused by the Stale/Phantom Session
- Advance Troubleshoot of Stale/Phantom Session
  - DART Bundle Collection
  - DART Bundle Analysis
  - Investigation on ISE Logs
  - Investigation on ISE Reports
- How to Identify When an Issue was Causedby the Absence of a Discovery Restart
- Advanced Troubleshoot The Absence of Discovery Restart

Solution

- Classical Approach - Issue Avoidance
  - Best Practices That Can Minimize The Amount of Stale or Phantom Sessions in The ISE Deployment
  - Posture Over VPN Use-case
  - Best Practices Can be Implemented to Minimize The Impact From The Absence of ISE Posture Module Discovery Restart
  - Reauthentication Implementation
  - Environments With Load Balancers
- Modern Approach - Posture State Sharing
  - Light Data Distribution Architecture
  - Posture Status Sharing Over RSD
  - Posture Status Sharing Over RSD - Stale/Phantom Session
  - Posture Status Sharing Over RSD - Failover Between PSNs

Introduction

This document describes the common Identity Service Engine (ISE) posture services problem such as **"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 an issue like this becomes extremely time-consuming, which causes serious impact.

This document explains:

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

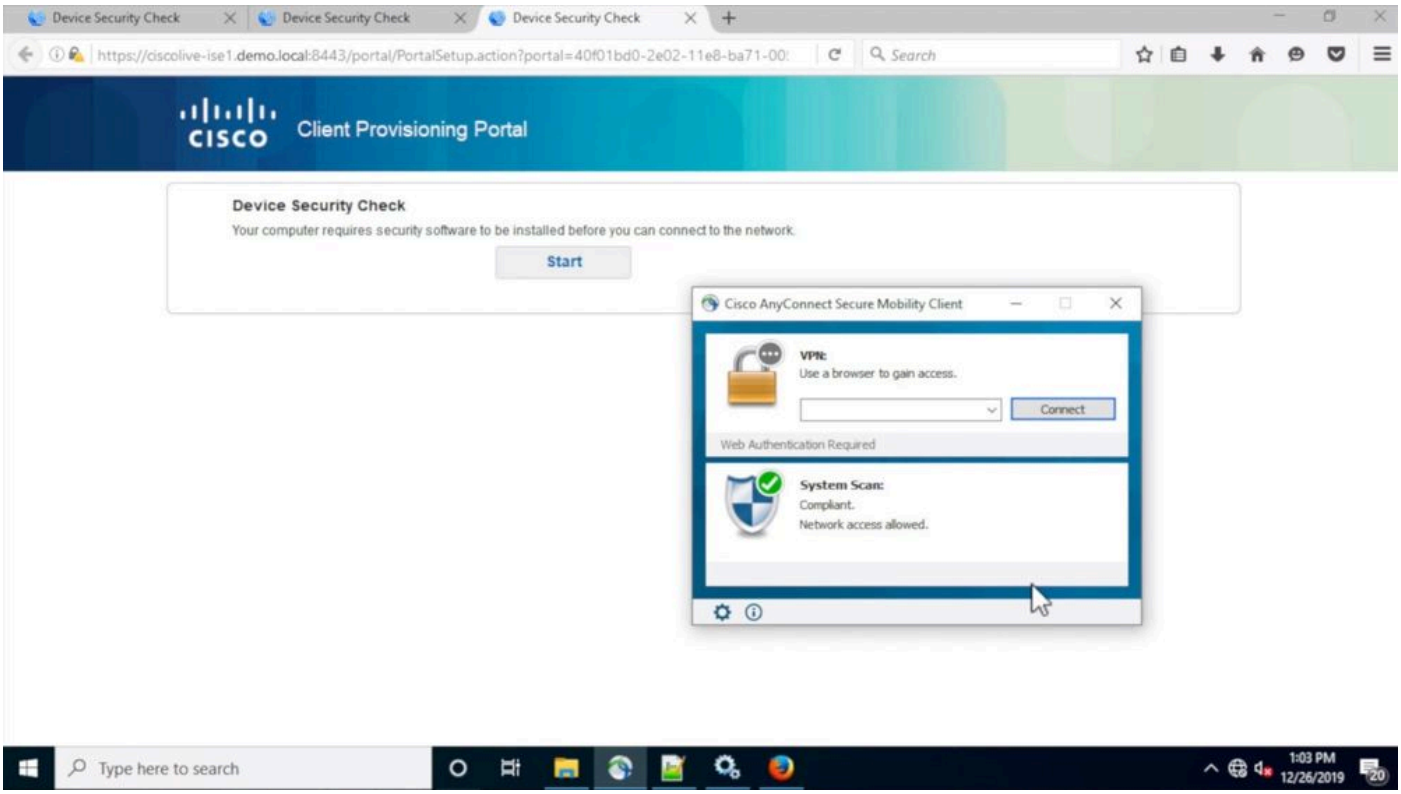
For a better explanation of the concepts described later, refer to ISE Posture Style Comparison for Pre and Post 2.2

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**.

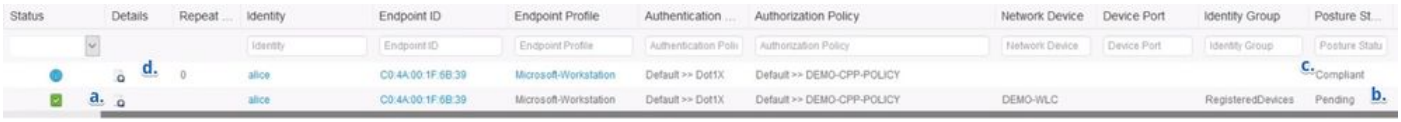
Typical end-user experience:



ISE Admin Experience

When initially triaging this issue, an ISE admin begins an investigation of the Radius Live logs to ensure an authentication hits ISE. The first symptom discovered in this stage indicates a mismatch in a posture status between the endpoint and ISE in the live logs. Or, the Radius authentication reports last successful authentication for the endpoint shows **Pending** posture status.

Typical ISE Admin experience:



- 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: In the example, 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 display. In this case, TAC normally works 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.

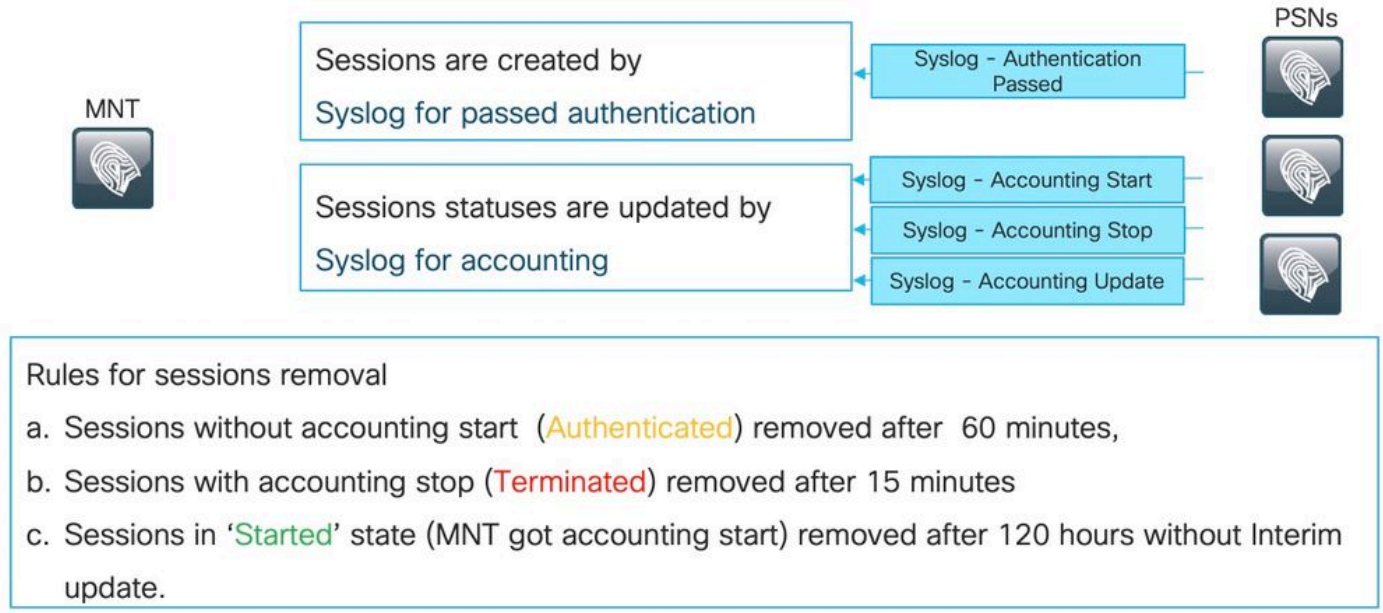
Stale/Phantom Session Problem

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 the issue, it is critical to understand the theory of session management on both personas.

MNT and Session Management



As explained in this image, the MNT node creates seasons based on the passed authentication Syslog messages that come from PSNs. Session status can be updated later by the Syslog for accounting.

Session removal on MNT happens in three 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 sessions in the 'Started' state for more than 5 days (120 hours.) A started state means the MNT node processed both authentication and accounting to start the Syslog session.

Note: These are the normal MnT cleanup timers, not guaranteed wall-clock deletion times. The removal can be delayed if:

- SessionPurgeJob is not running successfully.
- MnT is under resource or database pressure.
- The Accounting-Stop cannot be correlated to the existing session.

Examples of Syslog messages from PSN:

Messages are logged into the 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=0A3E946C000000

5200 NOTICE Passed-Authentication: Authentication succeeded

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

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, A

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

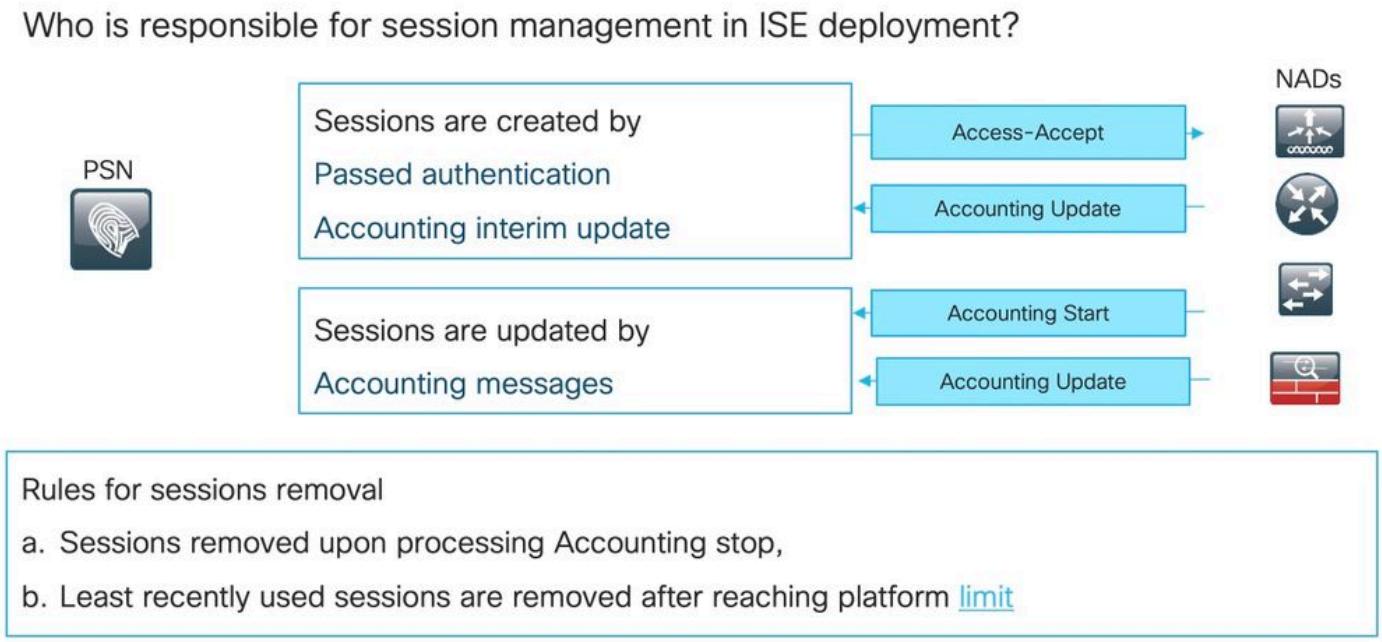
---

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 states from one node to another.

For every active session ID, PSN stores all attributes 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 such as Authentication, Authorization, Client Provisioning, and Posture.

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



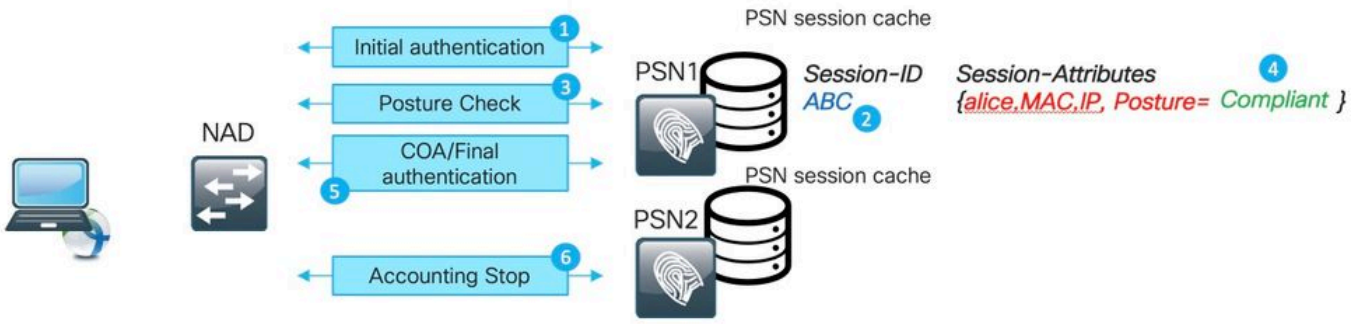
Current session processing logic creates a new entry in the session cache in two scenarios. Later details of existing sessions can be updated from accounting messages, which come from NADs.

- The session has been successfully authenticated on the PSN.
- PSN has an accounting interim update for the session that does not exist in the session cache. With session removal, PSN implements this logic:
    - Session cache entry is removed immediately after processing the accounting stop message.
    - PSN begins 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 that did not perform the actual authentication:

Example of the stale session:



---

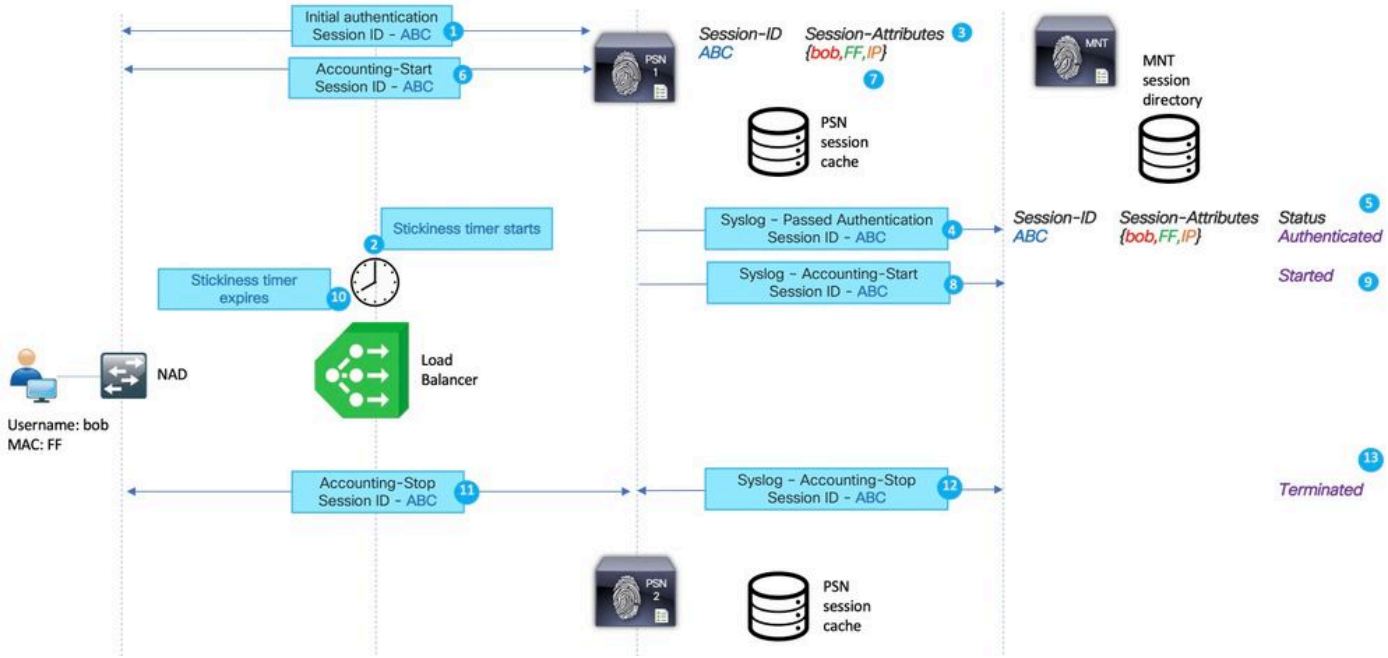
1. Successful authentication happens on PSN for session ABC.
2. PSN creates an entry in the session cache.
3. Posture assessment occurs.
4. Session is marked as **Compliant**.
5. Change of Authorization (COA) is triggered by the posture status change, which leads to reauthentication of the endpoint to apply the next access level.
6. Accounting stops for session ABC to PSN2.

Afterwards, ABC becomes stuck in a stale state on PSN1 as there is no accounting stop message processed on this PSN to remove it. The session is removed if deployment does not experience a high number of authentication attempts.

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

- The accounting stop was sent 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 caused NAD failover to the next PSN.

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



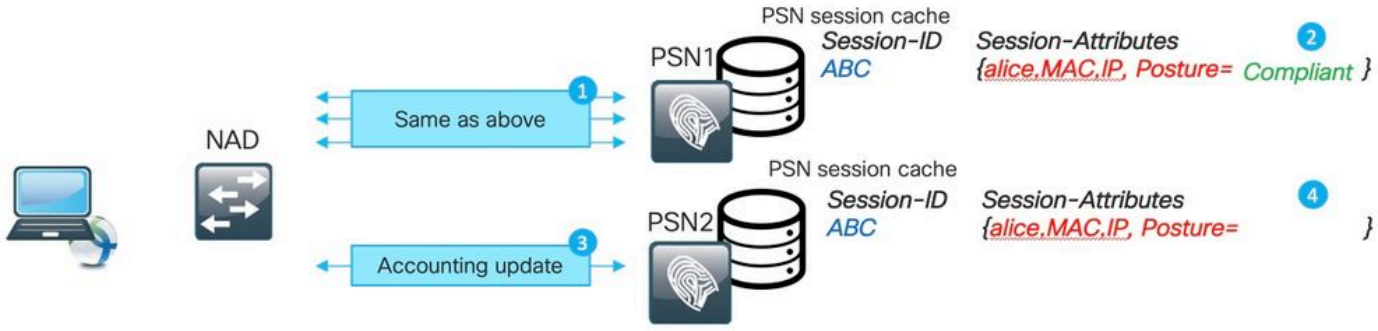
---

1. Initial authentication for Session ABC is performed by PSN1. This authentication initiates a stickiness timer on the load balancer.
2. PSN1 creates an entry for session ABC in the local cache.
3. The Syslog message for the passed authentication is transferred to the MNT node.
4. The entry for session ABC is created into the MNT session directory with the **Authenticated** state.
5. The accounting start message for session ABC lands on PSN1.
6. The session cache entry for session ABC is updated with information from Accounting-Start.
7. The Syslog message for Accounting-Start is transferred to the MNT node.
8. Session state is updated to **Started**.
9. Stickiness timer expires on the load balancer.
10. Accounting-Stop for session ABC is forwarded by the load balancer to PSN2.
11. The Syslog message for Accounting-Stop is forwarded by PSN2 to MNT.
12. Session ABC is marked as **terminated** on MNT.

Phantom Session on the PSN

The phantom session is a scenario when the accounting interim update comes to the PSN and it did not perform authentication for that session. In this scenario, a new entry is created in the PSN session cache. If PSN does not receive 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:



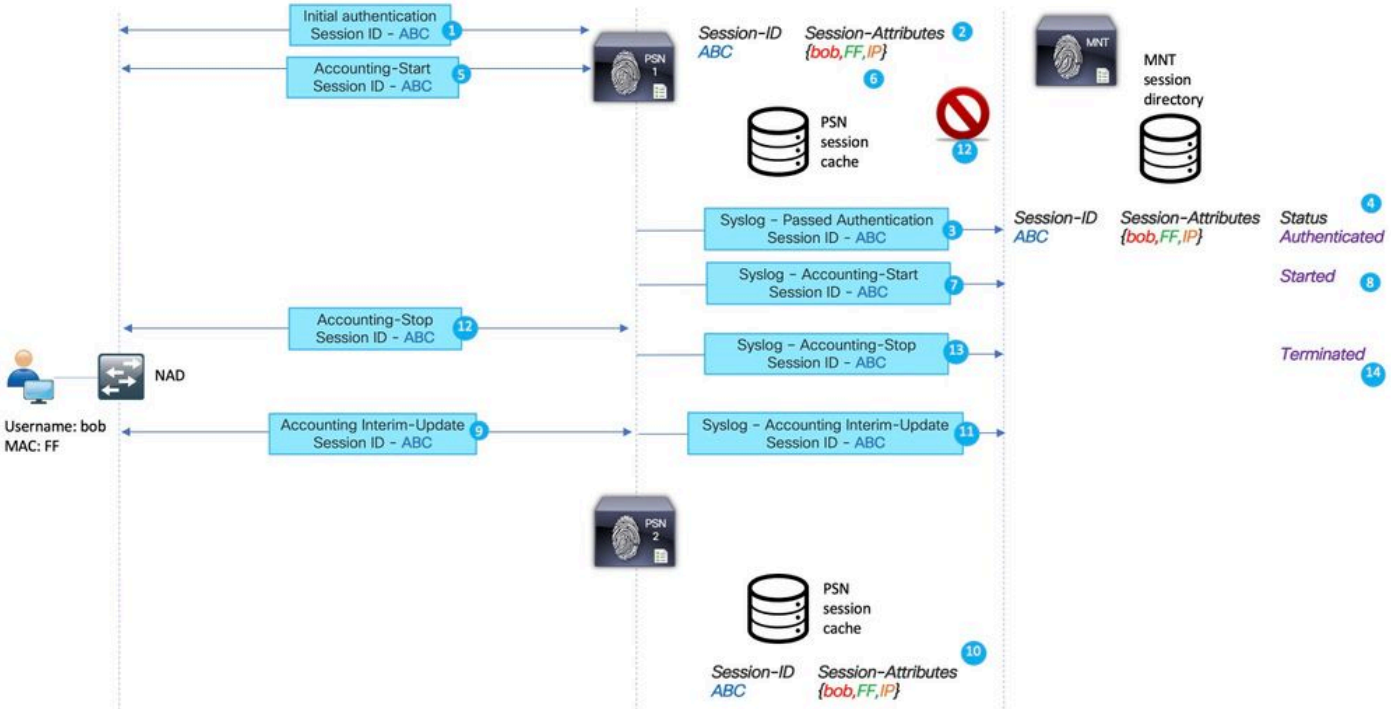
---

1. The same steps described in the stale session example occurs on PSN1 for the ABC session.
2. Session ABC has a status of **Compliant** in the PSN1 session cache.
3. An accounting interim update for session ABC hits PSN2.
4. A session entry for ABC is created on PSN2. Since the session entry is created from the accounting message, it has a limited number of attributes. For example, the posture status is not available for session ABC, and user groups or other authorization specific attributes are absent as well.

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

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

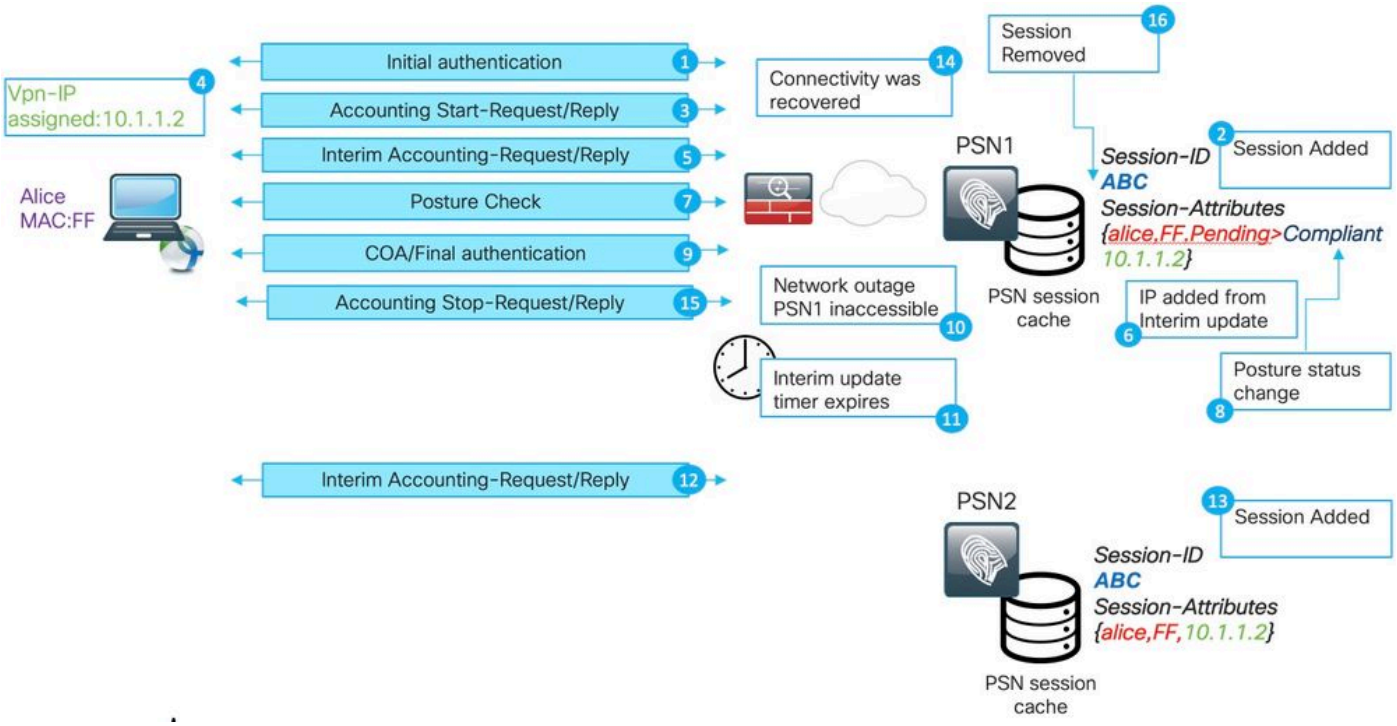
The next screenshot is an example of a phantom session where temporary issues on the network path towards PSN1:



---

1. Initial authentication for session ABC is performed by PSN.
2. PSN1 creates an entry for 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 PSN1 to MNT.
14. Session ABC is marked as terminated on MNT.

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



---

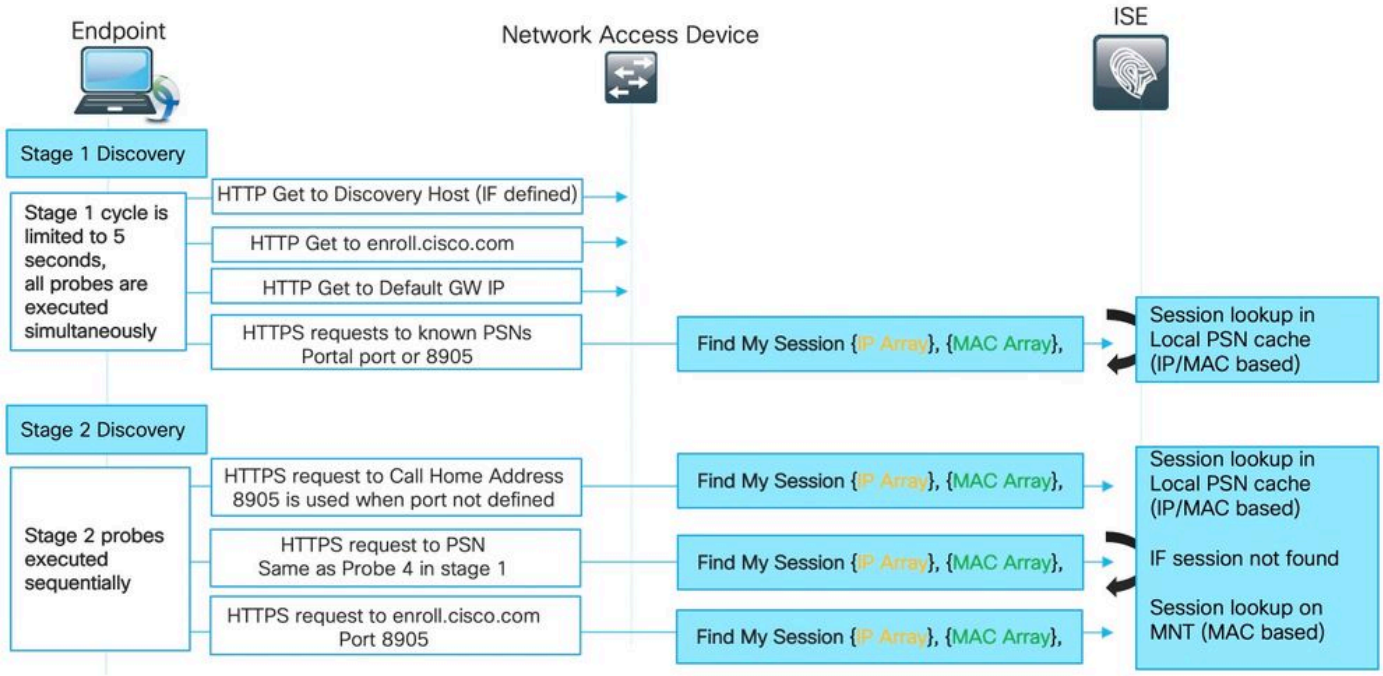
1. Initial authentication on PSN1.
2. Session ABC is created in the session cache.
3. Accounting starts the message processed by the PSN.
4. The new IP address is assigned to the Virtual Private Network (VPN) adapter.
5. An interim accounting update with the IP address information lands on the PSN.
6. The IP address information is added to the session cache.
7. Posture assessment occurs with PSN1.
8. The 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 PSN1 is inaccessible.
12. The interim accounting update arrives 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) 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 and the phantom sessions break the posture, you can review the AnyConnect ISE posture module discovery process:



Stage 1 Discovery:

During this stage, the ISE posture module executes four simultaneous probes to locate the PSN which authenticates the endpoint.

First, three 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 the redirected URL is taken from the NAD itself.

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

On the 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 results from probe four is used only if all other three probes fail or if the ISE posture module cannot establish proper communication with the PSN returned in the redirect URL within 5 seconds.

When probe four 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 receives multiple answers for probe four (**ConnectionData.xml** can contain more than one primary PSN), the fastest reply is always used.

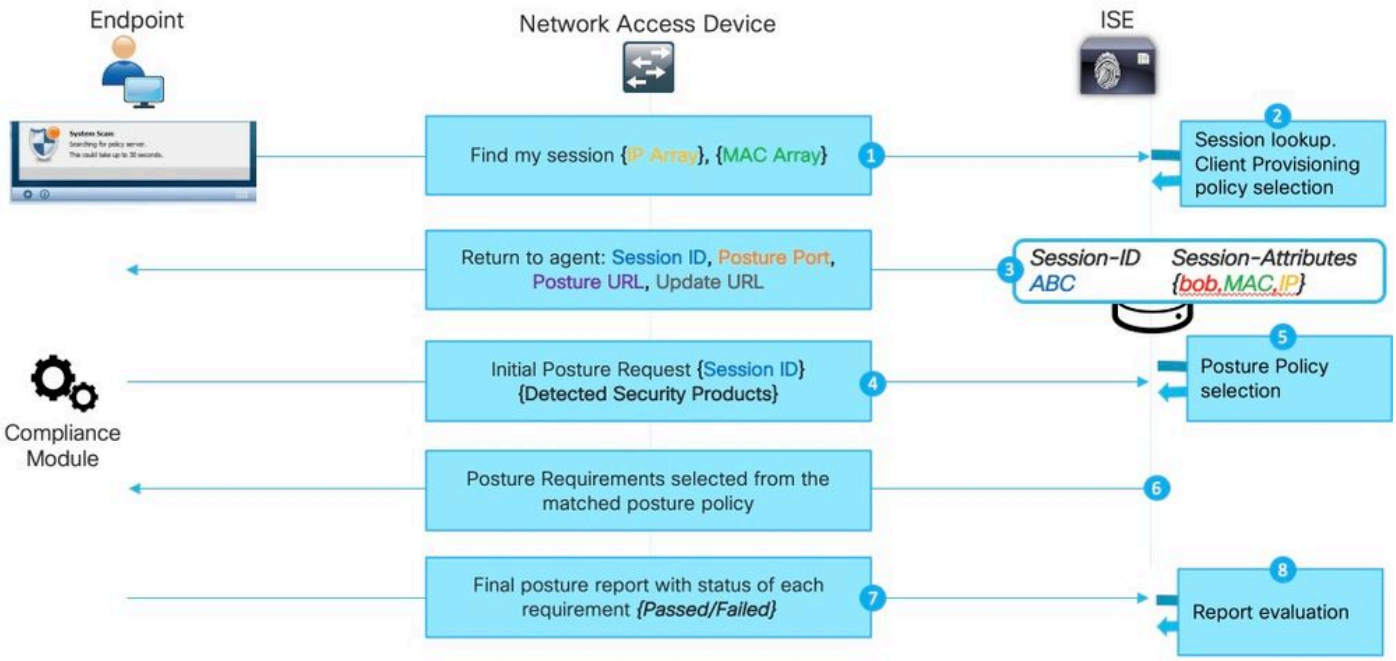
Stage 2 Discovery:

All stage 2 discovery probes are redirect-less, which means every probe triggers a session lookup on the destination PSN. If the PSN cannot locate the session in the local session cache, it must perform an MNT lookup (MAC address-based only) to find a session owner and return the owner name to the agent.

As all probes trigger session lookup, Stage 2 discovery can be significantly affected by issues as a result from stale or phantom sessions.

If PSN moves 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.

This example shows how posture appears when PSN holds a stale session or phantom session:



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. If a phantom session has a lack of authentication/authorization attributes and all policies configured by you are specific (for example, policies are created for specific Active Directory groups), PSN is cannot assign a correct client provisioning policy. This can manifest in the error message: "Bypassing AnyConnect scan your network is not configured to use Cisco NAC Agent".
    - In cases where client provisioning policies are generic (attributes available in the phantom session are enough to match the policy with an AnyConnect configuration) PSN replies with details required for the continuation of the assessment process.
    - At this step, when working with stale sessions, PSN replies immediately with a posture status of **Compliant** and all next steps are not completed. PSN does not send COA because it believes 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 on 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. The phantom session has a lack of attributes at this point, and there is no policy to match. In this case, PSN replies to the endpoint that it is compliant. This is default ISE behavior if the posture policy does not match.

Note: When there is a generic policy that can be selected from phantom session attributes, you can continue to Step 6.

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

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

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

8. The report evaluation occurs on ISE and the session status changes to **Compliant.**

Note: If posture issues are caused by the phantom session, the ISE administrator can notice some failed posture COAs. If so, the COA requests are executed from the incorrect PSNs and for the 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:

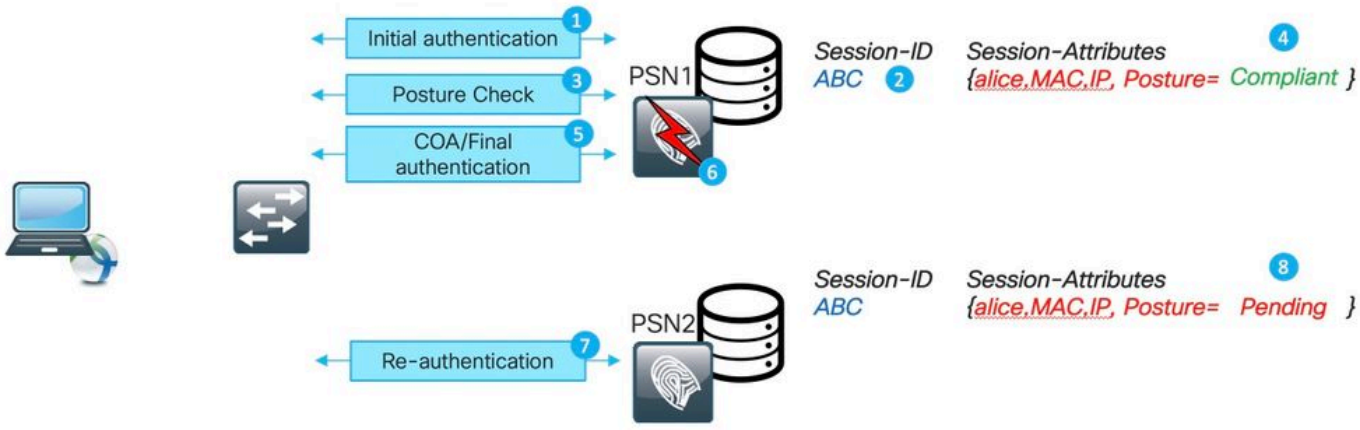
- Initial ISE posture module installation.
- User login.
- Power events.
- Interface status change.
- OS resume after sleep.
- Default Gateway (DG) change.
- Posture Reassessment (PRA) failure, see Cisco bug ID CSCvo69557. New dot1x authentication, PC unlock, and IP address change are not detected by the ISE posture module.

The ISE posture module cannot detect a new authentication or reauthentication attempt in these scenarios:

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

Reauthentication on Different PSN

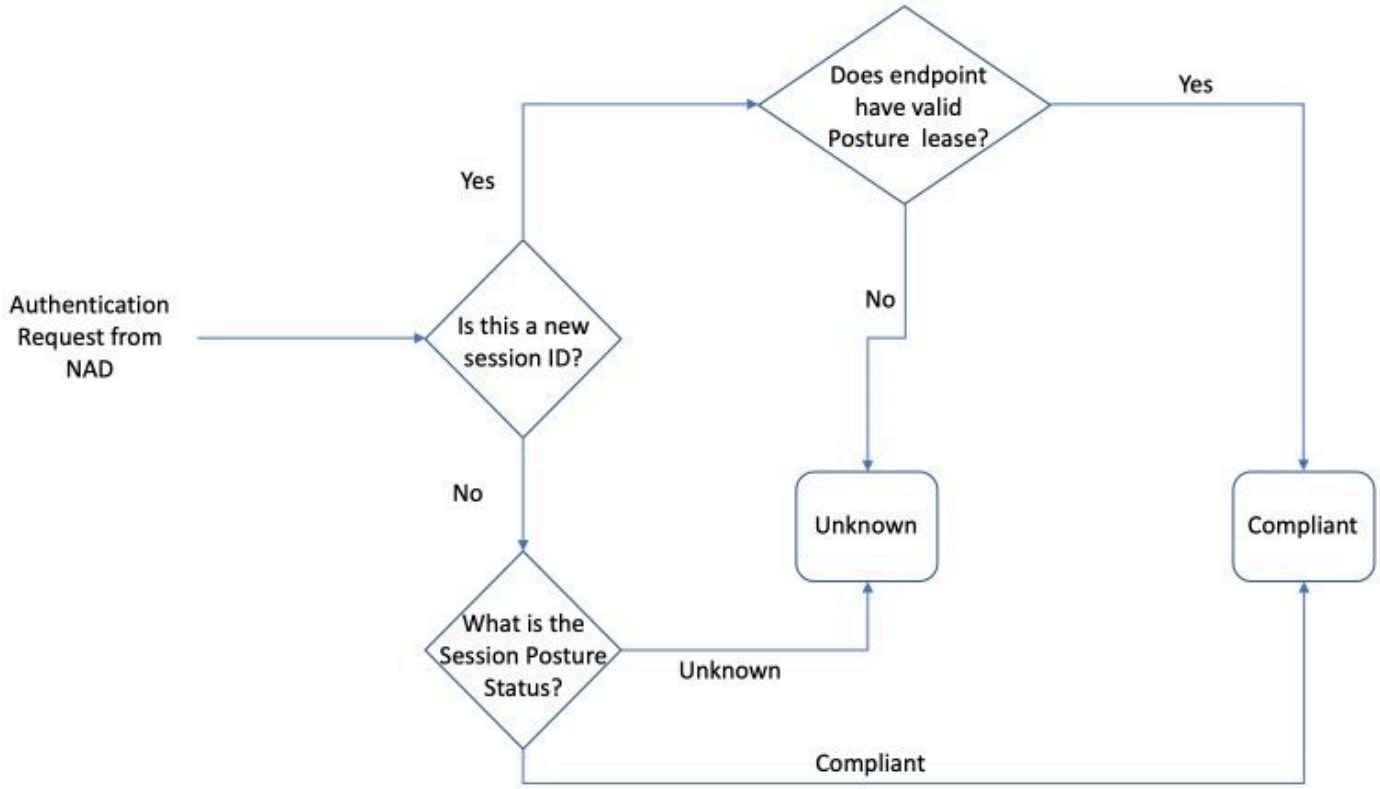
This diagram depicts an example of reauthentication on a different PSN caused by the outage of the original PSN. A scenario with a load balancer looks similar. In the case of a load balancer, reauthentication is directed to the different PSN as a result of a stickiness timer expiration.



---

1. Initial authentication on PSN1.
2. Session ABC is created in the PSN1 session cache.
3. Posture assessment is performed with PSN1.
4. Session ABC posture status moves to **Compliant.**
5. COA is triggered by the posture status change and leads to reauthentication of the endpoint to apply the next access level.
6. PSN1 becomes unavailable.
7. Reauthentication for session ABC hits PSN2.
8. Because it is a new session for PSN2, the posture status of the session changes to **Pending.**

The initial posture status is assigned by PSN to the session:



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

NAD Generates New Session-ID on Reauthentication

This can happen in the two most common scenarios:

- Reauthentication 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 reauthentication attempt. This is a potentially buggy behavior that must be investigated on NAD.
    - The new session ID can be generated in some other scenarios. For example, in some cases, wireless roaming can be a cause.
    - 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.

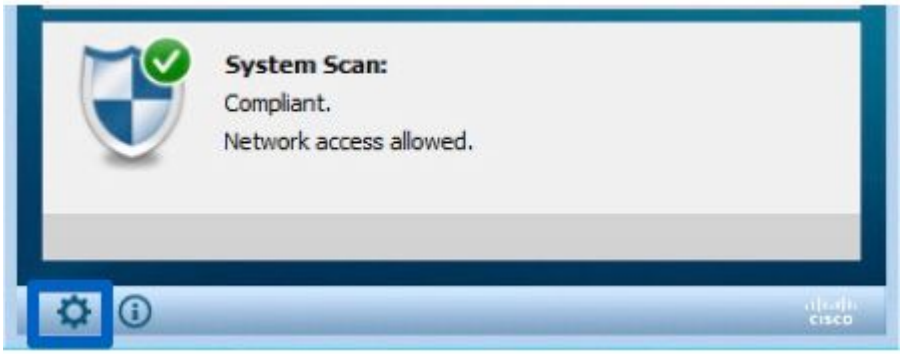
Quick Ways to Identify When an Issue was Caused by the Stale/Phantom Session

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

Investigate System Scan Details

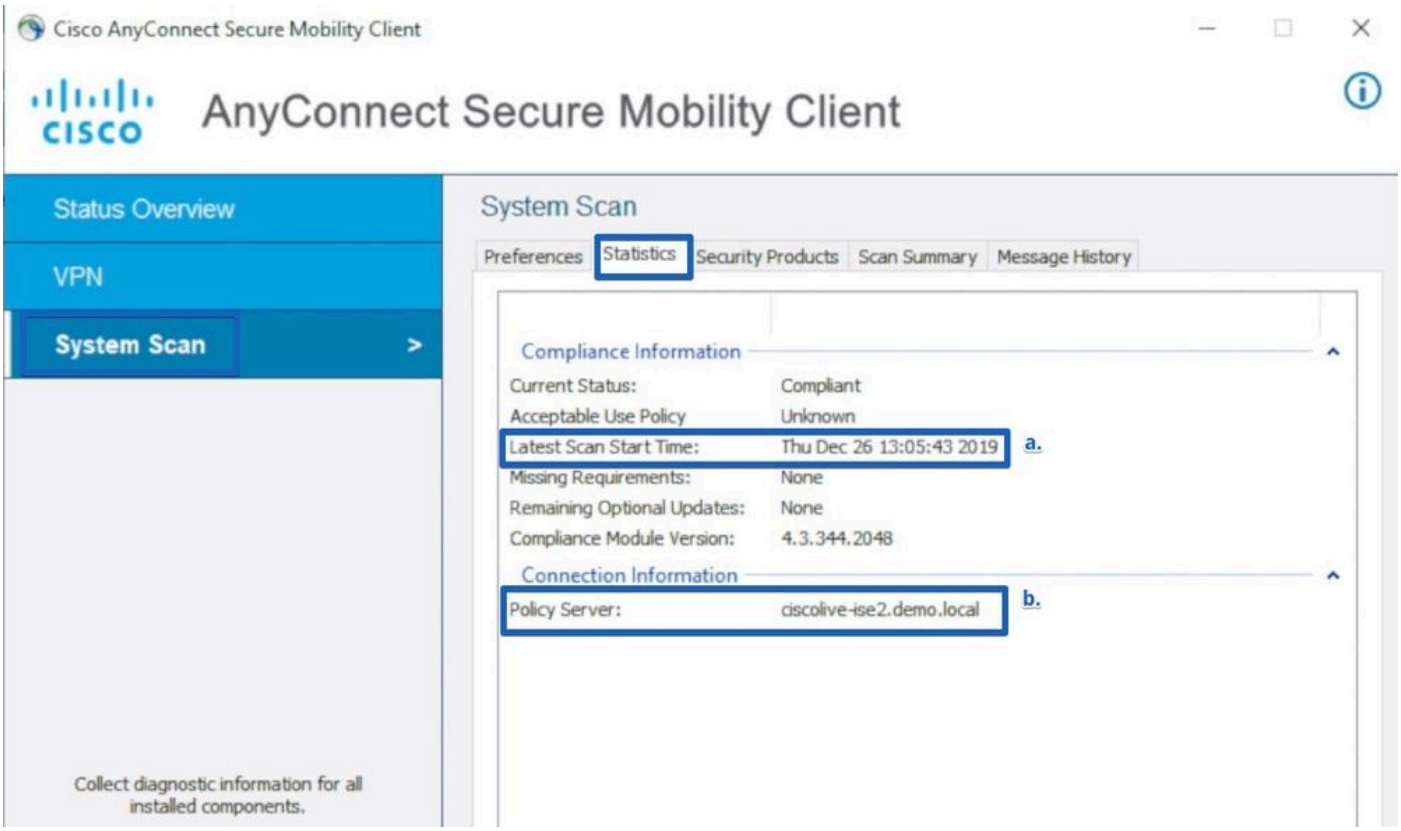
---

1. Click the gear icon in AnyConnect UI.



---

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



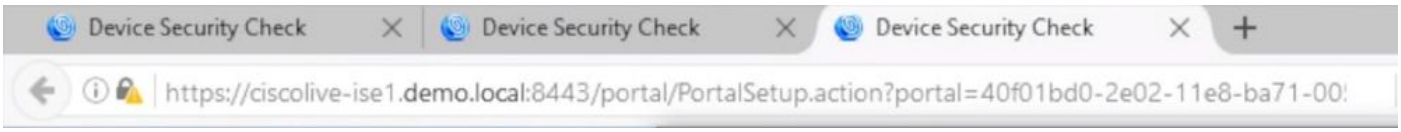
Next, pay attention to two important elements:

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

posture.)

---

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



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

Advance Troubleshoot of Stale/Phantom Session

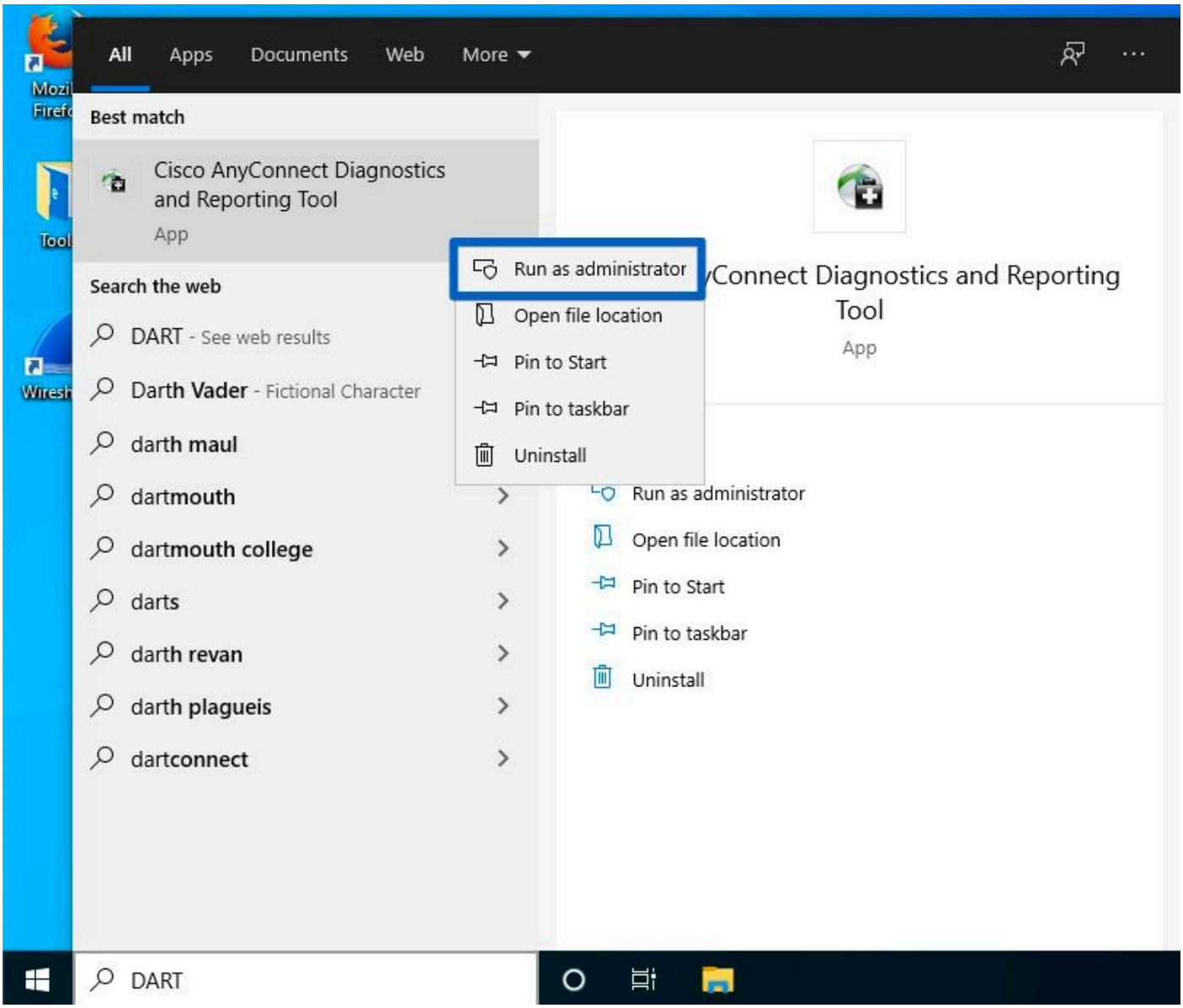
The previous example differentiates the issue of a stale or phantom session from the problem of the discovery process which did not start. At the same time, you must identify the actual session that triggered the problem to understand how it becomes a stale or phantom session issue. While in some scenarios stale and phantom sessions cannot be avoided, you must ensure best practices are implemented, to prevent stale/phantom sessions from being created in an environment.

DART Bundle Collection

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

---

1. Only keep important logs in DART and it is recommended to clear logs before the problem is reproduced.
2. To achieve this, the DART bundle utility must start as an Administrator and perform log clean up.
3. On Windows, navigate to Start and begin to type **DART > right-click, and choose - Run as Administrator**.



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



---

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

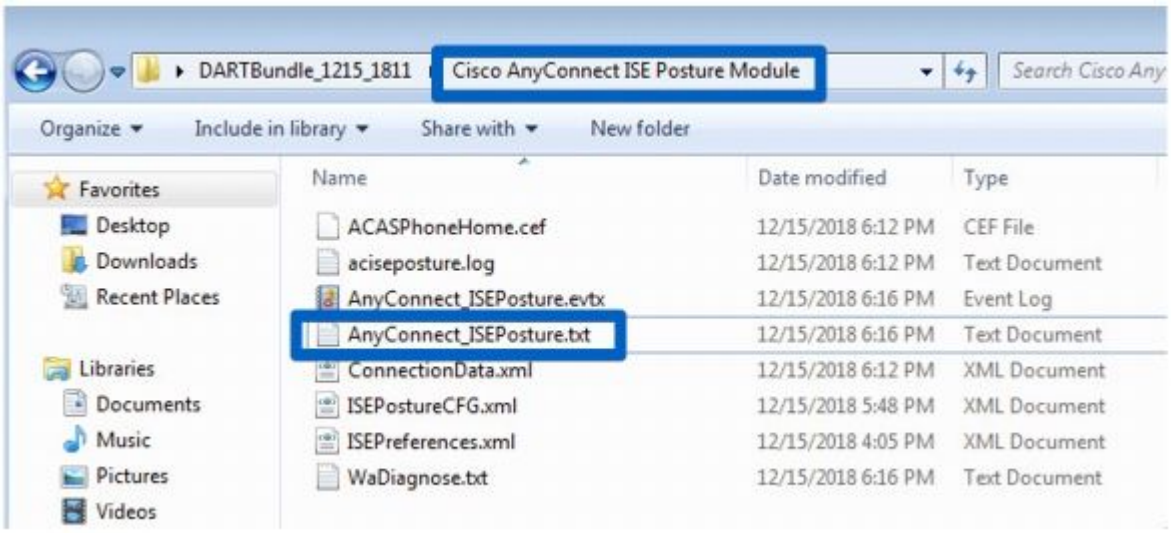


---

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

DART Bundle Analysis

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



---

1. Start troubleshooting and identify all moments of discovery restart. Keywords to search are **Restarting Discovery** or **HTTP Discovery**. Navigate to the line with discovery restart that occurred 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          1251 Level: info  Restarting Discovery.
Line 4214: 2018/12/15 18:00:54 <output  1251 Level: info  Restarting Discovery.
Line 4308: 2018/12/15 18:01:14 omitted> 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 the discovery restart, there is a line that 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,
```

---

3. 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 similar and can be hard to spot the difference.

---

4. Read the log files to see the results for each probe (this is an example of how a 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.>.
```

---

5. 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 the session became 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

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

---

2. PSN receives 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

---

3. The request contains arrays of IP addresses and MAC addresses. In this example, each array holds only one value. The log shows the session ID from the request is null, which indicates 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 process

```
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

---

4. After the line with keywords **Sent http response**, you can view content from the 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

Once the ID of the stale/phantom session is known, you can investigate the Radius Accounting report to gain a better understanding of what caused the session to become stale/phantom:

---

1. 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.

---

2. This is an example of a report that shows how the stale session was 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

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

How to Identify When an Issue was Caused by the Absence of a Discovery Restart

The same logic is applicable for the previous issue, however, the only difference is, you must focus on the Latest Scan Start Time. For this type of problem, the timestamp of the last scan is in the past.

Normally when an end-user discovers a problem, a scan has occured. 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

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

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

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

Solution

Classical Approach - Issue Avoidance

Historically, there were no features on ISE that resolved 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 the ISE side to 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 counter-argument to this recommendation is a bad user experience appears 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 ONLY redirect 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
```

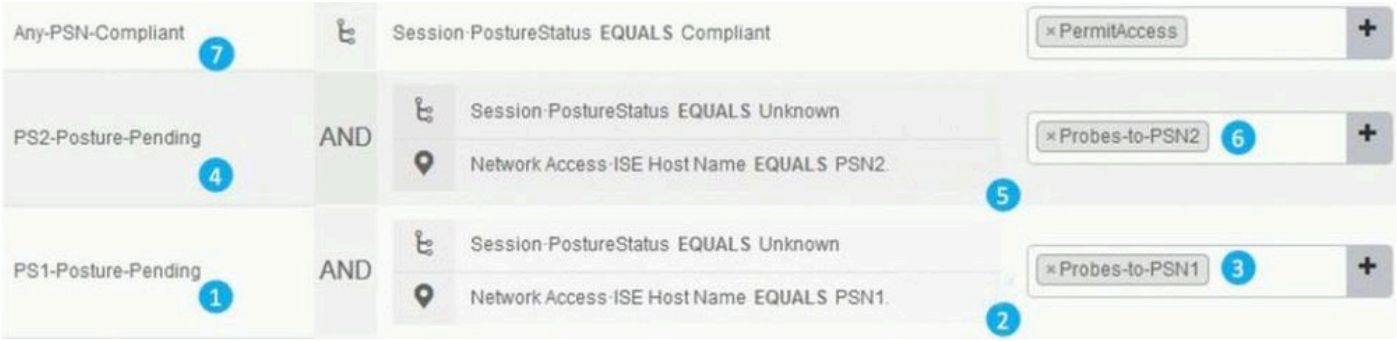
To keep an acceptable level of security, a redirect ACL can be combined with DACL assigned from ISE.

Pending State Allows Connections Only to PSN Where Endpoint was Authenticated

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

As a solution, implement multiple **Posture Pending** authorization policies (one per PSN.) Each policy must contain as one of the conditions, the name of the PSN where authentication took place. In the authorization profile, all PSNs must be blocked except for the node where authentication occurred.

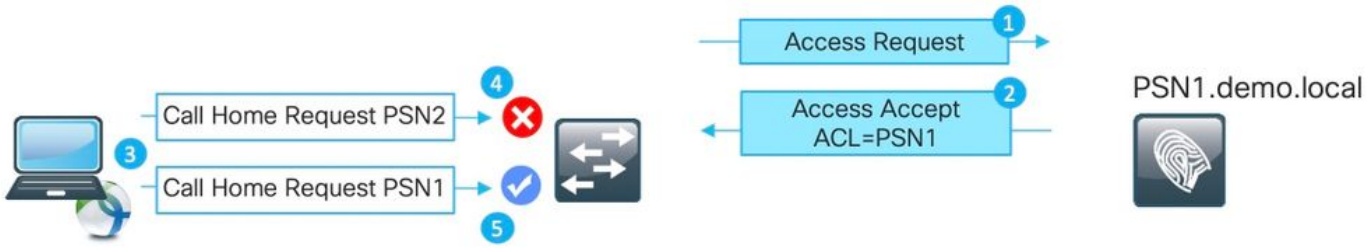
Create authorization policies for two nodes:



---

1. Posture **Pending** policy for PSN1.
2. PSN1 name is used as a condition in the policy.
3. The authorization profile with ACL blocks access to all PSNs except for PSN1.
4. The posture **Pending** policy for PSN2.
5. PSN2 name is used as a condition in the policy.
6. The authorization profile with ACL blocks access to all PSNs except for PSN2.
7. Posture 'Compliant' authorization policy.

The next figure explains how the approach works:



---

1. Authentication hits PSN1.
2. As a result of configured authorization policies, PSN1 assigns the 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. Refer to best practices for ISE with the Cisco & F5 Deployment Guide: ISE Load Balancing Using BIG-IP.
- Use stickiness timer longer than an average work-day to cover the moment when the PC moves into sleep mode (for example 10 hours instead of 8 hours.)
- If reauthentication is implemented, use the reauthentication timer slightly lower than the stickiness timer (use 8 hours if stickiness is configured for 10 hours.) This ensures the stickiness interval is prolonged by reauthentication.

Posture Over VPN Use-case

Ensure 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

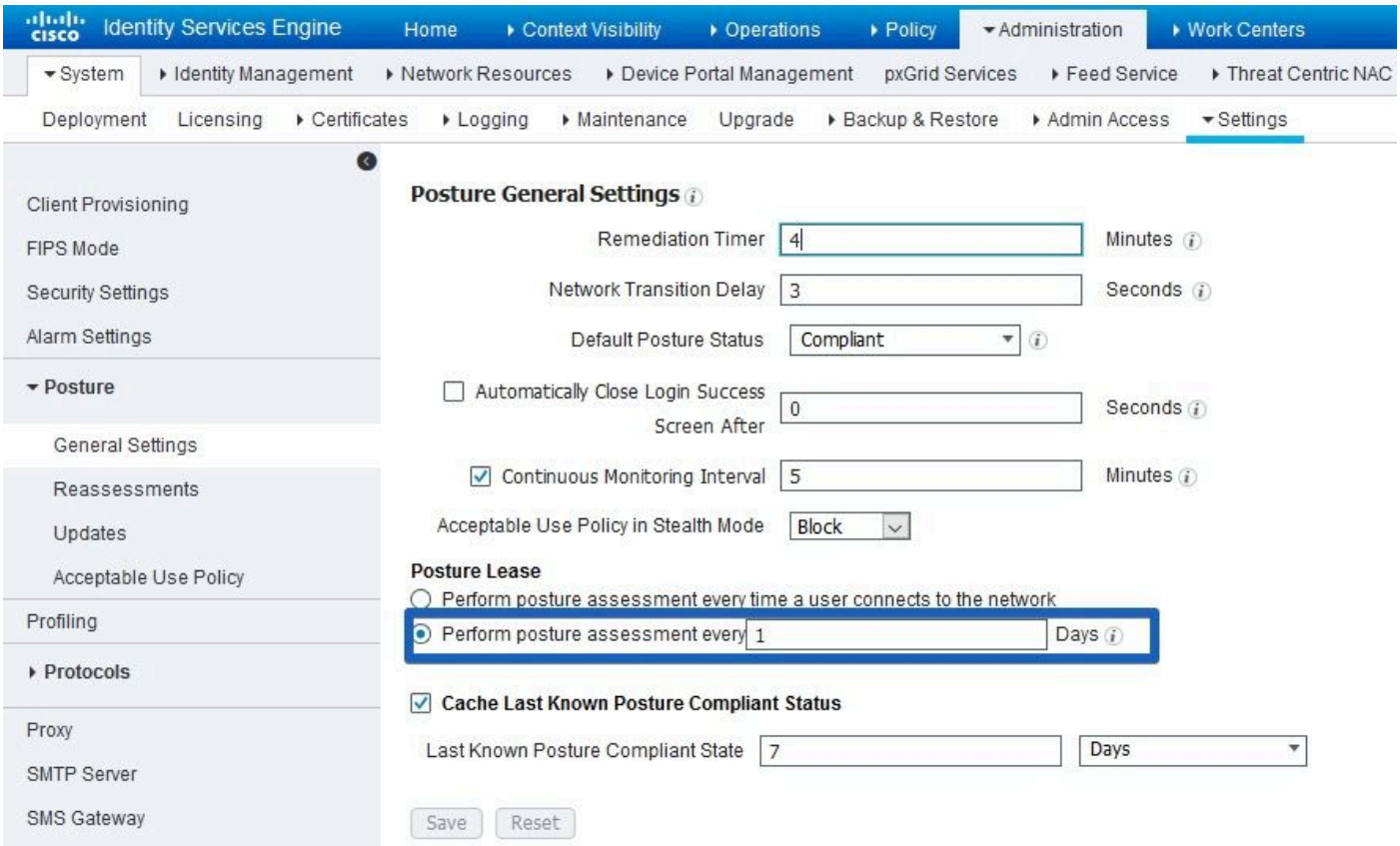
Enable Posture Lease

This is a feature on ISE which marks the endpoint as **compliant** for a defined period (1-365 days.)
The **Posture lease value** is an endpoint attribute which means it is stored on the ISE DB. All endpoint attributes that include the posture lease are replicated across all nodes in the ISE deployment.

When PSN receives a new session for the endpoint, the posture lease can be utilized to mark the session as **Compliant** immediately. To make this decision, PSN uses 3 values and those values are:

---

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



---

2. **Value of PostureExpiry attribute** is an endpoint attribute that 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 is 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

---

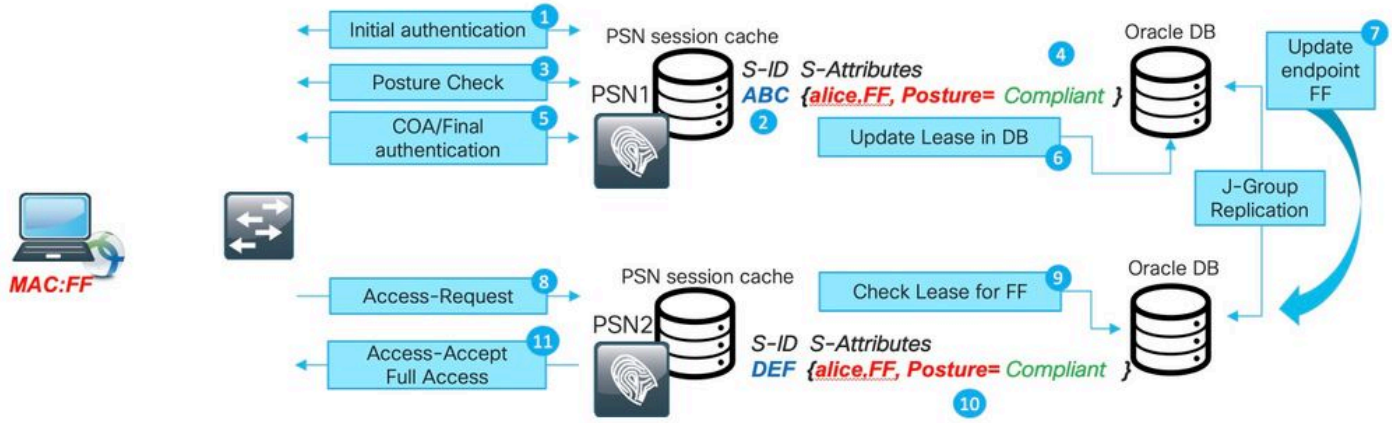
3. This value can be converted into the human-readable timestamp, for example, here - https://www.epochconverter.com/



---

3. When authentication for an endpoint with posture lease hits PSN, it uses PostureExpiry and system date to retrieve 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 receives a **Compliant** status. If the result value is higher than the lease value, the session receies 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:

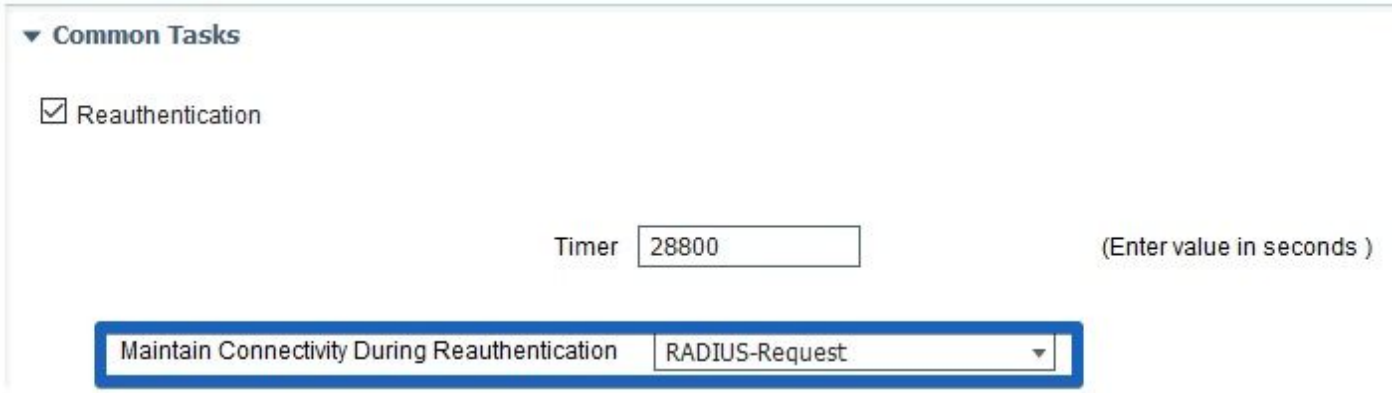


---

1. Initial authentication occurs with PSN1.
2. Session ABC is created in the session cache.
3. Posture assessment takes place.
4. Session status changes to **Compliant**.
5. COA is triggered by posture status change and leads to reauthentication of the endpoint to apply to next access level.
6. The 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.
10. Session is added to the session cache as **Compliant**.
11. Due to the valid lease, the session is created with the posture status of **Compliant**.

Reauthentication Implementation

Always push the reauthentication timer from ISE with the RADIUS-Request, which is selected in **Maintain Connectivity During Reauthentication**. This setting ensures NAD maintains the same session ID on reauthentication.



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 designs provide the opportunity to use different subnets such as **Pending** and **Compliant** states, this approach guarantees every change in the 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. When the original PSN is unavailable, PRA failure restarts the discovery process.

Modern Approach - Posture State Sharing

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

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

This new feature is based on Light Session Directory (LSD) mechanism which was introduced in ISE 2.6. In newer versions, this functionality was 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 are no full session context replications 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 must determine the current session owner. Owner lookup is required when COA flow starts. With LDD, every PSN can find an 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. These 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 happens
  - PSN FQDN
  - NAS-IP-Address
  - NAS-IPv6-Address
  - State - Authenticated, Started, Stopped

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

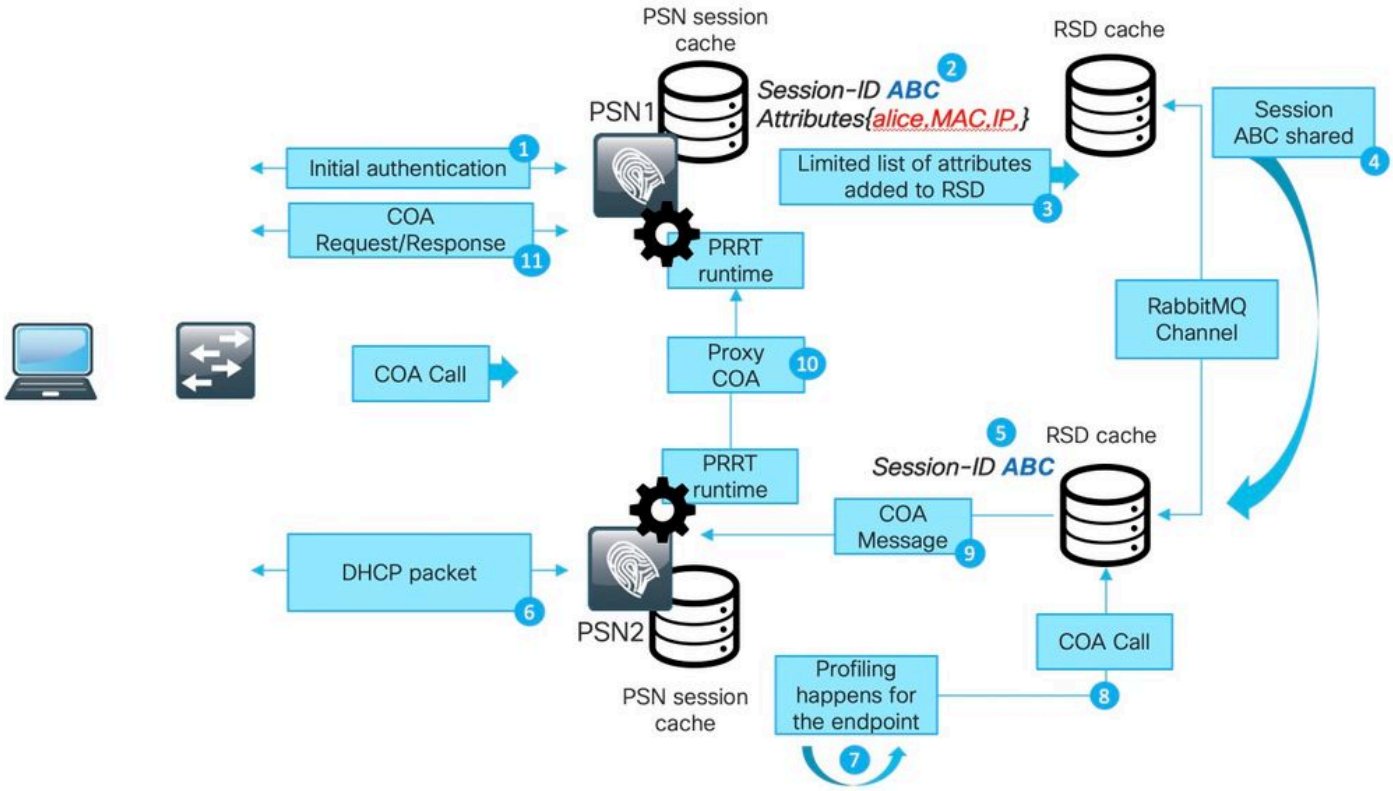
- Publisher: The Radius Session Directory represents a publisher. When a new successful authentication is processed by a PSN, a new session is created in the PSN session cache. For this session, a limited set of attributes is placed into the Radius Session Directory.

- Consumer: On all other nodes, the Radius Session Directory represents a consumer.



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

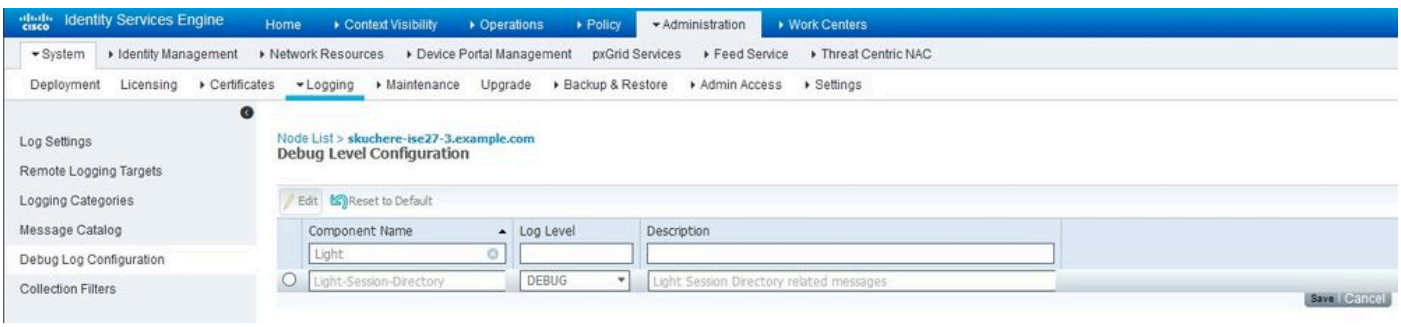
The next example explains how COA flow works with RSD cache:



---

1. Initial authentication occurs with PSN1.
2. Session ABC is created in the session cache.
3. Required attributes are saved in 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 moves from one ISE node to another and contains all details 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-Directory** component into DEBUG:



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

```
DEBUG  [pool-45-thread-6][] cisco.cpm.lsd.service.LSDRedisClient -::::- Mapping Session ID 0a3e94980000
DEBUG  [PrRTEvents-Executor-2][] cisco.cpm.lsd.service.LSDNetAccessEventListener -::::- Publishing sess
DEBUG  [PrRTEvents-Executor-2][] cisco.cpm.lsd.service.SessionPublisher -::::- Forwarding session 07a26
```

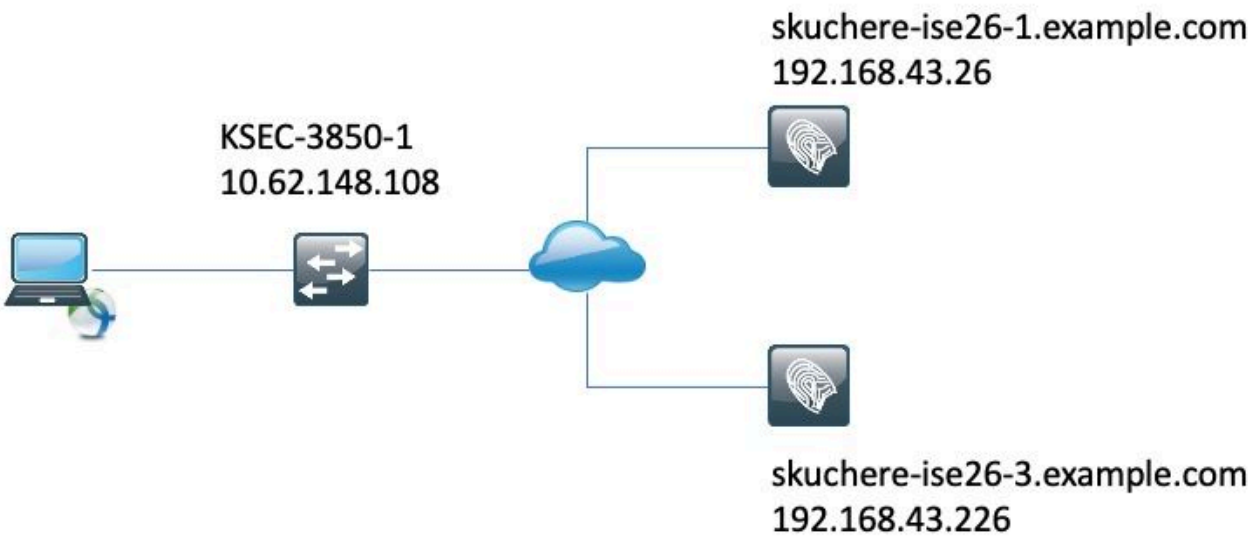
On all other ISE nodes, you see how a session was consumed:

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

Posture Status Sharing Over RSD

Posture status sharing solves issues when the root cause is either a Stale/Phantom session or reauthentication on a different PSN which did not trigger the discovery restart. As soon as the session becomes Compliant, this information places into the session RSD and later can be used by every PSN in the deployment.

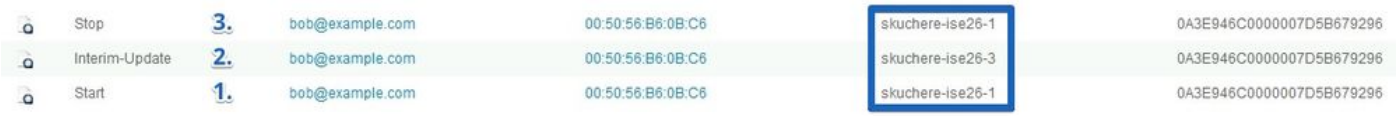
There are some other corner cases the described feature cannot solve. For example, when NAD runs reauthentication on the same PSN but with a different session ID. These scenarios can be handled with best practices described in this document. This figure demonstrates the topology used for a test of posture status sharing:



Posture Status Sharing Over RSD - Stale/Phantom Session

To create a stale session, authentication must be initially performed on **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**.

This image 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.

The endpoint connects to the network, but the redirection no longer works. In the guest.log of PSN for **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 it holds a stale/phantom session for the endpoint, it does not reply to the ISE posture module and this allows you to obtain information from PSN where the latest authentication occurred.

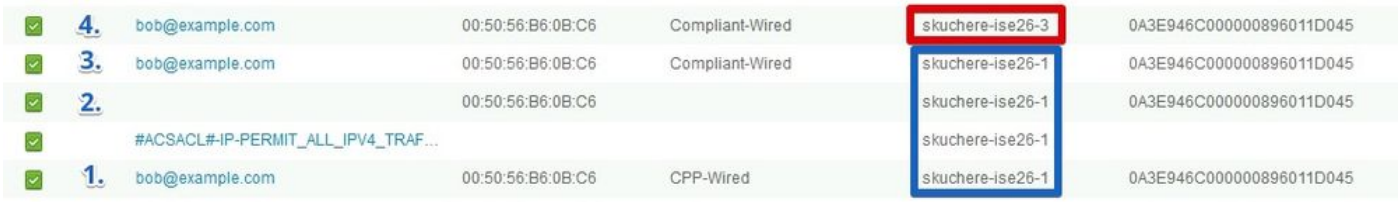
As a solution to the stale/phantom session problem at the time of the session lookup, PSN checks the presence of any new session for the endpoint in the RSD. If RSD contains a different session ID from what PSN has in the local session cache, it assumes the session (presented in the session cache) is stale.

Posture Status Sharing Over RSD - Failover Between PSNs

To reproduce this scenario, a short reauthentication 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 reauthentication timer expiration, the same session is unauthenticated on the different PSN.

The next 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**, the authorization profile with redirection is assigned.
2. COA after successful posture assessment.
3. Next authentication for the authorization profile to the compliant state is assigned.
4. Authentication hits a different PSN, however, it still receives an 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: {}

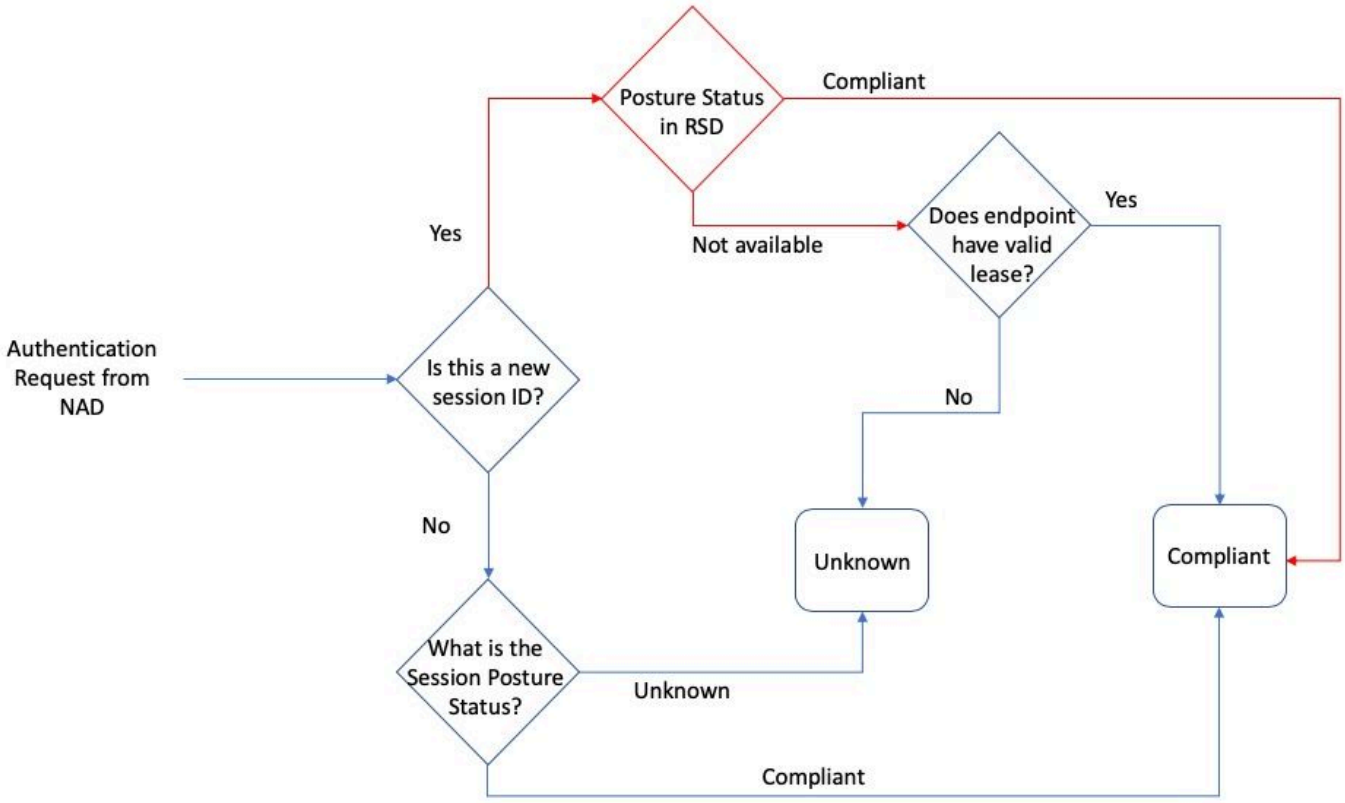
2020-04-09 11:06:42,177 DEBUG [Thread-7979][] cisco.cpm.posture.pip.PostureStatusPIP -::::-

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 resolved with the addition of extra logic into the posture status selection process. This figure demonstrates what was 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