HPE Networking SD-WAN Orchestrator and ClearPass Policy Manager

Integration Guide

Contents

1. Overview 4
2. Prerequisites 4
3. CPPM integration with Orchestrator RADIUS flow 5
4. Initial configuration requirements 6
5. Create an API client on the ClearPass Guest 7
6. Enable Insight in CPPM 8
7. Verify Async Network Services status in CPPM 9
7.1. CPPM cluster considerations 9
8. Configure Endpoint Context Server action 10
8.1. Add CPPM account in the Orchestrator 10
8.2. Pause Orchestration 12
8.3. Manually configure the Endpoint Context Server action 12
9. Prepare CPPM to POST login and logout events to Orchestrator 17
9.1. Create a post-authentication enforcement profile 17
9.2. Identify the ClearPass service used for user authentication 18
9.3. Apply enforcement profile to enforcement policy 18
9.4. Verify services configuration 19
9.5. View user details in CPPM and Orchestrator 19
10. Appendix A: Technical notes 23
10.1. Deep dive into post-authentication actions: CPPM header content 23
10.2. Default content of the login header 23
10.3. Default content of the logout header 23
10.4. Source for CPPM to build payload for POST 23
10.5. Reviewing user information in Access Tracker 24
11. Appendix B: RADIUS packet review 26
11.1. RADIUS request 26
11.2. RADIUS accept 26
11.3. Accounting-Request 26
12. Appendix C: Troubleshooting 28
12.1. Check on Orchestrator 28
12.2. Check on CPPM 28
12.3. Collect logs on CPPM to confirm that the HTTP POST was successful 30

Document Revision History

Project name: HPE Networking SD-WAN Orchestrator and ClearPass Policy Manager Integration Guide

Document status: Revision B

Document version	Date	Section and text revised
Revision B	August 14, 2026	— Full refresh to improve the text. — Added changes relevant to Manual Context server action, cppmDomain, and impact of pausing Orchestration.
Revision A	March 2023	Original document revision.

1. Overview

Integrating HPE Networking ClearPass Policy Manager (CPPM) with HPE Networking SD-WAN Orchestrator provides visibility into user authentication activity occurring in the network. Through this integration, authentication events such as user login and logout are shared with Orchestrator, along with associated user attributes such as username, IP address, role, device type, and location.

In a typical deployment, network clients such as wireless devices authenticate to the network through access infrastructure (for example, access points or switches) using ClearPass as the authentication server. When a client successfully authenticates, ClearPass records the authentication event and associated user information. Similarly, when a user disconnects from the network or the session ends, ClearPass records a logout event.

With this integration enabled, ClearPass sends these login and logout events to Orchestrator. Orchestrator uses this information in two ways:

1. **Authentication event visibility:** Login and logout events received from ClearPass are displayed on the ClearPass Policy Manager tab in Orchestrator, allowing administrators to monitor user authentication activity.
2. **Enhanced flow visibility:** User attributes such as username, role, and IP address are correlated with traffic flows and displayed in Orchestrator (Monitoring > Bandwidth > Flows). This provides additional context when analyzing traffic flows, enabling administrators to identify which user and role are associated with a particular flow.

Note

This integration enhances operational visibility and troubleshooting by providing user-level context for network traffic flows. While this information is visible within Orchestrator, it is not used by EdgeConnect appliances to enforce identity-based security or routing policies.

To build identity-aware policies based on username or user role, see the RADIUS Snooping Guide.

CPPM establishes a secure HTTPS WebSocket connection with Orchestrator to transmit user authentication events and associated user attributes. For this integration to function, RADIUS authentication traffic between network authenticators and ClearPass does not need to traverse the EdgeConnect appliances.

2. Prerequisites

1. Reachability between the CPPM Server and the Orchestrator.
2. Users and devices authenticated by CPPM using an existing ClearPass service policy.
3. Access to CPPM and the ability to configure API client and other profile or policy configurations in CPPM.
4. Any authenticator (access point, switch, wireless controller, or other network device) that sends RADIUS packets to CPPM must send a RADIUS accounting packet that includes the standard IETF RADIUS attribute `Framed-IP-Address`.
5. Access to an Orchestrator with admin credentials.
6. IP address or FQDN details of the CPPM.
7. IP address or FQDN details of the Orchestrator.

3. CPPM integration with Orchestrator RADIUS flow

CPPM integration with Orchestrator enables CPPM to do an HTTP POST of user information to Orchestrator. In this method, RADIUS packets do not have to flow through EdgeConnect appliances. Simply ensure that the CPPM is the authenticating server for wired/wireless devices. The authenticator (switch/access point/Aruba controller) sending the RADIUS packets to CPPM must support RADIUS accounting with the `Framed-IP-Address` attribute included in the accounting packets.

Notes

— ClearPass needs the `Framed-IP-Address` (user IP address) attribute to build the HTTP payload that will be sent to Orchestrator. Without this information, CPPM will not send the user details to Orchestrator.

— This integration requires RADIUS accounting support. As a result, it is not supported when EdgeConnect operates as the RADIUS authenticator using the Network Access Control (NAC) feature, as RADIUS accounting information is not currently supported in EdgeConnect.

Figure 1 shows three different topologies where users authenticate and RADIUS packets do not necessarily flow through an EdgeConnect appliance.

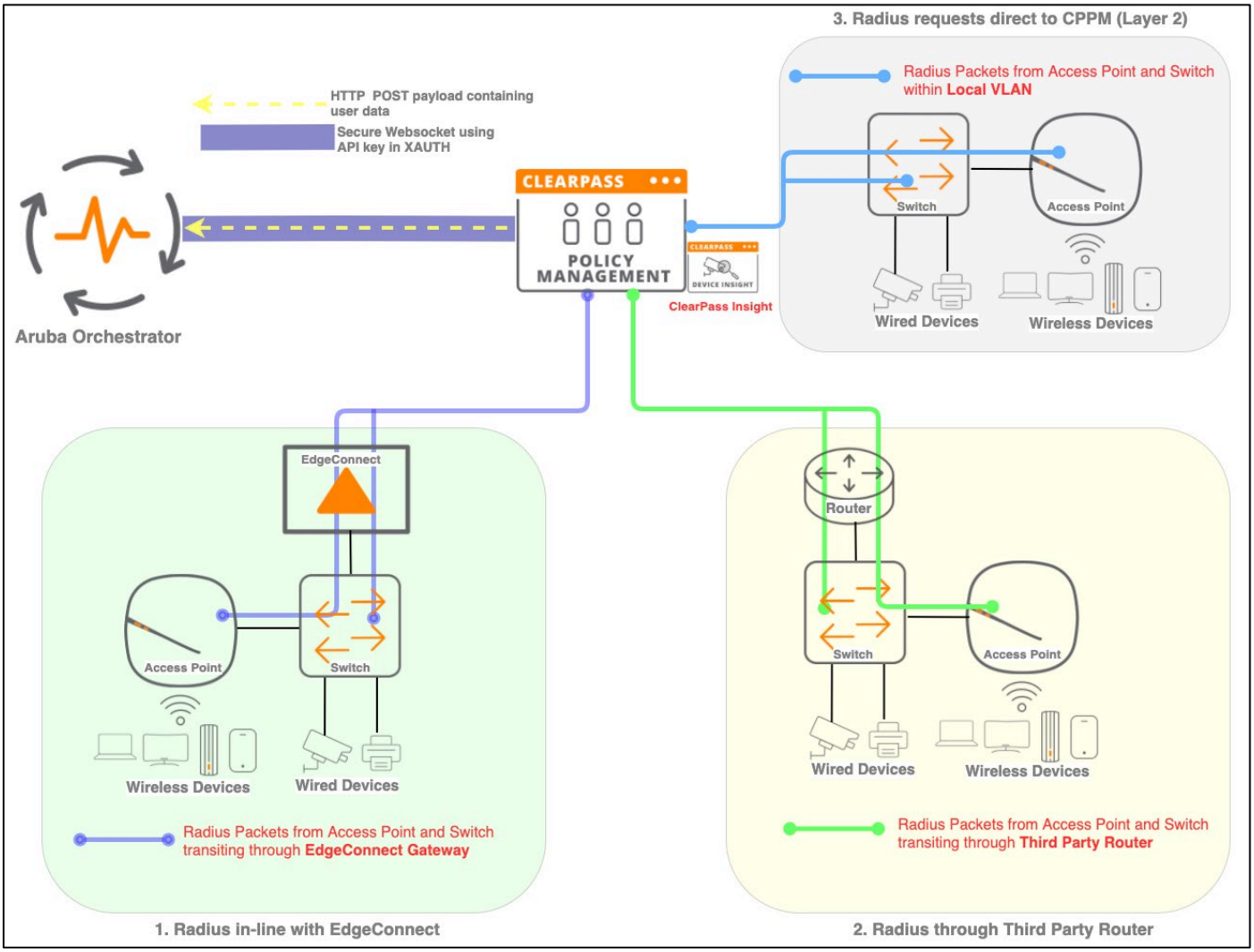


Figure 1: RADIUS flows in various environments feeding ClearPass Insight.

4. Initial configuration requirements

To enable the integration between CPPM and Orchestrator, certain configuration steps must be completed on both systems. While Orchestrator can automatically push some context server action configurations to CPPM, this automation alone is not sufficient to establish a fully functional integration. Without completing the additional prerequisites in CPPM, login and logout events may not be transmitted correctly, resulting data not being displayed on the CPPM integration page.

The key configuration tasks that must be completed include:

— Creating an API client in the ClearPass Guest.

— Enabling Insight in CPPM.

— Verifying that the Async Network Services process is running in CPPM.

— Configuring the Endpoint Context Server action in CPPM.

The following sections provide step-by-step guidance to configure the required settings on both CPPM and Orchestrator to ensure that the integration works as intended.

Note

Creating an API client on ClearPass is only required when using automated configuration from Orchestrator.

In deployments where Orchestrator is hosted as a cloud service (OaaS) and inbound firewall access to CPPM is not permitted, the integration relies on manual configuration of the Endpoint Context Server action in CPPM. In such cases, customers can skip Section 5 (Create an API client on the ClearPass Guest) and proceed directly to Section 6: Enable Insight on CPPM.

5. Create an API client on the ClearPass Guest

To allow Orchestrator to automatically configure the Endpoint Context Server action in CPPM, an API client must be created on the CPPM server.

This API client allows Orchestrator to securely authenticate and interact with CPPM through its REST APIs. The credentials generated during this step are required when adding the CPPM account in Orchestrator.

1. Log in to the CPPM administrative interface, and then navigate to **ClearPass Guest > Administration > API Services > API Clients**.

   Note

   API clients can only be created from the ClearPass Guest management interface.

2. Click **Create API Client**, and then add Orchestrator as the API client by providing the following information:
   a. **Client ID:** Enter a unique string.
   b. **Enabled:** Select the check box.
   c. **Operating Mode:** Select **ClearPass REST API - Client will be used for API calls to ClearPass**.
   d. **Operator Profile:** Select **Super Administrator**.
   e. **Grant Type:** Select **Client credentials (grant_type=client_credentials)**.

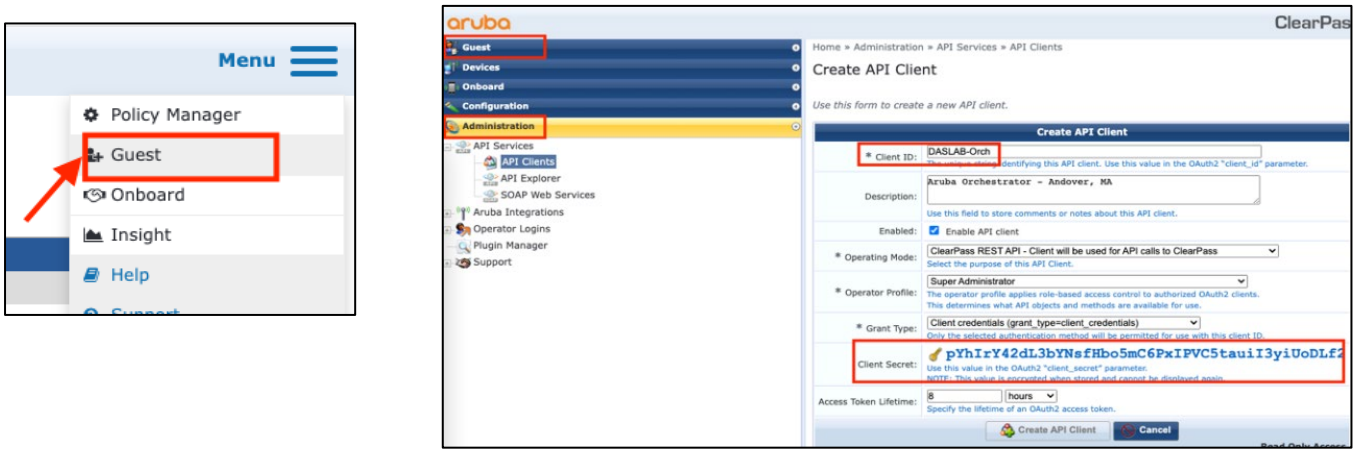


Figure 2: Creating an API client with a secret key on ClearPass Guest.

3. Note the following values:
   a. Client ID
   b. Client Secret
4. Click **Create API Client**.

Important

Copy and securely store the secret key when it is displayed. The secret key will not be visible again after the API client is created.

These credentials will be used in Orchestrator when adding the CPPM server account for the integration.

6. Enable Insight in CPPM

User Insight is required for CPPM to generate the authentication event information to shares with Orchestrator. In a CPPM cluster deployment, Insight services may run on either a Publisher node or a Subscriber node.

Note

This setting must be configured in the Policy Manager interface, not in the ClearPass Guest interface.

1. Log in to the CPPM administrative interface.
2. Navigate to **Administration > Server Configuration > System**.
3. In the Insight Setting field, ensure that the **Enable Insight** and **Enable as Insight primary server** check boxes are selected.

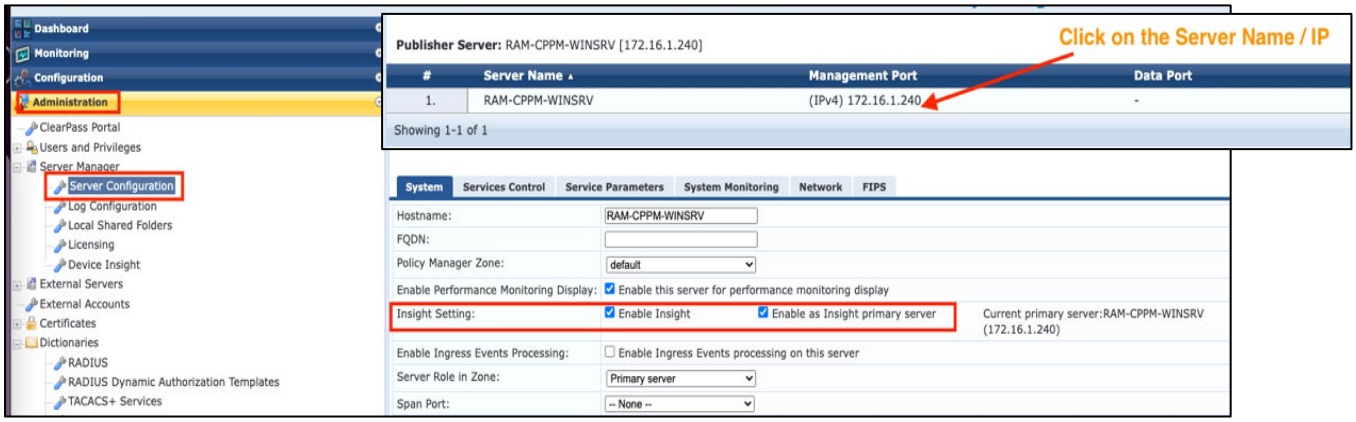


Figure 3: Enabling CPPM Insight.

7. Verify Async Network Services status in CPPM

For the Endpoint Context Server action to function correctly, the Async Network Services process must be running in CPPM. This process is responsible for processing and forwarding event notifications, including the login and logout events sent to Orchestrator. If the Async Network Services process is not running, CPPM will not generate or send the required authentication event notifications.

1. Navigate to **Administration > Server Manager > Server Configuration**.

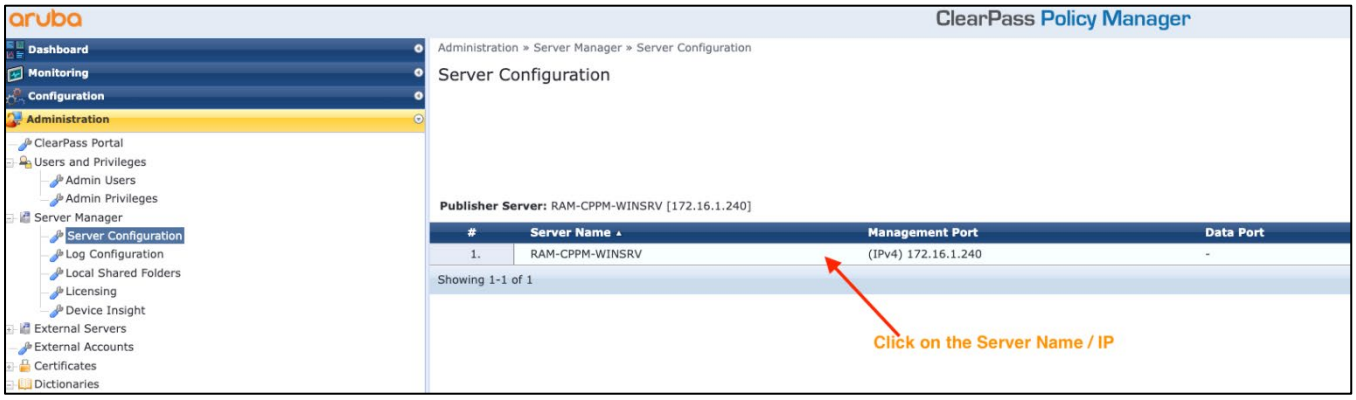


Figure 4: Navigating to the CPPM Server Configuration page.

2. If the service is not running, start the Async Network Services process.
3. Click the **Services Control** tab, and then verify that Async network services appears as *Running*.

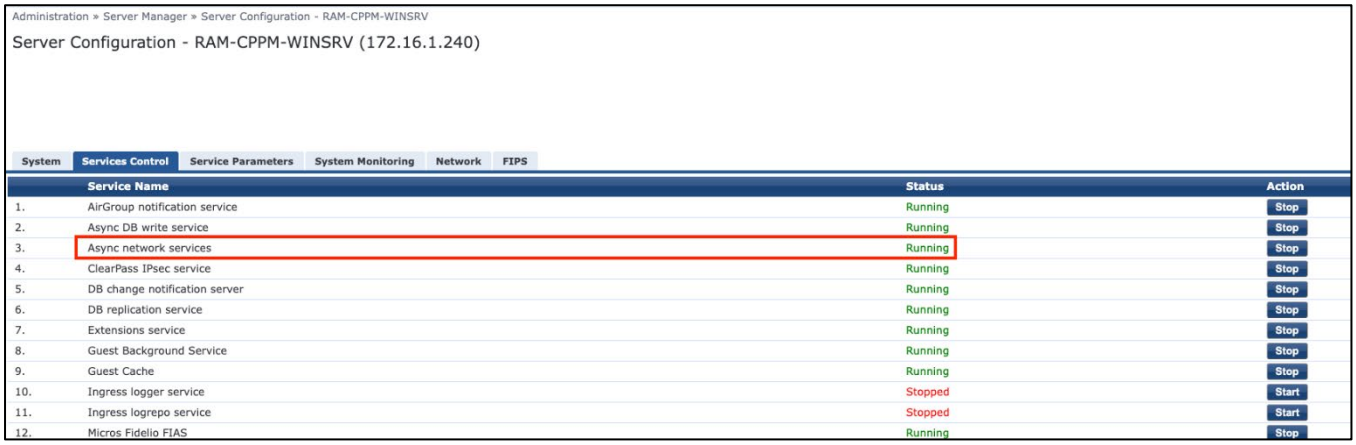


Figure 5: Verifying the status of Async network services.

7.1. CPPM cluster considerations

In a CPPM cluster deployment, Async Network Services typically runs on nodes responsible for processing authentication and event notifications.

Ensure that Async Network Services is running on the CPPM node responsible for generating Insight events. In most deployments, this is the same node where Insight services are enabled.

8. Configure Endpoint Context Server action

Configuring the Endpoint Context Server action in CPPM is a key requirement of this integration. This configuration allows CPPM to send user login and logout events to Orchestrator using an HTTP-based notification mechanism.

The Endpoint Context Server action defines the parameters required for CPPM to send authentication event data to Orchestrator. This includes:

— Orchestrator IP address or hostname

— Connection method (HTTP/HTTPS)

— Authentication method

— XAUTH token used for authentication

— User context attributes included in the event payload (such as username, role, and IP address)

When a login or logout event occurs, CPPM sends this information to Orchestrator using an HTTP POST request based on the parameters defined in the Endpoint Context Server action.

There are two supported methods to configure the Endpoint Context Server action in CPPM:

1. **Method 1: Automated configuration in Orchestrator**

   Orchestrator can automatically create the required Context Server Action in CPPM using an API-based configuration push.

   This method requires that Orchestrator can reach the CPPM server over TCP port 443 (HTTPS). In environments where CPPM is deployed behind a firewall, or when Orchestrator is hosted as a cloud service (OaaS), the firewall must allow inbound HTTPS (TCP 443) access from Orchestrator to the CPPM server.

   To use this method, you need to complete only the steps in Add CPPM account in the Orchestrator. Orchestrator then automatically creates and configures the Endpoint Context Server action in CPPM.

2. **Method 2: Manual configuration in CPPM**

   In some deployments, CPPM may be hosted on-premises while Orchestrator is deployed as a cloud service (OaaS). In these environments, you may prefer not to allow inbound firewall access from the public cloud to the internal CPPM server.

   For these scenarios, the Endpoint Context Server action must be configured manually on the CPPM server. This approach avoids the need for inbound connectivity from Orchestrator to CPPM, while still allowing CPPM to send authentication events to Orchestrator.

   To use this method, you must complete both of the following steps:

   - Add CPPM account in the Orchestrator
   - Manually configure the Endpoint Context Server action

8.1. Add CPPM account in the Orchestrator

1. Log in to Orchestrator, and then navigate to **Orchestrator > HPE Integration Services > ClearPass Policy Manager**.

Name: Provide a name for this account

Domain/IP: <IP/FQDN>

Client ID: <Provide Client ID obtained from CPPM>

Secret Key: <Secret key created obtained from CPPM>

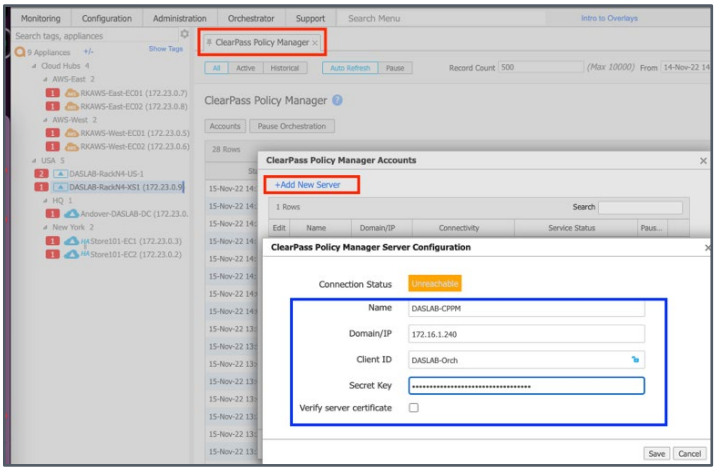


Figure 6: Adding CPPM as a new server in Orchestrator.

2. Click **Accounts**.
3. Click **+Add New Server**.
4. Enter a name, IP (see below), client ID, and secret key.

   For the Domain/IP filed, specify either the IP address or the Fully Qualified Domain Name (FQDN) of the CPPM server.

   a. In most lab or test environments, using the IP address is common.

   b. In production deployments, if a DNS entry is available, you can use the FQDN instead.

Important

Ensure that you use the same value (IP address or FQDN) consistently across the configuration. This value must match the one used in the Endpoint Context Server action JSON payload (when configuring it manually in CPPM).

Using inconsistent values—such as using an IP address in one configuration and an FQDN in another—may prevent Orchestrator from correctly correlating incoming login and logout events from CPPM. As a result, user data may not be displayed on the CPPM integration page.

---

5. Click **Save**.

   The Connection Status should appear as *Connected*.

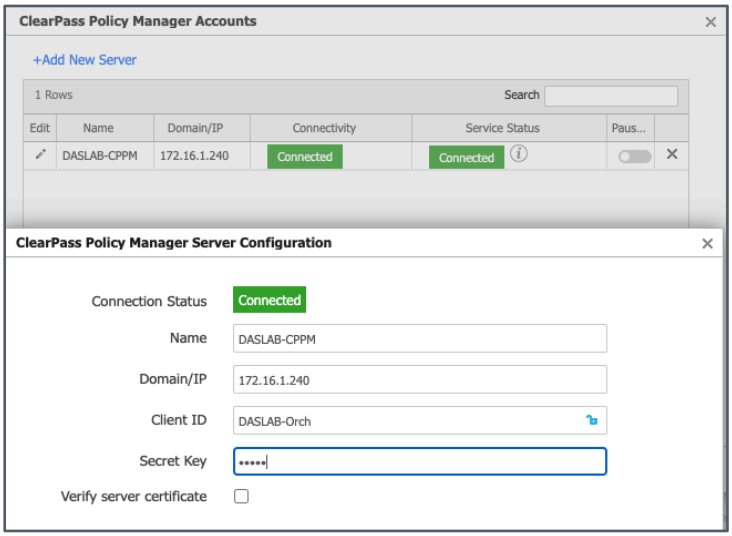


Figure 7: CPPM connection status appearing as *Connected*.

After the CPPM server is added, Orchestrator will automatically:

— Authenticate with CPPM using the provided API credentials.

— Create the required Endpoint Context Server action.

— Configure the necessary parameters, including:

- Orchestrator connection details
- Authentication (XAUTH token)
- User context attributes (such as username, role, and IP address)

Note

If there is a firewall or other connectivity restriction to reaching CPPM, this status may appear as *Unreachable*. To resolve this, fix the firewall or connectivity issues. If this restriction is due to expected firewall limitations, then it is acceptable for the Connection Status to remain unreachable. Proceed to Manually configure the Endpoint Context Server action.

8.2. Pause Orchestration

You can pause orchestration by clicking the **Pause Orchestration** button on the ClearPass Policy Manager tab in Orchestrator.

When this option is enabled, the Orchestrator temporarily stops processing HTTP POST messages related to user login and logout events. The ClearPass Policy Manager tab will not display any user event details while orchestration is paused.

Normal operation will resume only after orchestration is restarted.

8.3. Manually configure the Endpoint Context Server action

8.3.1. Generate an API key in Orchestrator

1. Log in to Orchestrator and navigate to **Orchestrator > Orchestrator Server > User & Authentication > API Keys**.
2. Click **New API Key**.
3. Configure the following:
   a. **Key Name:** Enter a descriptive name (for example, *clearpass-api*).
   b. **Permission:** Select **Read-Write**.

      Expiration: Set an expiration for the API key. Alternatively, clear the expiration if you prefer not to renew API keys frequently).
   c. **Active:** Select **Yes**.

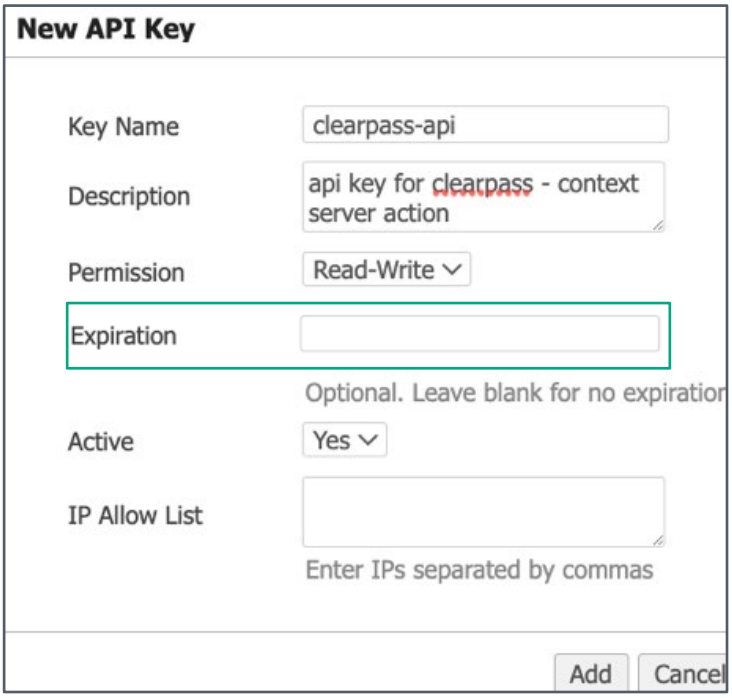


Figure 8: Adding a new API key in Orchestrator.

4. Click **Add**.
5. Click **Copy Key to Clipboard** and securely store the key to a text file. This API key will be required during CPPM configuration in the next step.

8.3.2. Add Orchestrator as a context server in CPPM

1. Log in to CPPM, and then navigate to **Administration > External Servers > Endpoint Context Servers**.
2. Click **Add**.
3. Configure the following settings:
   a. **Server Type:** Select **Generic HTTP Context Server**.
   b. **Server Name:** Enter the IP address or FQDN of the Orchestrator.
   c. **Server Base URL:** This auto-populates based on the server name.

   Note

   Use either an IP address or an FQDN consistently across all configurations. Mixing both may prevent proper communication between CPPM and Orchestrator.

4. Click **Save**.

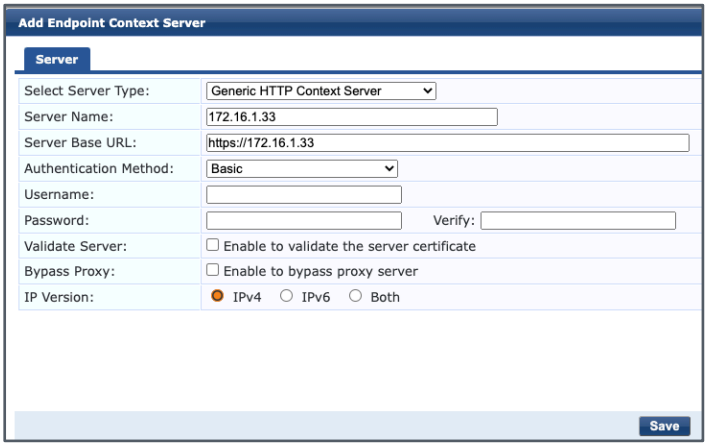
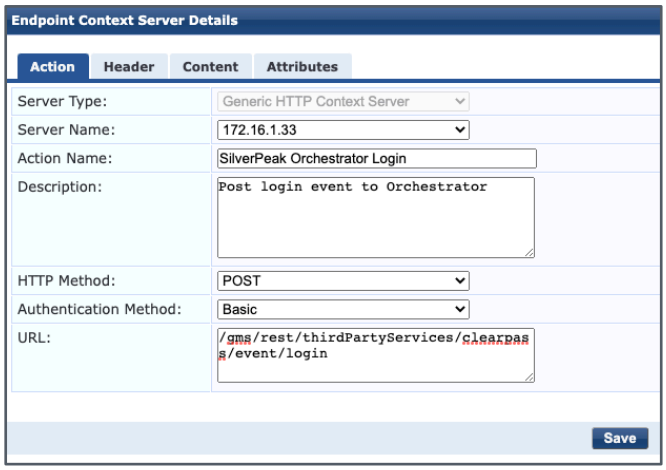


Figure 9: Adding Orchestrator as an HTTP context server in CPPM.

8.3.3. Configure context server actions

8.3.3.1. Create login action

1. Navigate to **Administration > Dictionaries > Context Server Actions**.
2. Click **Add**.
3. Configure a Login Context Server action by filling in the Action, Header, and Content tabs as per your deployment requirements. For reference, refer to Figure 10, Figure 11, and Figure 12.
4. Click **Save**.



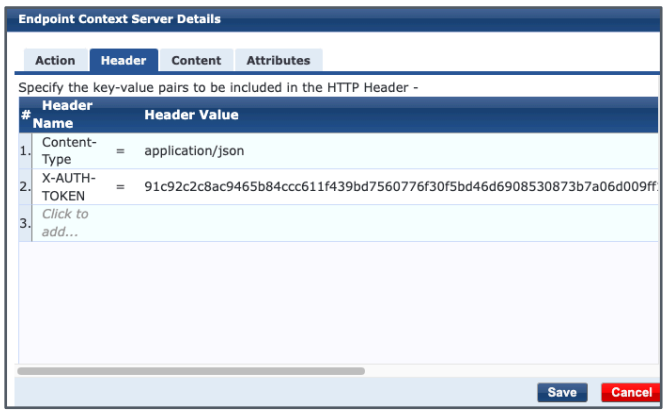
Server Type: Generic HTTP Context Server

Server Name: Select the Orchestrator IP from the drop down

Action Name: SilverPeak Orchestrator Login

URL: /gms/rest/thirdPartyServices/clearpass/event/login

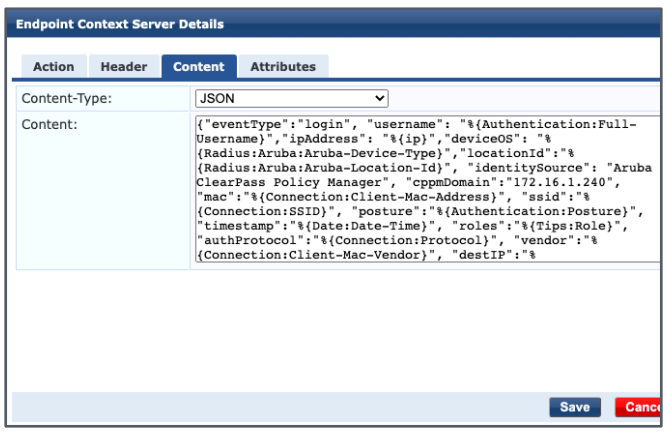
Figure 10: Endpoint Context Server – Action/Login



Content-Type: application/json

X-AUTH-TOKEN: <API key copied from Orchestrator>

Figure 11: Endpoint Context Server – Header/Login



Content-Type: JSON

Content:
```
{"eventType":"login", "username":
"%{Authentication:Full-Username}","ipAddress":
"%{ip}","deviceOS": "%{Radius:Aruba:Aruba-
Device-
Type}","locationId":"%{Radius:Aruba:Aruba-
Location-Id}", "identitySource": "Aruba
ClearPass Policy Manager",
"cppmDomain":"172.16.1.240",
"mac":"%{Connection:Client-Mac-Address}",
"ssid":"%{Connection:SSID}",
"posture":"%{Authentication:Posture}",
"timestamp":"%{Date:Date-Time}",
"roles":"%{Tips:Role}",
"authProtocol":"%{Connection:Protocol}",
"vendor":"%{Connection:Client-Mac-Vendor}",
```

Figure 12: Endpoint Context Server – Content/Login

Note

In the JSON content in Figure 12, the `cppmDomain` attribute must exactly match the value configured in the Domain/IP field in Add CPPM account in the Orchestrator. Using a different value—such as an IP address in one place and a domain name in another—may prevent the Orchestrator from recognizing incoming login or logout events from CPPM, resulting in data not being displayed on the integration page.

8.3.3.2. Create logout action

1. Click **Add**.
2. Configure a Logout Context Server action. Use similar settings as the Login action, with appropriate modifications for logout events. For reference, refer to Figure 13, Figure 14, and Figure 15.
3. Click **Save**.

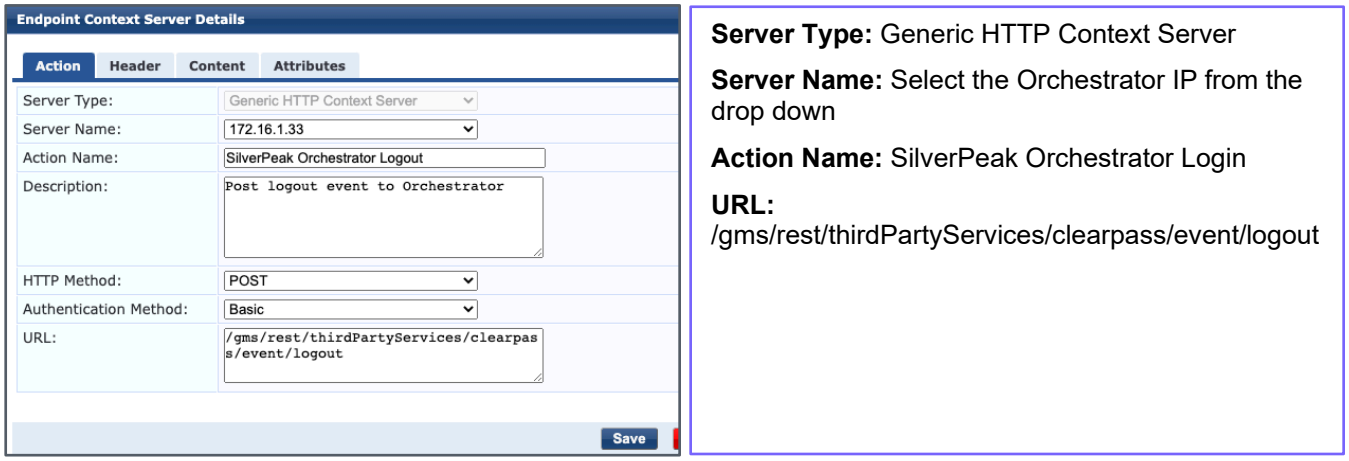


Figure 13: Endpoint Context Server – Action/Logout

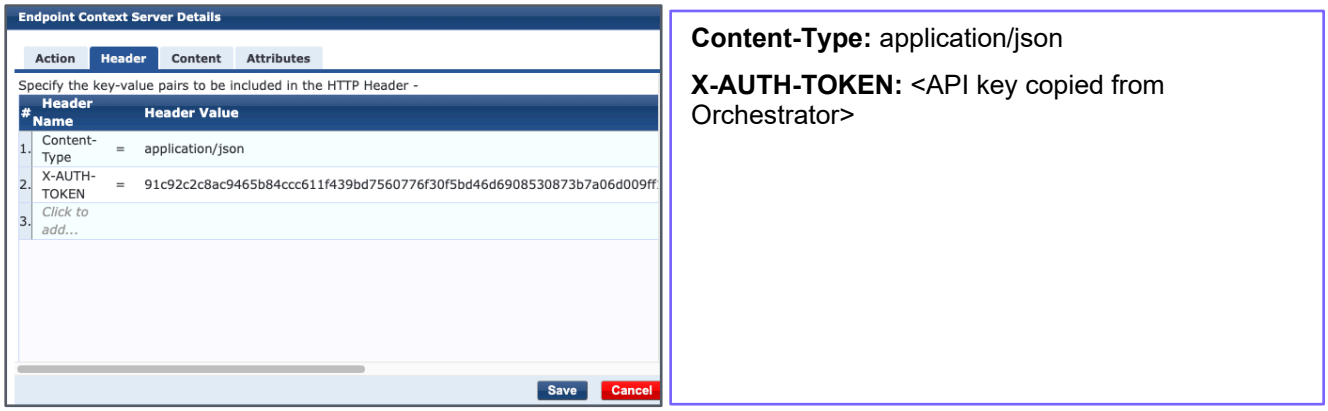
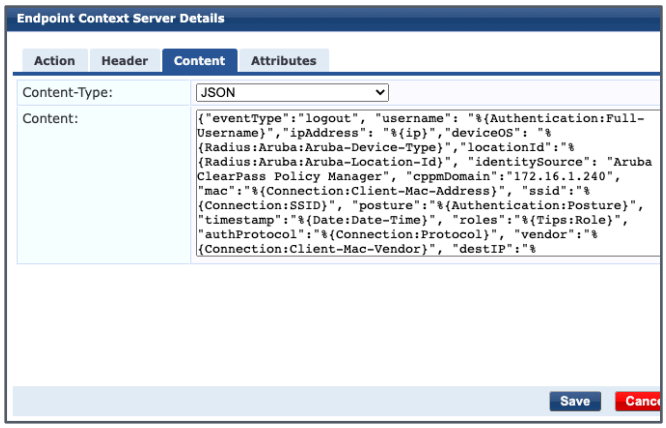


Figure 14: Endpoint Context Server – Header/Logout



Content-Type: JSON

Content:

```
{"eventType":"logout", "username":
"%{Authentication:Full-Username}","ipAddress":
"%{ip}","deviceOS": "%{Radius:Aruba:Aruba-
Device-
Type}","locationId":"%{Radius:Aruba:Aruba-
Location-Id}", "identitySource": "Aruba
ClearPass Policy Manager",
"cppmDomain":"172.16.1.240",
"mac":"%{Connection:Client-Mac-Address}",
"ssid":"%{Connection:SSID}",
"posture":"%{Authentication:Posture}",
"timestamp":"%{Date:Date-Time}",
"roles":"%{Tips:Role}",
"authProtocol":"%{Connection:Protocol}",
"vendor":"%{Connection:Client-Mac-Vendor}",
```

Figure 15: Endpoint Context Server – Content/Logout

Note

In the JSON content in Figure 15, the `cppmDomain` attribute must exactly match the value configured in the Domain/IP field in Add CPPM account in the Orchestrator. Using a different value—such as an IP address in one place and a domain name in another—may prevent the Orchestrator from recognizing incoming login or logout events from CPPM, resulting in data not being displayed on the integration page.

9. Prepare CPPM to POST login and logout events to Orchestrator

After configuring the required Endpoint Context Server actions, you must ensure that these actions are properly invoked within a ClearPass service.

To achieve this, the Endpoint Context Server actions must be associated with an enforcement profile applied as part of the ClearPass service configuration.

Important

Invoking Endpoint Context Server actions within the ClearPass service configuration is essential. This is what enables CPPM to POST user login and logout event details to the Orchestrator.

In the following steps, you will:

— Create an enforcement profile that includes the Endpoint Context Server actions.

— Attach this enforcement profile to an existing service used for user and device authentication.

9.1. Create a post-authentication enforcement profile

In this step, you will create an enforcement profile that triggers the required Endpoint Context Server actions. The profile will also include attributes such as Server IP and Server Type.

1. Log in to CPPM, and then navigate to **Configuration > Enforcement > Profiles**.
2. Click **Add**.
3. Create a new enforcement profile:
   a. **Template:** Select **Session Notification Enforcement**.
   b. Include all required attributes as captured in Figure 16.

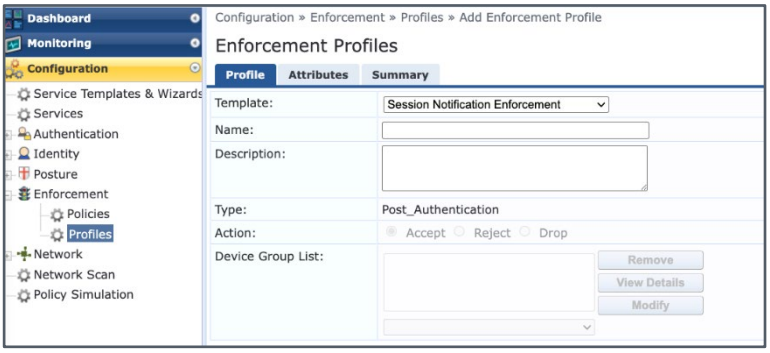


Figure 16: Creating an enforcement profile in CPPM.

   c. Update the Server IP to match your Orchestrator IP address/FQDN.

Figure 17: Viewing the Enforcement Profiles Summary tab.

9.2. Identify the ClearPass service used for user authentication

An enforcement profile does not take effect until it is mapped to an enforcement policy within an existing ClearPass service.

To enable CPPM to POST user information to Orchestrator after authentication, you must first identify which service is handling wired/wireless authentication in your environment. You can then modify that service's enforcement policy to include the newly created enforcement profile, which contains the post-authentication actions.

In the example below, multiple ClearPass services are configured. This example uses the *DASLAB-1X-CPPM* service for modifying an enforcement policy.

1. Navigate to **Configuration > Services**.
2. Select the service that needs to be modified—in this example, **DASLAB-1X-CPPM**.
3. View the enforcement policy attached to this service—in this example, *RK – Radius Policy*.

   The next step associates the newly created enforcement profile.

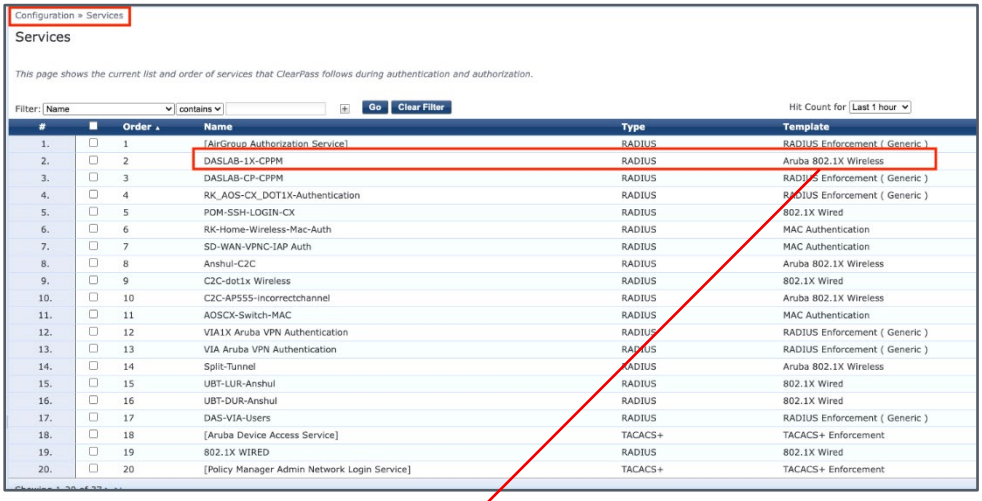


Figure 18: Viewing CPPM services.

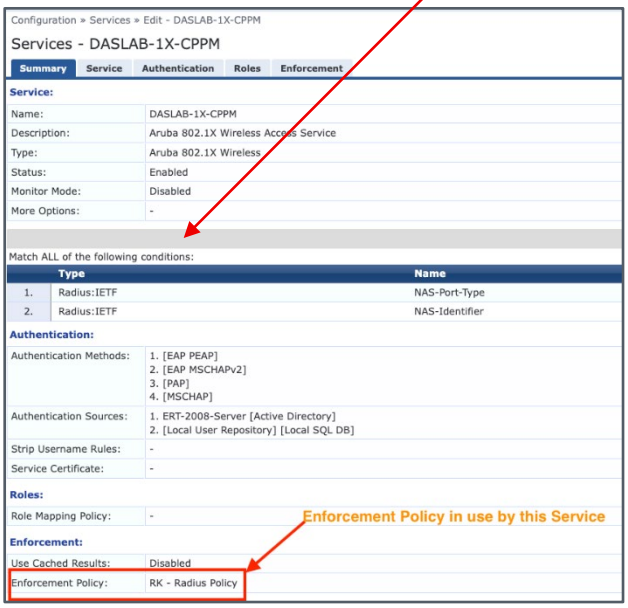


Figure 19: Viewing details of a service policy.

9.3. Apply enforcement profile to enforcement policy

The previous section identified the relevant service and, more specifically, the enforcement policy in use. This step modifies this enforcement policy to include the newly created enforcement profile.

The *RK – Radius Policy* referenced in Figure 19 performs two functions:

— Returns the `Aruba-User-Role` attribute as part of the RADIUS accept response. (Not required for this integration, but included in this example.)

— Applies the post-authentication enforcement profile to enable CPPM to perform HTTP POSTs to the Orchestrator.

To update the policy:

1. Navigate to **Configuration > Enforcement > Policies**.
2. Locate your existing enforcement policy, and then click **Edit Rule**.
3. Add the newly created enforcement profile. ***Do not remove*** any profiles already in use.

Note

If you are unsure which policy to modify or which enforcement policy is actively used for user/device authentication, contact your CPPM administrator before making changes.

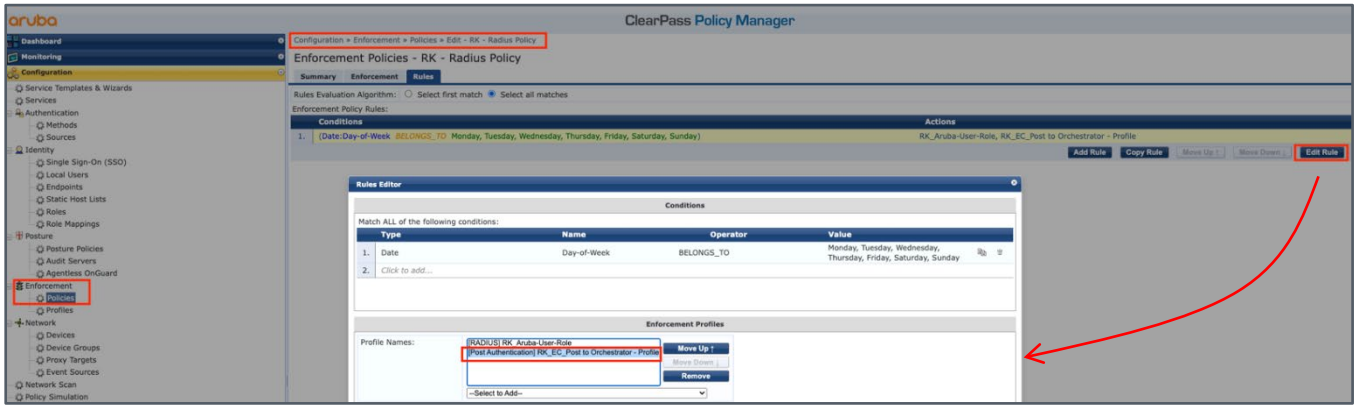


Figure 20: Associating the enforcement profile to the enforcement policy.

9.4. Verify services configuration

To view the enforcement profiles applied to a service policy, navigate to **Configuration > Services**, select a service, and then click the **Enforcement** tab.

In Figure 21, the service *DASLAB-1X-CPPM* uses the enforcement profile *RK_EC_Post to Orchestrator*, which was configured earlier to perform the POST action after authentication.

Figure 21: Verifying that the enforcement policy is part of services.

9.5. View user details in CPPM and Orchestrator

After the enforcement profile for posting login and logout events to the Orchestrator has been created and applied to the ClearPass service policy, the configuration must be validated.

To do this, simulate a user authentication event and verify that the user details are successfully transmitted to the Orchestrator.

1. Perform a user authentication that matches the configured ClearPass service policy.

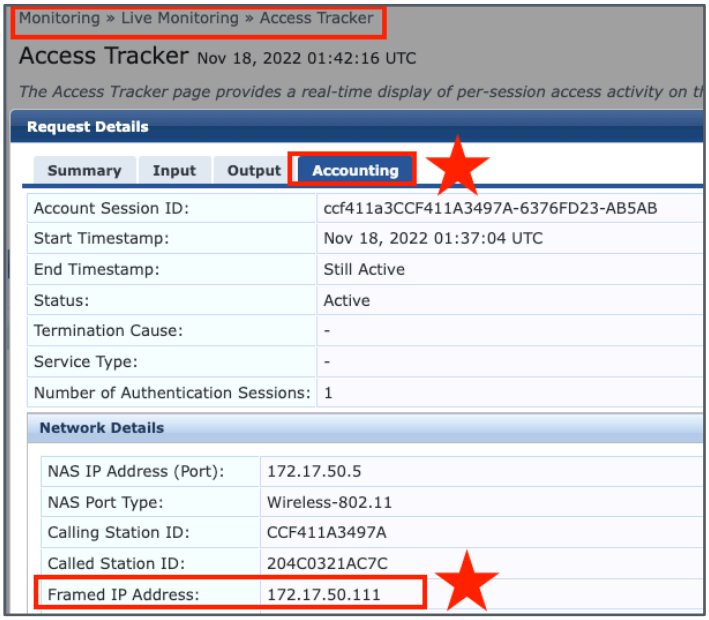


Figure 22: Viewing the Access Tracker – Accounting tab.

2. Verify that the user or device is successfully authenticated by navigating to **Monitoring > Live Monitoring > Access Tracker** and confirming that the IP address and other session details are visible.

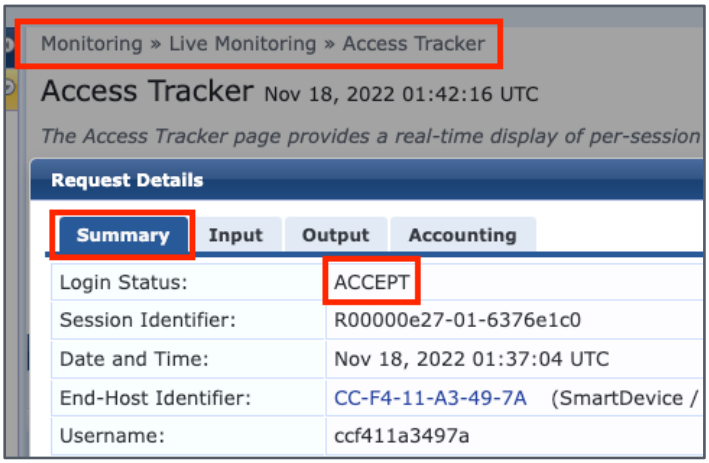


Figure 23: Verifying successful authentication via the CPPM Access Tracker.

3. Log in to Orchestrator, navigate to **Orchestrator > HPE Integration Services > ClearPass Policy Manager**, and then verify that the corresponding user login and logout event information is displayed.

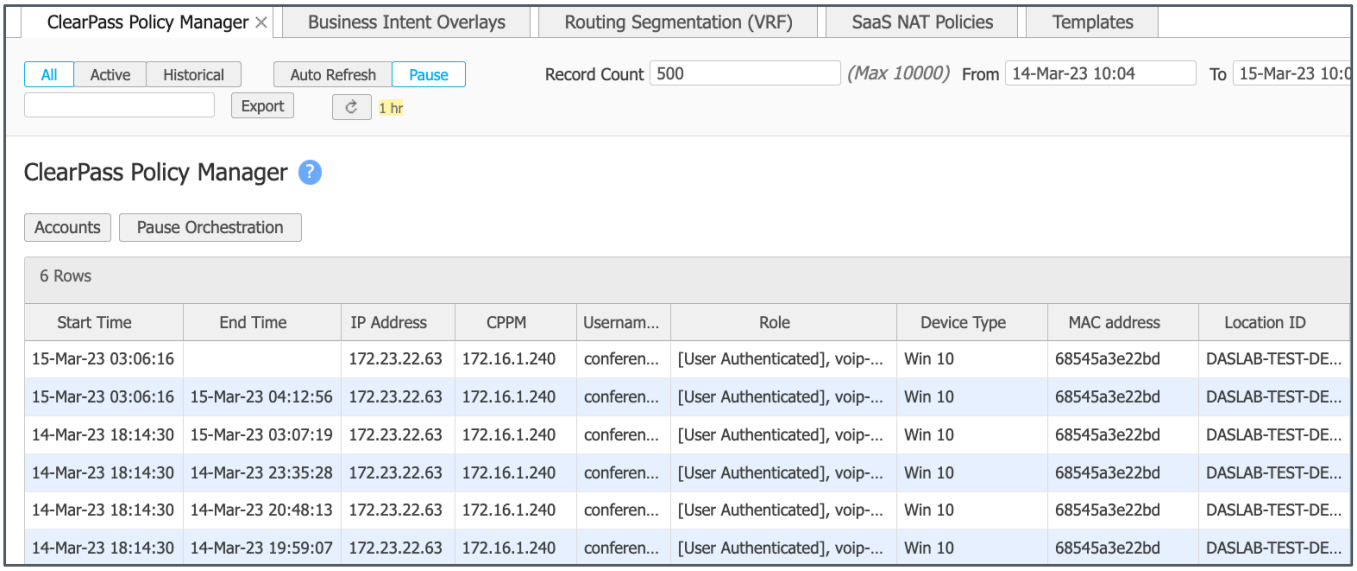


Figure 24: Viewing user details on the ClearPass Policy Manager tab in Orchestrator.

Note

The ClearPass Policy Manager tab occasionally shows raw variable names instead of actual values. Typically, this behavior occurs when CPPM does not have sufficient data available to populate all fields.

When CPPM sends context to Orchestrator, the username, IP address, and user role are the primary attributes expected to be populated. Other fields included as part of the Endpoint Context Server action are optional.

If CPPM has additional client information available through Insights, it includes those values as well. Otherwise, it is expected behavior for the UI to display the variable name itself rather than a resolved value.

Alternatively, view detailed user information on the Flow details dialog box by navigating to **Monitoring > Bandwidth > Flows**, and then clicking the **i** icon in the Details column of the specific flow for which user details were sent by CPPM.

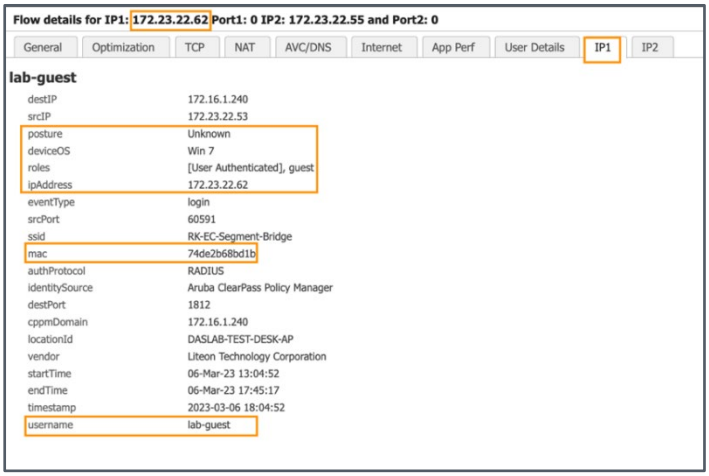


Figure 25: Viewing flow details for IP1 (source).

In the example Flow details dialog box seen in Figure 25, the IP1 (source) section shows the user details associated with the source IP address.

If both the source and destination IP addresses are authenticated by the same CPPM, the IP2 tab shows user details such as user role, username, MAC address, and other associated metadata for the destination IP.

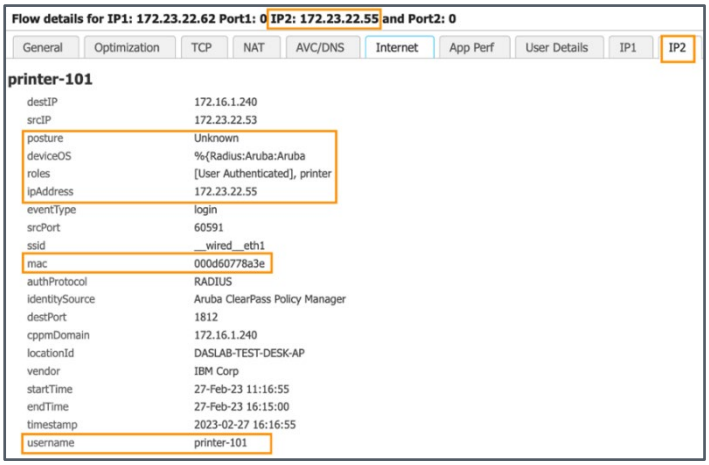


Figure 26: Viewing flow details for IP2 (destination).

This concludes the configuration required for CPPM integration with Orchestrator. The subsequent appendices are included to provide additional troubleshooting guidance and deeper insight into the integration behavior.

10. Appendix A: Technical notes

10.1. Deep dive into post-authentication actions: CPPM header content

This section provides a deeper look at how user information is packaged by CPPM before being transmitted to Orchestrator.

By default, header content includes several key attributes such as username, IP address, MAC address, and others. While these keys must remain intact, the corresponding values can be customized based on your requirements.

For example:

— **Key:** username

— **Value:** %{name}

This can be modified to use a different attribute, such as:

— **Key:** username

— **Value:** %{Authentication:Full-Username}

Note

Modifying these values requires a solid understanding of CPPM attribute handling and the available variable mappings. To avoid unexpected behavior, ensure that any changes align with your authentication and policy configuration.

10.2. Default content of the login header

```
{"eventType":"login", "username": "%{name}","ipAddress": "%{ip}","deviceOS": "%{Radius:Aruba:Aruba-Device-Type}","locationId":"%{Radius:Aruba:Aruba-Location-Id}", "identitySource": "Aruba ClearPass Policy Manager", "cppmDomain":"172.16.1.240", "mac":"%{Connection:Client-Mac-Address}", "ssid":"%{Connection:SSID}", "posture":"%{Authentication:Posture}", "timestamp":"%{Date:Date-Time}", "roles":"%{Tips:Role}", "authProtocol":"%{Connection:Protocol}", "vendor":"%{Connection:Client-Mac-Vendor}", "destIP":"%{Connection:Dest-IP-Address}", "srcIP":"%{Connection:Src-IP-Address}", "srcPort":"%{Connection:Src-Port}", "destPort":"%{Connection:Dest-Port}"}
```

10.3. Default content of the logout header

```
{"eventType":"logout", "username": "%{name}","ipAddress": "%{ip}","deviceOS": "%{Radius:Aruba:Aruba-Device-Type}","locationId":"%{Radius:Aruba:Aruba-Location-Id}", "identitySource": "Aruba ClearPass Policy Manager", "cppmDomain":"172.16.1.240", "mac":"%{Connection:Client-Mac-Address}", "ssid":"%{Connection:SSID}", "posture":"%{Authentication:Posture}", "timestamp":"%{Date:Date-Time}", "roles":"%{Tips:Role}", "authProtocol":"%{Connection:Protocol}", "vendor":"%{Connection:Client-Mac-Vendor}", "destIP":"%{Connection:Dest-IP-Address}", "srcIP":"%{Connection:Src-IP-Address}", "srcPort":"%{Connection:Src-Port}", "destPort":"%{Connection:Dest-Port}"}
```

10.4. Source for CPPM to build payload for POST

To gather the information required to construct the payload for the HTTP POST request, CPPM leverages the Insight application.

Policy Manager Insight is an application within CPPM that aggregates data from one or more Policy Manager appliances, including archived network access logs. This aggregated data serves as the foundation for building the payload sent to Orchestrator.

For more details, see the ClearPass Insight documentation.

10.5. Reviewing user information in Access Tracker

The integration fundamentally mirrors the information displayed in CPPM's Access Tracker for user login and logout events (limited to authenticated users/devices) from the ClearPass Policy Manager tab in Orchestrator.

When troubleshooting missing data in the Orchestrator, the source of truth is always the CPPM Access Tracker. Verify that the expected user information is present in the Access Tracker. Most importantly, confirm that the Accounting tab includes the user's IP address.

It is important to note that if the Accounting tab does not display the user's IP address, CPPM cannot send the user data to the Orchestrator. Therefore, accounting must be properly supported and configured on your network devices, and the user's IP address must be visible in the Access Tracker for the integration to function correctly.

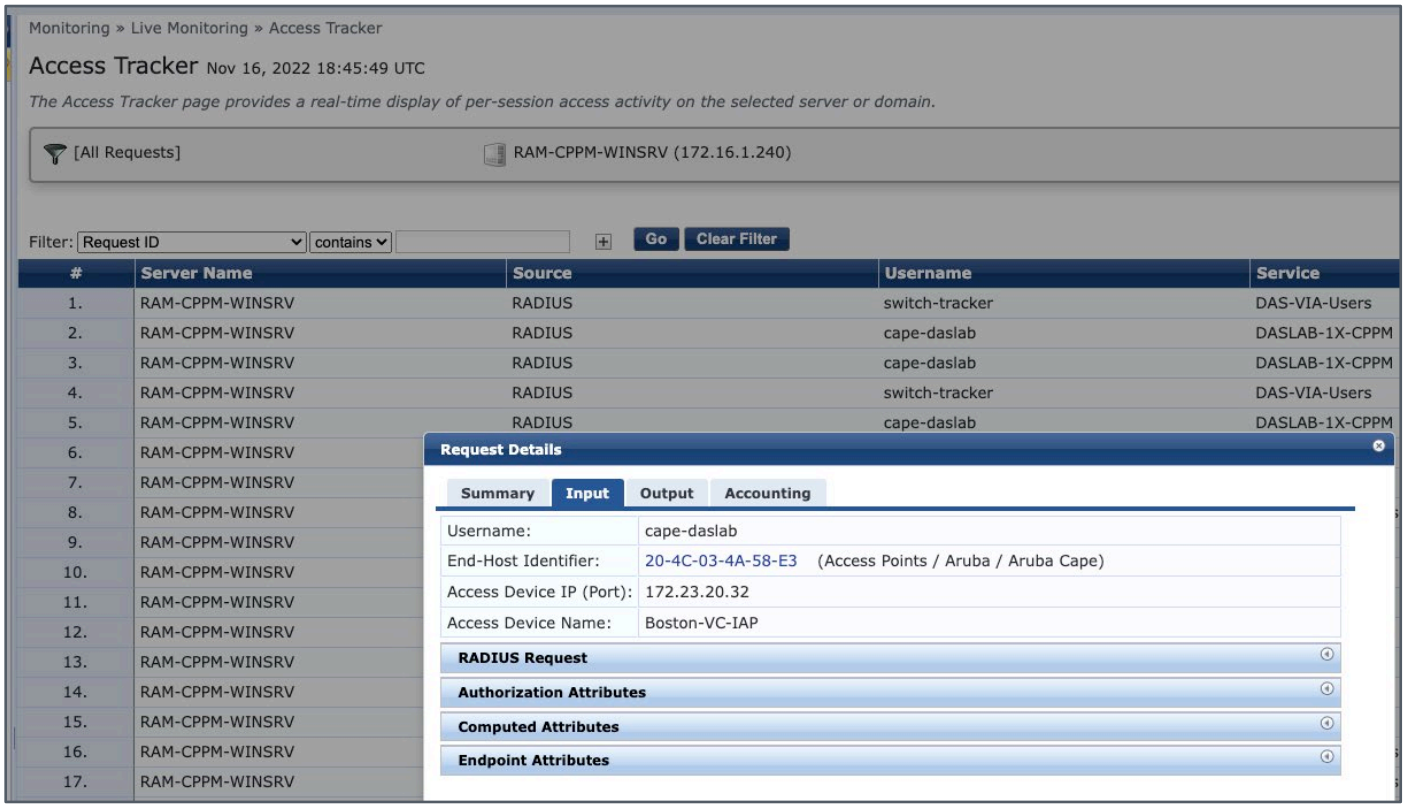


Figure 27: Viewing the Request Details dialog box in the CPPM Access Tracker.

10.5.1. RADIUS request

Attribute	Details
Radius:Aruba:Aruba-Device-Type	OS X
Radius:Aruba:Aruba-Essid-Name	RK-EC-Segmentation
Radius:Aruba:Aruba-Location-Id	AP555-RackN2
Radius:IETF:Calling-Station-Id	204c034a58e3
Radius:IETF:User-Name	cape-daslab

10.5.2. Computed attributes

Attribute	Details
Authentication:Username	cape-daslab
Connection:AP-Name	AP555-RackN2
Connection:Client-Mac-Address	204c034a58e3
Connection:Client-Mac-Vendor	Aruba, a Hewlett Packard Enterprise Company
Connection:Dest-IP-Address	172.16.1.240

Attribute	Details
Connection:Dest-Port	1812
Connection:NAD-IP-Address	172.23.20.32
Connection:Protocol	RADIUS
Connection:Src-IP-Address	172.23.20.32
Connection:Src-Port	56198
Connection:SSID	RK-EC-Segmentation
Date:Date-Time	2022-11-15 19:00:17

10.5.3. Endpoint attributes

Attribute	Details
MAC Vendor	Aruba, a Hewlett Packard Enterprise Company
Added by	Policy Manager
Status	Unknown
Device Category	Access Points
Device OS Family	Aruba
Device Name	Aruba Cape
MAC Address	204c034a58e3
IP Address	172.23.20.33
Hostname	cnhcklq0r4
Profiler Conflict	true
Other Category	Monitoring Devices
Other Family	Aruba
Other Name	User Experience Insight Sensor

10.5.4. Fingerprint details

Attribute	Details
DHCP Option55	["1,28,2,3,15,6,119,12,44,47,26,121,42,252"]
DHCP Option60	["Aruba-Sensor-Q9X65A"]
DHCP Options	["53,54,50,12,55,60"]
Host MAC Vendor	["Aruba, a Hewlett Packard Enterprise Company"]

11. Appendix B: RADIUS packet review

To better understand how information is populated in CPPM and Insight, review the actual RADIUS packets exchanged between the authenticators (switches/access points) and CPPM. These packets provide the foundational data used for authentication, authorization, and accounting.

11.1. RADIUS request

The RADIUS request packet is the initial authentication request sent by the authenticator to CPPM. It contains key attributes such as username, device MAC address, Aruba AP location ID, device type, and other identifying information related to the user or endpoint.

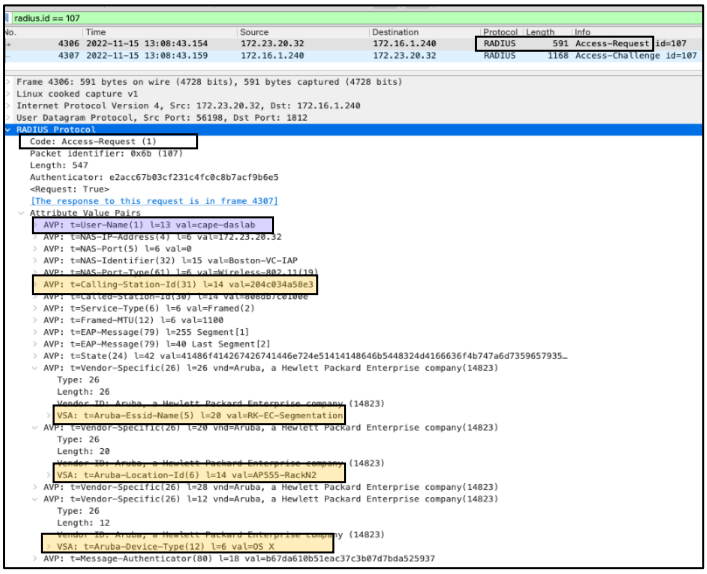


Figure 28: Viewing a RADIUS request packet after RADIUS packet capture.

11.2. RADIUS accept

The RADIUS accept packet indicates that the user or device authentication has been successfully validated by CPPM. This response may also include additional attributes, such as the `Aruba-User-Role`, if configured in the CPPM policy to be returned as part of the authentication process.

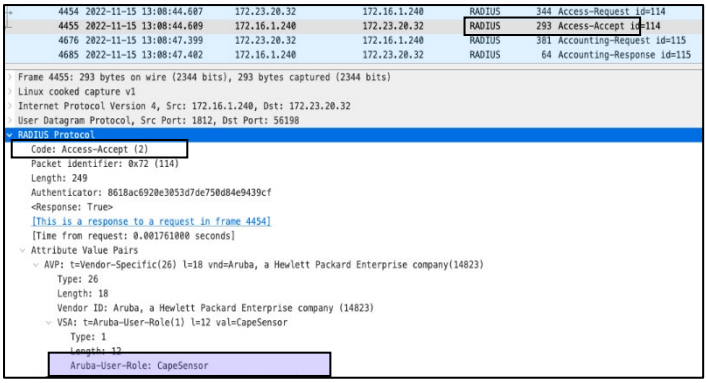


Figure 29: Viewing a RADIUS accept packet after RADIUS packet capture.

11.3. Accounting-Request

The RADIUS accounting-start packet is critical for this integration. It includes the `Framed-IP-Address` attribute, which represents the IP address assigned to the end user or device.

This IP address is essential, as it is used by CPPM to map user identity to network activity and to forward user information to Orchestrator. Without the `Framed-IP-Address` attribute in the accounting packet, CPPM cannot send user data to the Orchestrator.

Important

RADIUS accounting must be enabled on all authenticators (wired switches and wireless access points). Additionally, ensure that the user IP address is correctly captured and visible in CPPM.

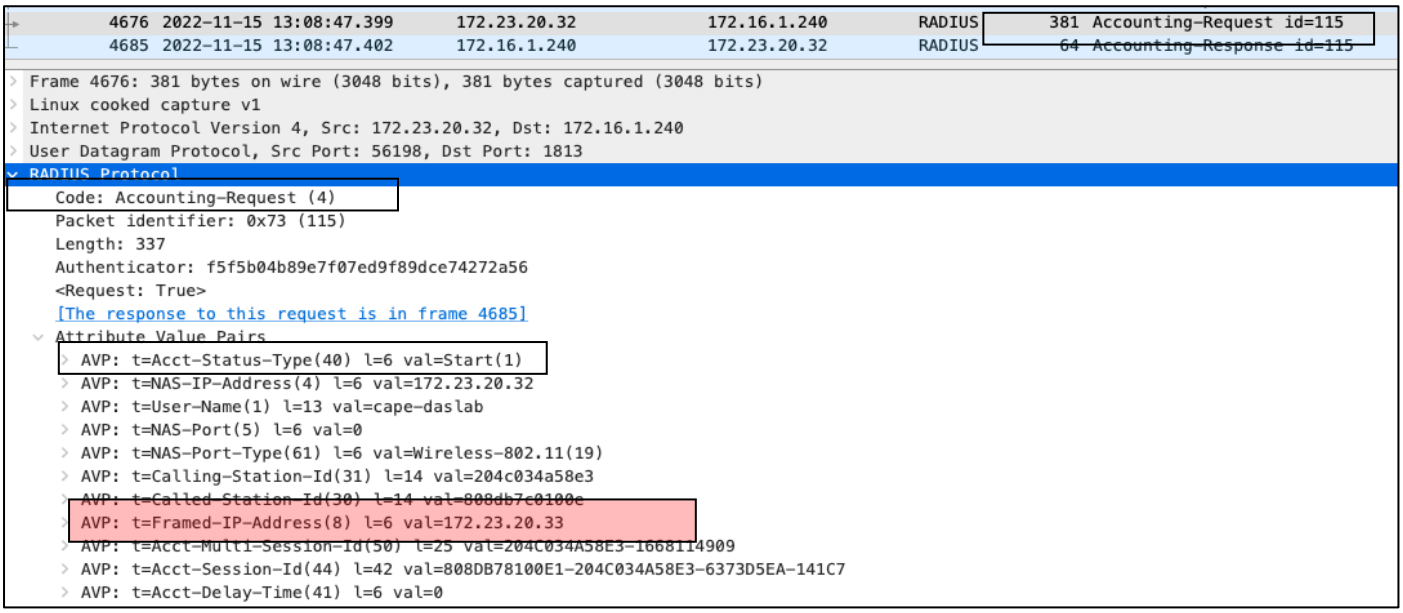


Figure 30: Viewing an accounting request after RADIUS packet capture.

12. Appendix C: Troubleshooting

Use the following steps to verify key components of the configuration and isolate issues related to missing user data or failed integration.

12.1. Check on Orchestrator

1. Ensure that CPPM account connectivity and service status appear as *Connected*.

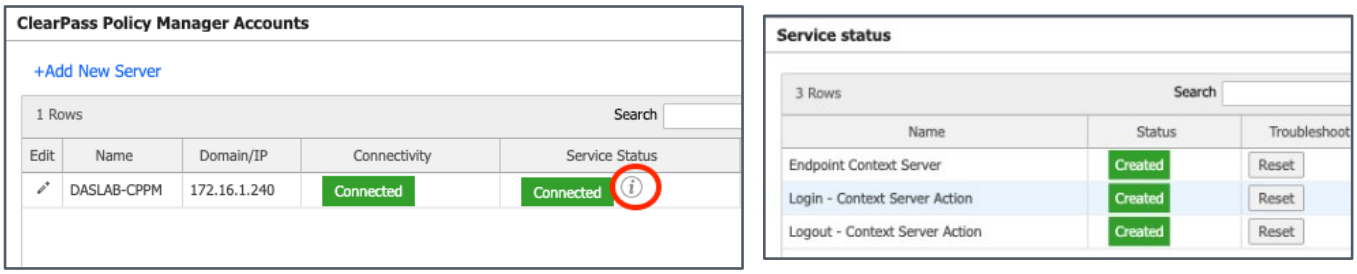


Figure 31: Viewing CPPM server connection status in Orchestrator.

2. If the status appears as *Unreachable* for the scenarios where firewall blocks inbound tcp443, ensure that Endpoint Context Server actions are configured (see Configure context server actions).
3. Verify that the API key created for ClearPass is present. Copy the key and verify it against the value (XAUTH token configured as part of Endpoint Context Server action) configured in CPPM.

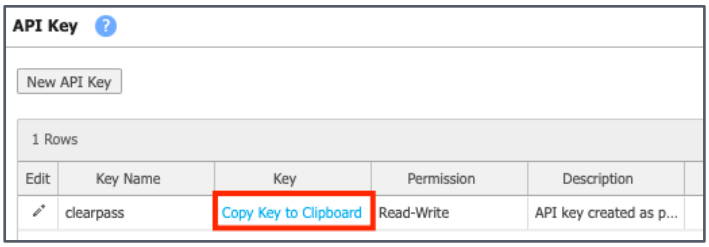


Figure 32: Viewing the API key in Orchestrator.

12.2. Check on CPPM

1. Review the Access Tracker (Monitoring > Live Monitoring > Access Tracker) and confirm that the user or device is successfully authenticated. On the Accounting tab, verify that the `Framed-IP-Address` field is populated with a valid IP address.

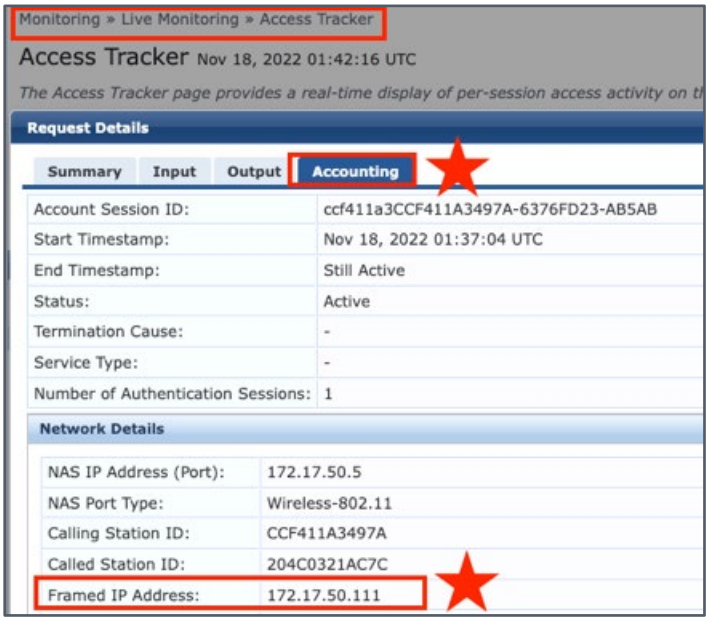


Figure 33: Verifying the Accounting tab in CPPM's Access Tracker.

2. Identify the ClearPass service used for the authentication. Navigate to **Configuration > Services**, click the **Enforcement** tab, and then confirm that the enforcement policy is applied to the service.

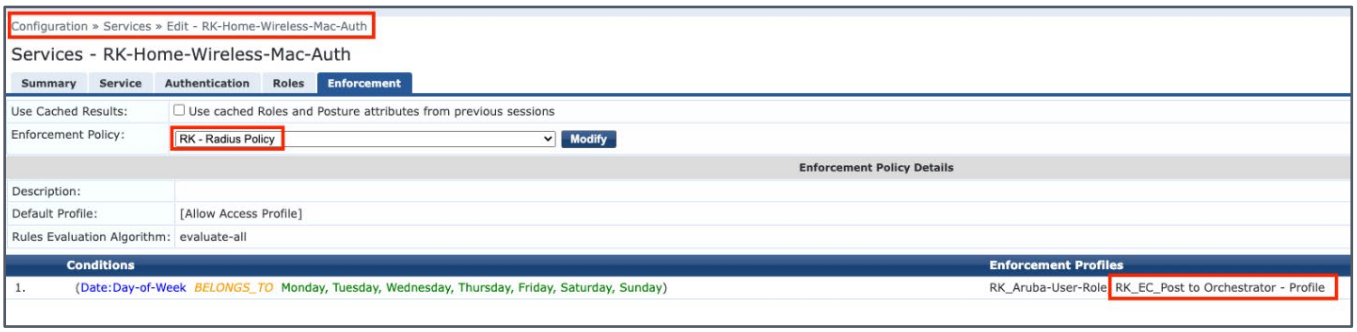


Figure 34: Confirming that the CPPM service has the correct enforcement policy.

3. Review the enforcement policy to ensure that it includes an enforcement profile that contains the required Endpoint Context Server actions.

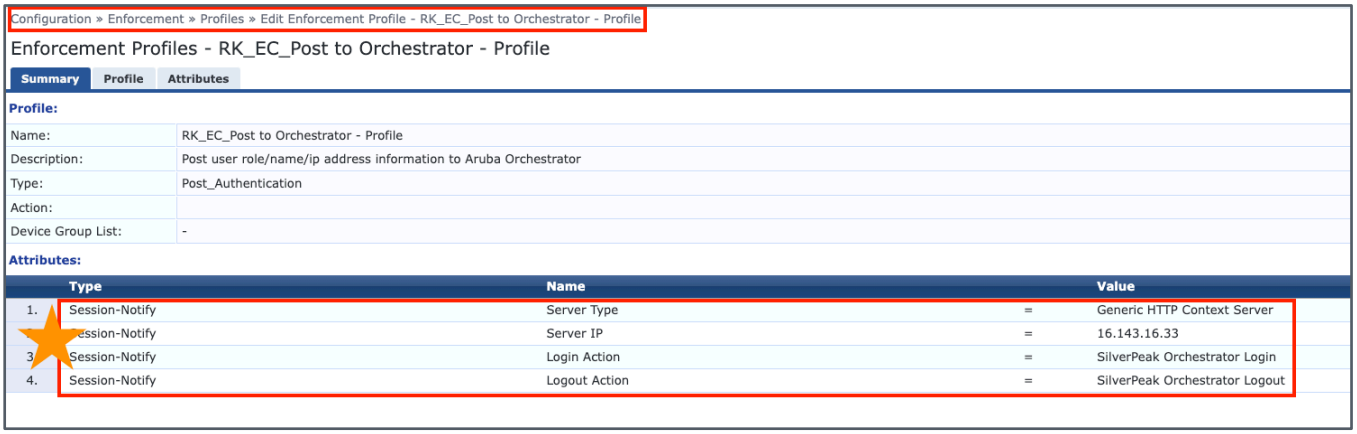


Figure 35: Reviewing enforcement profile details in CPPM.

4. Validate the **Endpoint Context Server Details – Action** tab by confirming that:
   a. The Orchestrator IP address is correct.
   b. The HTTP method is set to **POST**.
   c. The authentication method is set to **Basic**.
   d. The URL matches the expected configuration.

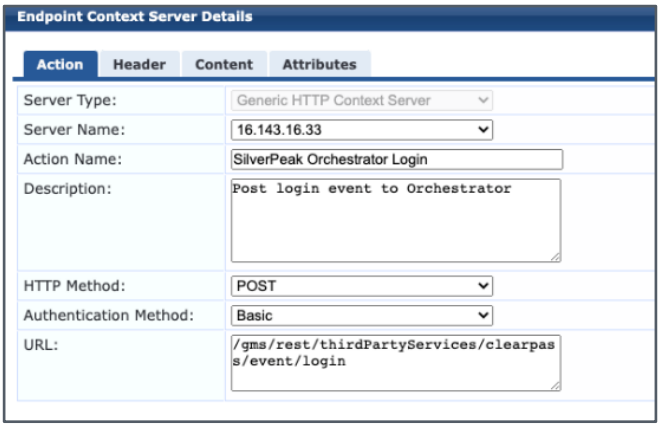


Figure 36: Confirming settings on the Endpoint Context Server Details – Action tab in CPPM.

5. Click the **Endpoint Context Server Details – Header** tab and confirm that the X-AUTH token matches the API key configured in Orchestrator.

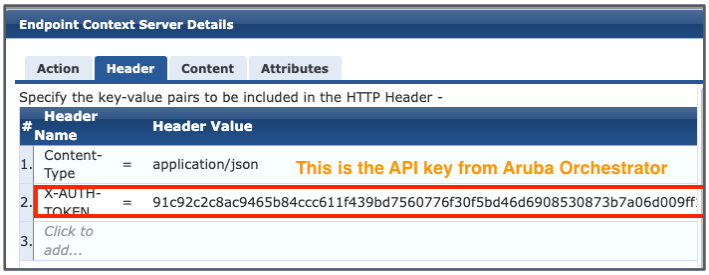


Figure 37: Confirming the X-AUTH token on the Endpoint Context Server Details – Header tab in CPPM.

6. Click the **Endpoint Context Server Details – Content** tab and confirm that all required keys are present and correctly defined, as shown in Figure 38. Verify that the `cppmDomain` attribute matches what was configured in Add CPPM account in the Orchestrator.

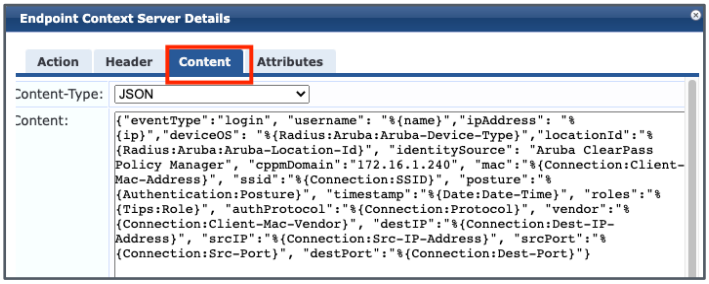


Figure 38: Confirming the required keys and attributes on the Endpoint Context Server Details – Content tab in CPPM.

12.3. Collect logs on CPPM to confirm that the HTTP POST was successful

Use the following steps to confirm whether the HTTP POST request from CPPM to Orchestrator was processed successfully.

12.3.1. Enable DEBUG logging for Async Network Services

The DEBUG setting increases log verbosity for asynchronous network operations, including outbound HTTP POST requests, to capture detailed HTTP transaction information.

To enable DEBUG-level logging, navigate to **Administration > Log Configuration**, click the **Service Log Configuration** tab, and configure the following settings:

— **Select Service:** Select **Async network services**.

— **Default Log Level:** Select **DEBUG**.

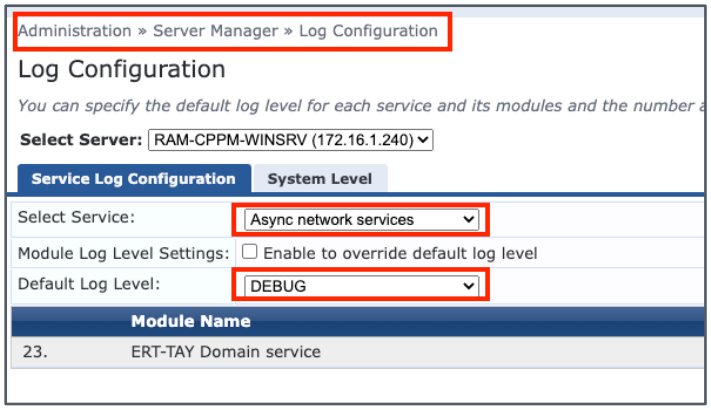


Figure 39: Enabling DEBUG for the Async Network Service in CPPM.

12.3.2. Collect logs from the CPPM server

After reproducing the issue, collect logs from the CPPM server for analysis.

1. Navigate to **Administration > Server Configuration > Collect Logs**.

2. When prompted, enter a password.
3. Select the **Logs from all Policy Manager services** check box.
4. Click **Start**.

The generated log bundle will include the Async Network Service logs required to confirm whether the HTTP POST request was sent successfully and how it was processed.

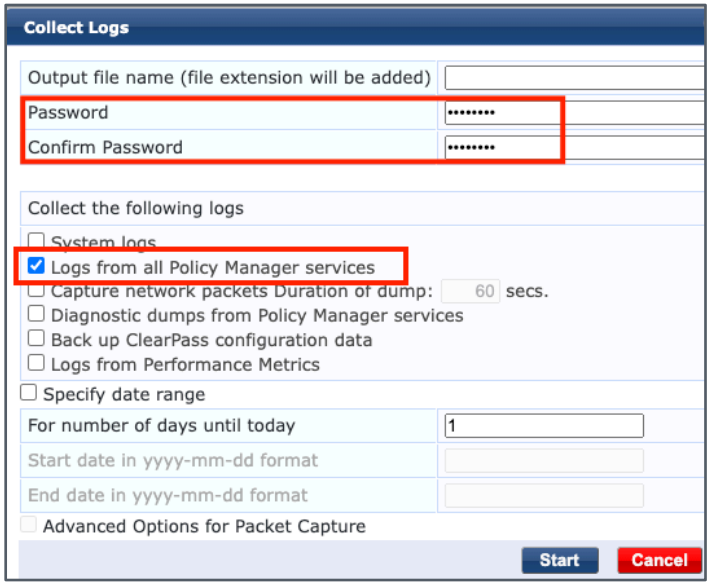


Figure 40: Collecting logs in CPPM.

12.3.3. Analyze CPPM logs to validate HTTP POST execution

When log collection is complete, download the generated log bundle and extract the contents.

1. Navigate to the log file at **PolicyManagerLogs > async-netd > postauth.log**.
2. Review the `postauth.log` entries corresponding to the transaction under investigation and confirm the following:
   a. The payload contains complete and expected user data.
   b. The HTTP response status code is `200`, indicating successful processing by Orchestrator.

A successful transaction shows a DEBUG entry similar to the following, confirming that the HTTP POST was sent and acknowledged:

Note

A `200` response code is required for every payload sent from CPPM. If a `500` response code is observed, confirm that the API key configuration on the Orchestrator is valid and active.

This is the API key for ClearPass, generated in Orchestrator

```
DEBUG ep.go:235:main.invokePlugin: --------- PLUGIN START {HTTP 16.143.16.33} ----------

DEBUG generic.go:167:postauth/plugins/generic.GenericHttpPlugin.ProcessData: Headers are
Content-Type and application/json

DEBUG generic.go:167:postauth/plugins/generic.GenericHttpPlugin.ProcessData: Headers are X-AUTH-
TOKEN and
91c92c2c8ac9465b84ccc611f439bd7560776f30f5bd46d6908530873b7a06d009ff1803a9bb4bd4b1344f407ee20b37
cc88e17e9cb34c39a8847f8ce30406ce

generic.go:244:postauth/plugins/generic.GenericHttpPlugin.ProcessData: Sending payload
{"eventType":"login", "username": "ccf411a3497a","ipAddress": "172.17.50.111","deviceOS":
"%{Radius:Aruba:Aruba-Device-Type}","locationId":"44:48:c1:c2:7d:7c", "identitySource": "Aruba
ClearPass Policy Manager", "cppmDomain":"172.16.1.240", "mac":"CCF411A3497A", "ssid":"vRASS",
"posture":"Unknown", "timestamp":"2022-11-18 01:37:04", "roles":"[User Authenticated]",
"authProtocol":"RADIUS", "vendor":"Google, Inc.", "destIP":"172.16.1.240", "srcIP":"172.17.50.5",
"srcPort":"44500", "destPort":"1812"} for SilverPeak Orchestrator Login to URL https://16.143.16.33

generic.go:262:postauth/plugins/generic.GenericHttpPlugin.ProcessData: Status code: 200,
Response: , URL: https://16.143.16.33/gms/rest/thirdPartyServices/clearpass/event/login

DEBUG ep.go:272:main.invokePlugin: --------- PLUGIN END {HTTP 16.143.16.33} --------
```

You must see response code of 200 for every payload sent from CPPM.

If you see response code of 500, then the API key is invalid. Check the API key in Orch/update CPPM.

Visit HPE.com

Chat now

© Copyright 2026 Hewlett Packard Enterprise Development LP. The information contained herein is subject to change without notice. The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty.
Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

Trademark acknowledgments, if needed. All third-party marks are property of their respective owners.

HEWLETT PACKARD ENTERPRISE

HPE.com