Change Log

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Modified By</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>V2020-01</td>
<td>Mar 2020</td>
<td>Danny Jump</td>
<td>First Published Version – Phase1 Visibility</td>
</tr>
</tbody>
</table>

Copyright

© Copyright 2020 Hewlett Packard Enterprise Development LP.

Open Source Code

This product includes code licensed under the GNU General Public License, the GNU Lesser General Public License, and/or certain other open source licenses. A complete machine-readable copy of the source code corresponding to such code is available upon request. This offer is valid to anyone in receipt of this information and shall expire three years following the date of the final distribution of this product version by Hewlett-Packard Company. To obtain such source code, send a check or money order in the amount of US $10.00 to:

Hewlett-Packard Company
Attn: General Counsel
3000 Hanover Street
Palo Alto, CA 94304
USA

Please specify the product and version for which you are requesting source code. You may also request a copy of this
Contents

Change Log .................................................................................................................................................. 2
Contents ...................................................................................................................................................... 3
Figures ......................................................................................................................................................... 4
Introduction ............................................................................................................................................... 5
Software Requirements ............................................................................................................................ 5
Installation and Deployment Guide ......................................................................................................... 5
Pictorial view of the Integration .............................................................................................................. 6
Configuration ............................................................................................................................................ 7

Configuration of ClearPass .......................................................................................................................... 7
  Create a ClearPass ‘API’ User ..................................................................................................................... 7
  Create a ClearPass Operator Profile .......................................................................................................... 8
  Create a ClearPass OAuth API Client ........................................................................................................ 10

Configuration of Medigate ......................................................................................................................... 11

Medigate sending enforcement data .......................................................................................................... 15

Monitoring/Reviewing ClearPass and Medigate communications .......................................................... 17
Figures

Figure 1: Pictorial view of ClearPass Policy Manager integration with Medigate ........................................ 6
Figure 2: Create an API level account in ClearPass ................................................................. 7
Figure 3: Creating a restricted access Operator Profile – Part1 ................................................. 8
Figure 4: Creating a restricted access Operator Profile – Part2 .................................................. 9
Figure 5: Create an API Client ................................................................................................. 10
Figure 6: Medigate Main Dashboard ......................................................................................... 11
Figure 7: Enabling ClearPass Policy Manager integration .................................................... 11
Figure 8: Configuring the ClearPass communication connector inside Medigate ..................... 11
Figure 9: Communication with ClearPass ............................................................................. 12
Figure 10: Endpoint Dictionary Attributes created by Medigate ............................................ 12
Figure 11: Extensive list of Endpoints created by Medigate ................................................ 13
Figure 12: Normalized Endpoint data created by Medigate ................................................... 13
Figure 13: Custom Endpoint attributes created by Medigate ................................................ 14
Figure 14: Example of Medigate context driving enforcement/segmentation ............................ 14
Figure 15: Medigate Endpoint Integration vendors .............................................................. 15
Figure 16: Medigate Endpoint Integration policy ................................................................. 16
Figure 17: Reviewing 'Last Sync' time to ClearPass ............................................................. 17
Figure 18: Example of API logs between Medigate and ClearPass ........................................ 17
Introduction

This Integration Guide covers the configuration and use of the integration between Medigate and ClearPass Policy Manager. The Medigate platform is focused on providing visibility into the Medical and Healthcare space, it discovers and precisely identifies every connected device on your clinical network. Utilizing industry-leading medical device signature database developed by Medigate Research Labs, they fingerprint each device with deep packet inspection (DPI) techniques, allowing dynamic inventory management and facilitating advanced detection and prevention capabilities.

This endpoint data is then shared directly with ClearPass via the ClearPass Security Exchange framework and the open API exposed under Policy Manager. Medigate will automatically update the ClearPass Policy Manager Endpoint Database with endpoint classification data and a number of custom security attributes, these attributes can then be used to drive role-mapping and or enforcement policies, a great source of context for segmentation policies.

This is Phase1 of our planned integration with Medigate, centralized visibility of network assets and endpoints across Medical IoT (MIoT) infrastructure. From here a centralized endpoint and edge security policy can be defined across standard IT, MIoT and your enterprise IoT from a single platform.

A Phase2 is under planning, check back for updates.

Software Requirements

At the time of writing, ClearPass version 6.8.4 is available and the recommended release. CPPM runs on hardware appliances with pre-installed software or as a Virtual Machine under the following hypervisors. Hypervisors that run on a client computer such as VMware Player are not supported.

- VMware ESXi 6.0 6.5, 6.6, 6.7 or higher
- Microsoft Hyper-V Server 2012, 2016 R2 or 2019
- Hyper-V on Microsoft Windows Server 2012, 2016 R2 or 2019
- Amazon EC2
- KVM on CentOS 7.5 or later.

The minimum version of Medigate required is, 2.5.1, this version or later supports the integration features.

Installation and Deployment Guide

The generic ClearPass installation and deployment guide is located here:

Pictorial view of the Integration

The diagram below shows a pictorial overview of the components and how they interact with each other.

**Figure 1: Pictorial view of ClearPass Policy Manager integration with Medigate**

Medigate synchronizes its asset inventory list into the ClearPass Endpoint Database. The device context can be used for Dynamic Segmentation leveraging policy.

Discovered Medical-IoT/IoT devices are pushed into the ClearPass Policy Manager Endpoint Database leveraging the endpoint REST APIs. The sync frequency can be configured.
Configuration

Configuration of ClearPass

Prior to creating and enabling the integration in Medigate a number of configuration elements need to be pre-created in ClearPass. Follow the below configuration steps carefully, collecting data as highlighted as you will need this in the following section when configuring Medigate to communicate with ClearPass.

Create a ClearPass ‘API’ User

As part of the communications channel between the two products, Medigate will use a number of API’s {both TIPS and REST}, access to the TIPS API’s is validated and authorized via Username/Password combination credentials. This UserId needs to have minimum levels of access, do not use a Super Administrator profile privilege level, use API Administrator as shown below. Access to the REST API’s is separately authenticated and authorized under OAuth2, details of this follow on the next page.

Create a user from Administration -> Users and Privileges -> +ADD -> {Create a user, ensure that you set a privilege level of API Administrator}

Make a note of the UserId and the password that was configured, ensure Privilege Level is API Administrator

Figure 2: Create an API level account in ClearPass
Create a ClearPass Operator Profile

Medigate also uses the REST API’s as part of the integration, REST API’s are authenticated under an OAuth2 framework, as part of the authorization behind OAuth ClearPass Policy Manager uses Operator Profiles to restrict access to only the necessary REST API’s. In summary all options are set as ‘No Access’ except for API Services and Policy Manager, where custom should be selected and then follow the specific as shown below. This will only allow the API’s to create endpoint Dictionary attributes and create/update endpoints. Create this under Guest -> Administration -> Operator Logins -> Profiles (Create a new operator profile)

Under API Services

- Allow API Access = Allow Access

Under Policy Manager

- Dictionaries – Attributes = Read, Write, Delete
- Dictionaries – Fingerprints = Read, Write, Delete
- Identity – Endpoints = Read, Write, Delete

Figure 3: Creating a restricted access Operator Profile – Part 1

Edit Operator Profile (API Profile)

Use this form to make changes to the operator profile API Profile.

Operator Profile Editor

* Name: API Profile
  Enter a name for this operator profile.

Description:
Operators with this profile can use the API to manage Extensions
Comments or descriptive text about the operator profile.

Access
These options control what operators with this profile are permitted to do.

Enabled:
If unchecked, operators with this profile will not be able to log in.

Administrator
Select operator permissions for system administration and management tasks.

- No Access

Advertising Services
Select operator permissions for managing advertising content and services.

- No Access

API Services
Select operator permissions for API access and management.

- Allow API Access

Operators with this privilege are permitted to make API calls. Additional privileges are also required, depending on the API.

- API Documentation
  Operators with this privilege can browse and interact with the API Explorer.

- Configure SOAP Web Services (Legacy)
  Operators with this privilege can change system settings for SOAP web services.

- List SOAP Web Services (Legacy)
  Operators with this privilege can browse the available SOAP web services and access the service definitions (WSDL).

- Manage API Clients
  Operators with this privilege may view and manage API clients (OAuth2 authentication).

- SOAP API (Legacy)
  Operators with this privilege can use SOAP web services to perform system functions. Additional privileges are also required, depending on the API.

- XML/RPC API (Legacy)
  Operators with this privilege can access system functions through the XML/RPC API. Additional privileges are also required, depending on the API.

Aruba Integrations

- No Access
Figure 4: Creating a restricted access Operator Profile – Part 2

<table>
<thead>
<tr>
<th>Policy Manager</th>
<th>Custom...</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Select operator permissions for Policy Manager</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Licenses</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage Application Licenses</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication - Methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage authentication methods</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certificate - Revocation List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage Revocation Lists</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certificate - Trust List</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage certificate trust lists</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Certificates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage certificates</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clearpass Portal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage Clearpass Portal</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration - Network Scan</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage Network Scan under Configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Configuration - Services</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage Services under Configuration</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dictionaries - Attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage attributes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dictionaries - Context Server Actions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage context server actions</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Dictionaries - Fingerprints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage fingerprints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Events - Login Audit</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage login audits</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Events - System Events</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage system events</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Servers - Endpoint Context Servers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage endpoint context servers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Servers - File Backup Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage file backup servers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Servers - SNMP trap receivers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage SNMP trap receivers</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Servers - Syslog Export Filters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage syslog export filters</td>
<td></td>
<td></td>
</tr>
<tr>
<td>External Servers - Syslog Targets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage syslog targets</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identity - Endpoints</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Operators with this privilege can manage endpoints</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Create a ClearPass OAuth API Client

Next create an API Client **Guest -> Administration -> API Services -> API Clients -> {Create API Client}**

Ensure you use the Operator Profile previously created to restrict the capabilities of the API Client.

Notice the highlighted configuration options needed, and set as appropriate

- **Operating Mode** = ClearPass REST API - Client will be used for API calls to ClearPass
- **Operator Profile** = Use the Profile you created in the previous section
- **Grant Type** = Client credentials (grant_type = client_credentials)

Ensure you record the Client Secret and the ACTUAL API-Client ID name i.e. medigate as below

**Figure 5: Create an API Client**

At this time all of the necessary config has been created in Policy Manager, ensure you collected the below list of information before proceeding to the next section.

- **CPPM UserID**
- **CPPM UserID Password**
- **CPPM OAuth2 API Client ID Name**
- **CPPM OAuth2 API Client Secret**
**Configuration of Medigate**

For this initial integration between the two products, there is limited configuration required on Medigate. As part of the communication setup between Medigate and ClearPass Policy Manager, Medigate will create a number of Endpoint custom attributes. After the configuration is complete the Medigate platform will continue to update the ClearPass Policy Manager EndpointDb as it discovers new endpoints after the initial sync. Follow the below to configure and enable the integration. From the Medigate main console click on the **Integrations -> ClearPass** from the navigation banner.

**Figure 6: Medigate Main Dashboard**

By default, the ClearPass Policy Manager integration is disabled, click on ‘**Activate Integration**’, to start the config.

**Figure 7: Enabling ClearPass Policy Manager integration**

Complete the configuration as necessary to setup the connection between Medigate and ClearPass, you'll require the **Client Id & Client Password** and **ClearPass Username & ClearPass Password** created in the previous section, adjust the Sync interval as appropriate.

**Figure 8: Configuring the ClearPass communication connector inside Medigate**
After clicking on Activate, the integration is enabled and stats related to sync state are displayed in the UI. An edit option is also available in case you need to modify any of the configuration, e.g. the Sync interval. Medigate will initially synchronize the endpoints it discovered and then continue to update new endpoint data directly into the ClearPass Policy Manager endpointDb based upon the sync interval.

**Figure 9: Communication with ClearPass**

As part of enabling and configuring the above integration, Medigate will create a number of custom Endpoint Dictionary attributes using the ClearPass REST API. These custom attributes can then be used for role-mapping/enforcement actions in a Service Policy and to drive segmentation workflows.

**Figure 10: Endpoint Dictionary Attributes created by Medigate**

The Endpoint data is updated by Medigate, it creates the Endpoints, sets the endpoint classification and configures the custom endpoint attributes. An example of the data is shown below.
Looking closer at the endpoint data we can see several important things, the mac-address, mac-vendor, and some endpoint classification as determined by Medigate, other valuable data such as the date the endpoint was added and profiled, said another way the time Medigate updated ClearPass with the devices data.

**Figure 12: Normalized Endpoint data created by Medigate**
In addition to the standard data, Medigate supplies other customer attributes, clicking on the Attribute tab provides the below attributes, any of this data could be used in Policy to trigger enforcement, role mapping or segmentations policy.

**NOTE** specifically that Medigate also supplies a risk-score based context on the device health, this could be very valuable to ClearPass Enforcement Policies.

**Figure 13:** Custom Endpoint attributes created by Medigate

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Source</td>
<td>Medigate</td>
</tr>
<tr>
<td>2. medigate_device_type</td>
<td>Medical Device Integrator</td>
</tr>
<tr>
<td>3. medigate_model</td>
<td>Neuron 2</td>
</tr>
<tr>
<td>4. medigate_os_version</td>
<td>7 Embedded</td>
</tr>
<tr>
<td>5. medigate_risk_score</td>
<td>2</td>
</tr>
<tr>
<td>6. medigate_vendor</td>
<td>Capsule</td>
</tr>
<tr>
<td>7. medigate_vlan</td>
<td>111</td>
</tr>
<tr>
<td>8. Click to add...</td>
<td></td>
</tr>
</tbody>
</table>

An example of a ClearPass Enforcement Policy utilizing the Medigate data is below.

**Figure 14:** Example of Medigate context driving enforcement/segmentation

Configuration > Enforcement > Policies > Edit - medigate-enforcement-and-segmentation

Enforcement Policies - medigate-enforcement-and-segmentation

**Enforcement:**
- Name: medigate-enforcement-and-segmentation
- Description:
- Enforcement Type: RADIUS
- Default Profile: [Deny Access Profile]

**Rules:**
- Rules Evaluation Algorithm: First applicable

<table>
<thead>
<tr>
<th>Conditions</th>
<th>Actions</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Endpoint:Source EQUALS Medigate)</td>
<td>medigate-cart</td>
</tr>
<tr>
<td>AND (Endpoint:medigate_device_type CONTAINS Anesthesia Cart)</td>
<td></td>
</tr>
<tr>
<td>(Endpoint:Source EQUALS Medigate)</td>
<td>medigate-nurse-vlan</td>
</tr>
<tr>
<td>AND (Endpoint:medigate_location CONTAINS Surg)</td>
<td></td>
</tr>
<tr>
<td>(Endpoint:Source EQUALS Medigate)</td>
<td></td>
</tr>
<tr>
<td>AND (Endpoint:medigate_os_version NOT_EQUALS 11.5.5)</td>
<td>[Deny Access Profile]</td>
</tr>
<tr>
<td>AND (Endpoint:medigate_device_type EQUALS Glucose Meter)</td>
<td></td>
</tr>
</tbody>
</table>
Medigate sending enforcement data

One of the advanced capabilities from Medigate is their ability to understand device protocols, and within a customer deployment understand where services are running, combining Medigate's ability to categorize endpoint and automatically build enforcement policies that can be used by ClearPass Administrators removes the issue of human error in Policy Manager.

Medigate simplifies and automates this process based upon its deep understanding of the endpoint and the protocols its running, this expands to be able to define an ACL that is representative of the endpoint. Today this is limited to an ACL based profile, in the future Medigate will be able to define and push an Aruba-Role definition into Policy Manager. Let's look at a device on Medigate.

**Figure 15: Medigate Endpoint Integration vendors**

![Medigate Endpoint](image)

Notice under the Integrations section, ClearPass is shown as configured, this now allows Medigate to automatically generate the enforcement-profile for this Siemens Symbia E scanner.

Click on View Policies and an example of automatically generated data is shown.
Figure 16: Medigate Endpoint Integration policy
Monitoring/Reviewing ClearPass and Medigate communications

Once the sync has started endpoint data will be populated directly into the Policy Manager EndpointDb, you can view the last update time from the integration configuration screen, see below for an example.

*Figure 17: Reviewing 'Last Sync' time to ClearPass*

If the Sync is not working or shows an error then it's likely you've missed capturing some of the information, recheck the data recorded, additionally you can view the API calls between Medigate and ClearPass from [Guest-> Administration-> Support-> Application Log](/guest/administration/support/application-log) below an example of API's from Medigate to ClearPass.

*Figure 18: Example of API logs between Medigate and ClearPass*

The events and messages generated by this application are logged here. For in-depth information about an event, click on it.

<table>
<thead>
<tr>
<th>Time</th>
<th>IP</th>
<th>User</th>
<th>Severity</th>
<th>Message</th>
</tr>
</thead>
<tbody>
<tr>
<td>2020-02-26 20:40:30</td>
<td>10.0.0.190</td>
<td>oauth2:Medigate-New</td>
<td>info</td>
<td>API call 'GET /api/session?filter={&quot;framedipaddress&quot;: &quot;10.0.0.190&quot;}&amp;sort=-id&amp;offset=0&amp;limit=25&amp;calculate_count=false' succeeded</td>
</tr>
<tr>
<td>2020-02-26 20:38:11</td>
<td>10.0.0.190</td>
<td>oauth2:Medigate-New</td>
<td>info</td>
<td>API call 'GET /api/session?filter={&quot;framedipaddress&quot;: &quot;10.2.100.203&quot;}&amp;sort=-id&amp;offset=0&amp;limit=25&amp;calculate_count=false' succeeded</td>
</tr>
<tr>
<td>2020-02-26 20:38:00</td>
<td>10.0.0.190</td>
<td>oauth2:Medigate-New</td>
<td>info</td>
<td>API call 'GET /api/session?filter={&quot;framedipaddress&quot;: &quot;10.2.100.202&quot;}&amp;sort=-id&amp;offset=0&amp;limit=25&amp;calculate_count=false' succeeded</td>
</tr>
<tr>
<td>2020-02-26 20:37:47</td>
<td>10.0.0.190</td>
<td>oauth2:Medigate-New</td>
<td>info</td>
<td>API call 'GET /api/session?filter={&quot;framedipaddress&quot;: &quot;10.2.100.201&quot;}&amp;sort=-id&amp;offset=0&amp;limit=25&amp;calculate_count=false' succeeded</td>
</tr>
<tr>
<td>2020-02-26 20:35:45</td>
<td>10.0.0.190</td>
<td>admin</td>
<td>info</td>
<td>API: Created new OAuth2 access token for client ID 'Medigate-New'</td>
</tr>
</tbody>
</table>