CyberHound

a Hewlett Packard
Enterprise company

ClearPass

Integration Guide

Change Log

Version	Date	Modified By	Comments
V2019-01	Feb 2019	Jason Atkins	Initial Document Release

Copyright

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 source code free of charge at HPE-Aruba-gplquery@hpe.com.

Contents

Introduction and Overview 5
Software Requirements 5
ClearPass Installation and Deployment Guide 6
CyberHound Prerequisites 7
CyberHound Initial Configuration 7
CyberHound IPS Configuration 8
ClearPass Ingress Event Engine 9
Check Inbound Event Notification is Enabled 9
Check the Ingress Services are running 10
Batch Processing Interval 10
Configure the Ingress Event Dictionary for CyberHound 11
Enable Radius Accounting and Change of Authorization on Network Device 12
Adding a CoA Delay (Optional) 12
Add an Event Source 13
Add Endpoint Dictionary Attributes 13
Building ClearPass Events and Actions 15
Create the Event Service Policy and Profiles 15
    Add New Enforcement Profiles 15
    Add a New Enforcement Policy 18
Create Event Service Policy 19
Create an Authentication Service Policy 20
Creating a Quarantine Captive Portal Page 21
    Configuring a ClearPass Captive Portal 21
Removing a Device from Quarantine 22
Access Tracker Results 23
Appendix 1 – IEE_CyberHound.xml file 25

Figures

Figure 1: CyberHound and ClearPass Integration Overview 5
Figure 2: CyberHound Enable ClearPass IP addressIntegration 7
Figure 3: CyberHound test settings 7
Figure 4: CyberHound IPS Category Configuration 8
Figure 5: ClearPass Enable Ingress Event Engine 9
Figure 6: ClearPass Services Control 10
Figure 7: ClearPass Batch Processing Interval 10
Figure 8: ClearPass Ingress Event Dictionaries 11
Figure 9: ClearPass Event Dictionary list 11
Figure 10: ClearPass CoA Delay 12
Figure 11: ClearPass Adding an Event Source 13
Figure 12: ClearPass Adding an Event Source 13
Figure 13: ClearPass New Endpoint Attributes 14
Figure 14: Example workflow 15
Figure 15: Add Post Authentication Enforcement Profile 15
Figure 16: ClearPass Enforcement Profile Examples 17
Figure 17: ClearPass New Event Enforcement Policy 18
Figure 18: ClearPass Event Enforcement Policy Rules 18
Figure 19: ClearPass Event Service Policy 19
Figure 20: ClearPass select the Enforcement Policy 19
Figure 21: ClearPass Authentication Service Policy Enforcement 20
Figure 22: ClearPass Find Endpoint 22
Figure 23: ClearPass Endpoint Attributes 22
Figure 24: ClearPass Access Tracker 23

a Hewlett Packard
Enterprise company

www.arubanetworks.com
1344 Crossman Avenue
Sunnyvale, CA 94089
Phone: 1-800-WIFI-LAN (+800-943-4526)
Fax 408.227.4550

Introduction and Overview

This guide covers the setup required to integrate CyberHound's Advanced Threat Management system with Aruba ClearPass Policy Manager.

The integration provides ClearPass with "real time" event messages for devices identified (by CyberHound) as generating malicious network traffic including Malware, Viruses, Botnets, Vulnerabilities, Worms and more.

Utilizing the Ingress Event Engine [IEE] available in Aruba ClearPass, the "real time" event messages from CyberHound can trigger event workflows in ClearPass to automate a response to Block or Quarantine authenticated devices on the network.

The diagram below describes the end-to-end workflow utilizing CyberHound to discover the malicious traffic and ClearPass to enforce the isolation of the device on the network.

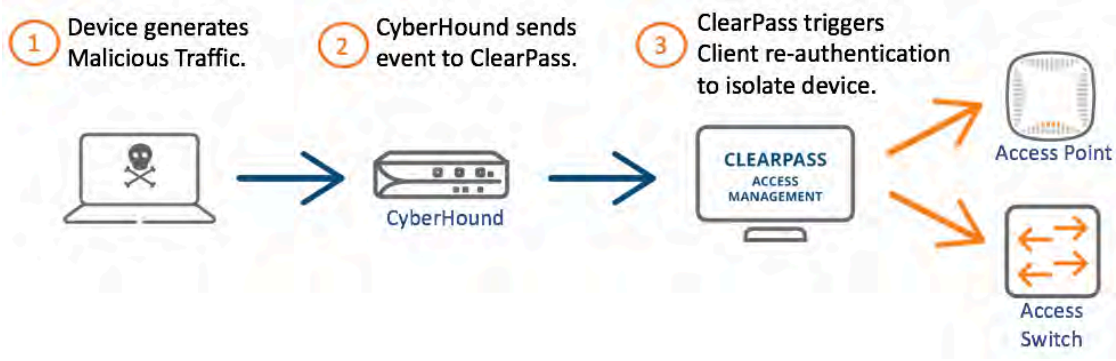


Figure 1: *CyberHound and ClearPass Integration Overview*

Software Requirements

The minimum software version required for ClearPass is 6.7.2. At the time of writing, ClearPass 6.7.9 is the latest available and recommended release. Any subsequent ClearPass software release will support this integration. ClearPass runs on either 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 5.0, 5.1, 5.5, 6.0, 6.5 or higher
- Microsoft Hyper-V Server 2012 or 2016 R2
- Hyper-V on Microsoft Windows Server 2012 or 2016 R2
- KVM on CentOS 7.5 or later

The minimum software release for CyberHound UTM is V31.7.1.

ClearPass Installation and Deployment Guide

This document assumes your ClearPass environment is already configured and operational. If you require assistance with basic deployment, refer to the following deployment guide:
http://www.arubanetworks.com/techdocs/ClearPass/Aruba_DeployGd_HTML/Default.htm

CyberHound Prerequisites

You will need the following in order to complete this integration.

- CyberHound version 31.7.1
- A valid CyberHound UTM license
- CyberHound IPS Module installed (please refer to the CyberHound documentation for minimum resource specification)

Please contact CyberHound Support if you require access to the CyberHound IPS module.

CyberHound Initial Configuration

To implement ClearPass integration and apply automated network policy actions, go to ***Advanced Firewall > IPS Configuration*** and select '***Enabled'*** for both ***IPS Rules Engine*** and ***Aruba ClearPass Integration***. Specify the host name or IP address of your ClearPass Policy Manager. Multiple ClearPass Policy Managers are support with comma separated entries.

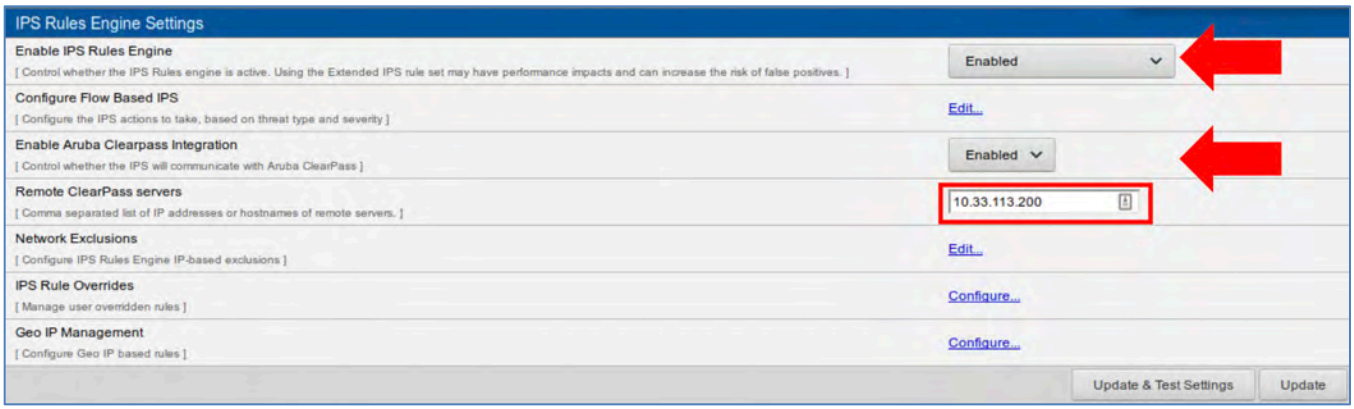


Figure 2: *CyberHound Enable ClearPass IP addressIntegration*

To finalize the configuration, Click Update and test the settings.

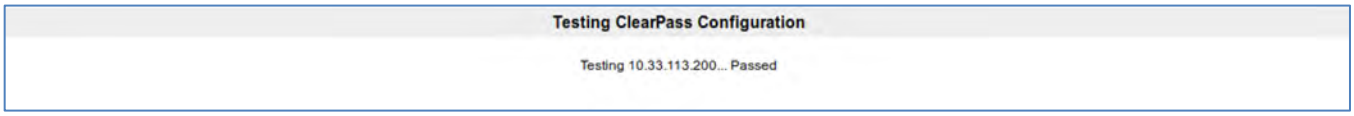


Figure 3: *CyberHound test settings*

CyberHound IPS Configuration

IPS and Network Policy actions may be applied on a 'per category' and 'severity' level basis.

Figure 4: *CyberHound IPS Category Configuration*

For example: Denial of Service threats with a severity level 0 (most severe) will be identified and dropped by the CyberHound. To enforce an additional Aruba Clearpass network policy such as quarantining or blocking the offending device from connecting to the network, simply select the desired Clearpass action.

The following matrix outlines the actions that are available to be sent to ClearPass Policy Manager.

Event Action	Description
Block	Instruct ClearPass to dynamically disconnect and block the infected device from rejoining the network.
Quarantine	Instruct ClearPass to dynamically transfer the infected device into a predefined quarantine network
Ignore	Will not send events to ClearPass

By Default, IPS rules are configured to alert only and no threat intelligence feed is transmitted to ClearPass.

NOTE

ClearPass actions for Block and Quarantine will be sent as syslog events to ClearPass for processing. Please ensure these events do not exceed 100 events per minute.

NOTE

Note: Optional 3rd party IPS event log management may be configured to use additional Remote syslog servers based on log source. This may be used in conjunction with the ClearPass integration. To configure go to ***Configuration > Remote Syslog*** and specify the 3rd party host.

ClearPass Ingress Event Engine

The integration between ClearPass Policy Manager and CyberHound is driven through a ClearPass capability known as the Ingress Event Engine [IEE]. This feature is part of the ClearPass Exchange Framework and provides ClearPass with the ability to consume inbound syslog messages. Once ingested, ClearPass will parse the message and trigger an event in the form of an Enforcement update for the device.

Check Inbound Event Notification is Enabled

By default, the IEE feature is disabled. To enable IEE go to **ClearPass Policy Manager > Administration > Server Manager > Server Configuration > [Your CPPM Node] > System**.

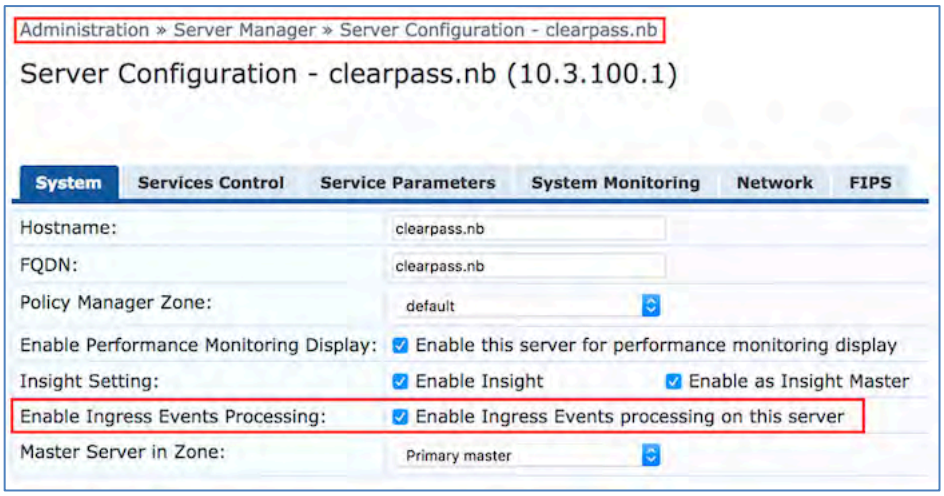


Figure 5: *ClearPass Enable Ingress Event Engine*

Note when enabling this feature, a warning message is displayed which highlights and warns of the potential consequences. Be aware that IEE processing can generate a significant CPU load on the node. Careful consideration needs to be used and we specifically do not want ClearPass to receive a constant stream of syslog messages that ClearPass will need to process. **ClearPass should only be receiving syslog messages, by exception, that it needs to take action on**. If you send a constant stream of syslog then the overhead will likely cause the node to become CPU bound and the potential failure/timeout of the primary function, the Authentication of Users/Computers. **Do not exceed 100 events per minute hitting the IEE**.

NOTE

Additional guidelines for enabling the IEE feature can be found in the Tech Note titled "ClearPass Ingress Event Engine" which is available for download on the Aruba Support web site (https://support.arubanetworks.com).

Check the Ingress Services are running

Please ensure the Ingress services have been started on the ClearPass node by clicking on the "Services Control" TAB.

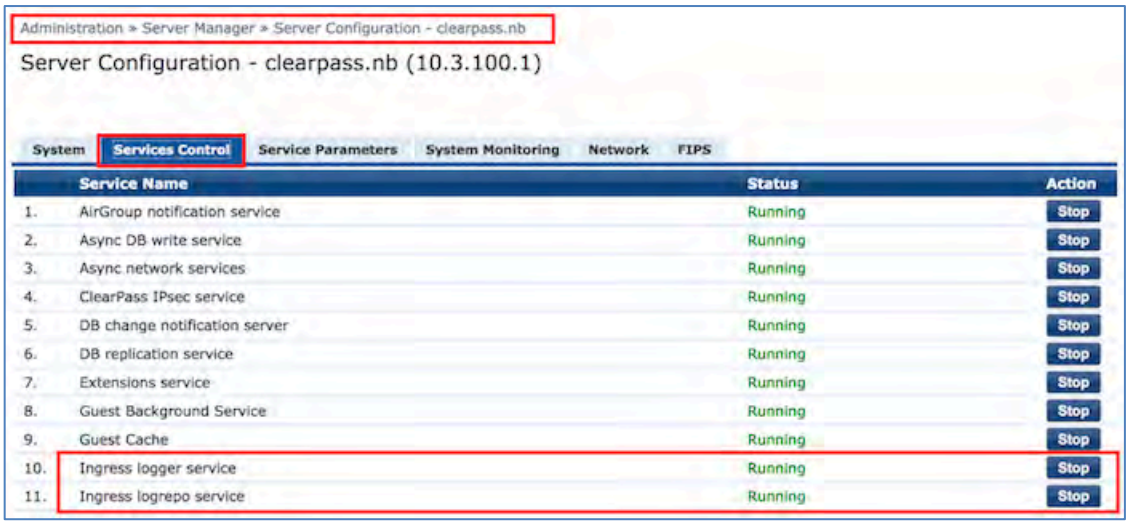


Figure 6: *ClearPass Services Control*

Batch Processing Interval

The batch processing interval determines the frequency for the event engine to check the EventDB for new entries. For testing purposes, you can reduce the batch processing interval to 10 seconds to make the processing of events more "real time". It's recommended to leave this value at 30 seconds for production environments.

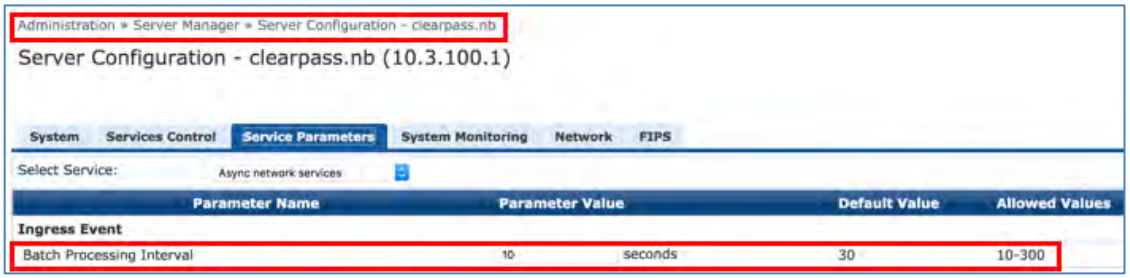


Figure 7: *ClearPass Batch Processing Interval*

Configure the Ingress Event Dictionary for CyberHound

As of ClearPass 6.7.9, the CyberHound IEE dictionary is NOT installed by default and will need to be imported before it can be used. As a reference the CyberHound dictionary is provided in the appendix of this document, however our experience has shown that copying and pasting the text and saving it to a local XML file can lead to spurious characters being added depending on the text application being used.

We prefer that you download the file from our support site directly to your local device, you can find it in this folder, Note: support credentials are required to access this location.

https://support.arubanetworks.com/DownloadSoftware/tabid/75/DMXModule/510/EntryId/22752/Default.aspx

Once download, the dictionary can be installed ***Administration > Dictionaries > Ingress Events > Import***

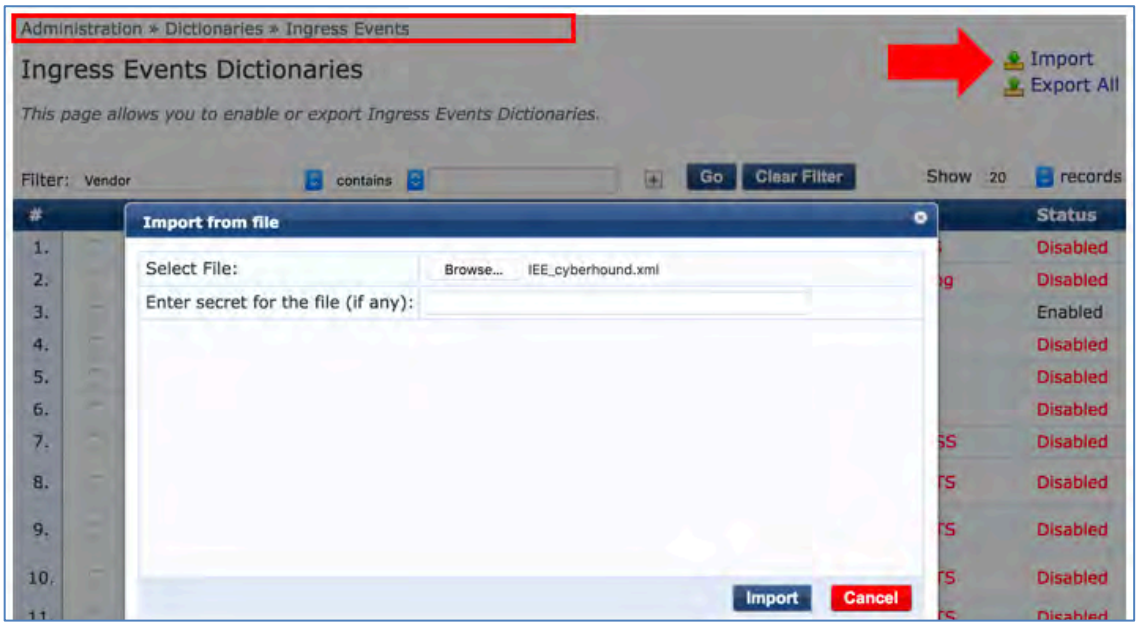


Figure 8: *ClearPass Ingress Event Dictionaries*

Once imported, the CyberHound dictionary should be automatically enabled and visible in the list.

Figure 9: *ClearPass Event Dictionary list*

Enable Radius Accounting and Change of Authorization on Network Device

ClearPass will receive the IP address of the malicious device via the Event message from CyberHound. ClearPass will use Radius Accounting to identify the current session for that IP address which, in turn, will provide the Policy Engine with the MAC Address of the malicious device. Once the MAC Address is known, ClearPass will be able to write details to the Endpoint Repository and also issue Radius Terminate message to disconnect the intended device.

Please ensure;

1. Network Access Device is configured to send Radius Accounting to ClearPass
2. Both the Network Access Device and ClearPass are configured for Radius CoA

Adding a CoA Delay (Optional)

If you would like to disconnect devices after an event is received then you may need to add a delay to the triggering of the CoA message. This will allow ClearPass to write the Endpoint Attributes to the database before the device automatically reconnects. This value may need to be tweaked depending on your cluster design. For this example the delay will be set to 10 seconds. Go to ***Administration > Server Manager > Server Configuration > [select your node]*** and open the **Service Parameters** TAB.

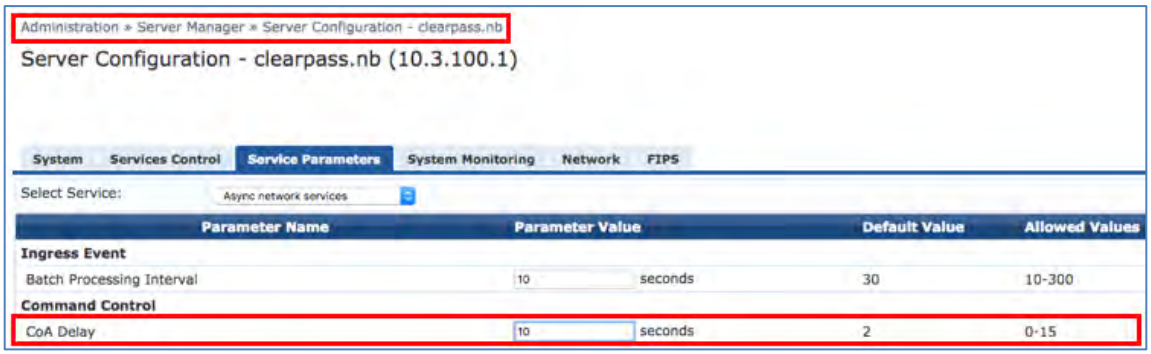


Figure 10: *ClearPass CoA Delay*

Add an Event Source

The IP address of the CyberHound appliance must be added before ClearPass before incoming events will be processed. The new event source can be configured under ***Configuration > Network > Event Sources > ADD.***

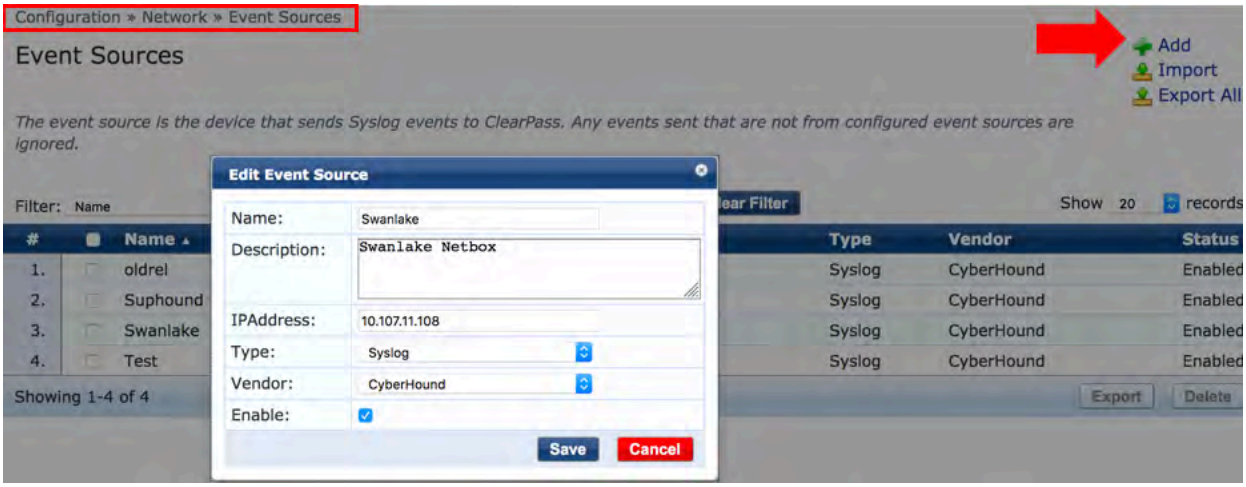


Figure 11: *ClearPass Adding an Event Source*

Add Endpoint Dictionary Attributes

To ensure that the CyberHound events do not affect other endpoint attributes it is recommended that you create three (3) new attributes specifically for the CyberHound integration.

Go to Administration > Dictionaries > Dictionary Attributes and click ADD.

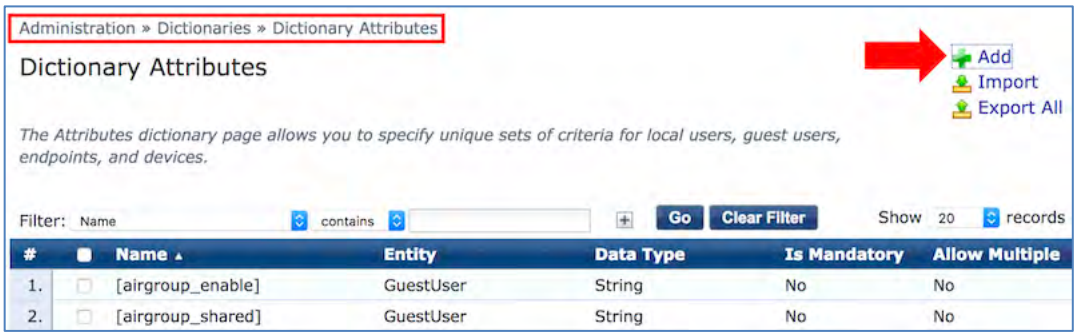


Figure 12: *ClearPass Adding an Event Source*

Next, create three (3) new Endpoint attributes (screen shots on next page)

1. **CyberHound_Event_Action** (as string)
2. **CyberHound_Event_DateTime** (as date-time)
3. **CyberHound_Event_ThreatName** (as string)

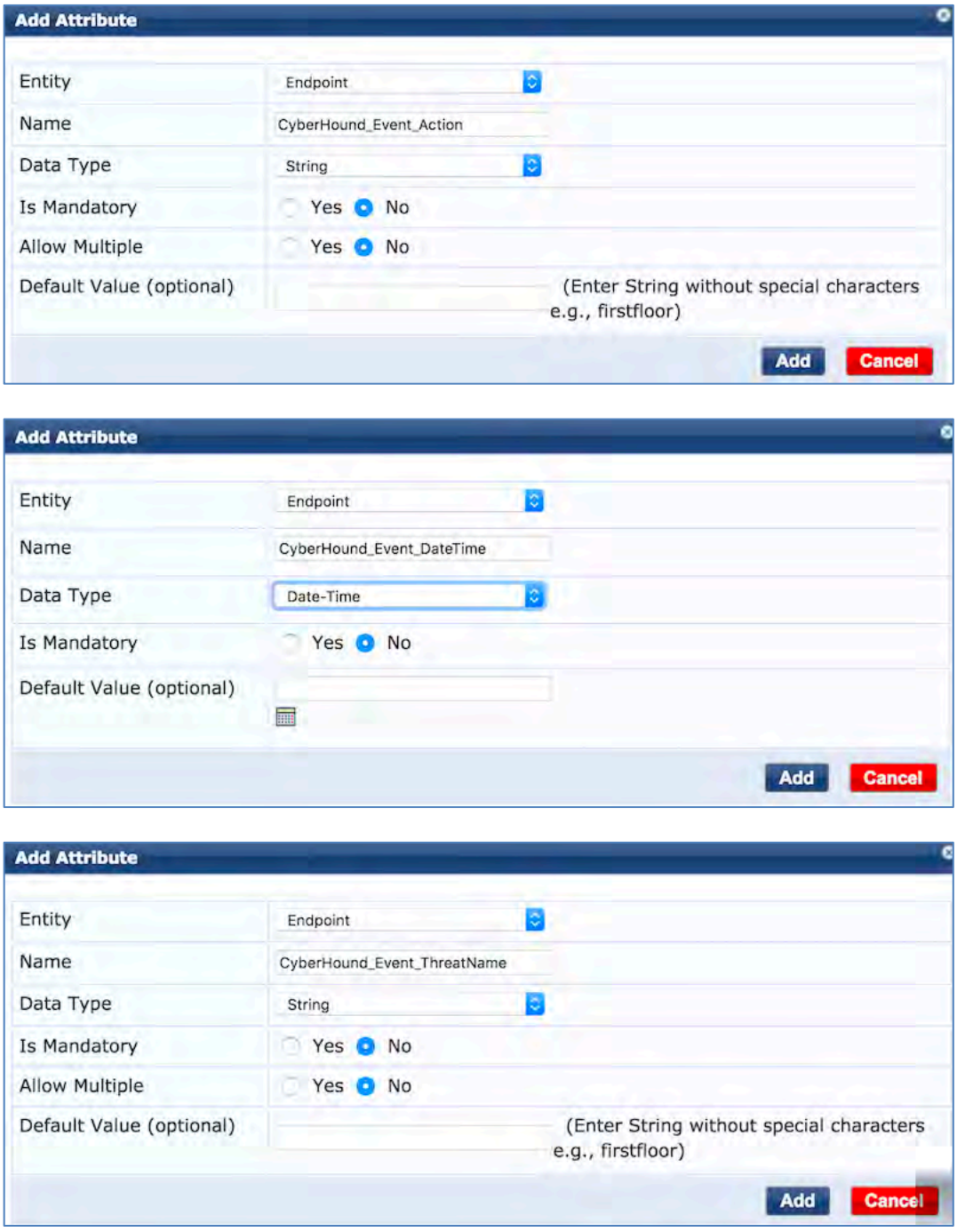


Figure 13: *ClearPass New Endpoint Attributes*

The initial setup of ClearPass is now completed.

Building ClearPass Events and Actions

The next phase of the configuration is to write a service policy to process the incoming Events and trigger the appropriate actions. This will involve the creation of Enforcement Profiles, Enforcement Policies and Event Services.

The following workflow will be used to demonstrate the integration between CyberHound and ClearPass. Your environment may require a modified workflow so please feel free to modify as required.

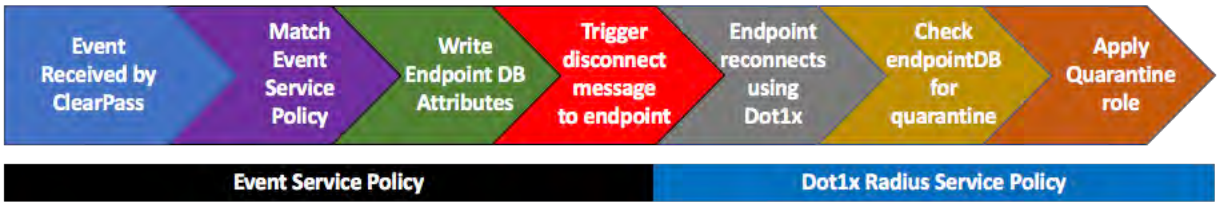


Figure 14: *Example workflow*

Create the Event Service Policy and Profiles

When creating the Event Service Policy it is best to work from right to left by creating the Enforcement Profiles, Enforcement Policies before finally creating the Event Service Policy.

Add New Enforcement Profiles

Go to ***Configuration > Enforcement > Profiles*** and click the ADD option. Use the "ClearPass Entity Update Enforcement" template to create a Post_Authentication enforcement profile.

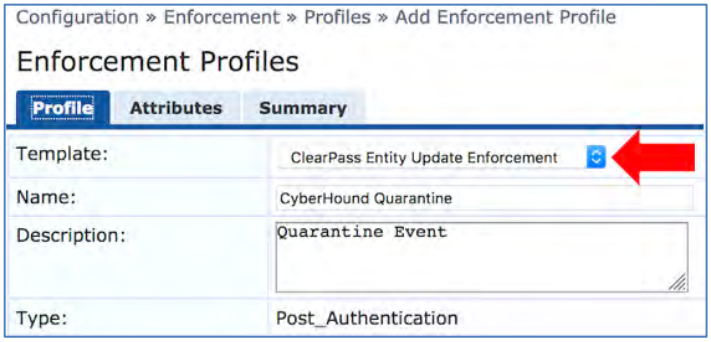


Figure 15: *Add Post Authentication Enforcement Profile*

Create three (3) enforcement profiles, one for each action sent by CyberHound (Quarantine, Block and Ignore). The Enforcement Profile will write attributes to the Endpoint repository which will be used to take action on the device.

Attribute	Value
Endpoint.CyberHound_Event_ThreatName	%{Event:CyberHound:threat}
Endpoint. CyberHound_Event_Action	Choose between Quarantine or Block
Endpoint. CyberHound_Event_DateTime	%{Event:CyberHound:eventdatetime}

Enforcement Profiles - CyberHound QUARANTINE

Summary | Profile | Attributes

Profile:

Name:	CyberHound QUARANTINE
Description:	Quarantine event
Type:	Post_Authentication
Action:	
Device Group List:	-

Attributes:

	Type	Name		Value
1.	Endpoint	CyberHound_Event_ThreatName	=	%{Event:CyberHound:threat}
2.	Endpoint	CyberHound_Event_Action	=	Quarantine
3.	Endpoint	CyberHound_Event_DateTime	=	%{Event:CyberHound:eventdatetime}

Enforcement Profiles - CyberHound BLOCK

Summary | Profile | Attributes

Profile:

Name:	CyberHound BLOCK
Description:	Block event
Type:	Post_Authentication
Action:	
Device Group List:	-

Attributes:

	Type	Name		Value
1.	Endpoint	CyberHound_Event_ThreatName	=	%{Event:CyberHound:threat}
2.	Endpoint	CyberHound_Event_Action	=	Block
3.	Endpoint	CyberHound_Event_DateTime	=	%{Event:CyberHound:eventdatetime}

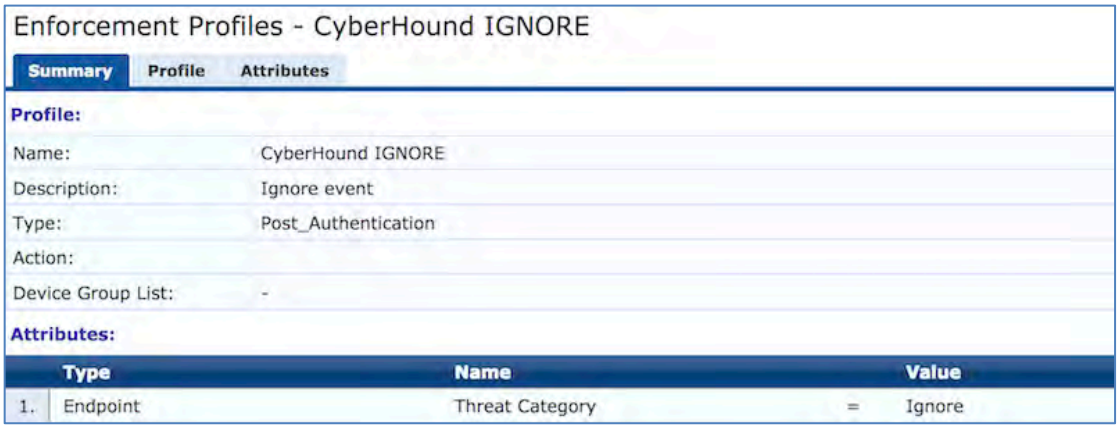


Figure 16: *ClearPass Enforcement Profile Examples*

Add a New Enforcement Policy

Create a new enforcement policy that will react to the "action" received from CyberHound. If the event action is Quarantine or Block then database update will occur and then a terminate message will also be sent to force the device to disconnect from the network.

When creating the Enforcement Policy, make sure you select the Enforcement Type of EVENT. See image below for clarification.

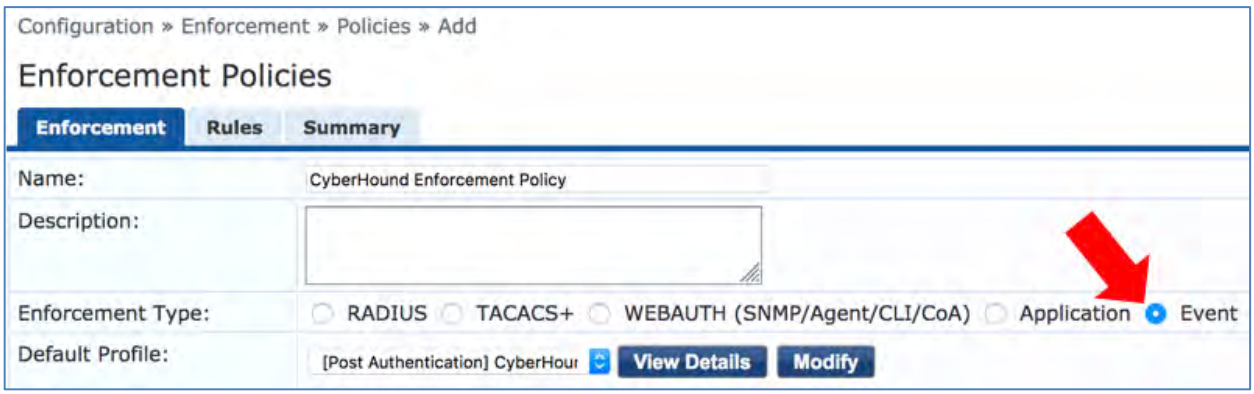


Figure 17: *ClearPass New Event Enforcement Policy*

Set the Default Profile to the "CyberHound IGNORE" Profile

Go to the Rules TAB and configure the policy rules to match the CyberHound Actions.

Figure 18: *ClearPass Event Enforcement Policy Rules*

Create Event Service Policy

Go to ***Configuration > Services > Add***. Select the type of "Event-based Enforcement"

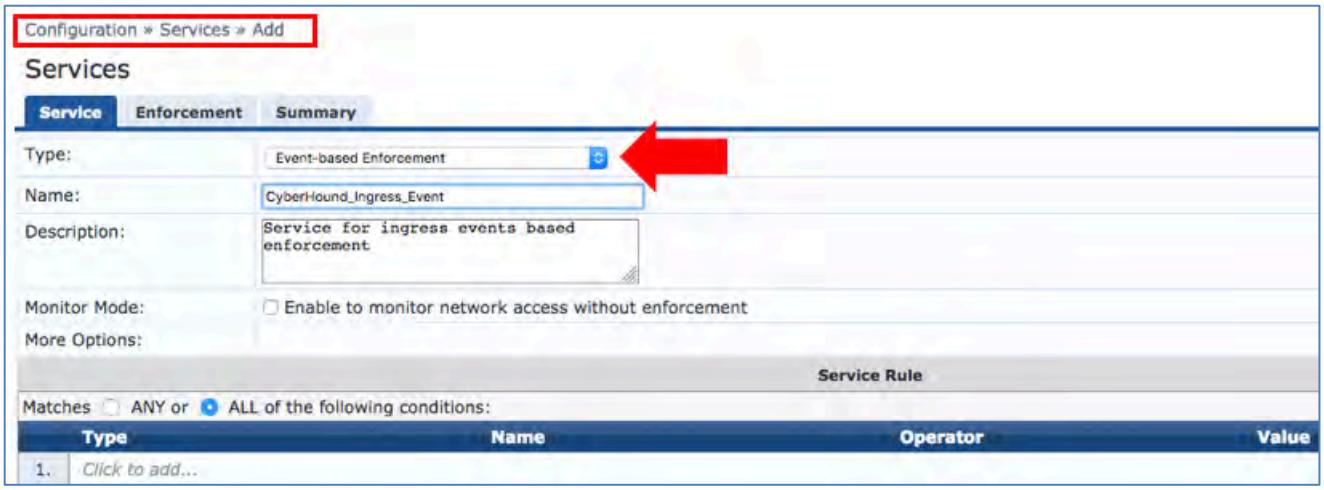


Figure 19: *ClearPass Event Service Policy*

Go to the Enforcement TAB and select the Enforcement policy you created earlier.

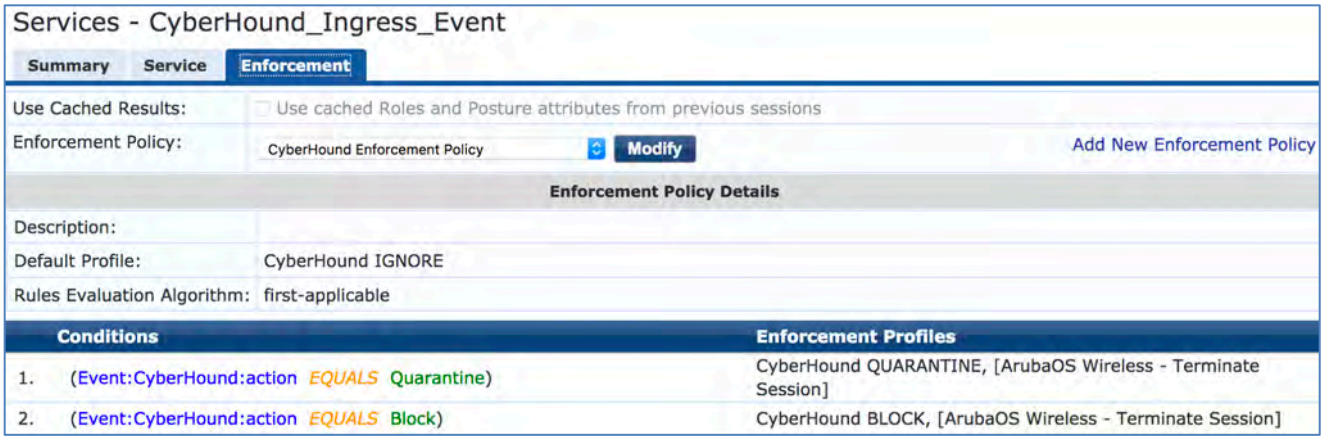


Figure 20: *ClearPass select the Enforcement Policy*

NOTE

When using the *EQUALS* operator in the condition statement, the text entered is the value is case sensitive.

Create an Authentication Service Policy

Use the ClearPass Service Template & Wizard to create a working authentication service policy that matches your requirements. Configuring the service policy from scratch is not detailed in this document.

For this example I have created a simple 802.1X Service Policy that authenticates users that are in the "staff" AD group.

Next, I will modify the Enforcement Policy to check the "***Endpoint.CyberHound_Event_Action***" field in the Endpoint Repository.

NOTE

Make sure you add the Endpoint Repository as an Authorization Source to your Service Policy.

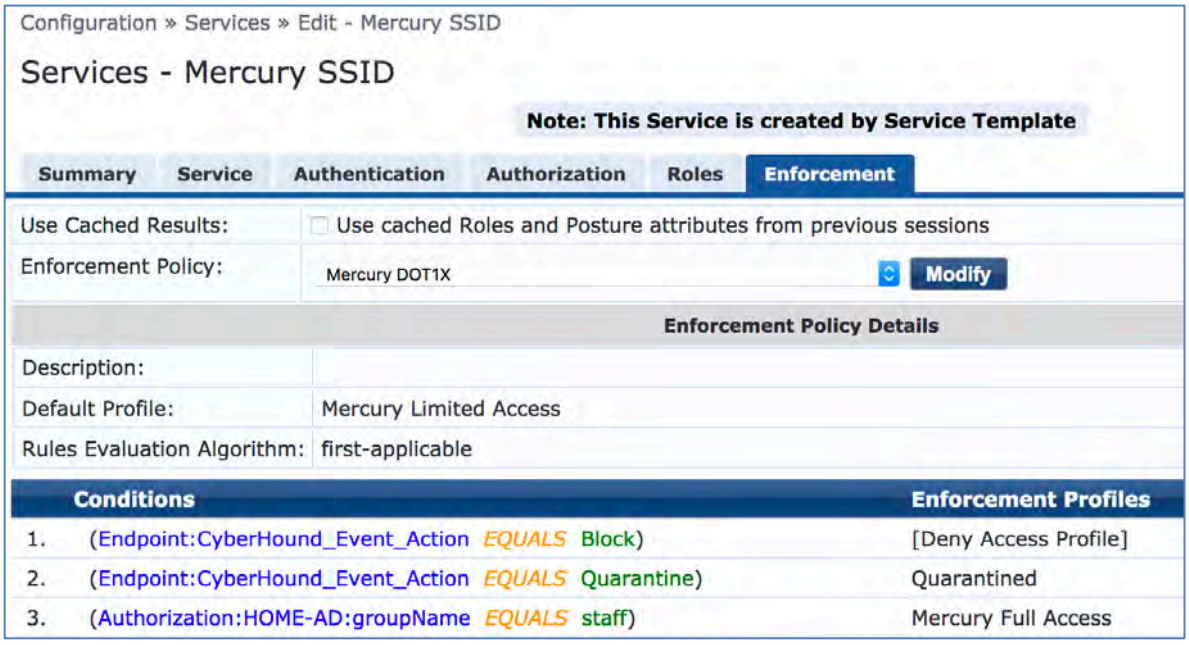


Figure 21: *ClearPass Authentication Service Policy Enforcement*

Creating a Quarantine Captive Portal Page

The result of quarantining clients on the network will vary depending on your specific requirements. In any case, the Enforcement Profile will return a radius attribute to the network access device which will place the device in a quarantined user role or maybe a quarantine VLAN.

The quarantine user role on an Aruba Mobility Controller or Aruba Instant AP may be configured to forward the user to a Captive Portal.

When redirecting users to a Captive Portal it is a good idea to display a message that identifies the issue so they can easily convey this information to the help desk.

Configuring a ClearPass Captive Portal

Our preference is to re-use one of the default pages to create the quarantine page.

1. Go to ***ClearPass Guest*** then select ***Configuration > Pages > Web Pages***.
2. Duplicate the "WebCC Blocked" page and call it **cyberhound_quarantined**.
3. Edit your new page and set the Title field to "Quarantined!".
4. Copy the following HTML code into the HTML field.

```
<link href="external/font-awesome/css/font-awesome.min.css" rel="stylesheet"
type="text/css">
<div style="width: 300px; margin: 0 auto; text-align: center;">
 <p>
  <!--
    refer to http://fortawesome.github.io/Font-Awesome/examples/ for icon docs
    -->
  <span class="fa-stack fa-lg" style="font-size: 100px; color: #FFF;">
   <i class="fa fa-cloud fa-stack-2x" style="color: #9FD4C9;"></i>
   <i class="fa fa-ban fa-stack-1x"></i>
  </span>
 </p>
 <h1 style="margin: 20px 5px;">Device Quarantine</h1>
 <p>
  You have been Quarantined due to the network traffic generated by your
device.</p><p>Please contact a network administrator to resolve this issue.
 </p>
 <p>
  <b>Device IP Address:</b> {$smarty.server.REMOTE_ADDR|escape}<br />
  <b>Device MAC Address:</b> {$_endpoint.mac_address|escape}<br />
  <b>Threat Name:</b> {$_endpoint.CyberHound_Event_ThreatName|escape}<br />
 </p>
</div>
```

Removing a Device from Quarantine

The example workflow created in this document relies on the **manual clearing** of the device before connectivity can be restored for the device in question. To do this, the ClearPass Administrator will be required to find the device in the EndPoint Repository using it's MAC address and then clear the "Threat" attributes.

Firstly, go to ***Configuration > Identity > Endpoints*** and search for the MAC Address.

Alternatively, you can list all endpoints with the **CyberHound_Event_Action** flag set by searching for an attribute.

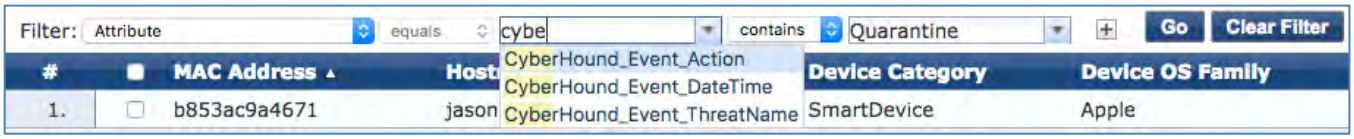


Figure 22: *ClearPass Find Endpoint*

Click the Endpoint and select the Attributes TAB. You will need to DELETE the Threat Attributes and bounce the device off/on the network.

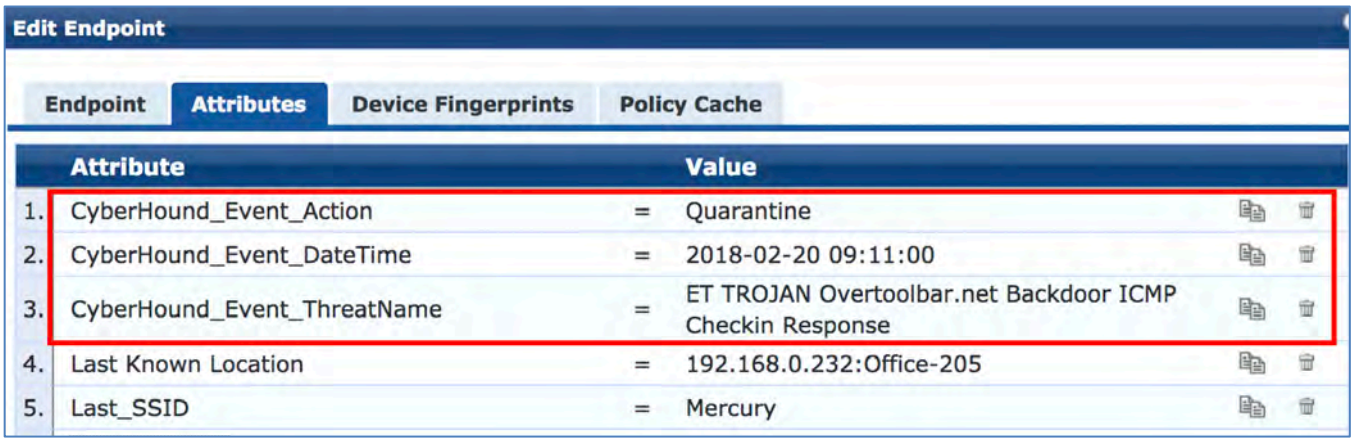
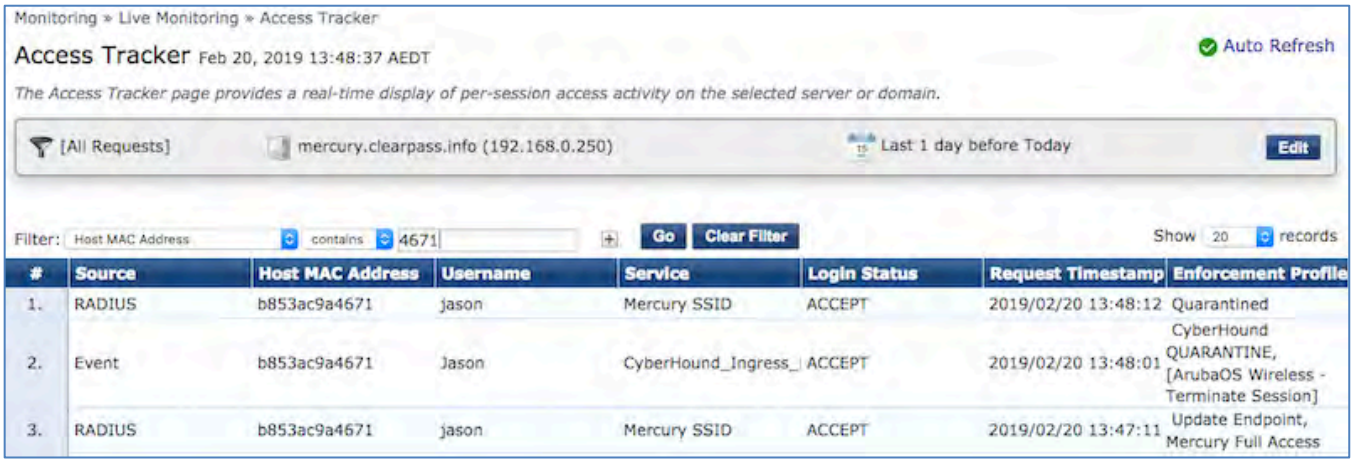


Figure 23: *ClearPass Endpoint Attributes*

Access Tracker Results

Access Tracker will display incoming Radius authentications and Events from CyberHound.



#	Source	Host MAC Address	Username	Service	Login Status	Request Timestamp	Enforcement Profile
1.	RADIUS	b853ac9a4671	jason	Mercury SSID	ACCEPT	2019/02/20 13:48:12	Quarantined
2.	Event	b853ac9a4671	Jason	CyberHound_Ingress_	ACCEPT	2019/02/20 13:48:01	CyberHound QUARANTINE, [ArubaOS Wireless - Terminate Session]
3.	RADIUS	b853ac9a4671	jason	Mercury SSID	ACCEPT	2019/02/20 13:47:11	Update Endpoint, Mercury Full Access

Figure 24: *ClearPass Access Tracker*

Let me explain the workflow depicted in the Access Tracker images above. The #number below indicates the line number for the messages seen above.

#3 – User 'jason' connects to the Mercury SSID using EAP-TLS and is granted "Mercury Full Access".

In the background a Radius Accounting Start message hits ClearPass to map the MAC Address 'b853ac9a4671' to IP Address '192.168.0.37'.

User 'jason' is going about his work until CyberHound notices some malicious traffic coming from his IP Address. CyberHound fires a syslog event at ClearPass

#2 – ClearPass receives an Event message via the IEE. The message has an action of 'Quarantine' and provides the IP Address of the device causing the issue.

ClearPass performs a lookup of the active sessions and returns the MAC Address (which can be seen in #2 above).

ClearPass writes the attribute 'Threat Category' = 'Quarantine' to the Endpoint DB.

ClearPass issues a Radius CoA to the Access Point to terminate the session for the MAC Address in question.

#1 – The device automatically reconnects to the Mercury SSID. The Endpoint Repository is used as an Authorization Source and the 'Threat Category' attribute will contain 'Quarantine'. The Enforcement Profile results in an Aruba-User-Role = 'Quarantine' being returned to the Access Point.

The Access Point has a User Role called 'Quarantine' which places the device in a Captive Portal role which points to the 'cyberhound_quarantined' web page.

Appendix 1 – IEE_CyberHound.xml file

The following text can be copied and saved as an XML file for upload to ClearPass as an Event Dictionary.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<TipsContents xmlns="http://www.avendasys.com/tipsapiDefs/1.0">
  <TipsHeader exportTime="Mon Feb 18 13:54:39 AEST 2019" version="6.7"/>
  <IngressEvents>
    <IngressEvent>
      <Vendor>CyberHound</Vendor>
      <Description>CyberHound Structured SysLog Format</Description>
      <FormatName>CyberHound-SIEM-Structured-Syslog</FormatName>
      <Format>syslog_priority,version,syslog_datetime,host1,host2,host3,var1,eventdatetime,action,infectedhostIP,threat</Format>
      <Prefix>CyberHound</Prefix>
      <Enabled>true</Enabled>
      <Sample>&lt;134&gt;1 10 Oct 12 11:40:03 swanlake ips-clearpass[18760]: {ips-clearpass} XI 2018-10-10 15:50:37 BLOCK 10.3.103.100 "dave" ET TROJAN Overtoolbar.net
Backdoor ICMP Checkin Request</Sample>
      <Filter>filter {
          grok {
                match =&gt; { 'message' =&gt;
'&lt;%{POSINT:syslog_priority}&gt;%{POSINT:version} %{TIMESTAMP_ISO8601:syslog_ts} %{DATA:host1} %{DATA:host2} %{POSINT:num1} - (?&lt;syslog_meta&gt;-
|(\[[^\[\]]*\])*) %{DATA:host3} %{DATA:var1} %{TIMESTAMP_ISO8601:eventdatetime} %{DATA:action} %{IPV4:infectedhostIP} "%{GREEDYDATA:username]" %{GREEDYDATA:threat}'}
                add_tag =&gt; [ "CyberHound" ]
          }
          if ("CyberHound" in [tags]) {
              mutate {
                add_field =&gt; [ 'Event:priority', '%{syslog_priority}' ]
                add_field =&gt; [ 'Event:version', '%{version}' ]
                add_field =&gt; [ 'Event:Username', '%{username}' ]
                add_field =&gt; [ 'Event:Source-IP-Address', '%{infectedhostIP}' ]
                add_field =&gt; [ 'Event:Event-Name', '%{threat}' ]
                add_field =&gt; [ 'Event:Pattern-Name', 'CyberHound-SS' ]
                add_field =&gt; [ 'Event:Timestamp', '%{syslog_datetime}' ]
                add_field =&gt; [ 'Event:device-vendor', 'CyberHound' ]
                add_field =&gt; [ 'Event:action', '%{action}' ]
               }
           ruby {
                  code =&gt; "
                      data = event.clone.to_hash;
                      data.each do |k,v|
                          if (!k.start_with?('Event:') and !k.start_with?('@'))
                              newFieldName = 'Event:CyberHound:'+ k
                              event[newFieldName] = v
                              event.remove(k)
                          end
                      end
                      tstamp = Time.now.to_i
                      tstamp_str = Time.at(tstamp).strftime('%Y-%m-%d %H:%M:%S')
                      event['Event:Timestamp'] = tstamp_str
                  "
              }
          }
      }</Filter>
      <FieldMapping>
        <Field AllowedValues="" DataType="Integer" Name="syslog_priority"/>
        <Field AllowedValues="" DataType="Integer" Name="version"/>
        <Field AllowedValues="" DataType="String" Name="syslog_datetime"/>
        <Field AllowedValues="" DataType="String" Name="host1"/>
        <Field AllowedValues="" DataType="String" Name="host2"/>
        <Field AllowedValues="" DataType="String" Name="host3"/>
        <Field AllowedValues="" DataType="String" Name="var1"/>
        <Field AllowedValues="" DataType="String" Name="eventdatetime"/>
        <Field AllowedValues="" DataType="String" Name="action"/>
        <Field AllowedValues="" DataType="IPv4Address" Name="infectedhostIP"/>
        <Field AllowedValues="" DataType="String" Name="threat"/>
      </FieldMapping>
      <GenericFieldMapping>
        <Field GenericName="Event-Name" Name="action"/>
        <Field GenericName="Timestamp" Name="eventdatetime"/>
        <Field GenericName="Source-IP-Address" Name="infectedhostIP"/>
      </GenericFieldMapping>
    </IngressEvent>
  </IngressEvents>
</TipsContents>
```