**ClearPass with Fortinet (Fortimanager) integration**

**Tim Friesen – ACDX 887 / ACCX 1175**

**Summary**

After going through the exercise of integrating ClearPass Policy Manager along with Fortimanager while referring to the Fortinet documentation I realized that their documentation lacks a lot of background information that would be helpful in setting up this integration, as well as notes on particular caveats that were encountered while testing. While the information provided by Fortinet is technically correct, my attempt here is to streamline it for someone setting something like this up for the first time, provide explanations for why certain things are set the way they are, as well as provide insight into any “gotcha!” moments where further clarification or particular configuration points are required in order to have a successful integration.

For further reference, here is the original document by Fortinet: <https://docs.fortinet.com/document/fortimanager/6.2.2/administration-guide/733863>

**Topology and Versions**

Our topology for this exercise is quite simple; we have CPPM running in a cluster with a VIP at 10.10.10.5, and Fortimanager running at 10.10.10.130. They do not have to be in the same VLAN, but for this document they are. You will also need a switch or wireless controller at some other address, added as device in Clearpass and a RADIUS service for some endpoint authentication. Setting that up is beyond the scope of this document.

For this howto, ClearPass is on version 6.9.1, Fortimanager is on version 6.4.x. The original Fortinet document uses CPPM 6.7.0.

**Required API Account**

You need an API account to be present on the CPPM cluster for this integration to work. You can create a local account, or you could authenticate against another source as well. Below we create an account locally.

Under Administration -> Admin Users, create a new account with the role “API Administrator.” This account will be used by the Fortimanager to connect to the CPPM API.



**API Service**

You’ll need to create a new Service for the Fortimanager API account to login to. Here’s how I set mine up:



The info above should be enough to get it going, but just in case...

The Fortimanager will login via OAuth2, so I just created a service rule to categorize only those logins with:

Type: Aruba Application Authentication

Application Name Equals OAuth2

Authentication Source: Admin User Repository (where we created the API account)

Role Mapping Policy: Not Required

Enforcement Policy: Use the built-in Guest Operator Logins policy.

Order the Service however you want. If you have other OAuth2 Services, make sure the Service rules will not conflict.

**Add Fortimanager as an Endpoint Context Server in ClearPass**

Navigate to Administration -> External Servers -> Endpoint Context Servers

Add the Fortimanager as shown below. I used the Fortimanager Admin account but another account with adequate permissions should work fine.



Now add some Context Server Actions at Administration -> Dictionaries -> Context Server Actions.

We need two, one for Login, and one for Logout. Enter the information as shown below for **Login**:



On the Header tab:



On the Content tab, enter the following:

Content-type: JSON

Paste the following text (without my arrow notes):

{

 "adom": "root", < ----- change to adom name where Clearpass connector is configured.

 "connector": "CPPM", < ----- change to name of the connector on Fortimanager

 "user": "%{Authentication:Username}",

 "role": "%{Tips:Role}",

 "ip-addr": "%{ip}"

}

For the Logout action, create another Context server action the same as the first, but modify the name and the URL…screenshot below:



Go back to the Fortimanager Endpoint Context Server, open it up and verify that the two actions are now on the Actions tab:



**Create an Enforcement Profile**

Now we need an Enforcement Profile to tag onto our Enforcement Policies to notify the Fortimanager that a user has been logged in/out when they authenticate.

Go to Configuration -> Enforcement -> Enforcement Profiles:

Add a new one of type “Session Notification Enforcement” and enter the information as shown:



This enforcement profile will be tagged to each line of your enforcement policy where you want to notify the Fortimanager that a user has been logged in. Here is an example from an 802.1X service:



**Create an API client**

Last part of the CPPM configuration side. Open the Guest dashboard and navigate to Administration -> API services -> API clients. We need to create an API client to allow access to the CPPM API from Fortimanager.

You can mirror the settings in the screenshot below or change the Client ID as you like, just make note of it as this will be used on the Fortimanager side.



That’s it for the Clearpass configuration.

**Fortimanager Setup**

Log into your Fortimanager appliance and select the ADOM where you want to install the ClearPass connector. Select Fabric View and click “Create New”, select type ClearPass. Make sure you use the ADOM that you specified in your Context Server Actions created earlier; or update it to match afterward.



Here we need to make sure some information matches across Fortimanager and CPPM.

Name: This is the name for the connector. It needs to match the connector name used in your Context Server Actions that we created earlier.

Status: ON (obviously!)

Server: IP of CPPM (VIP)

Client: this is the Client ID of the API client that we created in the last step (see previous page)

User/Password: This is the API user we created in the Admin User Repository on ClearPass.

Click OK. Then open the connector again and hit “Apply + Refresh.” The connector should reach out to ClearPass and download all of the roles from the cluster. On the CPPM side, you should see your Oauth2 service hit by Fortimanager using the API username that we created. You won’t see any NAD information though, but you can find the source IP under the Input tab Computed Attributes field. You can also update your OAuth2 service rules with the IP address if you like.



If you got this far and the Fortimanager downloaded the roles, everything should be good to go! You can now map those SSO groups from ClearPass into Local user Groups on the Fortigate, and start using them in policies.

**Special Notes/Caveats**

**ClearPass Roles and Fortimanager Sync**

The Fortimanager connector sync will grab all Roles that it finds configured on the CPPM cluster. For this reason, you should create Roles inside the Cluster at Configuration -> Identity -> Roles for every Role you intend to use in your policies. The connector will \*not\* grab any roles that are dynamically assigned via Authorization (ie. converting Active Directory groups into Roles automatically during policy evaluation).

**Role Mapping Policies**

A common issue with Fortigate integrations has been the ability for Fortigate to parse roles sent to it by CPPM when multiple roles result from policy evaluation. Although CPPM is sending the role information to the Fortigate, multiple roles are in the accounting packets. Fortigate will register the first role it sees, which can result in the user being placed into the wrong role at the Fortigate. This problem is exacerbated by the fact that using “Select all matches” in your Role Mapping Policy will ALWAYS result in the default role for the role mapping policy being tagged along with the authentication results.

For this reason, it is important to use only the “Select First Match” option on the Role Mapping policy, and carefully construct your role mapping statements such so that there is only 1 role result for every policy combination. This will result in only one role being sent to the Fortimanager for the session.

Other tricks are out there via Accounting Proxy, setting endpoint attributes and others, however they have been found to be less than totally reliable in getting the session information to the Fortimanager. Ensuring your Role Mapping Policies have only 1 result was found to be the most reliable way. (Note: The built-in [User Authenticated] and [Machine Authenticated] roles did not produce any conflict in testing.

**IP Addresses**

Lastly, the client IP address is also included in what is sent to the Fortimanager for authenticated sessions. On most wireless networks and controllers this information is readily available, however in the case of certain wired networks, this information may not be as easy to obtain. For instance, most current Aruba switches contain the “ip client-tracker trusted” feature which will help obtain the IP addresses of most connected devices. If you have a legacy switch that does not have this feature, DHCP snooping can help. In my testing, I was running a legacy Aruba switch that only had AOS-Switch firmware up to 16.02.xxxx, which lacked the ip client-tracker feature. Without client IP address information, the Fortimanager would not register the SSO session from the logged-on client. After enabling DHCP snooping, the client Framed-IP-Address attribute was present in subsequent RADIUS accounting updates, and was then forwarded to the Fortimanager, resulting in a proper SSO session.