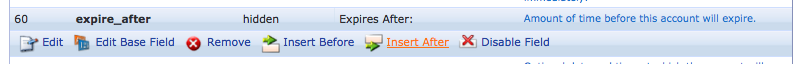
I guess that many of you have deployed guest WLANs where the password is sent over SMS to the user. By doing this we make sure whoever logs into the network has to provide a valid phone number that can be tracked in case there's been an improper use of the network. An interesting alternative could be to verify the email address of the guest using the sponsor approval workflow. The user would have limited access to the network while he validates his email and full access once that's done.

This is how I've been doing it lately:

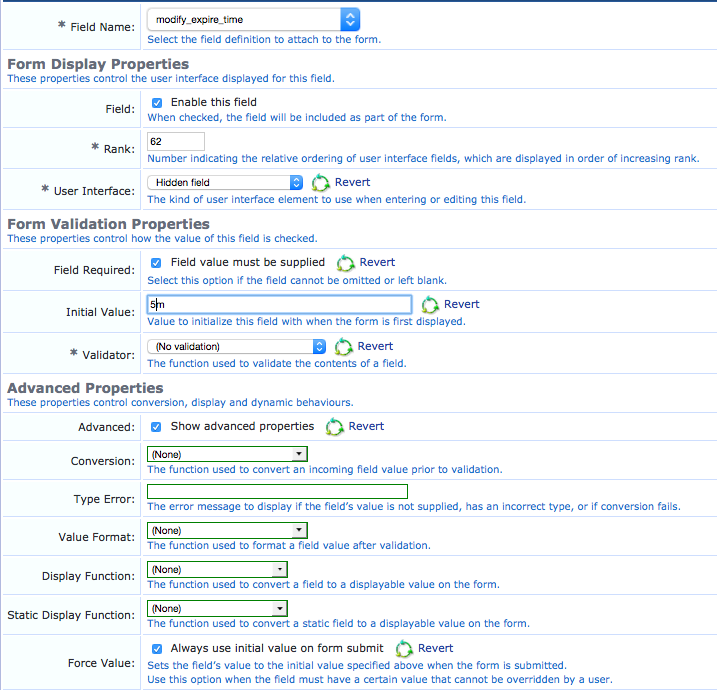
**Clearpass Guest Configuration**

I've created a self-registration where the guest gets 5 minutes of internet access to validate it's email address. This lets the guest receive the email to self-sponsor himself. The config process would be the following:

First of all, modify expire time so that the initial duration is 5 minutes. I've used the "modify\_expire\_time" field and set it to 5m:

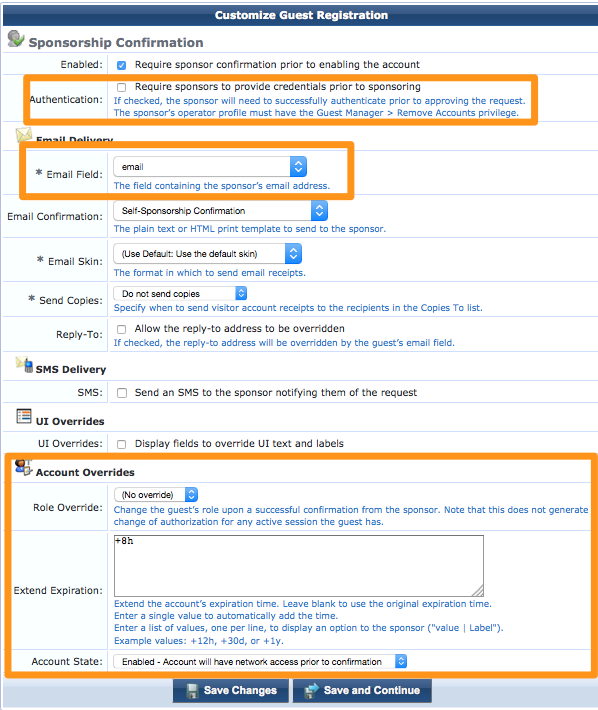


I first click "Insert\_After" to add a new field and then add the "modify\_expire\_time" field with the following configuration:



As you can see, it’s a hidden field with the initial value set to 5m and we’re forcing it to always use its initial value.

Next step would be to send a “self-sponsor” email to the user. In order to do that we configure “sponsor approval” with no sponsor auth, send approval request to the user’s own email, an 8h increment if the account duration and we set the initial state to “enabled”:

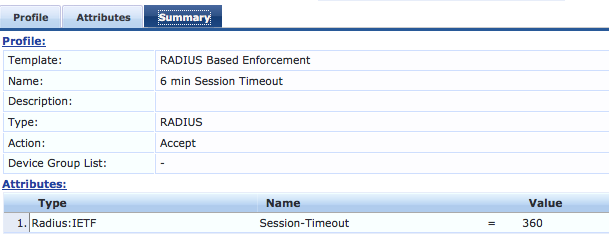


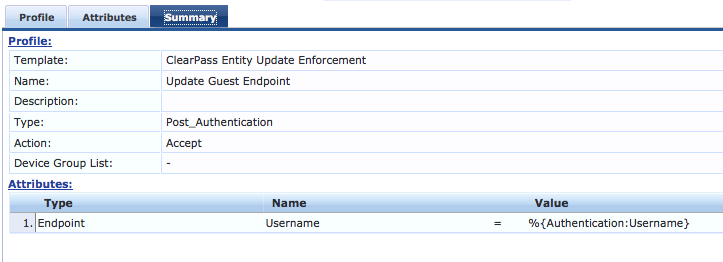
This would be the basic config we would need in Clearpass Guest. It admits a lot of small imprivements such as modifying the confirmation email to have a more appropriate text or the page seen by the “self-sponsor”. In order to keep things simple, we’ll leave those out for the moment.

**ClearPass Policy Manager Configuration**

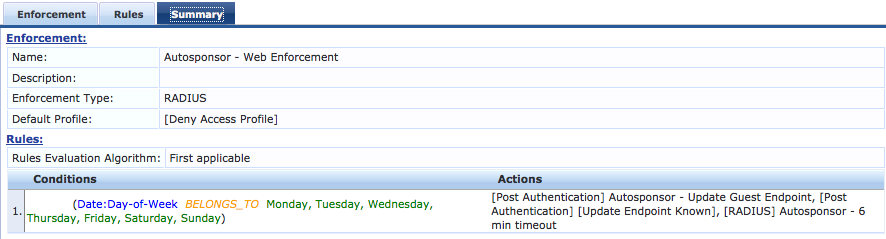
What we’ll do in CPPM is similar to what we do when configuring MAC Caching with the only difference that after the web authentications a new authentication will be triggered after 5 minutes. In order to do that, we’ll create 2 services, one for the RADIUS authentication from the web login and one for the subsequent MAC authentications.

Let’s focus first on the web authentication service. We need a generic RADIUS auth service where we validate the auth comes from the right SSID and so on. The key part relies in the Enforcement policy, were we’ll have the following:

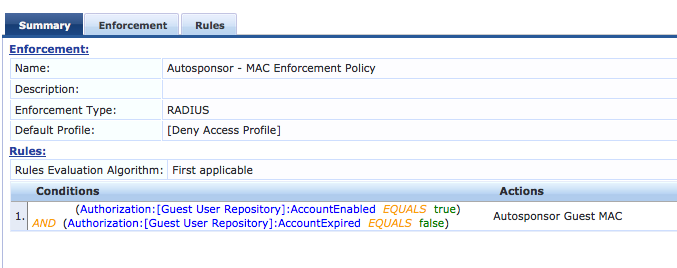
* [Update Endpoint Known] to mark the device as known.
* 5-6 Min Session timeout to expire the user session after 5 minutes have passed. This would be done in the following way:
* Update Guest Endpoint to save the user data in the endpoint



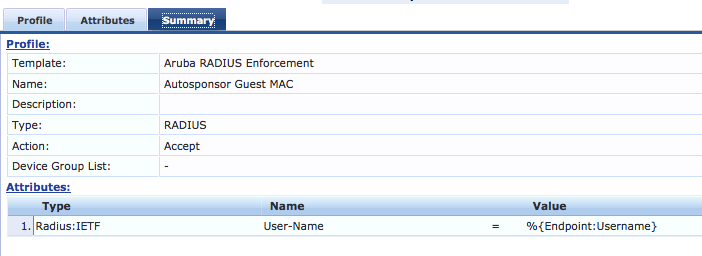
With this configuration, our enforcement profile should look more or less like this:



Now that the web authentication is set, we just need to take care of the MAC authentication service. This will allow us to cache the device for as long as we want the user session to be open. We just need to validate that the guest account tied to the endpoint exists and **is not expired**. The enforcement policy should look like this:



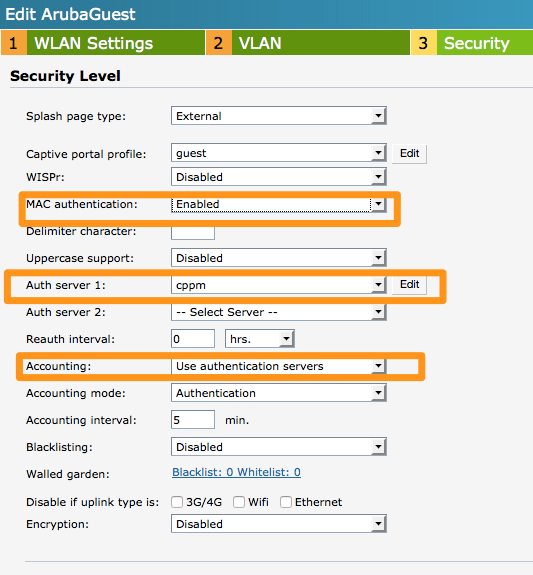
There are two key aspects here. First of all, we’ll need to add the [Guest User Repository] as an authorization source and second, we need to send the username back to the controller/ap. We don’t want the MAC address to appear as the username, we want the user id from the initial web login (which we’ve saved in the endpoint). This allows us to track the user session even when he’s being cached:



This would be all the required configuration in Clearpass, now we just need to take care of the AP/controller config.

**AP Configuration**

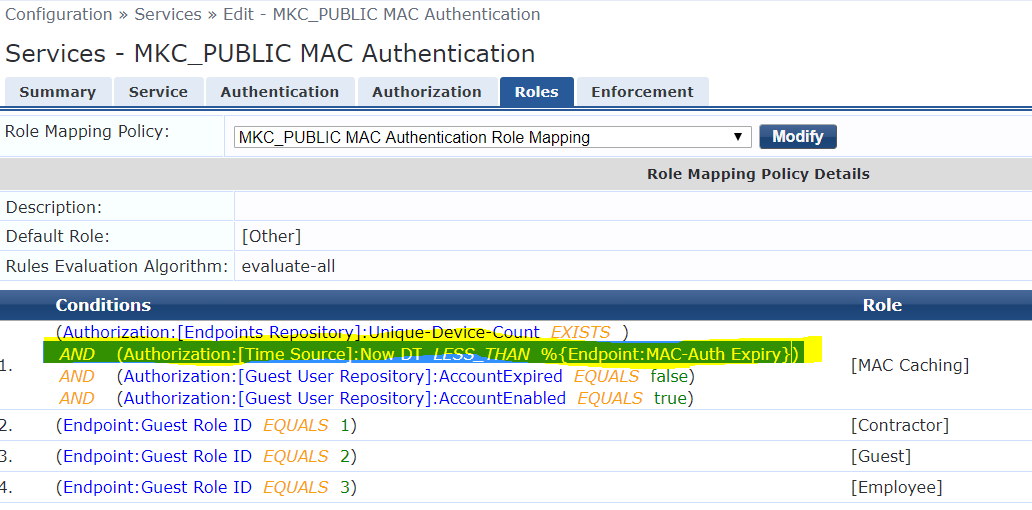
Since we plan on validating users based on their MAC addresses, we need to add MAC authentication to the guest authentication we usually have in these type of scenarios. This is how my IAP config looks like:



As you can imagine, this is just a sample configuration that will allow you to get started on this, and it leaves room for a lot of minor improvements. Nevertheless, I think this sets some basic foundations for the auto-sponsor login that could be useful to many. Give it a go and tell us what you think about it :)

If you have any issues with client timeout try the following:

Remove the following condition:



Then test again and see if users are able to connect without their session getting timeout.

Guest Manager:

Changed value to 2 active sessions

Expire action: disable and logout at specified time.

Enable insight master and insight

“MAC caching policy” removed session timeout limit from enforcement policy on mac caching service