Clearpass MAC Caching Service

Summary
This article describes an alternative MAC Caching service for Clearpass. Although the MAC Caching Service created by the service templates works fine, some find it difficult to comprehend and do not want to depend on Insight as authorization source.

The MAC Caching service discussed here does not use Insight as authorization source. Instead, it makes use of an Endpoint attribute containing the MAC expiry date. This attribute is checked against the authentication date. If the authentication date is before the Expiry date then access is granted, otherwise denied (or redirected to a captive portal).

In this article we assume two types of users for which MAC caching is enabled:
- Guests: users defined in, and authenticated against the Guest User Database and have the role [Guest]. The MAC Expiry will be set to the Guest Account Expiry
- Employees: defined in, and authenticated against an external database, like Active Directory and have the role [Employee]. The MAC expiry will be set to a fixed interval, for example 6 Months.

The flow will be discussed in 'reverse order' and not in the configuration order. At the end of this article, the steps will be listed in the right order.

Description
This service makes use of an Endpoint attribute holding the MAC Cach expiry date.

Because this solution uses Endpoint attributes, care should be taken when using this solution with other systems updating Endpoint attributes. An API call to update an Endpoint attribute may not take into account existing Endpoint attributes. And example is MDM systems updating Endpoint objects.

MAC Authentication Policy
The policy will simply look like this:

![Policy Image]

The Policy will only allow authentications which have the role [MAC Caching].

If MAC Caching is applied, different enforcement profiles are used depending on the role. In the example above, an employee will have the aruba user-role 'MAC-Staff' applied and guest will have the aruba-user-role 'MAC-Guest' applied. This can be entirely customised according to the customer's policy and equipment.

The default profile is [Deny Access Profile] in the above example. Alternatively, the default profile can be set to an enforcement profile which enforces a captive portal. For Aruba controllers this can be achieved by returning an aruba-user-role='guest-logon' for example.
Role Mapping policy

As you can see, the Role Mapping uses a couple of new attributes to determine if the role [MAC Caching] is assigned.

Endpoint Attribute

%{Endpoint:MAC-Auth Expiry} is a new attribute defined in the Endpoint. Goto Administration -> Dictionaries - Attributes and add an Endpoint attribute as below:

This attribute is updated by a Post Authentication Enforcement Policy in the Policy of the Web Login Service.

Post Authentication Enforcement Profiles

For Guests, the MAC Expiry will be set to the same value as the Guest Account Expiry:

Enforcement Profiles - BvZ Guest MAC Caching

Note that ‘ExpireTime’ needs to be added to the the [Guest User Repository]. More about that later.

For Employees, authenticating against another auth source, the account expiry is not available. Therefore the MAC Expiry will be set to a fixed interval determined by the customer's security policy. In this example, the customer has decided that MAC addresses for employees are allowed to be cached 6 months after the Web Login.
Clearpass MAC Caching Service

In the above example, the MAC Expiry is set to a fixed interval after the Web login authentication time. See hereafter.

### Authentication/Authorization Sources

`%{Authorization:[Time Source]:Today}` is a new attribute defined in the Authentication Source `[Tme Source]`.

The attribute **Today** is defined as:

The SQL: `select localtimestamp(0) as today;`

The attribute `Six Months From Now` is defined as:
The SQL: `select localtimestamp(0) + interval '6 months' as sixmonths;`

You can define other intervals as you wish by changing the interval in the SQL Query. For example if you want to set the MAC Auth Expiry to 7 days, the SQL query will be like:

`select localtimestamp(0) + interval '7 days' as sevendays;`

Next map the 'sevendays' to the Alias "Seven Days From Now" for example.

As mentioned earlier, the Guest User Account Expiry time needs to be made available from the [Guest User Repository]:

Add the highlighted string (expire_time::timestamp) to the existing Authentication query and map this to Alias ExpireTime as shown below:

**Putting it all together.**

- Add the Endpoint attribute MAC-Auth Expiry
- Add the ExpireTime attribute to the authentication source [Guest User Repository]
- Add the attributes today and a fixed interval attribute to the Authentication source [Time Source]
- In the existing Web Login Service, add the post authentication enforcement to update the Endpoint attribute MAC-Auth Expiry
- In the existing Web Login Service, add [Time Source] as an authorization source. You can remove [Insight] as authorization source
- Create the MAC Authentication policy:
• Ensure the Authentication source [Time Source] is added as an authorization source