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 configuation order. At the end of this article, the steps will be listed in the right order

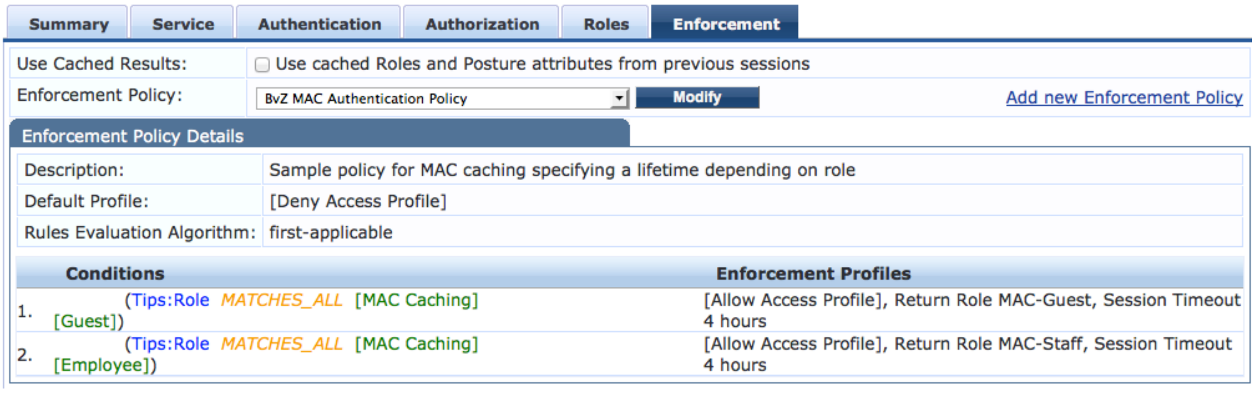
Description

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:



Use Cached Results: ☐ Use cached Roles and Posture attributes from previous sessions

Enforcement Policy: BvZ MAC Authentication Policy

Enforcement Policy Details

Description:	Sample policy for MAC caching specifying a lifetime depending on role
Default Profile:	[Deny Access Profile]
Rules Evaluation Algorithm:	first-applicable

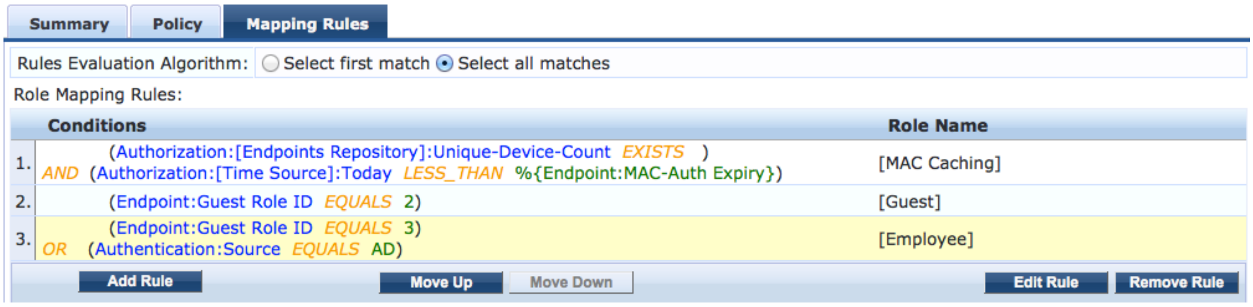
	Conditions	Enforcement Profiles
1.	(Tips:Role MATCHES_ALL [MAC Caching] [Guest])	[Allow Access Profile], Return Role MAC-Guest, Session Timeout 4 hours
2.	(Tips:Role MATCHES_ALL [MAC Caching] [Employee])	[Allow Access Profile], Return Role MAC-Staff, Session Timeout 4 hours

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 accodrding 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 enforcemnt 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



Rules Evaluation Algorithm: ○ Select first match ● Select all matches

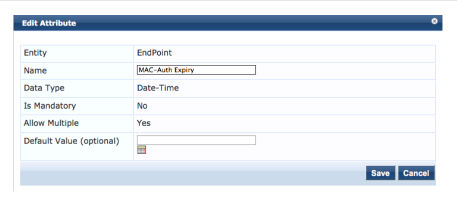
Role Mapping Rules:

	Conditions	Role Name
1.	(Authorization:[Endpoints Repository]:Unique-Device-Count EXISTS) AND (Authorization:[Time Source]:Today LESS_THAN %{Endpoint:MAC-Auth Expiry})	[MAC Caching]
2.	(Endpoint:Guest Role ID EQUALS 2)	[Guest]
3.	(Endpoint:Guest Role ID EQUALS 3) OR (Authentication:Source EQUALS AD)	[Employee]

As you can see, the Role Mapping uses a couple of new atributes 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:

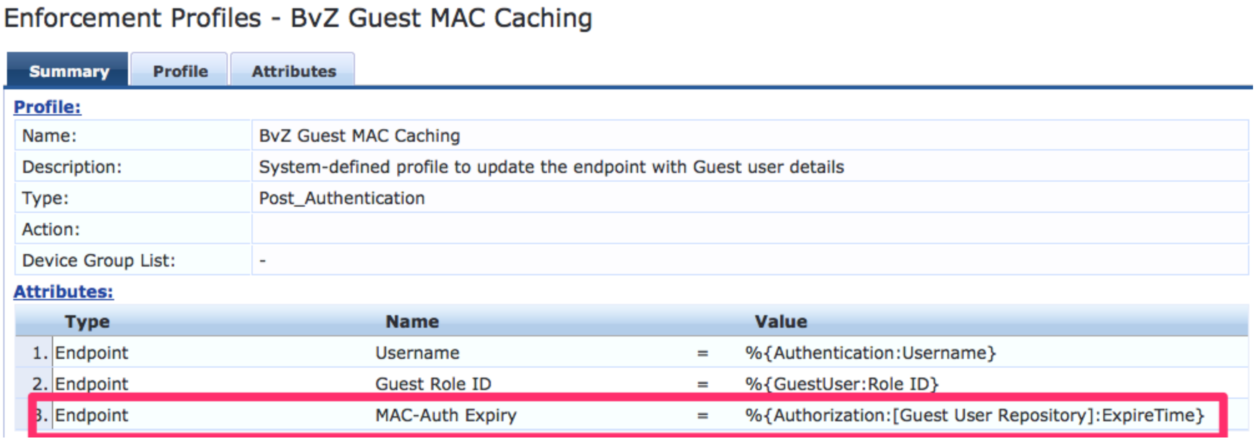


Entity	EndPoint
Name	MAC-Auth Expiry
Data Type	Date-Time
Is Mandatory	No
Allow Multiple	Yes
Default Value (optional)	

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

Profile:

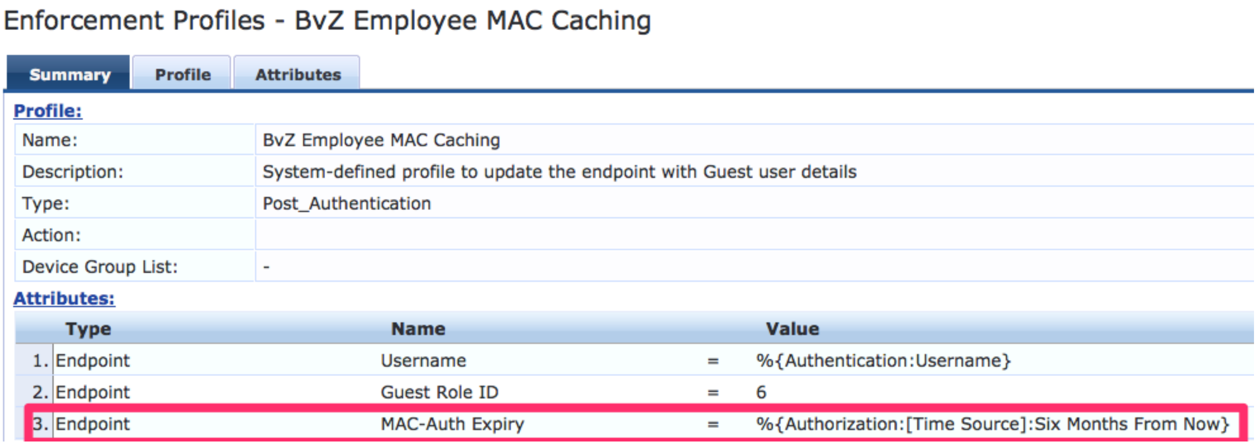
Name:	BvZ Guest MAC Caching
Description:	System-defined profile to update the endpoint with Guest user details
Type:	Post_Authentication
Action:	
Device Group List:	-

Attributes:

	Type	Name		Value
1.	Endpoint	Username	=	%{Authentication:Username}
2.	Endpoint	Guest Role ID	=	%{GuestUser:Role ID}
3.	Endpoint	MAC-Auth Expiry	=	%{Authorization:[Guest User Repository]:ExpireTime}

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.



Enforcement Profiles - BvZ Employee MAC Caching

Profile:

Name:	BvZ Employee MAC Caching
Description:	System-defined profile to update the endpoint with Guest user details
Type:	Post_Authentication
Action:	
Device Group List:	-

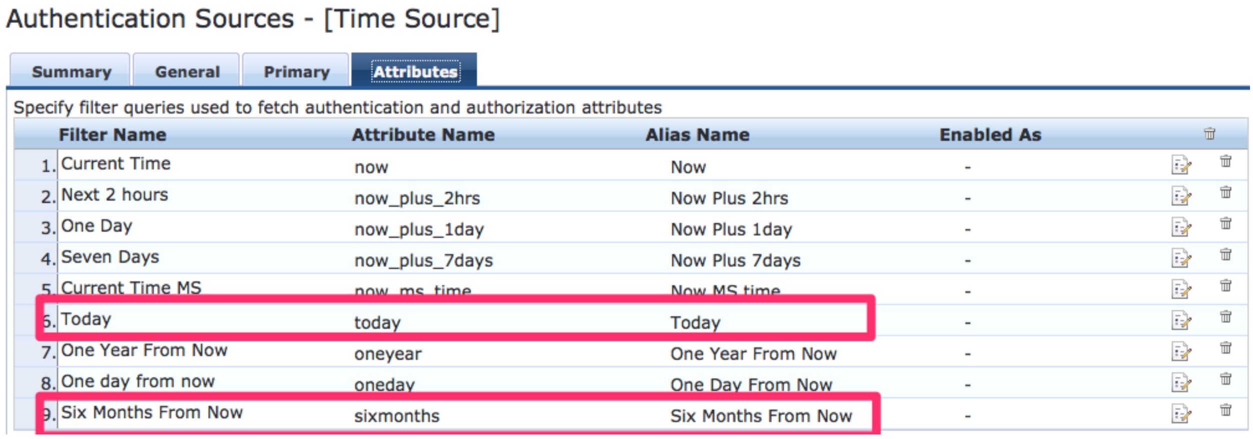
Attributes:

	Type	Name		Value
1.	Endpoint	Username	=	%{Authentication:Username}
2.	Endpoint	Guest Role ID	=	6
3.	Endpoint	MAC-Auth Expiry	=	%{Authorization:[Time Source]:Six Months From Now}

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].

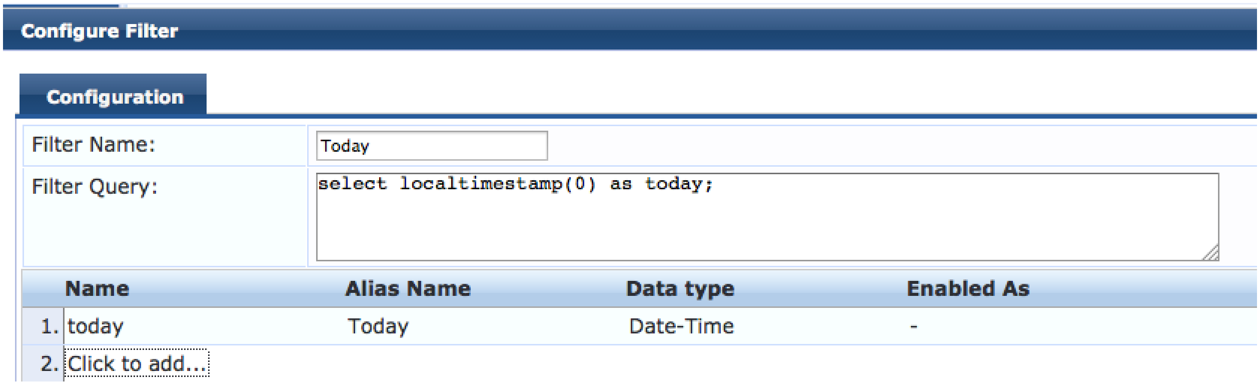


Authentication Sources - [Time Source]

Specify filter queries used to fetch authentication and authorization attributes

	Filter Name	Attribute Name	Alias Name	Enabled As
1.	Current Time	now	Now	-
2.	Next 2 hours	now_plus_2hrs	Now Plus 2hrs	-
3.	One Day	now_plus_1day	Now Plus 1day	-
4.	Seven Days	now_plus_7days	Now Plus 7days	-
5.	Current Time MS	now_ms_time	Now MS time	-
6.	Today	today	Today	-
7.	One Year From Now	oneyear	One Year From Now	-
8.	One day from now	oneday	One Day From Now	-
9.	Six Months From Now	sixmonths	Six Months From Now	-

The attribute **Today** is defined as:



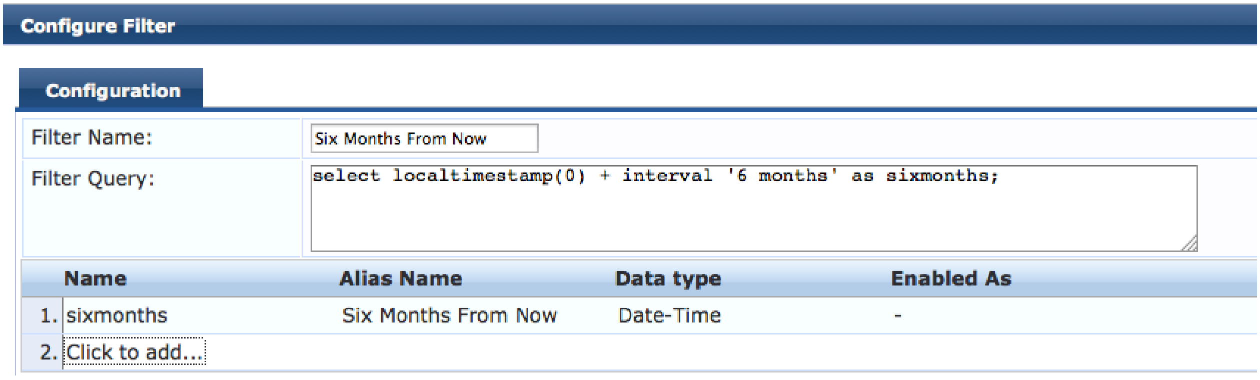
Configure Filter

Filter Name:	Today
Filter Query:	select localtimestamp(0) as today;

	Name	Alias Name	Data type	Enabled As
1.	today	Today	Date-Time	-
2.	Click to add...			

The SQL: select localtimestamp(0) as today;

The attribute ' *Six Months From Now*' is defined as:



Filter Name:	Six Months From Now
Filter Query:	select localtimestamp(0) + interval '6 months' as sixmonths;

	Name	Alias Name	Data type	Enabled As
1.	sixmonths	Six Months From Now	Date-Time	-
2.	Click to add...			

The SQL: select localtimestamp(0) + interval '6 months' as sixmonths;

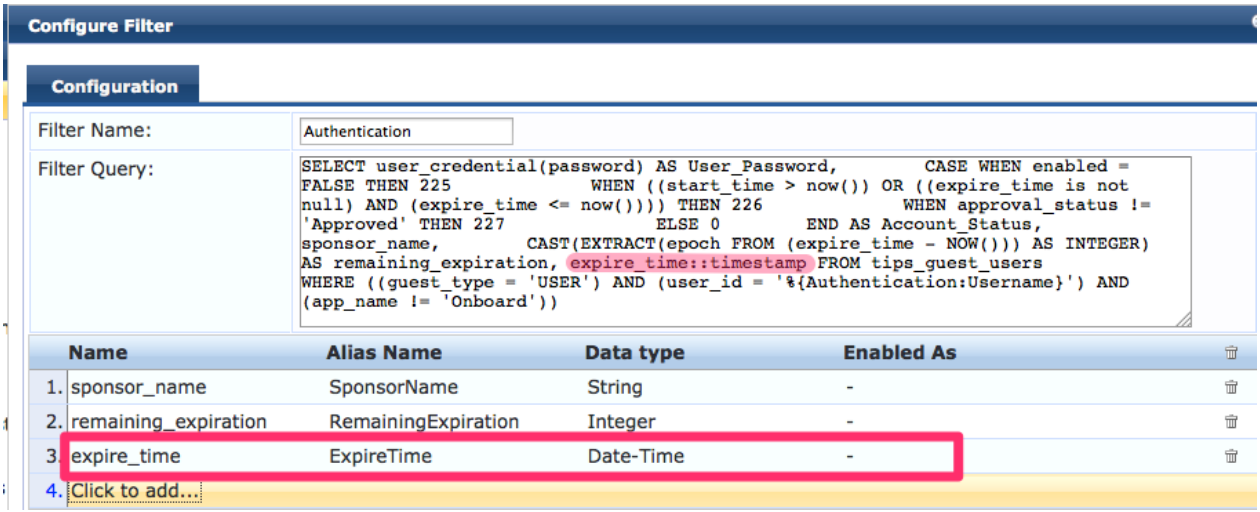
You can define other intervals as you wish by changing the interval in the SQL Query. For example if you want to set the MAC Auth Expiry to 7 days, the SQL query will be like:

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 Acount Expiry time needs to be made avaiable 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:



Filter Name: Authentication

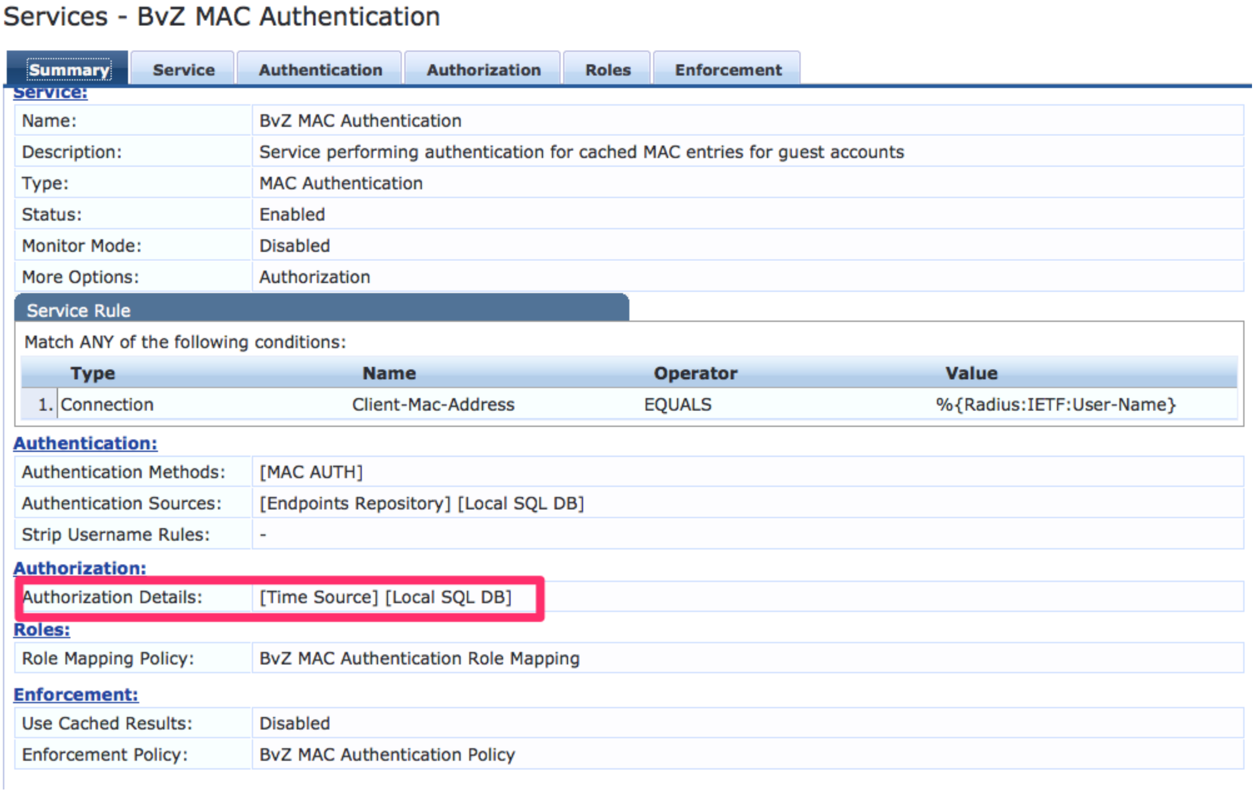
Filter Query:

```
SELECT user_credential(password) AS User_Password,        CASE WHEN enabled =
FALSE THEN 225              WHEN ((start_time > now()) OR ((expire_time is not
null) AND (expire_time <= now()))) THEN 226            WHEN approval_status !=
'Approved' THEN 227             ELSE 0        END AS Account_Status,
sponsor_name,        CAST(EXTRACT(epoch FROM (expire_time - NOW())) AS INTEGER)
AS remaining_expiration, expire_time::timestamp FROM tips_guest_users
WHERE ((guest_type = 'USER') AND (user_id = '%{Authentication:Username}') AND
(app_name != 'Onboard'))
```

	Name	Alias Name	Data type	Enabled As
1.	sponsor_name	SponsorName	String	-
2.	remaining_expiration	RemainingExpiration	Integer	-
3.	expire_time	ExpireTime	Date-Time	-
4.	Click to add...			

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 enforment 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 Athentication policy:



Services - BvZ MAC Authentication

Service:

Name:	BvZ MAC Authentication
Description:	Service performing authentication for cached MAC entries for guest accounts
Type:	MAC Authentication
Status:	Enabled
Monitor Mode:	Disabled
More Options:	Authorization

Service Rule

Match ANY of the following conditions:

	Type	Name	Operator	Value
1.	Connection	Client-Mac-Address	EQUALS	%{Radius:IETF:User-Name}

Authentication:

Authentication Methods:	[MAC AUTH]
Authentication Sources:	[Endpoints Repository] [Local SQL DB]
Strip Username Rules:	-

Authorization:

Authorization Details:	[Time Source] [Local SQL DB]

Roles:

Role Mapping Policy:	BvZ MAC Authentication Role Mapping

Enforcement:

Use Cached Results:	Disabled
Enforcement Policy:	BvZ MAC Authentication Policy

- Ensure the Authentication source [Time Source] is added as an authorization source