Building Apache HTTPD development module on RHEL /CentOS/SL 6 manually

The Apache HTTP server is the Apache Software Foundation's web server. See the project's website for more details.

Contents

> ⊘ **Apache Moonshot module information**
>
> These instructions relate to manually building the Apache Moonshot module (development version). This version provides support for exporting the SAML attributes received from the IDP as environment variables, thus making them available for the web applications (e.g. OpenStack).

> ⚠ All of the instructions below assume that you have root access, and will work as the root user (either directly or using sudo).

1. System Preparation

1.1. Add the Moonshot libraries

If you have not already done so, you first need to follow the instructions on how to install the Moonshot Libraries on RHEL/CentOS/SL 6.

1.2. Install some prerequisites

Building the Apache mod_auth_gssapi module requires you to have several packages already installed on the machine. To install them:

```
$ yum install autoconf automake krb5-devel git httpd httpd-devel gcc
```

1.3. Build the module

We are now ready to build the Apache module.

1. Get a copy of the code via git:

```
$ git clone https://bitbucket.org/umujisc/mod_auth_kerb.git
```

2. Enter the directory that just got created:

```
$ cd mod_auth_kerb
```

3. Run autoreconf:

```
$ autoreconf -fi
```

4. Build the software:

```
$ ./configure --sysconfdir=/etc --prefix=/usr && make && make install
```

> ✅ **Tip**
>
> This will install the module to `/usr/lib/httpd/modules`. On 64-bit platforms, you should run the following configure command in step 6:
>
> ```
> $ ./configure --sysconfdir=/etc --prefix=/usr --libdir=/usr/lib64
> ```

2. Installation Instructions

1. To enable the Apache module, add the following line to `/etc/httpd/conf/httpd.conf`:

   ```
   LoadModule auth_gssapi_module /usr/lib64/apache2/modules/mod_auth_gssapi.so
   ```

2. Ensure that the certificates referenced in `/etc/radsec.conf` can be read by the Apache user:

   ```
   $ su - --shell=/bin/bash apache
   $ cat path_to_ca.pem
   $ cat path_to_client.pem
   $ cat path_to_client.key
   ```

3. Verify that the `KeepAlive` option is enabled in the Apache configuration file `/etc/httpd/conf/httpd.conf`:

   ```
   KeepAlive On
   ```

4. Restart Apache:

   ```
   $ service httpd restart
   ```

3. Configuration Instructions

> ⊘ **Shibboleth2 Apache module incompatibility**
>
> Please note that this module is currently not compatible with the Shibboleth2 service provider Apache module. When testing or using the Moonshot module, disable the Shibboleth module and restart the webserver before attempting your test. We are attempting to resolve this problem.

3.1. Protecting a location with Moonshot

To protect a particular location on your Apache server, you must configure it with an AuthType of "Negotiate".

> ✅ **Example**
>
> To allow anyone with a valid Moonshot account to access `/wherever`, you would do the following:
>
> ```
> <Location "/wherever">
>     AuthType Negotiate
>     Require valid-user
> </Location>
> ```

3.2. Exporting GSS API attributes as environment variables

The module includes an option called GssapiNameAttributes that allows controlling which GSS API attributes (either SAML or RADIUS) are exported as environment variables. It is used as follows:

```
GssapiNameAttributes ENV_VAR_NAME ATTRIBUTE_NAME
```

This option can be specified multiple times, once for each attribute to expose. The Special value "json" is used to expose all attributes in a json formatted string via the special environment variable GSS_NAME_ATTRS_JSON.

Example:

```
<Location "/wherever">
    AuthType Negotiate
    Require valid-user
    GssapiNameAttributes json
    GssapiNameAttributes RADIUS_USER_NAME urn:ietf:params:gss:radius-attribute 1
    GssapiNameAttributes EPPN urn:ietf:params:gss:federated-saml-attribute urn:oasis:names:tc:SAML:2.0:attrname-
format:uri urn:oid:1.3.6.1.4.1.5923.1.1.1.6
</Location>
```

The special environment variable **GSS_NAME_ATTR_ERROR** is set with the GSS API returned error string in case the inquire name function fails to retrieve attributes, and with the string "0 attributes found", if no attributes are set.

In addition to this, in the event of an authentication failure, the module exports an environment variable called **MAG_ERROR** which contains one of the following values:

- *"NO_AUTH_DATA"* when the client did not send any authentication data (usually because the appropriate libraries are not installed on the browser).
- *"UNSUP_AUTH_TYPE"* when the client sent authentication data of an invalid type.
- *"GSS_MECH_ERROR"* when the GSS mechanism failed for some reason (e.g. invalid credentials).

Finally, whenever MAG_ERROR takes a value of "*GSS_MECH_ERROR*", an additional environment variable named **GSS_ERROR_STR** is sourced. This variable contains the result of the gss_display_status() call and may help web developers to show a more appropriate error page/string to the user.

3.3. Populating REMOTE_USER

Web services often rely on the REMOTE_USER Apache environment variable for user information, such as a local user account or a pseudonymous identifier.

To populate REMOTE_USER, update the reply from the RP Proxy with the User-Name RADIUS attribute in the RP Proxy's post-auth section:

```
update reply {
        User-Name := "content"
}
```

3.4. HTTPS Internet Explorer compatibility

For updated best practice with Internet Explorer connections, you should also read Microsoft's HTTPS and Keep-Alive Connections article.