Building Apache HTTPD 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.

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-workstation 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 http://www.project-moonshot.org/git/mod_auth_kerb.git
   ```

2. Enter the directory that just got created:

   ```
   $ cd mod_auth_kerb
   ```

3. Run autoconf:

   ```
   $ autoconf
   ```

4. Currently, there is a problem with the automake configuration. Until this is addressed, run `autoreconf` as follows:

```
$ autoreconf -vfi
```

5. The apxs script is not set as executable. Fix it manually:

```
$ chmod +x ./apxs.sh
```

6. 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, remove the comment from the below line in `/etc/httpd/conf/httpd.conf`:

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

2. Add a dummy Kerberos key to make the module happy:

```
$ ktutil
ktutil:  addent -password -p HTTP/localhost@YOUR-WEBSERVER-HOSTNAME -k 1 -e aes256-cts
<enter any password>
ktutil:  wkt /etc/httpd/krb5.keytab
ktutil:  quit
```

3. Export the location of the keytab file into Apache's config:

```
$ echo export KRB5_KTNAME=/etc/httpd/krb5.keytab >> /etc/httpd/envvars
```

> **Alternative**
>
> Alternatively, you can use the `GSSKrb5Keytab` configuration option in the `Location` directive in Section 3.1 to specify the keytab.

4. Assign the correct permissions to the keytab file:

```
$ chown apache.apache /etc/httpd/krb5.keytab
```

5. 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
```

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

```
KeepAlive On
```

7. 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. 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"
}
```

3.3. HTTPS Internet Explorer compatibility

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