

# Apache HTTPD on RHEL/CentOS/SL 6

The Apache HTTP server is the [Apache Software Foundation](#)'s web server. See [the project's website](#) for more details.

## Contents

- 1. System Preparation
  - 1.1. Turn off SELinux
    - 1.1.1. Temporarily
    - 1.1.2. Permanently
  - 1.2. Add the Moonshot libraries
- 2. Installation Instructions
- 3. Configuration Instructions
  - 3.1. Protecting a location with Moonshot
  - 3.2. Populating REMOTE\_USER
  - 3.3. Accessing Moonshot attributes
- 4. HTTPS Internet Explorer compatibility



## Apache Moonshot module information

These instructions relate to using the Apache Moonshot module package. To build the module manually, see the [Building Apache HTTPD module on RHEL/CentOS/SL 6 manually](#) page.



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. Turn off SELinux

Currently, Moonshot will not work while SELinux is in enforcing mode. Until we resolve this, simply turn SELinux to permissive mode. This can be done temporarily (i.e., on reboot it will be turned back on), or permanently (the change will persist).

#### 1.1.1. Temporarily

The following command will turn Enforcing mode off:

```
$ echo 0 > /selinux/enforce
```

#### 1.1.2. Permanently

Edit `/etc/sysconfig/selinux` and change "SELINUX=enforcing" to "SELINUX=permissive". Reboot the system.

### 1.2. 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](#).

## 2. Installation Instructions

1. To use the Apache module, install it and the MIT Kerberos client package:

```
$ yum install mod_auth_gssapi krb5-workstation
```

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 directive 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. If they cannot be read, add the Apache user to the group that has read access to the certificates:

```
$ usermod -a -G <group> apache
```

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

```
KeepAlive On
```

8. Restart Apache:

```
$ service httpd restart
```

## 3. Configuration Instructions



### Shibboleth2 Apache module incompatibility

Please read Section 6.2 in [Apache HTTPD](#) on module incompatibilities.

### 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".

The `/etc/httpd/conf.d/auth_gssapi.conf` file contains a sample configuration that can get you started.



### 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 FreeRADIUS 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. Accessing Moonshot attributes

The Moonshot module can use either the Shibboleth attribute resolver library to map RADIUS and SAML attributes to internal Shibboleth attributes, and then to environment variables, or use its own internal JSON attribute resolver to map either RADIUS attributes or SAML attributes to environment variables. Read more at [Configure a Linux Server's Attribute Resolution](#) about how to configure Shibboleth or the internal JSON attribute resolvers.

We are working on enhancements that allow the Moonshot module to expose attributes in the same way as the RedHat module.

## 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.