# Dave Mulford's Blog:Configuring Red Hat Enterprise Linux 7.4 and mod_auth_mellon for Microsoft ADFS

*Posted by Dave Mulford Jan 18, 2018*

This is part 4 in a series about configured Red Hat's mod_auth_mellon with Microsoft Active Directory Federation Services. It is assumed you've followed the previous steps in the posts listed below.

Part 1 - Active Directory Domain Services Setup
Part 2 - Active Directory Enterprise Certification Authority Setup
Part 3 - Active Directory Federation Services Setup

## Install the Necessary Packages

Install httpd, mod_ssl, and mod_auth_mellon:

```
# yum install httpd mod_ssl mod_auth_mellon
```

At the time of writing, the mod_auth_mellon scratch builds provided by John Dennis were used which adds the **MellonDiagnostics** and **MellonSignatureMethod** configuration options.

## Setup /etc/hosts to simplify hostname resolution

Setting up proper DNS on a Windows Domain Controller can be difficult. To make things easier, the /etc/hosts and the corresponding file in Windows (C:\Windows\System32\drivers\etc\hosts) were modified to provide local hostname resolution.

Here is my sample /etc/hosts file:

```
# cat /etc/hosts
127.0.0.1 localhost localhost.localdomain localhost4
localhost4.localdomain4
::1 localhost localhost.localdomain localhost6 localhost6.localdomain6

192.168.122.250 win-9jtgtdecjic win-9jtgtdecjic.mydomain.com
192.168.122.117 mellonprovider mellonprovider.mydomain.com
```

## Generate a Certificate Signing Request (CSR)

Use the commands below. When running the second command, be sure the **Common Name** you enter is your RHEL server's fully-qualified domain name (eg: mellonprovider.mydomain.com)

```
# openssl genrsa -des3 -out my_privkey.key 2048

# openssl req -new -sha256 -key my_privkey.key -out cert.csr
```

- Copy cert.csr to your Windows server. I used FileZilla to do so.

## Sign the CSR using the Active Directory Certification Authority

On your Windows 2012 R2 server, open a command window and run the following command. Be sure to specify the absolute path to cert.csr.

```
certreq -submit -attrib “CertificateTemplate:WebServer” C:\Users\Administrator\Downloads\cert.csr
```

Save the certificate using the window that is displayed after you enter the command above.

Copy the saved certificate back to your RHEL server. Save it, and the key, to the following locations.

- The certificate should be saved at: /etc/pki/tls/certs
- The key should be saved at: /etc/pki/tls/private

## Configure httpd to use the new certificate

Open /etc/httpd/conf.d/ssl.conf and change the following items:

- **ServerName** should be set to your fully-qualified domain name, like mellon.mydomain.com
- **SSLCertificateFile** should be the absolute path to the certificate file you saved, like /etc/pki/tls/certs/mellonserver.crt
- **SSLCertificateKeyFile** should be the absolute path to the key file you saved, like /etc/pki/tls/private/mellonserver.key

# Create and setup the mellon metadata on RHEL 7

Create the following directories on your RHEL server.

- The mellon directory will be used by mod_auth_mellon for storing session information.
- The private directory is what we will be protecting with mod_auth_mellon and ADFS.

Use this command to create the directories:

```
# mkdir /var/www/html/{mellon,private}
```

Create the directory that will be used to store the mellon metadata.

```
# mkdir /etc/httpd/saml2
```

Enter the directory you created in the previous step, then generate the mellon metadata. I used the mellon_create_metadata.sh script provided by the mod_auth_mellon package.

```
# cd /etc/httpd/saml2

# /usr/libexec/mod_auth_mellon/mellon_create_metadata.sh "https://mellon.mydomain.com" "https://mellon.mydomain.com/mellon"
```

Still in the /etc/httpd/saml2 directory, request the ADFS metadata. Change the windows hostname to match your environment.

```
# curl -kL -o https_win-9jtgtdecjic.mydomain.com.xml https://win-9jtgtdecjic.mydomain.com/FederationMetadata/2007-06/FederationMetadata.xml
```

Configure mod_auth_mellon to use the metadata. Here is my full /etc/httpd/conf.d/auth_mellon.conf:

```
MellonCacheSize 100

MellonLockFile "/run/mod_auth_mellon/lock"

<Location />

 MellonEnable info

 MellonEndpointPath /mellon/

 MellonDiagnosticsEnable On

 MellonSignatureMethod rsa-sha256

 # The mellon metadata

 MellonSPMetadataFile /etc/httpd/saml2/https_mellonprovider.mydomain.com.xml
 MellonSPPrivateKeyFile /etc/httpd/saml2/https_mellonprovider.mydomain.com.key
 MellonSPCertFile /etc/httpd/saml2/https_mellonprovider.mydomain.com.cert

 # The ADFS metadata
 MellonIdPMetadataFile /etc/httpd/saml2/https_win-9jtgtdecjic.mydomain.com.xml
</Location>

<Location /private>
 AuthType Mellon
 MellonEnable auth
 Require valid-user
</Location>
```

## Setup a Relying Party Trust in ADFS

Setup the corresponding /etc/hosts on your Windows server. That file is C:\Windows\System32\drivers\etc\hosts. Here is mine:

```
192.168.122.250 win-9jtgtdecjic win-9jtgtdecjic.mydomain.com
192.168.122.117 mellonprovider mellonprovider.mydomain.com
```

Start httpd on your RHEL server:

```
# systemctl start httpd.service
```

On your Windows server, open **Server Manager**, then go to **Tools** -> **AD FS Management** to open the **Active Directory Federation Services** window.

In the left pane, expand the **Trust Relationships** section, then select **Relying Party Trusts**.

In the right pane, click **Add Relying Party Trust** to open the Add Relying Party Trust wizard.

In Add Relying Party Trust wizard:

Click **Start** to get past the **Welcome** screen

On the **Select Data Source** screen, choose the **Import data about the relying party published online or on a local network** option and enter the mellon metadata URL as shown below. Remember to replace the server name with your mellon server's FQDN.

Example metadata URL:

https://mellonprovider.mydomain.com/mellon/metadata

Click **Next,** then click **OK** on the warning about some features not being supported.

On the **Specify Display Name** screen, click **Next** to accept the default values.

Click **Next** to accept defaults on the **Configure Multi-Factor Authentication** screen.

Click **Next** on the **Choose Issuance Authorization Rules** screen.

On the **Ready to Add Trust** screen, feel free to browse the tabs, particularly the **Endpoints** tab as this shows that the mellon metadata was read successfully. Click **Next** when you're ready.

On the **Finish** screen, click **Configure**.

## Add a Claim Rule to populate the transient NameID field requested by mod_auth_mellon

mod_auth_mellon requests that ADFS specify a value for the NameID field. This is requested as a transient value, not a persistent value. ADFS does NOT populate this field by default, so you must tell it to do so.

On your Windows server, open **Server Manager**, then go to **Tools** -> **AD FS Management** to open the **Active Directory Federation Services** window.

In the ADFS window, right-click the **Relying Party Trust** for your mellon provider and choose **Edit Claim Rules**.

Make sure the **Issuance Transform Rules** tab is active, then click the **Add Rule** button.

On the **Choose Rule Type** screen, choose **Transform an Incoming Claim** and click **Next**.

On the **Configure Claim Rule** screen, choose the following values.

- Claim rule name: Name your rule. I named mine "Send Windows Name as NameID"
- Incoming claim type: Windows account name
- Outgoing claim type: Name ID
- Outgoing name ID format: Transient Identifier
- Make sure that the **Pass through all claim values** option is chosen.

Click **Finish** to close the wizard, then click **OK** to close the **Edit Claim Rules** window.

I hope this series has helped you in setting up your own mod_auth_mellon and ADFS environment. I'm fairly sure that most of this can be automated using Ansible and Windows PowerShell, but that's for another day.

Feel free to reach out via the comments on any of the posts in this series!
47 Views  Tags: active-directory, mod_auth_mellon, windows-server

There are no comments on this post