

FERMILAB CENTRAL WEB HOSTING  
SINGLE SIGN ON (SSO) ON CWS LINUX  
WITH SAML AND MOD\_AUTH\_MELLON

## Contents

Information and Security Contacts: .....	3
1. Introduction .....	4
2. Installing Module .....	4
3. Create Metadata and Certificates .....	4
4. Create POST replay directory .....	5
5. Configure Apache .....	5
6. Handler URLs .....	5
7. Virtual Hosts .....	6
8. Registering with IdP .....	7
9. Protecting Directories .....	8
10. Testing .....	8
Appendix A: mellon_create_metadata.sh .....	9
Appendix B: Example Apache configuration .....	10

## Information and Security Contacts:

Title	Name	Email	Telephone	Initials
Service Owner / Service Provider	Peter J. Rzeminski II	ptr@fnal.gov	630.840.5524	
System Managers – Central Web Hosting	Andrew Duranceau John Inkmann	adurance@fnal.gov inkmann@fnal.gov	630.840.6457 630.840.6508	
Service Owner – AD	Al Lilianstrom	lilstrom@fnal.gov	630.840.2028	
System Managers – AD	Olga Terlyga	terlyga@fnal.gov	630.840.4685	
Management Contact	Jon Bakken (Division) Michael Rosier (ESO Dept) Peter J. Rzeminski II (Group)	bakken@fnal.gov mrosier@fnal.gov ptr@fnal.gov	630.840.4790 630.840.8385 630.840.5524	

## 1. Introduction

Unlike `mod_shib` which requires the use of a standalone daemon, `mod_auth_mellon` is a pure Apache module. All of the configuration is done through Apache and `.htaccess` with the exception of a directory to store certificates and IdP metadata.

Before getting started with `mod_auth_mellon`, it is best to understand the flow of a SAML login.

For this example, a user visits `https://example.fnal.gov/internal/index.html` which has been protected using `mod_auth_mellon` in the `.htaccess` file.

The Apache module intercepts this request and checks the value of a cookie called `mellon-cookie`. If valid, the request is allowed to continue and the page is displayed. The SAML attributes are provided to that page in the form of HTTP header environment variables. If the cookie is missing or expired, the user is redirected to the IdP's login page.

When the user submits their credentials to the IdP, it redirects back to a handler page on the original server with a POST request containing the SAML attributes.

The module steps in again and verifies the signature of the POST data. If it's genuine, `mellon-cookie` is set and the user is redirected back to the original requested URL.

It is also possible to manually trigger the login sequence. See [6. Handler URLs] for more information.

The official `mod_auth_mellon` documentation is located at [https://github.com/UNINETT/mod\\_auth\\_mellon](https://github.com/UNINETT/mod_auth_mellon).

When debugging login sequences, the Firefox plugin "SAML tracer" is very useful.

## 2. Installing Module

The `mod_auth_mellon` module is available through `yum` on RedHat 6.6 and later.

To install the module, simply run:

```
$ sudo yum install mod_auth_mellon.x86_64
```

The module should automatically be enabled in Apache. If not, load it with the following directive in your Apache configuration:

```
LoadModule auth_mellon_module modules/mod_auth_mellon.so
```

## 3. Create Metadata and Certificates

SAML uses XML metadata to define relationships between the IdP (authentication service provider) and the SP (protected web zone). The metadata contains the public key portion of a certificate used to verify authentication requests originating from your server. Before you can register your server with an IdP, metadata and certificates must be created.

The script provided in Appendix A creates both the metadata and the certificates and places them in `/etc/mod_auth_mellon`. If the directory does not exist, it will be created.

To use the script, simply call the script with the full hostname. For Central Web, use the VIP address.

```
$ sudo ./mellon_create_metadata.sh FULLY.QUALIFIED.DOMAIN
```

For clustered systems, such as Central Web, the metadata should be generated on one server and then replicated to the other servers in the cluster.

For servers with virtual hosts on different domains, you may need a separate set of certificates and metadata for each domain name. In this case, call the script with the `-secondary` flag. The generated files will be placed in `/web/etc/mod_auth_mellon`.

```
$ sudo ./mellon_create_metadata.sh FULLY.QUALIFIED.DOMAIN -secondary
```

The IdP metadata is provided by the Authentication and Directory Services group. The links are included below for convenience. In most cases, you will use the production metadata.

PingFederate development metadata: <https://pingdev.fnal.gov:9031/files/pingdev-metadata.xml>

PingFederate production metadata: <https://pingprod.fnal.gov:9031/files/pingprod-metadata.xml>

This should be placed in `/etc/mod_auth_mellon`.

For Central Web, it is located in `/web/etc/mod_auth_mellon`.

#### 4. Create POST replay directory

A directory on the server's disk is required to store POST data submitted from unauthenticated forms or forms with an expired session to a protected script. This directory should be owned and readable only by the apache user.

```
$ sudo mkdir /var/cache/mellon_post_data
$ sudo chown apache:apache /var/cache/mellon_post_data
$ sudo chmod 600 /var/cache/mellon_post_data
```

#### 5. Configure Apache

The Apache configuration consists of global settings that are used by all virtual hosts and web directories, as well as directory- or path-specific access control settings.

The global configuration is typically placed in `conf.d/auth_mellon.conf`. Most of these settings are contained in a `<Location />` block and are inherited by virtual hosts.

The example configuration in Appendix B should work as is on Fermilab systems.

Apache should be fully restarted the first time after saving the configuration changes. A graceful reload will cause Apache to hang. Subsequent changes can be applied with a graceful reload.

#### 6. Handler URLs

The next steps will require the use of `mod_auth_mellon` handler URLs. These URLs are always intercepted by the module and display no content. They will redirect on success or return a basic HTTP

error. Each SP has its own set of URLs. For most servers with only one SP, they are built on the server's hostname. For these examples we will use the server web1001.fnal.gov.

**Login Handler:** https://web1001.fnal.gov/mellon/login?ReturnTo=**URL**

This handler can be used to manually trigger the login sequence. A URL to redirect back to after successful login is required.

**Logout Handler:** https://web1001.fnal.gov/mellon/logout?ReturnTo=**URL**

This handler can be used to logout and invalidate the mellon-cookie. A URL to redirect back to after logout is required.

**Response Handler:** https://web1001.fnal.gov/mellon/postResponse

This handler receives the POST data from the IdP. If the data is genuine, mellon-cookie is set in the response.

**Metadata Handler:** https://web1001.fnal.gov/mellon/metadata

This handler displays the XML metadata needed by the IdP to create a relationship with your server.

You MUST always use HTTPS for handler URLs or undefined behavior will occur.

For clustered systems such as Central Web, the vip address should be used instead of the server address.

## 7. Virtual Hosts

For virtual hosts on the same domain as the server, no additional configuration is required.

However, consider the following example:

A user visits a protected page at http://www.scientificlinux.org which is hosted on the server web1001.fnal.gov. When the SAML login sequence takes place, mellon-cookie will be set for \*.fnal.gov. After the user is redirected back to the original URL, no cookie will be sent by the browser and an infinite loop will occur.

To get around this, any domains that cannot be matched by the default MellonCookieDomain will require their own set of metadata. See [3. Create Metadata and Certificates] for more information.

The metadata, certificate and cookie settings must be overridden for the VirtualHost.

```
<VirtualHost>
...
  <Location />
    # A wildcard is still used so multiple subdomains
    # can be access after a single login sequence.
    MellonCookieDomain .DOMAIN

    # To avoid collisions, the cookie name must also be unique
    # for this set of metadata
    MellonVariable DOMAIN

    # In a clustered system, this domain will be served
```

```
# by multiple servers. Therefore, the metadata must
# be stored in a centralized location.
MellonSPCertFile /web/etc/mod_auth_mellon/sp-cert_DOMAIN.pem
MellonSPPrivateKeyFile /web/etc/mod_auth_mellon/sp-key_DOMAIN.pem
MellonSPMetadataFile /web/etc/mod_auth_mellon/sp-metadata_DOMAIN.xml
</Location>
...
</VirtualHost>
```

## 8. Registering with IdP

Now that your server is almost ready to go, you must register it with the IdP. Use the Federation Services Interconnect form in ServiceNow under the Service Request Catalog.

* Requested for
<input type="text"/>
* Choose environment
-- None --
* URL for service provider page to federate
<input type="text"/>
Logout URL for service provider (optional)
<input type="text"/>
* Metadata URL
<input type="text"/>
Additional information (list of attributes required)
<input type="text"/>

In most cases, you should use the Production environment.

The URL is the HTTPS address of the server or cluster you are registering. For example:

<https://vip-webt1c01.fnal.gov>

The logout URL is the handler mentioned earlier, such as:

<https://vip-webt1c01.fnal.gov/mellon/logout>

If logout functionality is needed it should be requested under Additional Information. Note that logout applies to the entire SSO session, not just the sites on the server you are setting up.

The metadata URL is also provided by a handler:

<https://vip-webt1c01.fnal.gov/mellon/metadata>

The default Fermilab attributes are:

```
urn:oasis:names:tc:SAML:2.0:nameid-format:transient (NameID)
urn:oid:1.3.6.1.4.1.5923.1.1.1.6 (EPPN)
urn:mace:dir:attribute-def:mail (Email)
urn:oid:0.9.2342.19200300.100.1.3 (Email)
urn:mace:dir:attribute-def:givenName (First Name)
urn:oid:2.5.4.42 (First Name)
urn:mace:dir:attribute-def:sn (Last Name)
urn:oid:2.5.4.4 (Last Name)
urn:mace:dir:attribute-def:uid (UID)
urn:oid:0.9.2342.19200300.100.1.1 (UID)
urn:gluu:dir:attribute-def:memberOf (Group List)
urn:oid:memberOf (Group List)
```

Once the request is fulfilled, your server is registered. No additional files need to be installed.

## 9. Protecting Directories

Basic access control can be done in the .htaccess file. To restrict a directory to logged in users, simply add a line to .htaccess:

```
MellonEnable "auth"
```

If you wish to limit access to specific users or groups, use MellonCond.

```
# Limit to several users and a group
MellonCond "SSO_USERID" "adurance" [MAP,OR]
MellonCond "SSO_USERID" "ptr" [MAP,OR]
MellonCond "SSO_FNAL_GROUPS" "CN=WebTeam,OU=Distribution
Groups,OU=Exchange,DC=services,DC=fnal,DC=gov" [MAP,SUB]
```

More information on this topic is available in the mod\_auth\_mellon README.

## 10. Testing

Create a file in a SSO protected web directory called sstest.php containing the following code:

```
<table>
<?php
foreach ($_SERVER as $key => $value) {
    if (stripos($key, 'SSO') !== FALSE || stripos($key, 'SHIB') !== FALSE || stripos($key,
'MELLON') !== FALSE) {
        echo "<tr><td><b>$key</b></td><td>$value</td></tr>";
    }
}
?>
</table>
```

This page will display any available SSO attributes.

## Appendix A: mellon\_create\_metadata.sh

This script generates certificates and metadata for use by the SP. See [3. Create Metadata and Certificates] for usage.

```
#!/bin/sh

FQDN="$1"
SECONDARY="$2"
ENTITYID="https://$FQDN"
BASEURL="$ENTITYID/mellon"
ALTNAME="DNS:$FQDN,URI:$ENTITYID"
DAYS=3650
OUT="/etc/mod_auth_mellon"
OUTSECONDARY="/web/etc/mod_auth_mellon"

if [ "$SECONDARY" == "-secondary" ]; then
    OUT="$OUTSECONDARY"
    KEY="$OUT/sp-key_$FQDN.pem"
    CERT="$OUT/sp-cert_$FQDN.pem"
    METADATA="$OUT/sp-metadata_$FQDN.xml"
else
    KEY="$OUT/sp-key.pem"
    CERT="$OUT/sp-cert.pem"
    METADATA="$OUT/sp-metadata.xml"
fi

if [ -s $KEY -o -s $CERT ] ; then
    echo "The files $KEY and/or $CERT already exist!"
    exit 2
fi

echo "Creating directory if it does not already exist"
mkdir -p $OUT

echo "Generating certificate and key"

SSLCNF="$OUT/sp-cert.cnf"
cat >$SSLCNF <<EOF
# OpenSSL configuration file for creating sp-cert.pem
[req]
prompt=no
default_bits=2048
encrypt_key=no
default_md=sha1
distinguished_name=dn
# PrintableStrings only
string_mask=MASK:0002
x509_extensions=ext
[dn]
CN=$FQDN
[ext]
subjectAltName=$ALTNAME
subjectKeyIdentifier=hash
EOF

touch $KEY
chmod 600 $KEY
openssl req -config $SSLCNF -new -x509 -days $DAYS -keyout $KEY -out $CERT
rm $SSLCNF
```

```

CERTDATA="$(grep -v '^-----' "$CERT")"

echo "Generating metadata"

cat >"$METADATA" <<EOF
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<EntityDescriptor
  entityID="$ENTITYID"
  xmlns="urn:oasis:names:tc:SAML:2.0:metadata">
  <SPSSODescriptor
    AuthnRequestsSigned="true"
    WantAssertionsSigned="true"
    protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <KeyDescriptor use="signing">
      <ds:KeyInfo
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>$CERTDATA</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <KeyDescriptor use="encryption">
      <ds:KeyInfo
        xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>$CERTDATA</ds:X509Certificate>
        </ds:X509Data>
      </ds:KeyInfo>
    </KeyDescriptor>
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
      Location="$BASEURL/logout" />
    <SingleLogoutService
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
      Location="$BASEURL/logout" />
    <NameIDFormat>urn:oasis:names:tc:SAML:2.0:nameid-format:transient</NameIDFormat>
    <AssertionConsumerService
      index="0"
      isDefault="true"
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
      Location="$BASEURL/postResponse" />
    <AssertionConsumerService
      index="1"
      Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Artifact"
      Location="$BASEURL/artifactResponse" />
  </SPSSODescriptor>
</EntityDescriptor>
EOF

echo Done
exit 1

```

## Appendix B: Example Apache configuration

This configuration should work as-is on Fermilab systems.

```

# MellonCacheSize sets the maximum number of sessions which can be active
# at once. When mod_auth_mellon reaches this limit, it will begin removing
# the least recently used sessions. The server must be restarted before any
# changes to this option takes effect.

```

```
# Default: MellonCacheSize 100
MellonCacheSize 100

# MellonLockFile is the full path to a file used for synchronizing access
# to the session data. The path should only be used by one instance of
# apache at a time. The server must be restarted before any changes to this
# option takes effect.
# Default: MellonLockFile "/var/run/mod_auth_mellon.lock"
MellonLockFile "/var/run/mod_auth_mellon/lock"

# MellonPostDirectory is the full path of a directory where POST requests
# are saved during authentication. This directory must be writeable by the
# Apache user. It should not be writeable (or readable) by other users.
# Default: None
# Example: MellonPostDirectory "/var/cache/mod_auth_mellon_postdata"
MellonPostDirectory /var/cache/mellon_post_data

# MellonPostTTL is the delay in seconds before a saved POST request can
# be flushed.
# Default: MellonPostTTL 900 (15 mn)
MellonPostTTL 900

# MellonPostSize is the maximum size for saved POST requests
# Default: MellonPostSize 1073741824 (1 MB)
MellonPostSize 1073741824

# MellonPostCount is the maximum amount of saved POST requests
# Default: MellonPostCount 100
MellonPostCount 100

#####
## We apply everything to <directory> AND <location>
## To ensure these settings are valid for both real files and proxied/rewritten requests
#####

<Directory />
    # Enable mellon for this server, but do not force authentication by default
    MellonEnable "info"

    # Certificates and metadata should be generated once PER CLUSTER
    # and replicated across all servers in that cluster
    # The entity and endpoint URLs should be the cluster VIP address
    # such as https://vip-webint.fnal.gov
    MellonSPCertFile /etc/mod_auth_mellon/sp-cert.pem
    MellonSPPrivateKeyFile /etc/mod_auth_mellon/sp-key.pem
    MellonSPMetadataFile /etc/mod_auth_mellon/sp-metadata.xml

    # IdP Metadata provided by Authentication and Directory Services
    # Reference: https://pingprod.fnal.gov:9031/files/pingprod-metadata.xml
    MellonIdPMetadataFile /etc/mod_auth_mellon/pingprod-metadata.xml

    # Wildcard is required for FNAL VirtualHosts to work without individual
    # registrations with the IdP
    # Can be overridden by non-FNAL VirtualHosts, but they must be registered
    MellonCookieDomain ".fnal.gov"

    # Where mod_auth_mellon's internal requests are sent.
    # This is not immune to RewriteRules! Be sure to exclude it!
    MellonEndpointPath /mellon

    # POST requests sent without session are authenticated and then replayed
```

```

# hopefully transparent to the user
MellonPostReplay On

# This option prevents failure due to IPv6 and IPv4 address mismatch
MellonSubjectConfirmationDataAddressCheck Off

# Invalid pre-0.10
# Enabling may break applications designed to use the old method
MellonMergeEnvVars On

MellonSetEnvNoPrefix "SSO_Session_ID"          "NAME_ID"
MellonSetEnvNoPrefix "SSO_USERID"             "urn:oid:0.9.2342.19200300.100.1.1"
MellonSetEnvNoPrefix "SSO_FNAL_GROUPS"        "urn:oid:memberOf"
MellonSetEnvNoPrefix "SSO_NAME_FIRST"         "urn:oid:2.5.4.42"
MellonSetEnvNoPrefix "SSO_NAME_LAST"          "urn:oid:2.5.4.4"
MellonSetEnvNoPrefix "SSO_EMAIL"              "urn:oid:0.9.2342.19200300.100.1.3"
MellonSetEnvNoPrefix "SSO_EPPN"               "urn:oid:1.3.6.1.4.1.5923.1.1.1.6"

# REMOTE_USER is set to SSO_EMAIL
MellonUser "urn:oid:0.9.2342.19200300.100.1.3"

# The cookie name must be unique per cluster to avoid
# overlapping with other sessions.
MellonVariable "sso_%HOSTNAME%"
</Directory>

<Location />
# Enable mellon for this server, but do not force authentication by default
# MellonEnable "info"
# This part should not be enabled at the <Location> level because it overrides
.htaccess in the merge order

# Certificates and metadata should be generated once PER CLUSTER
# and replicated across all servers in that cluster
# The entity and endpoint URLs should be the cluster VIP address
# such as https://vip-webint.fnal.gov
MellonSPCertFile /etc/mod_auth_mellon/sp-cert.pem
MellonSPPrivateKeyFile /etc/mod_auth_mellon/sp-key.pem
MellonSPMetadataFile /etc/mod_auth_mellon/sp-metadata.xml

# IdP Metadata provided by Authentication and Directory Services
# Reference: https://pingprod.fnal.gov:9031/files/pingprod-metadata.xml
MellonIdPMetadataFile /etc/mod_auth_mellon/pingprod-metadata.xml

# Wildcard is required for FNAL VirtualHosts to work without individual
# registrations with the IdP
# Can be overridden by non-FNAL VirtualHosts, but they must be registered
MellonCookieDomain ".fnal.gov"

# Where mod_auth_mellon's internal requests are sent.
# This is not immune to RewriteRules! Be sure to exclude it!
MellonEndpointPath /mellon

# POST requests sent without session are authenticated and then replayed
# hopefully transparent to the user
MellonPostReplay On

# This option prevents failure due to IPv6 and IPv4 address mismatch
MellonSubjectConfirmationDataAddressCheck Off

```

```
# Invalid pre-0.10
# Enabling may break applications designed to use the old method
MellonMergeEnvVars On

MellonSetEnvNoPrefix "SSO_Session_ID"      "NAME_ID"
MellonSetEnvNoPrefix "SSO_USERID"         "urn:oid:0.9.2342.19200300.100.1.1"
MellonSetEnvNoPrefix "SSO_FINAL_GROUPS"   "urn:oid:memberOf"
MellonSetEnvNoPrefix "SSO_NAME_FIRST"     "urn:oid:2.5.4.42"
MellonSetEnvNoPrefix "SSO_NAME_LAST"     "urn:oid:2.5.4.4"
MellonSetEnvNoPrefix "SSO_EMAIL"         "urn:oid:0.9.2342.19200300.100.1.3"
MellonSetEnvNoPrefix "SSO_EPPN"          "urn:oid:1.3.6.1.4.1.5923.1.1.1.6"

# REMOTE_USER is set to SSO_EMAIL
MellonUser "urn:oid:0.9.2342.19200300.100.1.3"

# The cookie name must be unique per cluster to avoid
# overlapping with other sessions.
MellonVariable "sso_%HOSTNAME%"
</Location>
```