MDQ via Apache

This page documents an exploration of using Apache to serve MDQ queries. The challenge in this is with the forward slash character in entityIDs. By default Apache will not server URLs whose path includes the forward slash character encoded as "%2F". Instead, it returns a 404 to these URLs. The Apache directive AllowEncodedSlashes allows these URLs to pass through.

Started with a basic CentOS 7 host. First update the packages and install `httpd`:

Update and install commands

```
sudo yum update -y
sudo yum install -y httpd
reboot
```

Next add `mdq.conf`, an apache configuration to serve MDQ files from a configured directory. This file must be placed in the directory `/etc/httpd/conf.d`. The contents of that file are:

/etc/httpd/conf.d/mdq.conf

```
<VirtualHost *:80>
  DocumentRoot /var/www/html/mdq

  # Allow %2F in URLs, and pass them through as '%2F'
  AllowEncodedSlashes NoDecode

  # Use the correct content-type for the metadata
  ForceType application/samlmetadata+xml
</VirtualHost>
```

Create the entities directory and restart httpd:

```
sudo mkdir -p /var/www/html/mdq/entities

sudo systemctl restart httpd
```

Grab a few select files from the MDQ beta server:

```
cd /var/www/html/mdq/entities

# Fetch the InCommon federation manager
sudo wget -O https:%2F%2Ffm.incommon.org%2Fsp  http://mdq-beta.incommon.org/global/entities/https%3A%2F%2Ffm.incommon.org%2Fsp
# Fetch a mace entity ID
sudo wget -O urn:mace:incommon:osu.edu  http://mdq-beta.incommon.org/global/entities/urn%3Amace%3Aincommon%3Aosu.edu

# Create a symbolic link for the SHA1 alias, this one for osu.edu
sudo ln -s urn:mace:incommon:osu.edu {sha1}d4d1ca999a5a9352fa618075eac456eb67956c74
```

Now you should be able to access the metadata using the MDQ protocol:

Sample queries to localhost

```
curl http://localhost/entities/https%3A%2F%2Ffm.incommon.org%2Fsp
curl http://localhost/entities/urn%3Amace%3Aincommon%3Aosu.edu
curl http://localhost/entities/%7Bsha1%7Dd4d1ca999a5a9352fa618075eac456eb67956c74
```

Notes

- My host did not have a firewall running. If yours does you may have to open up port 80
- I'm not sure we've got all the query types. Are there more besides http(s), urn:mace: and {sha1}?
- ~~It would be better to make the sha1-entries symbolic links to their named entry, which may require an Apache directive to follow links~~
  - This works

Attachments

File	Modified
File mdq-mirror.py	Jul 29, 2016 by tmitchell@unitedid.org