Fermilab Central Web Hosting

Using Single Sign On (SSO) Authentication with .htaccess

Contents

Information and Security Contacts: 3
Using .htaccess with SSO 4
Allow all authenticated users 4
Allow all Fermilab users 4
Allow users in a SERVICES group 4
Allow specific users 5
Other MellonCond usage 5
Using scripts with SSO 6
PHP – Require Authenticated User 6
PHP – Require Group 7
Perl – Require Authenticated User 7
Perl – Require Group 7
Python – Require User 8
Python – Require Group 8

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	

Using .htaccess with SSO

Access can be controlled based on SSO attributes using a .htaccess file placed in the directory you wish to protect.

If you have an existing .htaccess file, be sure to comment out or remove any “Auth” or “Require” directives as they may interfere with SSO.

The two directives you will use are MellonEnable and MellonCond.

The values for MellonEnable are:

off – SSO is disabled completely
info – SSO is enabled but authentication is not required (mostly useful for scripts)
auth – SSO is enabled and authentication is required

MellonCond is optionally used to set conditional statements for allowing access. It can match values against the following attributes:

SSO_USERID	Username without domain
SSO_FNAL_GROUPS	SERVICES groups that the user belongs to
SSO_NAME_FIRST	First name
SSO_NAME_LAST	Last name
SSO_EMAIL	Email address

The following examples are provided for common scenarios.

Note the use of the flags in brackets at the end of each MellonCond line.

The MAP flag is always required.
The REG flag specifies the use of regular expressions instead of basic string matching and is optional.
The OR flag allows the user through if ANY of the lines match, with the default behavior requiring ALL lines to match.

Allow all authenticated users

Live example: http://tools.fnal.gov/examples/sso/require_authenticated/

```
MellonEnable "auth"
```

Allow all Fermilab users

Live example: http://tools.fnal.gov/examples/sso/require_fermi/

```
MellonEnable "auth"
MellonCond SSO_EMAIL "@fnal.gov$" [MAP,REG]
```

Allow users in a SERVICES group

Live example: http://tools.fnal.gov/examples/sso/require_group/

```
MellonEnable "auth"
MellonCond SSO_FNAL_GROUPS "CN=web-wp-
administrator,OU=Central_Web,OU=FermiGroups,DC=services,DC=fnal,DC=gov" [MAP]
```

Allow specific users

Live example: http://tools.fnal.gov/examples/sso/require_user/

```
MellonEnable "auth"
MellonCond SSO_EMAIL "adurance@fnal.gov" [MAP,OR]
MellonCond SSO_EMAIL "ptr@fnal.gov" [MAP]
```

Other MellonCond usage

The full documentation for mod_auth_mellon is located at https://github.com/UNINETT/mod_auth_mellon/blob/master/README. The section involving MellonCond is included here for convenience.

```
        # MellonCond provides the same function as MellonRequire, with
        # extra functionality (MellonRequire is retained for backward
        # compatibility). The syntax is
        # 'MellonCond <attribute name> <value> [<options>]'
        #
        # <value> is an attribute value to match. Unlike with MellonRequire,
        # multiples values are not allowed.
        #
        # If the [REG] flag is specified (see below), <value> is a regular
        # expression. The syntax for backslash escape is the same as in
        # Apache's <LocationMatch>'s directives.
        #
        # Format strings are substituted into <value> prior evaluation.
        # Here are the supported syntaxes:
        #    %n       With n being a digit between 0 and 9. If [REG,REF]
        #             flags (see below) were used in an earlier matching
        #             MellonCond, then regular expression back references
        #             are substituted.
        #    %{num}   Same as %n, with num being a number that may be
        #             greater than 9.
        #    %{ENV:x} Substitute Apache environment variable x.
        #    %%       Escape substitution to get a literal %.
        #
        # <options> is an optional, comma-separated list of option
        # enclosed with brackets. Here is an example: [NOT,NC]
        # The valid options are:
        #    OR  If this MellonCond evaluated to false, then the
        #        next one will be checked. If it evaluates to true,
        #        then the overall check succeeds.
        #    NOT This MellonCond evaluates to true if the attribute
        #        does not match the value.
        #    SUB Substring match, evaluates to true if value is
        #        included in attribute.
        #    REG Value to check is a regular expression.
        #    NC  Perform case insensitive match.
        #    MAP Attempt to search an attribute with name remapped by
        #        MellonSetEnv. Fallback to non remapped name if not
        #        found.
        #    REF Used with REG, track regular expression back references,
        #        So that they can be substituted in an upcoming
        #        MellonCond directive.
        #
        # It is allowed to have multiple MellonCond on the same
        # attribute, and to mix MellonCond and MellonRequire.
        # Note that this can create tricky situations, since the OR
        # option has effect on a following MellonRequire directive.
        #
        # Default: none set
        # MellonCond "mail" "@example\.net$" [OR,REG]
```

```
        # MellonCond "mail" "@example\.com$" [OR,REG]
        # MellonCond "uid" "superuser"
```

Using scripts with SSO

The SSO attributes are made available to scripts through environment variables. The available variables are:

SSO_Session_ID	A unique identifier for each SSO session
SSO_USERID	Username without domain
SSO_FNAL_GROUPS	SERVICES groups that the user belongs to
SSO_NAME_FIRST	First name
SSO_NAME_LAST	Last name
SSO_EMAIL	Email address
REMOTE_USER	Same as SSO_EMAIL **(recommended for identifying users)**

Some attributes may contain more than one value, such as SSO_FNAL_GROUPS. When this is the case, they will be split into separate numbered variables. For example:

SSO_FNAL_GROUPS_0: CN=Services-Orion-CCD,OU=FermiGroups,DC=services,DC=fnal,DC=gov
SSO_FNAL_GROUPS_1: CN=All_Active_Users,OU=FermiGroups,DC=services,DC=fnal,DC=gov

There are two ways to integrate the sign on process with your scripts. If your entire application should be protected, you can simply add a .htaccess file containing:

```
MellonEnable “auth”
```

Otherwise, you can initiate the sign on through your application logic by redirecting to a URL:

```
Location: /mellon/login?ReturnTo=RETURN_URL_HERE
```

The user will be redirected back to the return URL once the sign on is complete. At this time, the environment variables will be available to your script.

PHP, Perl, and Python example scripts are provided below.

PHP – Require Authenticated User

Live example: http://tools.fnal.gov/examples/sso/require_user.php

```
<?php

if (isset($_SERVER['SSO_Session_ID'])) {
        foreach ($_SERVER as $header => $value) {
                if (stripos($header, 'SSO_') === 0) {
                        echo "$header: $value <br/>";
                }
        }
} else {
        /* No SSO session found, send them to the login page! */
        $current_url = (isset($_SERVER['HTTPS']) ? 'https' : 'http') . '://' . $_SERVER['HTTP_HOST'] .
$_SERVER['REQUEST_URI'];
        header('Location: /mellon/login?ReturnTo='.urlencode($current_url));
```

```
}
?>
```

PHP – Require Group

Live example: http://tools.fnal.gov/examples/sso/require_group.php

```
<?php

$required_group = 'CN=web-wp-administrator,OU=Central_Web,OU=FermiGroups,DC=services,DC=fnal,DC=gov';

$authenticated = FALSE;
foreach ($_SERVER as $header => $value) {
        if (stripos($header, 'SSO_FNAL_GROUPS') === 0 && $value == $required_group) {
                $authenticated = TRUE;
                break;
        }
}

if ($authenticated) {
        echo "This script succeeds because you are a member of $required_group! <br/><br />";

        foreach ($_SERVER as $header => $value) {
                if (stripos($header, 'SSO_') === 0) {
                        echo "$header: $value <br/>";
                }
        }
} else {
        echo "Not in required group or not authenticated";
}

?>
```

Perl – Require Authenticated User

Live example: http://tools.fnal.gov/cgi-bin/examples/sso/require_user.pl

```
#!/usr/bin/perl

use strict;
use warnings;

if (exists $ENV{'SSO_Session_ID'}) {
        print "Content-type: text/html\n\n";
        foreach my $header (keys(%ENV)) {
                if ($header =~ /^SSO_/) {
                        print "$header = $ENV{$header}<br/>";
                }
        }
} else {
        my $current_url = (exists $ENV{'HTTPS'} ? 'https' : 'http') . '://' . $ENV{'HTTP_HOST'} .
$ENV{'REQUEST_URI'};
        print "Location: /mellon/login?ReturnTo=$current_url\n\n";
}
```

Perl – Require Group

Live example: http://tools.fnal.gov/cgi-bin/examples/sso/require_group.pl

```
#!/usr/bin/perl

use strict;
use warnings;
```

```
my $required_group = 'CN=web-wp-administrator,OU=Central_Web,OU=FermiGroups,DC=services,DC=fnal,DC=gov';


print "Content-type: text/html\n\n";

foreach my $header (keys(%ENV)) {
        if ($header =~ /^SSO_FNAL_GROUPS/ && $ENV{$header} eq $required_group) {
                print "This script succeeds because you are a member of $required_group! <br/><br />";

                foreach my $header (keys(%ENV)) {
                        if ($header =~ /^SSO_/) {
                                print "$header = $ENV{$header}<br/>";
                        }
                }

                exit 0;
        }
}

print "Not authenticated or not in group";
```

Python – Require User

Live example: http://tools.fnal.gov/cgi-bin/examples/sso/require_user.py

```
#!/usr/bin/env python

import os
import cgi
import urllib

if 'SSO_Session_ID' in os.environ:
        print "Content-type: text/html"
        print
        for header in os.environ:
                if header.startswith('SSO_'):
                        print header + ": " + os.environ[header] + "<br/>"
else:
        current_url = ('https' if 'HTTPS' in os.environ else 'http') + '://' + os.environ['HTTP_HOST'] +
os.environ['REQUEST_URI']
        print "Location: /mellon/login?ReturnTo="+urllib.quote(current_url)
        print
```

Python – Require Group

Live example: http://tools.fnal.gov/cgi-bin/examples/sso/require_group.py

```
#!/usr/bin/env python

required_group = 'CN=web-wp-administrator,OU=Central_Web,OU=FermiGroups,DC=services,DC=fnal,DC=gov'

import os
import cgi
import urllib

print "Content-type: text/html"
print

for header in os.environ:
        if header.startswith('SSO_FNAL_GROUPS') and os.environ[header] == required_group:
                print "This script succeeds because you are a member of "+required_group+"! <br/><br />"
                for header in os.environ:
                        if header.startswith('SSO_'):
                                print header + ": " + os.environ[header] + "<br/>"
```

```
                  break
else:
        print "Not in required group or not authenticated"
```