HOWTO Configuration guide to enable Shibboleth on EZproxy

Author: Barbara Monticini, GARR

Table of contents

Configuration guide to enable Shibboleth on EZproxy

The main goal of the guide is to enlight most of the directive provided by EZproxy to make it act as Shibboleth Service Provider and accept SSO login from authenticated users.

The first assumption is to have already deployed EZproxy in the standard way (local login, Database stanzas to resources already set up)

Pre-requisites

1. An identity management system already in place;
2. A Shibboleth identity Provider (IdP) tied to the environment in 1. from wich Services Providers (SP) can obtain identity information on users that request access to resources;
3. An EZproxy installation that provides authenticated remote access to library resources;
4. Being part of a national Federation and have joined eduGAIN interfederation to wider the potential portfolio of SAML-enabled resourses available.

For the successfull deployement of the guide it is recommended to have Shibboleth Identity Provider version 2.x or version 3.x and to be ready to share information with the Institution IdP manager/s.

The recommended method for proxying access to remote resources in EZproxy and reduce firewall issues is called “Proxy by Hostname”. This method implies the use of wildcard certificates and instead of using different ports it is based on the conversion of URL from the requested www.reource.com to www.reource.com.your-ezproxy.your.org in the URL bar of the browser. This has been written testing on a “Proxy by Hostname” configuration.

Configuration of user authentication method based on Shibboleth in EZproxy

The next step in integrating Shibboleth and EZproxy is to configure EZproxy to be a Shibboleth Service Provider. This is also referred to as Shibboleth-enabling EZproxy.

The official documentation from OCLC provides all the instructions needed to perfom the job:

http://www.oclc.org/support/services/ezproxy/documentation/usr/shibboleth.en.html

and everything suggested in the present guide is taken mosty from there.

Self-signed certificate creation

Open the EZproxy Administration page (https://login.ezproxy.your.org/admin) and select “Manage

SSL (https) certificates", then select the "Create new SSL certificate" link.

Complete the form with values relevant to your Institutional library and with the following suggested values:

```
Key size: 2048

Certificate name: [no wildcard]

Expiration (for self-signed only): 10 years
```

For finalizing the creation of the certificate click on "Self-Signed Certificate".

Next screen will show the details of the newly created certificate.

Read carefully: do not type ACTIVE in the dialogue box when prompted

Return to the "Manage SSL (https) certificates" page and take note of the ID number assigned to the new certificate.

Set-up Shibboleth built-in support of EZproxy

EZproxy contains built-in support that allows EZproxy to act as a Shibboleth 1.3/2.x Service Provider (SP), allowing EZproxy to accept user authentication and authorization information from your institution's Identity Provider (IdP) and to map that information into corresponding EZproxy authorizations.

Open the EZproxy configuration file config.txt Add the following directives below which are explained hereafter:

```
## Shibboleth settings

ShibbolethDisable 1.3

ShibbolethMetadata \
   -EntityID=EZproxyEntityID \
   -File=MetadataFile \
   [-Cert=EZproxyCertNumber \]
   [-URL=MetadataURL \]
   [-URLValidate=URLValidateFile \]
   [-AuthnContextClassRef=AuthnContext]
```

Import: The entire ShibbolethMetadata directive with all options can appear on a single line or it can broken across multiple lines as shown. When breaking options across multiple lines, all but the last such line must end with a space followed by a backslash (\).

This directive may appear multiple times to link EZproxy to multiple federations.

Explanaition of single options:

ShibbolethDisable 1.3 Disables the older 1.3 version of Shibboleth and forces EZProxy to use the newer version of SAML protocol (SAML2)

EntityID is a URL that uniquely identifies your EZproxy SAML service, the URL does not have to resolve to any particular content. It is something that as to be created here for the first time and will be the EntityID value in the SP metadata to be used by the Idp to communicate with EZproxy.

File This file contains the locally cached copy of a wider metadata containing both idp and sp metadata, for example a Federation metadata . The file must be manually downloaded once and placed in the main EZproxy directory (e.g. /usr/local/ezproxy).

Cert (optional) Is the ID number of the self signed certificate created in the previous steps. It can be omitted if the SP is configured to use in SAML the same certificate used for https. Usually the certificate for https is suggested to be wildcard (*.) if EZproxy is in Proxy by Hostname mode and wildcards can be refused from certains idps.

URL (optional) This URL is the right place to point to the metadata of all participating services and identity providers in a given federation, it must contain the metadata of idps the sp wants to talk to.

URLValidateFile (Optional) This optional parameter specifies the file containing an X.509 (SSL) certificate that can be used to verify that a MetadataFile retrieved from a MetadataURL is authenticated. The default directory for URLValidateFile is the directory where EZproxy is installed.

AuthnContext (Optional) This optional parameter only applies to Shibboleth 2.1 and specifies an authentication context class reference to include in the authentication request to the Identity Provider. Most institutions will not need to include this value. One possible value for this parameter is urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport.

Example used in the pilot installation:

```
## Shibboleth settings

ShibbolethDisable 1.3

ShibbolethMetadata \
   -EntityID=https://ezproxy.fi.infn.it/sp \
   -File=metadata.xml  \
   -Cert=3
```

Once you add this directive to config.txt and restart EZproxy, you should access the EZproxy administration page where you will find a new link titled “**Manage Shibboleth**”.

Open the Shibboleth management page and click on **Certificate Metadata**.

The content that appears can be used to register your EZProxy SAML Service Provider within a Federation or can be sent directly to the admin of the idp to be connected to (it includes the entityID

you chose in the earlier steps).

In the "**Manage Shibboleth**" page it is also possible to select from a list of idps (taken from the metadata file configured earlier with *ShibboletMetadata* directive in config.txt) in order to perform a test on attributes released from any single idp to your EZproxy sp. The test rely on the configuration of the SSO authentication avilable in the next section.

Set-up user authentication from Identity Provider or from a Federation

There are a number of different options for routing users to your Identity Provider for authentication. The examples of these options employ the following abbreviations:

Abbreviation	Description
DSURL	A URL for a Shibboleth 2.x Discovery Service.
EZproxyEntityID	An Entity ID used to identify your EZproxy server.
IDPEntityID	An Entity ID used to identify an Identity Provider.
WAYFURL	A URL to a Shibboleth 1.x Where Are You From (WAYF) service.

The configuration of user authentication is usually placed in the main installation directory (eg. /usr/local/ezproxy/user.txt)

Shibboleth 2.x Configurations

To route users to a specific Shibboleth 2.x Identity Provider, use a configuration similar to this in user.txt:

```
::Shibboleth
IDP20 IDPEntityID
/Shibboleth
```

To route users to a Shibboleth 2.x Discovery Service, use a configuration similar to this in user.txt:

```
::Shibboleth
DS20 DSURL # first form
DS20 -EntityID=EZproxyEntityID DSURL # second form
/Shibboleth
```

If all of your ShibbolethMetadata directives use the same -EntityID, then you can use the first form of DS20. If you have multiple ShibbolethMetadata directives with varying -EntityID values, you must use the second form and explicitly specify the EntityID of your EZproxy server for the specific Discovery Service.

At the end of the configuration of user.txt file it is possibile to perform the test on the "**Manage Shibboleth**" section "**Shibboleth 2.0 Attributes**"to Show Shibboleth 2.0 Attributes released from the selected IdP.

It is possible to mix traditional login to SSO login. In order to do that edit the login.htm file and add a link similar to this:

```
<a href="/login?auth=shibboleth&url=^U">Login with Shibboleth</a>
```

and then modify user.txt to be similar to this:

```
::Shibboleth
If login:auth eq "shibboleth"; IDP20 IDPEntityID
/Shibboleth
```

adding

```
If login:auth eq "shibboleth";
```

in front of whichever Shibboleth routing method you use.

Users who want to login with Shibboleth will click the Institutional special link, whereas other users will use the login form for local access.

EZProxy test installation for AARC project SA1 Libraries Pilots

Local access to Library services:	Institutional access to Library services:
Please enter your username: Please enter your password: Login	Login with Institution credentials

Configure Authorization based on user attributes

shibuser.txt is the file to allows the use of Shibboleth attributes to make EZproxy authorization decisions. The samples available on the official EZproxy guide demonstrates how to: restrict access to a specific Identity Provider, block alumni, place employees into an additional EZproxy group, place people affiliated with the law.example.edu scope into an additional EZproxy group, designate a specific user as an EZproxy administrator, and map an attribute for EZproxy to use as the username for logging, enforcing transfer limits, etc.

```
If !(auth:issuer eq "https://idp.yourlib.org/shibboleth/idp");
   Deny unaffiliated.html
If auth:urn:mace:dir:attribute-def:eduPersonPrimaryAffiliation eq
"alum";
   Deny alum.html
Group Default
If Any(auth:urn:mace:dir:attribute-def:eduPersonAffiliation,
"employee");
   Group +Employee
If Count(auth:urn:mace:dir:attribute-
def:eduPersonScopedAffiliation@law.example.edu) > 0;
   Group +Law
If auth:urn:mace:dir:attribute-
def:eduPersonPrincipalName@example.edu eq "ezmgr";
   Admin
Set login:loguser = auth:urn:mace:dir:attribute-
def:eduPersonTargetedID
```

This means that, once configured, EZproxy can make decisions as to which databases a user can gain access, based on user attributes made available to EZproxy by the authenticating IdP.

There are two types of configurations that are useful to understand. The first is the ability to deny access to all databases based on a user's attributes. The second is to selectively allow users to access certain databases based on specific user attributes (e.g., nursing student's access to journal x).

http://www.oclc.org/support/services/ezproxy/documentation/usr/common.en.html

http://www.oclc.org/support/services/ezproxy/documentation/expressions.en.html

Configure EZproxy to seamlessly enable Shibboleth access to Service Providers

The very interesting and innovating part of configuring SSO in EZproxy is the chance to take advantage of SSO-enabled login already in place in most of the on-line librarian resourses. Institutional logins are increasing in number and is available in almost the majority of principal on-line publishers and librarian resourses. Some example are: EBSCO, Elsevier, Thomson Reuters. The idea behind this section is to use the already established Shibboleth session of the user to redirect the request to the SSO SessionInitiators provided by Resource Provider instead of just proxing the content requested by the user. The advantages of this mechanism are: reduction of the amount of traffic generated by the proxy and the possibility, through Shibboleth, for resource providers to create personalized services for users in their interfaces (if available and based on personal attribute release by the Idp)

The directive provided by EZproxy to perform such mechanism is named **SPUEdit**. On the web guide located in:
http://www.oclc.org/support/services/ezproxy/documentation/cfg/spuedit.en.html

To understand the directive the suggestion is to review the concepts of starting point URLs:
http://www.oclc.org/support/services/ezproxy/documentation/cfg/understanding-urls.en.html

SPUEdit directive instructs EZproxy to edit starting point URLs using regular expressions to allow URLs to be rearranged and then users to be rerouted to the rearranged URLs.

Syntax

The **SPUEdit** directive takes the following form:

```
SPUEdit / find/ replace/ girs
```

Qualifier	Description
/	Any non-alphabetic, non-digit character.
find	A Perl-style regular expression to match.
replace	The string to use to replace the expression in find. This string may contain variables including:

	• $0: matches the entire string that matched find • $1 through $9: based on parenthesis matching • ${spueditvar}: a value defined by SPUEditVar • e: may appear after $ to indicate that the replacement text needs to be percent-escaped as appropriate for use in URLs, such as $e0 to substitute the percent-escaped version of the find string
girs	An optional combination of letters used to indicate that: • g: the changes should be applied globally through the URL • i: the find string match should be case-insensitive • r: the users should be redirected to the rewritten URL • s: SPUEdit processing should stop at this point

It could be very useful to use another directive named **SPUEditVar** to easily and clearly write a SPUEdit directive.

Examples from our test instance with Science Direct (Elsevier) Session Initiator:

```
SPUEditVar IdPProviderID=https://idp3.idem.garr.it/idp/shibboleth
```

```
SPUEdit
@^http://www\.sciencedirect\.com.*$@https://auth.elsevier.com/ShibAuth/institutionLogin?entityID=${IdPProviderID}&appReturnURL=http://www.sciencedirect.com@ir
```

Note the use of @ instead of / between the find, replace, and options. If / had been used, each of the slashes that appear in the find and replace strings would need to be specified using \/ instead of just /.

The directive **SPUEdit** can be combined with other EZproxy directives (AutoLoginIP, ExcludeIP, IncludeIP) usually used to identify computers that should be automatically logged into EZproxy (for example on campus users could potentially be automatically login to EZproxy and when requesting SSO-enabled resourses redirected to the SP for federated login without forcing them to authenticate to EZproxy too).

References

EZproxy documentation

http://www.oclc.org/support/services/ezproxy/documentation.en.html

In Common Federation - EZproxy HOW-TO

https://spaces.internet2.edu/display/inclibrary/Shibboleth+-+EZproxy+HOW-TO

Edugate Federation (HEAnet) - Using Shibboleth instead of LDAP to authenticate EZproxy users

https://edugate.heanet.ie/rr3/p/page/ezproxy