EZproxy v6.2.2 Release Notes

Release Date: November 2016

Last updated: December 15, 2016

Table of Contents

Operating System Requirements

EZproxy is supported under three different operating systems:

- Linux
- Solaris (x86)
- Windows

The supported versions of these operating systems along with their minimum hardware requirements can be found at EZproxy: Hardware and Operating System Requirements.

Recommended Actions

For this release, we recommend that you review the following checklists and complete the relevant tasks. These checklists identify updates that we have determined as significant for most institutions. We encourage you to review all of the items in the release notes to determine whether there are other items that might require additional action or follow up by your institution.

Action
❑ If you are upgrading from an EZproxy version earlier than V6.0, you will need to request an EZproxy Web Services Key (WSKey). To request a WSKey, you will need to have a current, annual subscription. EZproxy moved to the annual subscription model in July 2013, so if you purchased your EZproxy subscription prior to that time, you will need to update. To purchase an annual subscription, you can request a quote, and you will be provided with a quote and information about how to subscribe. If you are uncertain if your subscription is current, please email orders@oclc.org. If you have already upgraded to V6.x, your existing WSKey will work with this upgrade.
❑ Review EZproxy and OpenSSL, especially if you are upgrading from a version older than V5.7.44. EZproxy V6.2.x has many security updates that may make previous configurations in your config.txt file unnecessary, and you can remove certain directives after installing V6.2.x.

Release Notes

Administrative Updates

More Granular Permissions for EZproxy Administration (JIRA 1605)

Previously, the EZproxy Administration features were an (almost) all or nothing proposition in which users either had total administrative privilege or none. The only exception was the ability to give users access to the Token cross-reference feature.

The majority of options on the Administration page can now be granted to users individually by assigning them to special groups. When setting up this type of access, the historical Admin command is no longer used, but instead users are placed into special groups that correspond to the URL over the Admin feature. For example, the Audit page is available from /audit, so the group that grants access to this is Admin.Audit.

The groups available are:

- Admin.Audit
- Admin.DecryptVar
- Admin.Groups
- Admin.Intrusion
- Admin.LDAP
- Admin.Messages
- Admin.Restart
- Admin.Shibboleth
- Admin.SSLUpdate
- Admin.SSLView
- Admin.StatusUpdate
- Admin.StatusView
- Admin.Token
- Admin.Usage
- Admin.UsageLimits
- Admin.User
- Admin.Variables

The /admin page automatically adjusts based on group membership to display the options that correspond to these group memberships.

The SSL and Status pages have the ability to change key aspects of EZproxy's behavior, so these features have been divided into Update and View groups. Users in the Update group have the full functionality available in previous versions, whereas users in the View group are only able to view information on these pages.

Users who are full administrators through the classic Admin command or who have the Admin.Groups privilege can see a list of all of these groups at the /groups URL.

Admin users are assigned to these groups via user.txt. They cannot be used within config.txt.

Do not assign individuals to groups as follows:

```
someuser:somepass:group=Admin.StatusView
```

The above entry is equivalent to:

```
::group=Admin.StatusView
someuser:somepass
```

which tells EZproxy that all users from that point forward should be assigned into the Admin.StatusView group.

Instead, add users to groups following this example:

```
::group=+Admin.StatusView
someuser:somepass
otheruser:otherpass
::group=-Admin.StatusView
```

This would assign both `someuser` and `otheruser` into the `StatusView` group in addition to any other groups already set up, while ensuring that users who follow will not be in this special group.

Within an authentication method such as LDAP, sample usage would be:

```
::LDAP
BindUser CN=ezproxy,CN=users,DC=yourlib,DC=org
BindPassword verysecret
URL ldap://ldapserv.yourlib.org/CN=users,DC=yourlib,DC=org?
  sAMAccountName?sub?(objectClass=person)
IfUnauthenticated; Stop
IfUser jdoe; Group +Admin.StatusView
/LDAP
```

in which specific users are identified and have the special group enabled.

When initially deploying groups, OCLC recommends using:

```
Audit Most Login.Success.Groups
```

This tells EZproxy's Audit feature to include the groups to which a user is assigned in the Other column, making it easy to determine if users are being assigned to the expected groups.

This enhancement originated via the OCLC Community Center. For more information, see Admin Users.

Session and VirtualHost Details Logged at Startup (JIRA 1530)

When EZproxy starts up, it will log the value for MaxSessions and MaxVirtualHosts to messages.txt. If either or both of these are at their limits, a warning will be logged to messages.txt about this as well. The following messages are logged:

Startup with values at default:

```
2016-07-14 09:35:01 MaxVirtualHosts set to default 200
2016-07-14 09:35:01 MaxSessions set to default 500
```

Startup with values overridden:

```
2016-07-14 09:30:18 MaxVirtualHost (MV) changed from 200 to 400
2016-07-14 09:30:18 MaxSessions (MS) changed from 500 to 1000
```

Startup when the maximum number of sessions or virtual hosts already exists:

```
2016-07-14 09:30:19 WARNING: All 200 virtual hosts are active;
MaxVirtualHosts may need to be increased
2016-07-14 09:30:19 WARNING: All 500 sessions are active;
MaxSessions may need to be increased
```

For more information, see MaxVirtualHosts (MV). Information about MaxSessions is available in the EZproxy Reference Manual.

Library Versions Display on About Page (JIRA 1568)

To make it easier for users to determine what code library versions are used by a specific version of EZproxy, the administrative /about page will now show the library versions compiled into EZproxy.

EZproxy Now Uses zlib 1.2.8 (JIRA 1607)

EZproxy now uses version 1.2.8 of the zlib library for compression and decompression. This version made available several bug fixes and other improvements.

Authentication Updates

::Relogin Support Added for Shibboleth (JIRA 905)

Previous versions of EZproxy did not support use of the `::Relogin` directive, which forces users to re-authenticate after a certain number of minutes, with Shibboleth authentication. Support for this function has been added. For example, adding the following Directive to user.txt:

```
::Relogin=120
```

will now force users authenticated via Shibboleth to re-enter their credentials after two hours.

Support Added for Authentication via Shibboleth V3.x (JIRA 1478)

EZproxy now supports authentication via Shibboleth V3.x. For more information, see Shibboleth.

Support Added for Multiple Shibboleth Certificates (JIRA 836)

In a Shibboleth configuration, EZproxy acts as a Service Provider (SP). It is common for an SP to have multiple signing and/or encryption certificates associated with it, especially when transitioning from an old certificate to a new one. A single EZproxy server was previously incapable of recognizing and supporting two certificates at the same time; it now can with this update.

In the config.txt `ShibbolethMetadata` directive, to associate more than one certificate with the EZproxy server, provide a list of certificate numbers from the /ssl administrative page, separated by commas, such as:

```
ShibbolethMetadata \
   -EntityID=https://ezproxy.yourlib.org/sp \
   -File=metadata.xml \
   -Cert=1,2
```

The details on the Manage Shibboleth administrative page have been slightly reorganized and include a new option (“EZproxy Metadata”) which displays the complete Shibboleth metadata for the EZproxy server, including multiple certificates when they are in use.

For more information, see Shibboleth Authentication.

III Username Character Limit Increased (JIRA 1339)

Previous versions of EZproxy imposed a 20 character limit on the username for sites using III authentication. This limit has been increased to 128 characters, but can be reduced if needed.

For more information, see III Authentication.

Configuration Updates

AJAX Headers Processed by Default (JIRA 1445)

Due to the growing popularity of AJAX, EZproxy now processes AJAX HTTP headers by default. In other words, the following HTTPHeader Directive no longer needs to be declared explicitly in config.txt:

```
HTTPHeader X-JSON
```

AJAX headers can still be blocked for individual resources. For more information, see HTTPHeader.

Security Updates

EZproxy Now Uses OpenSSL 1.0.2j (JIRA 1626)

EZproxy 6.2.2 was built with OpenSSL 1.0.2j, which was released on September 26, 2016. OpenSSL 1.0.2j addressed vulnerabilities and bug fixes from previous versions of OpenSSL.

For more information, see EZproxy & OpenSSL.

Support for Load Balancer SSL Management (JIRA 1599)

Some load balancers decrypt SSL client requests before forwarding them to EZproxy. Previous versions of EZproxy required the load balancer to re-encrypt the content before forwarding it to EZproxy. It is now possible to declare that a port will listen using http even though it should be considered an SSL request by adding the option -http to LoginPortSSL such as:

```
LoginPortSSL -http 443
```

When using this syntax, EZproxy does not know whether or not the load balancer is presenting a proper wildcard certificate (such as *. followed by the name of the EZproxy server). The user must explicitly indicate the type of certificate used on the load balancer by specifying **one** of the following Directives:

```
Option ForceWildcardCertificate
Option IgnoreWildcardCertificate
```

in config.txt before the LoginPortSSL -http directive.

In the most advanced scenario, a load balancer may be receiving http requests for EZproxy on port 80 and https requests on port 443 using a proper wildcard certificate with proxy by hostname, but it may also remap those requests to port 8080 for http and 8081 for https using http. In this scenario, an appropriate configuration may be:

```
Name ezproxy.yourlib.org
Option ProxyByHostname
Option ForceWildcardCertificate
LoginPort -virtual 80
LoginPortSSL -virtual 443
LoginPort 8080
LoginPortSSL -http 8081
```

Stop Logging "HTTP over HTTPS" Unless Debugging Enabled (JIRA 1122)

When SSL support was first added to EZproxy, the following diagnostic message was logged to messages.txt:

```
HTTP over HTTPS
```

whenever EZproxy received a request for http traffic on a port configured for https. There is no need to constantly log these connection errors, so this functionality has been disabled unless the following Directive is added to config.txt:

```
DebugLevel 1
```

Bug Fixes

CAS Requests Including "renew=true" Handled Properly (JIRA 1622)

The CAS authentication protocol supports a "renew" request parameter, which, when set to "true", forces the user to re-authenticate. When this option was included, previous versions of

EZproxy entered an endless loop, forcing the user to authenticate over and over. This has been corrected.

Improved Performance for SAML Metadata Retrieval (JIRA 1620)

Some sites reported performance problems when EZproxy attempted to retrieve large SAML metadata files from identity federations. The problem was caused by inadvertent parallel processing of multiple requests for these files. This has been corrected.

Option AcceptX-Forwarded-For Handles IP Addresses with Trailing Spaces (JIRA 1608)

EZproxy 6.0 introduced a bug in which, if Option AcceptX-Forwarded-For is active, IP addresses in incoming X-Forwarded-For headers were ignored when followed by trailing spaces. This has been corrected.

EZproxy Reports Invalid Obscure LDAP Passwords (JIRA 1582)

In LDAP, if BindPassword -Obscure is specified with a password that is not a valid, obscure password, previous versions of EZproxy crashed instead of reporting the issues. This has been corrected.

ebrarySite Correctly Specifies Default Port (JIRA 1578)

EZproxy 6.0 stopped providing the correct default port for the ebrarySite -URL option, causing a "connection refused" error to occur unless the required port appeared explicitly in the provided URL. This has been corrected. It is no longer necessary to specify the default http port (80), as in the example below:

```
ebrarySite –URL=http://ebookcentral.proquest.com:80 sitecode
```

Status Page View of Session Variables (JIRA 1198)

From the EZproxy /status page, there is a link to view details of each session, and from the session details, there is a link to view the session variables for that session. When this option was selected, previous versions of EZproxy showed the session variables of the user who is logged in instead of the user whose session was selected. This has been corrected.

Login User Can Now Be Set to Empty String (JIRA 904)

In user.txt, it is possible to override the value of the user field from the login form using the `login:user` variable. If this value was set to the empty string (""), previous versions of EZproxy would crash. This has been corrected.

For sites using Shibboleth authentication, setting `login:user` to the empty string in shibuser.txt similarly led to undesirable results. This value now defaults to “shibboleth” and can be changed to any other value besides the empty string.

Important Links

Product website

More product information can be found at: https://www.oclc.org/ezproxy.en.html

Support websites

Support information for this product and related products can be found at:

- Documentation: http://www.oclc.org/support/services/ezproxy.en.html
- Release notes: http://www.oclc.org/support/services/ezproxy/release-notes.en.html