EZproxy v6.3.5 release notes

Last updated: Tue, 29 May 2018 13:41:58 GMT

Release Date: November 2017

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.

Administrative actions

These items require immediate action or decisions.

<table>
<thead>
<tr>
<th>ACTION</th>
</tr>
</thead>
</table>

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 V6.3.x has many security updates that may make previous configurations in your config.txt file unnecessary, and you can remove certain directives after installing V6.3.x.

### Administrative updates

#### Community Center Enhancement: Additional Audit Search Options

Previous versions of EZproxy allowed four search combinations from the audit page. In the updated version, four still show by default along with a new "Add Row" button. When "Add Row" is pressed, a new row is added that has the same field and condition as the previous last row with the value blank and the cursor focused for input. After entering a value in the last row, it is possible to press tab and then enter to create the next row for data entry.

See also: [original EZproxy Community Center enhancement request](https://help.oclc.org/Library_Management/EZproxy/EZproxy_release_notes_and_known_issues/2017_Release_notes).  

#### Login and Menu Page Display Improved on Mobile Devices

EZproxy includes very simple templates for its login page and default menu. The templates have been updated to specify a default value to improve appearance on mobile devices.

Since sites can customize these files, EZproxy does not overwrite them on install. These changes will only be observed in the new installs.

#### New Functions to Expose Latitude and Longitude from GeoIP

The following functions have been added for use in user.txt authorization decisions:

- **Latitude()**

  This numeric function returns the latitude associated with the user's IP address based on information provided by Location directives in config.txt.

- **Longitude()**

  This numeric function returns the longitude associated with the user's IP address based on information provided by Location directives in config.txt.

- **LatLongDistance(latitude, longitude, default)**

  This numeric function returns the great-circle distance in miles from the specified latitude and longitude to the latitude and longitude associated with the user's IP address based on information provided by Location directives in config.txt or
the default value specified if there is no latitude and longitude information associated with the user's IP address.

**LatLongDistance(latitude1, longitude1, latitude2, longitude2)**

This numeric function returns the great-circle distance in miles from the first specified latitude and longitude to the second specified latitude and longitude.

In addition, the Location config.txt directive has been extended to accept -Latitude and -Longitude to allow a latitude and longitude to be associated with a manually specified location such as:

```
```

which would indicate that anything that EZproxy sees originating from an address starting 192.168 should be considered to be located at the specified latitude and longitude when using any of the functions above.

user.txt example:

```::Common
If LatLongDistance (40.102604, -83.126973, 500) < 20; {
   Msg "You appear to be located within 20 miles of OCLC"
}
/Common```

**New Function to Test Validity of Starting Point URLs**

The following function has been added for use in user.txt:

**ValidStartingPointURLTarget (url)**

This Boolean function returns 1 if the specified URL is valid to use as the target of a starting point URL and 0 if not.

For example:

```::Common
Msg -expr ValidStartingPointURLTarget("http://www.oclc.org/")
Mst -expr ValidStartingPointURLTarget("https://www.oclc.org/")
/Common```

For more information, see [Expressions](#).

**New Function to Count Active Sessions on EZproxy Server**

A new function has been added to indicate how many active sessions are associated with a particular username to support alternate method of limiting how often a single account is allowed into EZproxy.

**SessionCount (str)**

This numeric function, for use in user.txt, returns the number of active EZproxy sessions whose usernames match the
specify str. If str is the empty string (""), a count of all active sessions is returned. When used as part of initial user login processing, the user who is logging in is not included in this count as the user does not yet have an active session. During relogin processing, the user will have an active session included in this count.

user.txt example:

::Common
If login:user ne "" { 
   msg -expr login:user . " has " SessionCount (login:user) . " active sessions"
}
/Common

For more information, see Expressions.

Authentication Updates

Beta Support for Shibboleth Single Logout

EZproxy now provides beta support for Shibboleth Single Logout (SLO). Please note that this feature may still require some refinement.

To enable Single Logout (SLO), the new option -SLO must be included in the ShibbolethMetadata directive such as:

ShibbolethMetadata \
   -EntityID=EZproxyEntityID \ 
   -SLO \ 
   -File=MetadataFile \ 
   -URL=MetadataURL

When this option is included, the metadata for the EZproxy server changes to add entries for SingleLogoutService to specify the URLs at which Identity Providers can communicate with EZproxy to coordinate Single Logout. Since this option changes the metadata, the updated metadata must be provided to the Identity Provider to enable this connection.

In addition, the metadata for the Identity Provider must contain SingleLogoutService entries as well. If it does not, EZproxy will not be able to initiate or respond to a Single Logout request.

When this option is fully enabled, the following actions occur:

- If the user accesses the EZproxy/logout URL such as https://login.ezproxy.yourlib.org/logout and if the user originally authenticated via Shibboleth, EZproxy will send a Single Logout request to the Identity Provider and expects to receive a Single Logout response.
- If the user accesses the logout option of their Identity Provider, the Identity Provider can send a Single Logout request to EZproxy and EZproxy will send back a Single Logout Response.
- In either scenario, the Identity Provider is responsible for coordinating the logout requests to any other servers that
have authenticated in the same session.

The exact page seen by the user after a single logout may be provided by the Identity Provider or by EZproxy. If EZproxy presents the final page, it defaults to sending the contents of the logout.htm file from the docs subdirectory. For installations that have a mixture of Shibboleth and non-Shibboleth authentication, it is possible to create a shiblogout.htm file and place this in the docs subdirectory. If this file exists, it will be sent after single logout completes.

Generating metadata from SSL Certificate page

The SSL Certificate page provides the option to generate the metadata needed for Shibboleth before config.txt has been edited. To support non-SLO and SLO options, this page now includes different links for each of these options.

SAML AuthnRequest Documents Can Now Be Signed

EZproxy can now check an IdP’s metadata for the presence of the WantAuthnRequestsSigned attribute, and if present, sign created AuthnRequest documents.

In config.txt on the ShibbolethMetadata directive, adding -SignAuthnRequest tells EZproxy to sign its AuthnRequest documents before sending them to the IdP. It also changes the metadata displayed by EZproxy in the /shibboleth page so that SPSSODescription element includes AuthnRequestsSigned="true" which also tells the IdP to expect the requests to be signed.

Configuration Updates

Improvements to X-Forwarded-For Handling & CIDR Notation Support

When EZproxy is installed behind load balancers and network proxies, the IP address of the incoming connection to EZproxy may reflect the address of these devices instead of the remote user’s IP address. These devices are often able to insert a header to indicate the remote user’s IP address, typically using the header X-Forwarded-For.

EZproxy supports "Option AcceptX-Forwarded-For" to tell EZproxy to trust this header regardless of the source IP address from which it arrives. This directive continues to be available to enable the old behavior. If any RemoteIPInternalProxy or RemoteIPTrustedProxy directives are present, they automatically override "Option AcceptX-Forwarded-For" with the new behavior.

New config.txt directives have been created to make the decision on when to accept these headers and how to parse them.

RemoteIPHeader header

This directive tells EZproxy what header to examine for IP addresses. If not specified, the default header used is X-Forwarded-For. This presence of this directive alone does not enable X-Forwarded-For processing.

RemoveIPIInternalProxy range [range...]

RemoteIPTrustedProxy range [range...]
RemoteIPInternalProxy and RemoteIPTrustedProxy accept one or more IP address ranges which can be specified as a single IP address (e.g., 192.168.1.10), two IP addresses separated by a hyphen (e.g., 192.168.1.10-192.168.1.15), or an IP address range in CIDR notation (e.g., 192.168.1.0/24).

Both directives authorize EZproxy to evaluate an X-Forwarded-For header that appears in a request from an IP address that appears within one of the specified ranges.

When EZproxy is authorized to evaluate X-Forwarded-For, it examines the IP addresses present in the header from right to left. Each IP address is evaluated under these conditions:

1. If the IP address is invalid, EZproxy ignores the X-Forwarded-For header and stops processing.
2. If the IP address does not fall within one of the ranges specified by these directives, EZproxy uses this as the remote user's IP address and stops processing.
3. If the IP address is a private address and does not match a RemoteIPInternalProxy range, EZproxy uses this as the remote user's IP address and stops processing.
4. EZproxy ignores the current address. If there is another address to the left, it is evaluated using the previous steps.
5. If all addresses are skipped, EZproxy ignores the X-Forwarded-For header and stops processing.

These directives are modeled on corresponding Apache directives. The Apache directives accept CIDR notation. To ensure that such directives that are copied from existing configuration to EZproxy are supported, support was added for CIDR not only in these directives, but in other config.txt directives that accept IP address ranges such as AutoLoginIp, ExcludeIP, and RejectIP.

These older directives that use ranges:

ExcludeIP 10.0.0.0-10.255.255.255
ExcludeIP 192.168.0.0-192.168.255.255

Can also be stated in CIDR notation as:

ExcludeIP 10.0.0.0/8
ExcludeIP 192.168.0.0/16

Character Set Can Now be Specified

By default, EZproxy does not specify the character set (charset) of its own pages such as the login page, menu page, and administration pages. The new config.txt Charset directive allows the user to specify what charset should be sent to allow proper display of extended characters in pages such as the menu page and status page. For example:

Charset utf-8

To avoid overriding any existing configuration, EZproxy will not include a charset unless the Charset directive is present with a specific value.

Security Updates
EZproxy Now Uses OpenSSL 1.0.2m

This result of EZproxy was built with OpenSSL 1.0.2m, which was released on November 2, 2017. OpenSSL: 1.0.2m addressed vulnerabilities and bug fixes from previous versions of OpenSSL.

For more information, see EZproxy & OpenSSL.

Community Center Enhancement Request: Disable RC4 in Default SSLCipherSuite Settings

Based on a survey in the OCLC Community Center, RC4 has been disabled in the default SSLCipherSuite settings for EZproxy 6.3. 50% of respondents to the survey indicated that they had already disabled RC4 without causing access problems to proxied resources.

The default SSLCipherSuite for EZproxy 6.3 is:

```
SSLCipherSuite ALL:!EXPORT:!LOW:!aNULL:!eNULL:!SSLv2:!RC4
```

At release date, this setting earns an "A" rating at Qualys SSL Labs.

If config.txt already contains a SSLCipherSuite directive, the value specified in that directive takes priority over this new default value.

See also: original Community Center Enhancement Request.

New Logic to Detect and Block CSRF Login Attempts

EZproxy now allows enabling logic to detect and block Cross Site Request Forgery (CSRF) attempts to log in. Enabling this logic requires the following changes to login.htm, loginbu.htm, and logup.htm files:

1. Locate the form tag and verify that its value is exactly:
   ```html
   <form action="/login" method="POST">
   
   2. Immediately following the form tag, add a new line with just the caret symbol followed by the letter F as:
   ```
   ^F
   ```

   With these lines in place, edit config.txt and add:

   ```
   Option CSRFToken
   ```

   and then restart EZproxy.

   Once this logic is in place, EZproxy will send a cookie named ezproxycsrftoken with a value that must match up to a hidden field that is added to the login form where the ^F is located. These values must match for authentication to succeed.

   This option is incompatible with CAS, CGI, Shibboleth, and Ticket authentication since the actual authentication step occurs external to EZproxy. EZproxy automatically overrides this option for these methods.
If the user blocks the ezproxysrftoken cookie, the system administrator fails to put ^F in the log* .htm files, or if an attempt made it to circumvent the CSRF logic, EZproxy will display the message:

```
Login failed due to missing or invalid ezproxysrftoken cookie
```

This message can be overridden by creating the file csrf.htm in the docs directory. With this option enabled, it is possible to override the requirement for specific lines in user.txt by adding IgnoreCSRFToken such as:

```
someuser:somepass:IgnoreCSRFToken
```

This option is mainly intended for special lines that override access to the EZproxy administrative interface such as:

```
someadmin:somepass:Admin,IgnoreCSRFToken
```

**Bug fixes**

**RedirectSafe Directive Priority Corrected**

According to the documentation for RedirectSafe, if the hostname in a starting point URL exactly matches both a RedirectSafe directive and a Host/URL directive, the host should be proxied. However, RedirectSafe has not been taking priority incorrectly in this scenario. This discrepancy has been corrected.

If any configurations have been relying on this incorrect behavior, either the stanza can be removed or the RedirectSafe can be changed to NeverProxy, as this is the defined behavior for NeverProxy.

**Report Name of Non-Existent IncludeFile to messages.txt**

Starting with EZproxy 5.5, if a config.txt file tried to include a non-existent or permission-restricted file using IncludeFile, EZproxy stopped recording the name of the file to messages.txt before shutting down. The previous behavior of reporting the name of the file has been restored.

**AutoLoginIP Directives - Support for Long Usernames Restored**

Starting in EZproxy 5.7.20, some AutoLoginIP directives specifying long usernames caused starting point URLs to land at the /menu page instead of the proxied version of the resource. This has been corrected.

**Links on Default Menu Page Corrected**

Previous versions of EZproxy contained a now out-of-date link to database stanza documentation, as well as an out-of-date email address for OCLC Support, on the default /menu page. These links have been corrected.

**MaxVirtualHosts Error Link Updated**

When EZproxy has reached its MaxVirtualHosts limit, it displays an error on the /admin page along with a link to the documentation to fix the error. This link needed to be updated to reflect the current location of the documentation.
IP Addressed with Leading Zeros Parsed Correctly on Linux

IP address ranges with leading zeros were not parsing correctly on Linux, such as:

ExcludeIP 192.168.000.000-192.168.255.255

This has been corrected.

Vulnerability for Sites Using SAML-based Authentication Fixed

In October 2017, OCLC verified a vulnerability in EZproxy versions 5.4 through 6.2.2 for customers using SAML-based authentication, such as Shibboleth, ADFS, Azure, or Okta. In rare cases, this vulnerability could result in unauthorized users obtaining a logged-in session on the EZproxy server. A hot fix was previously made available, and the vulnerability has been fixed in EZproxy 6.3.

HTML Generated in Browser Redirect Now Properly Encoded

When EZproxy generated redirects, the Location header directs the browser to the new location. EZproxy also generates an HTML document in case the location header is ignored, even though no browser is known to fall back to the body.

It was theoretically possible to give EZproxy a URL that would generate an inappropriate HTML body. Although the body was never actually processed by a browser, security scanners noticed this and reported exceptions.

EZproxy now ensures that URLs in the generated bodies are appropriately encoded.

Downloads Over 2GB Work Correctly

Previous versions of EZproxy encountered problems when downloading files over 2GB in size. This has been corrected.

SIP Response Character Limits Increased

Previous versions of EZproxy imposed a limit of 512 bytes on SIP responses, which caused some responses to be truncated. This has been corrected.

Important links

Product website

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

Support website(s)

Support information for this product and related products can be found at:
• Documentation: https://www.oclc.org/support/services/ezproxy.en.html
• Release notes: https://www.oclc.org/support/services/ezproxy/release-notes.en.html