Reverse Proxy Auth Plugin

Plugin Information
View Reverse Proxy Auth on the plugin site for more information.

Older versions of this plugin may not be safe to use. Please review the following warnings before using an older version:

- Authorities cache persisted on disk

This plugin lets you delegate the authentication to the reverse proxy that you run in front of Jenkins. It also includes Authorisation, which is done via LDAP groups loaded from the HTTP header or LDAP search - based on the username.This plugin is useful in an environment where you have a reverse proxy, such as Apache, already available and configured to perform necessary user authentication. This reverse proxy must pass the authenticated user name in an HTTP header of a fixed name. With this plugin, Jenkins that run behind it will simply look at this header and use its value as the user name. In the newest release, version 1.3, this plugin also offers Authorisation mechanism. The user can have Role Based Matrix Authorization configured, which will look up into LDAP groups that can be loaded into Jenkins either via HTTP header groups field or LDAP search.

The default values for the HTTP header fields are:

1. Header User Name: X-Forwarded-User
2. Header Groups Name: X-Forwarded-Groups
3. Header Groups Delimiter: |## In case no LDAP server is informed the plugin will try to take the information from the HTTP header. When no header groups information can be retrieved, in case the user wants to do authentication only, and there is no LDAP server configured, the user retrieved from the header will have only Authenticated authority available.

Apache Configuration Example

```
<VirtualHost *:80>
  ProxyPreserveHost On
  ProxyRequests     Off
  AllowEncodedSlashes NoDecode
  Timeout 5400
  ProxyTimeout 5400

  <Proxy "*">
    Order deny,allow
    Allow from all
    Authtype BASIC
    AuthName "Please sign in with your Apache user name and password"
    # file created with htpasswd
    AuthUserFile /usr/local/apache2/conf/passwd
    Require valid-user

    # Remove these header before to set the right value after, it prevent the client from setting this header
    RequestHeader unset "X-Forwarded-User"
    RequestHeader unset "X-Forwarded-Groups"
    # Remove the basic authorization header to avoid to use it in Jenkins
    RequestHeader unset "Authorization"

    # Adds the X-Forwarded-User header that indicates the current user name.
    # this portion came from http://old.nabble.com/Forcing-a-proxied-host-to-generate-REMOTE_USER-td2911573.
html#a2914465
    RewriteEngine On

    # User to use to login in Jenkins
    RequestHeader set "X-Forwarded-User" "%{RU}e"
    # Groups are separated by |
    RequestHeader set "X-Forwarded-Groups" "%{RU}e|users"

    # strip the REALM of Kerberos Login
    # RequestHeader edit X-Forwarded-User "@REALM$" ""

    # see the Apache documentation on why this has to be lookahead
    RewriteCond %{LA-U:REMOTE_USER} (.+)
    # this actually doesn't rewrite anything. what we do here is to set RU to the match above
    # "NS" prevents flooding the error log
    RewriteRule .* - [E=RU:%1,NS]
  </Proxy>

  # send you to the Jenkins instance
  ProxyPass "/jenkins" "http://jenkins.example.com:8282/jenkins" nocanon
  ProxyPassReverse "/jenkins" "http://jenkins.example.com:8282/jenkins"
</virtualhost>
```

Notes

- Make sure that clients cannot bypass the reverse proxy. If they can send requests directly to Jenkins, then a malicious client can send in arbitrary header name with arbitrary value, thus compromising the security of Jenkins
- Make sure you configure the reverse proxy to erase the header that you use to pass the authenticated user name. This prevents malicious client from setting the header name with arbitrary value, which would ruin the security.
- If your authorisation need is simple (for example, every valid user gets full access and everyone else gets no access), then you need not use this plugin, as you can do both authentication and authorisation in the reverse proxy.
- Hit http://yourserver/whoAmI to see the actual HTTP headers your Apache is sending to Jenkins. This is useful for trouble-shooting.

Jenkins says my reverse proxy setup is broken...

Since Jenkins 1.572 this message can also appear if you don't access Jenkins through a reverse proxy: Make sure the Jenkins URL configured in the System Configuration matches the URL you're using to access Jenkins.

Symptoms

An error message is displayed in the "Manage Jenkins" page - "It appears that your reverse proxy set up is broken"

Background

For a reverse proxy to work correctly, it needs to rewrite both the request and the response. Request rewriting involves receiving an inbound HTTP call and then making a forwarding request to Jenkins (sometimes with some HTTP headers modified, sometimes not). Failing to configure the request rewriting is easy to catch, because you just won't see any pages at all.

But correct reverse proxying also involves one of two options, EITHER

- rewriting the response (for more information see Hyperlinks in HTML). The primary place where this needs to happen is the "Location" header in the response, which is used during redirects. Jenkins will send back "Location: http://actual.server:8080/jenkins/foobar" and the reverse proxy needs to rewrite this to "Location: http://nice.name/jenkins/foobar". Unfortunately, failing to configure this correctly is harder to catch; OR
- Setting the `X-Forwarded-Host` (and perhaps `X-Forwarded-Port`) header on the forwarded request. Jenkins will parse those headers and generate all the redirects and other links on the basis of those headers. Depending on your reverse proxy it may be easier to set `X-Forwarded-Host` and `X-Forwarded-Port` to the hostname and port in the original `Host` header respectively or it may be easier to just pass the original `Host` header through as `X-Forwarded-Host` and delete the `X-Forwarded-Port` header from the request. You will also need to set the `X-Forwarded-Proto` header if your reverse proxy is changing from `https` to `http` or vice-versa

So Jenkins has a proactive monitoring to make sure this is configured correctly. It uses XmlHttpRequest to request a specific URL in Jenkins (via relative path, so this will always get through provided the request is properly rewritten), which will then redirect the user to another page in Jenkins (this only works correctly if you configured the response rewriting correctly), which then returns 200.

This error message indicates that this test is failing - and the most likely cause is that the response rewriting is misconfigured. See the **Server Configuration Guides** (below) for additional tips about configuring a reverse proxy.

Note. The reverse proxy tests were improved in release 1.552 so users with previously working proxy setups may start to receive proxy warnings.

Be sure to set the `X-Forwarded-Proto` header if your reverse proxy is accessed via HTTPS and then Jenkins itself is accessed via HTTP i.e. proxying HTTPS to HTTP.

Changing the context path of Jenkins with a reverse proxy is fraught with danger. There are lots of URLs that you need to rewrite correctly, and even if you get the ones in HTML files you may miss some in javascript, CSS or XML resources.

The recommendation is to ensure that Jenkins is running at the context path that your reverse proxy is serving Jenkins at. You will have the least pain if you keep to this principle.

While it is technically possible to use rewrite rules to change the context path, you should be aware that it would be a lot of work to find and fix everything in your rewrite rules and the reverse proxy will spend most of its time rewriting responses from Jenkins. Much easier to change Jenkins to run at the context path your reverse proxy is expecting, e.g. if your reverse proxy is forwarding requests at https://manchu.example.org/foobar/ to Jenkins then you could just use `java -jar jenkins.war --prefix /foobar` to start jenkins using `/foobar` as the context path

Further Diagnosis

For further diagnosis, try using cURL:

```
curl -iL -e http://your.reverse.proxy/jenkins/manage \
   http://your.reverse.proxy/jenkins/administrativeMonitor/hudson.diagnosis.ReverseProxySetupMonitor/test
```

(assuming your Jenkins is located at http://your.reverse.proxy/jenkins/ - and is open to anonymous read access)

Server Configuration Guides

While the pages talk primarily about Apache / NGinX / HAProxy / Squid, they also have information that applies to other reverse proxies.

- Running Jenkins behind Apache
- Running Jenkins behind Nginx
- Running Jenkins behind HAProxy
- Running Jenkins behind Squid
- Running Jenkins behind IIS

If using Apache check that *nocanon* is set on *ProxyPass* and that *AllowEncodedSlashes* is set as per the Apache link above.

AllowEncodedSlashes is not inherited in Apache configs, so this directive must be placed inside the *VirtualHost* definition.

Changelog

Version 1.6.3 (2018, Feb 07)

- JENKINS-49274 - Run reverse-proxy servlet filter only after the default filter so that the authentication gets right authorities (regression in 1.6.0)

Version 1.6.2 (2018 January 30)

- JENKINS-49238 - Prevent ClassCastException when processing authorities in DefaultReverseProxyAuthenticator (regression in 1.3?)

Version 1.6.1 (2018 January 29)

- JENKINS-49236 - Prevent NullPointerException when null authContext is passed to the AuthoritiesPopulator (regression in 1.6.0)

Version 1.6.0 (2018 January 29)

- JENKINS-22402/JENKINS-48970 - Stop storing authentication context and caches on the disk
  - The change also fixes compatibility with JEP-200 in Jenkins 2.102+
  - More info: Plugins affected by fix for JEP-200
- JENKINS-31612 - Fix handling of UI filters in the plugin so that it does not cause integration issues when using other ones
- JENKINS-32909 - Prevent NullPointerException when using BASIC auth and when the user does not exist
- PR #24 - Add configuration option for groupNameAttribute to use fields other than CN as group lookup
- PR #25 - Add support of custom log output redirect
- PR #26 - Add support of custom login URL
- PR #33 - Plugin now requires Jenkins core 1.625.3 or above

Version 1.5 (2016 January 22)

- Adding LDAP connection retries
- Adding robust handling of authorisation headers for API tokens
- Adding email and name attributes to LDAP configuration
- Fixed NPE when forwarded user was not present

For more details, please checked the closed pull requests on Github: https://github.com/jenkinsci/reverse-proxy-auth-plugin/pulls

Version 1.4.0 (2014 May 27)

- Fixed JENKINS-22402 - The authorities of each user are not required in the config.xml
- Adding group membership filter setting
- Adding Cache Update Interval so Jenkins can reload user's LDAP groups on the fly, no need to restart Jenkins if users are added to new groups.

Version 1.3.3 (2014 March 14)

- The user retrieved from the HTTP header is needed when the plugin does not use the LDAP advanced options.

Version 1.3.2 (2014 March 5)

- Fixed concurrent problem with instance variable that was not being used any more, although it could cause issues with users' rights visibility.

Version 1.3.1 (2014 January 8)

- Fixed the load user by name method in the Reverse Proxy Security Realm when LDAP is activated.

Version 1.3 (2014 January 7)

- Including Authorisation via both HTTP header groups field and LDAP search.

Version 1.2 (2013 December 20)

- Including Authorisation via LDAP groups performing search based on user name.

Version 1.0.1 (2013 May 7)

- list all unprotected root actions (URLs) in the configuration, so the admin gets a hint which URLs should not be protected by the reverse proxy (supported with Jenkins core 1.495+)

Version 1.0 (2011 March 26)

- Initial release