Bug 569855 (CVE-2020-27225) - Vulnerability in Eclipse livehelp.

Status: RESOLVED FIXED

Alias: CVE-2020-27225

Product: Platform
Component: User Assistance (show other bugs)
Version: 4.18
Hardware: PC Windows 10

Importance: P3 normal (vote)
Target Milestone: 4.19 RC1
Assignee: Andrew Johnson ECA
QA Contact:

URL:
Whiteboard:
Keywords: security

Depends on:
Blocks:

Reported: 2020-12-21 05:47 EST by Andrew Johnson ECA
Modified: 2021-06-22 04:44 EDT (History)
CC List: 9 users (show)

See Also:
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Git Commit
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Git Commit
- Git Commit
- Git Commit
- Git Commit
- Git Commit
- Gerrit Change
- Git Commit
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Git Commit
- Git Commit
- Git Commit
- Git Commit
- Git Commit
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Gerrit Change
- Git Commit
- Git Commit
- Git Commit
- Git Commit
- Git Commit

Attachments		
Prototype patch for securing live help (9.36 KB, patch) 2021-02-14 12:14 EST, Andrew Johnson ECA	*no flags*	Details | Diff
dbg_aix_cmui_msg.jpg (2.29 MB, image/jpeg) 2021-04-12 14:27 EDT, Kit Lo ECA	*no flags*	Details
Add an attachment (proposed patch, testcase, etc.)		View All

Note
You need to log in before you can comment on or make changes to this bug.

Andrew Johnson ECA 2020-12-21 05:47:59 EST Description

```
I have found a possible security vulnerability in Eclipse, in the help component
(Eclipse Project/Platform/User Assistance). I opened this bug in
Community/Vulnerability Reports for visibility, but please move it if required.

I have found it is possible to run commands on the Eclipse application/system from
an unauthenticated browser running on the same system. This could be a problem on
multi-user systems as another user could close the Eclipse application, or bring up
an export page or preferences page, which could be annoying.

The problem appears to be with the liveAction.js JavaScript calling the livehelp
endpoint which allows the running of ILiveHelpAction actions or any Eclipse
command.

For Eclipse applications using ILiveHelpAction in their HTML help pages
<a class="xref"
href="javascript:liveAction(%22org.eclipse.MyPlugin.ui%22,%22org.eclipse.MyPlugin.ui.SampleAction%22,%22sampleargs%22)"
shape="rect">click here</a>

A general Eclipse applications using live help inside the help HTML has code like:
<a class="xref" shape="rect"
href="javascript:executeCommand(&quot;org.eclipse.ui.file.exit()&quot;)">click
here</a>

These call methods in
https://git.eclipse.org/r/plugins/gitiles/platform/eclipse.platform.ua/+/refs/heads/master/org.eclipse.help/livehelp.js
which has this line:
url=url+"livehelp/?
pluginID="+pluginId+"&class="+className+"&arg="+encodedArg+"&nocaching="+Math.random();

so issuing a get to
http://127.0.0.1:49834/help/livehelp/?
pluginID=org.eclipse.help.ui&class=org.eclipse.help.ui.internal.ExecuteCommandAction&arg=org.eclipse.ui.file.exit()&nocaching=123
can shut down the Eclipse application. (Modify the port number to the current port
number of the help system, easily found by a port scan.)

https://github.com/eclipse/eclipse.platform.ua/blob/master/org.eclipse.help.webapp/plugin.xml
shows livehelp goes to
https://github.com/eclipse/eclipse.platform.ua/blob/master/org.eclipse.help.webapp/src/org/eclipse/help/internal/webapp/servlet/LiveHelpServlet.java
which calls
BaseHelpSystem.runLiveHelp(pluginID, className, arg);
https://github.com/eclipse/eclipse.platform.ua/blob/6fa77889b2d1ce2d8f07b2b4a5d576fb4db6cb11/org.eclipse.help.base/src/org/eclipse/help/internal/base/BaseHelpSystem.java#L316

There is bug 223520 but that is not the same. [The topic behaviour is still there,
which allows queries like: http://127.0.0.1:52054/help/index.jsp?
topic=/org.eclipse.ui/plugin.xml which then allows a quick way to find all the
commands that might be issued by livehelp to the plugin.]

There is bug 569560 Integrated web server should only listen on localhost.
Actually, I think it does only listen on 127.0.0.1 which is good as then this bug
is not directly exploitable over the local network or the Internet.

The bug is exploitable though by browsing a malicious webpage which uses a img
element referring to the livehelp endpoint. Often, but not always, the help server
runs on a random port, which slows down an attack. A malicious HTML email
containing a remote image reference to the livehelp endpoint might also be a
vector.

The bug is also exploitable by another local user as normally there is no user-
```

based security on local TCP/IP ports.

There are variety of Eclipse commands which could cause problems - for example select all, cut, delete, save, save all, exit which might cause denial of service (if the application is shut down) or data corruption (if text is deleted in a text editor). It might be possible to issue commands to overwrite files in the system, or to save data in a place accessible to another user on the system. Commands with command parameters are particularly interesting.

It may be possible to disable live help using a plugin customization
org.eclipse.help.base/activeHelp=false
but I don't have a working example yet of how to do that.

If help is not started then the attack doesn't work, but closing the help browser after opening it does not prevent the attack.

Removing the livehelp.js file will not prevent the attack.

Removing org.eclipse.help.ui.internal.ExecuteCommandAction prevents the issue of Eclipse commands from live help but still leaves ILiveHelpAction classes which could also do bad things if invoked maliciously.

The livehelp endpoint should be a POST rather than a GET because the live help action could change the state of the system.

Ideally the livehelp endpoint would use a session cookie so that only a web browser launched by the help system could launch live help in that Eclipse application. That might not stop a user from going to a malicious webpage from inside the help browser though so we would need to consider cross-site scripting. There is then the question of whether the cookie can be intercepted but a the data just goes through the loopback interface this might not be a big risk. Using https for Eclipse local help seems hard work.

Andrew Johnson ECA 2021-01-13 06:35:14 EST Comment 1

Any updates on this? It would be nice to have this fixed for 2021-03.

Wim Jongman ECA 2021-01-13 07:23:32 EST Comment 2

(In reply to Andrew Johnson from comment #0)

Thanks for this Andrew. Does this also affect people running the live help as an infocenter like we have here:

https://remainsoftware.com/docs/openapi/help/index.jsp

Andrew Johnson ECA 2021-01-13 07:56:59 EST Comment 3

Re: comment 2. That should be safe.

Calling the live help endpoint at:
https://remainsoftware.com/docs/openapi/help/index.jsp

gives a ServletException because BaseHelpSystem.getMode() ==
BaseHelpSystem.MODE_INFOCENTER

https://github.com/eclipse/eclipse.platform.ua/blob/02a1a93aa74af33e4b7b9ca8df6db03eb9de4736/org.eclipse.help.webapp/src/org/eclipse/help/internal/webapp/servlet/LiveHelpServlet.java#L36

Though we should consider modes:
BaseHelpSystem.MODE_STANDALONE
BaseHelpSystem.MODE_WORKBENCH

The exception backtrace displayed from ServletException from the info center is like this:

```
HTTP ERROR 500 javax.servlet.ServletException
URI:    /help/livehelp/
STATUS: 500
MESSAGE:        javax.servlet.ServletException
SERVLET:
org.eclipse.equinox.http.jetty.internal.HttpServerManager$InternalHttpServiceServlet-
52759574
CAUSED BY:      javax.servlet.ServletException
Caused by:

javax.servlet.ServletException
        at
org.eclipse.help.internal.webapp.servlet.LiveHelpServlet.init(LiveHelpServlet.java:36)
        at javax.servlet.GenericServlet.init(GenericServlet.java:244)
        at
org.eclipse.equinox.http.registry.internal.ServletManager$ServletWrapper.initializeDelegate(ServletManager.java:198)
...
```

While this is potentially useful to developers, it does give a little bit more information to an attacker. This would be a different problem though.

Wim Jongman ECA 2021-01-13 09:28:53 EST Comment 4

(In reply to Andrew Johnson from comment #3)
> Re: comment 2. That should be safe.

Thanks, Andrew. I am impressed, and I'm glad you are on *our* team. :)

Is it possible for you to provide a patch?

Andrew Johnson ECA 2021-01-13 10:09:04 EST Comment 5

A simple way to disable live help is to have the line

org.eclipse.help.base/activeHelp=false

in the plugin_customization.ini, or start Eclipse with the option
-pluginCustomization=pc.ini

where pc.ini contains the configuration that you want.

To properly fix the problem could be done if live help could check the session ID compared to the session ID of when the internal or external help browser was started. That could be a bit hard to find out in a secure fashion - so perhaps the internal/external help browser needs to be started referring to an end point with a couple of query parameters - the required starting help page and a securely generated non-guessable session id. The session ID is stored to be accessible by the livehelp servlet by the code starting the browser. The browser then makes a request to the endpoint which sets a (HttpOnly? SecureSite?) cookie value with the supplied session ID to be returned to the browser, and returns with a redirect to the required web page. When a webpage with livehelp is reached then the livehelp JavaScript accesses the livehelp endpoint, and the livehelp session cookie is passed across by the browser. The livehelp plugin then checks the incoming session cookie against the stored value.

We need to be aware of attacks where the user in the proper help browser goes to a malicious web page which attempts to get the livehelp session cookie - or just access the endpoint and the browser supplies the cookie. Beware session hijacking and cross-site request forgery. I don't think we can insist on https but for local access that shouldn't be a problem - if someone has OS privileges to sniff loopback then he or she probably has administrator access already.

This gets a bit beyond my experience level - and the help starting code then gets a dependency on how the livehelp code works, so really needs help from the user assistance team.

Wim Jongman ECA 2021-01-13 10:38:31 EST Comment 6

Adding Holger for comments.

Wayne Beaton ECA 2021-01-14 16:18:08 EST Comment 7

If the team believes that this vulnerability is significant and needs to be disclosed to adopters, then we should issue a CVE. For that, I need some information.

https://www.eclipse.org/projects/handbook/#vulnerability-cve

Andrew Johnson ECA 2021-02-10 12:01:58 EST Comment 8

Any more thoughts on this one? It's not long before 2020-03 M3

Is someone going to accept this bug and move it / create a new one at Platform / User Assistance?

I can provide a test case on request to a committer who can work on this; I won't attach it to this bug as I can't delete it later.

Andrew Johnson ECA 2021-02-10 12:02:38 EST Comment 9

Correction: It's not long before 2021-03 M3

Eclipse Genie ECA 2021-02-11 12:32:35 EST Comment 10

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/176135

Holger Voormann ECA 2021-02-11 13:08:14 EST Comment 11

I don't know a clean fix for this vulnerability, only a rough workaround that uses a system property to signal when the browser is opened to display help content:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/176135

Alternatively, the workbench should pass a security token (session ID?) to the browser that is opened to display the help (depending on the configuration, the SWT browser widget in the help window or an external system browser). As far as I know, this is only possible via the URL and I do not know how to do this ("...;jsessionid=..." does not override an existing "JSESSIONID" cookie; and then how to redirect to remove the session ID from the URL again?). But I have neither experience nor knowledge here.

By the way, the following property can be used to specify a fixed port:
-Dserver_port=<port_number>

Andrew Johnson ECA 2021-02-11 16:49:45 EST Comment 12

I think this is better, but there may still be a race condition. Can a malicious application running on the same machine, or a web page in another browser, make continual requests to the prospective help web server port and make the first request before the real browser makes its first request?

Does this idea cope with the user closing then reopening the external or internal web browser, or switching preferences between them, or does live help stop working until Eclipse is restarted.

Does this protect against cross-site request forgery? If the user in the help browser goes to a malicious external web page, can that page make a request with a valid session ID?

I'll see if I can get a prototype to work as with comment 5 as an alternate approach and we can compare.

Holger Voormann ECA 2021-02-12 04:32:45 EST Comment 13

(In reply to Andrew Johnson from comment #12)
Thanks for the reply.

Yes, also with this change there would be a vulnerability by continual requests or by cross-site request forgery.

The question is, how can this be fixed? Is there a secure way to open a URL and then be sure that subsequent HTTP requests come from the same browser, not triggered via cross-site forgery? Is there a best practice that could be followed here?

The help web server is started when the "Help" view or the help browser ("Help > Help Contents") is opened for the first time. It continues running even if the help view and help browser are closed.

The only idea I have (not being a web developer), is to add a token (e.g. UUID.randomUUID().toString()) to the URL (e.g. as URI path segment parameter) to open the help browser. For example something like:
127.0.0.1:55055/help/index.jsp;7a2e6231-fd9f-4ec5-ad9a-1681792c3b11?topic=/org.eclipse.platform.doc.user/gettingStarted/qs-02a.htm
And the JavaScript of the livehelp reads the token from the path or from a cookie where it has been stored and creates an HTTP request URL containing that token as URI path segment parameter:
http://127.0.0.1:55055/help/livehelp;7a2e6231-fd9f-4ec5-ad9a-1681792c3b11?...
or as query parameter:
http://127.0.0.1:55055/help/livehelp?...&token=7a2e6231-fd9f-4ec5-ad9a-1681792c3b11
Would that be safe?

Andrew Johnson ECA 2021-02-12 10:56:38 EST Comment 14

I've got something now based on comment 5 and comment 13.

With a help request to the local machine, HelpDisplay generates a secure random token, stores it in BaseHelpSystem and appends it to the help URL as &token=<UUID>

On the server, index.jsp checks the token matches BaseHelpSystem.
If so, it generates another secure token and stores it in a per port cookie XSESSION-nnnn
and also stores the value in a session attribute XSESSION

livehelp_js.jsp is updated by the server to rewrite the page to append a &token=<UUID> from the token parameter on each live help request. I think this is needed rather than relying on the token parameter later when it is called as the individual frames might not have the token.

LiveHelpServlet checks
 the JSESSIONID cookie against the session id
 the XSESSION-nnnn cookie against the session attribute XSESSION
 the token parameter against the BaseHelpSystem value

A request from another browser on the local machine fails with
HTTP ERROR 403 JSESSIONID
as there is no JSESSIONID cookie

This cookie is marked as HttpOnly:false and SameSite:"None" so might be accessed by other malicious webpages from the same browser.

XSESSION-nnnn is HttpOnly:true and SameSite:"Strict" which should stop external web pages

I'm not sure about malicious web pages served by another webserver (e.g. another Eclipse) on localhost - the cookies would be sent, but the token query parameter might protect against that.

Is it okay to put this on Gerrit - or is that too public?

If it works then we should have some tests too.

I'd also appreciate a web app developer review of this idea.

Holger Voormann ECA 2021-02-14 10:07:12 EST Comment 15

(In reply to Andrew Johnson from comment #14)
@Andrew That sounds great!

@Wayne Could you please tell us how to proceed? I abandoned my Gerrit change, but I cannot delete it. Please tell me whether any further action is required from my side.

Andrew Johnson ECA 2021-02-14 12:14:56 EST Comment 16

Created attachment 285548 [details]
Prototype patch for securing live help

Uses query parameters and then cookies to secure the
session.

Wayne Beaton ECA 2021-02-14 14:39:15 EST Comment 17

> @Wayne Could you please tell us how to proceed? I abandoned my Gerrit
> change, but I cannot delete it. Please tell me whether any further action is
> required from my side.

My sense is that the risk/reward is low and that you should just push to Gerrit and that we move quickly to disclose the vulnerability and issue a CVE.

Andrew Johnson ECA 2021-02-15 08:36:48 EST Comment 18

The code is attachment 285548 [details] is slightly different from the idea comment 14.

With a help request to the local machine, HelpDisplay generates a secure random token, stores it
in BaseHelpSystem and appends it to the help URL as &token=<UUID>

On the server, index.jsp checks the query token matches the token stored in BaseHelpSystem.
If so, it generates another secure token and stores it in a per port cookie XSESSION-nnnn and also stores the value in a session attribute XSESSION
It generates another secure token and stores it in a session attribute LSESSION
The token in BaseHelpSystem is removed so this logon only works once.
The token query parameter is removed from the redirect URL as it is not needed any more.

livehelp_js.jsp is updated by the server to rewrite the page to append a &token=<UUID>
from the session attribute LSESSION. I think this is
needed rather than relying on the token parameter later when it is called as
the individual frames might not have the token.

LiveHelpServlet checks
 the JSESSIONID cookie against the session id
 the XSESSION-nnnn cookie against the session attribute XSESSION
 the token parameter against the session attribute LSESSION

The JSESSIONID check protects against one-off live help GET request coming
from somewhere on the local machine. It's not a secure check as it the
cookie doesn't have the right HttpOnly SameSite attributes.

The XSESSION-nnn cookie protects against another browser making
a valid live help request. There might be a way in the same browser
of making a request, but it would have to come from the same site.
It's a bit tricky, but could an attacker on the same machine, but
different user account persuade the user in the help browser to click
on a web page served on a different port on 127.0.0.1
This could either directly make a live help request, or the
XSESSION-nnn cookie would be sent to the attacker's local server for
use later.

The session attribute LSESSION helps protect against the above.
This is also only in the page from the valid session which has included
livehelp_js.jsp which has been updated by the server with a query parameter.
Does this protect against a local attacker who can get the user's
browser to request this page again? Does this need to be generated
one time only? It would break livehelp if the help is reloaded?

Also the new help uses livehelp.js so this change could stop
live help working with new help.

Andrew Johnson ECA 2021-02-16 08:06:44 EST Comment 19

Should I submit my suggested changes to Gerrit?

Holger Voormann ECA 2021-02-16 09:46:59 EST Comment 20

(In reply to Andrew Johnson from comment #16 and #18)
> Created attachment 285548 [details]
> Prototype patch for securing live help
>

Thanks Andrew!

The patch looks good to me. To my understanding, knowledge and testing, it is also save against cross-site attacks. Could you please provide it as Gerrit change?

Here are my comments:

(1) HelpDisplay
Since the help URL might not have parameters (e.g. "Help > Help Contents"), the line
```
        helpURL += "&token=" + sessid; //$NON-NLS-1$
```
should be
```
        helpURL += (helpURL.indexOf('?') < 0 ? '?' : '&') + "token=" + sessid;
//$NON-NLS-1$
```

(2) BaseHelpSystem
Instead of getter/setter I would prefer push/pop for the one-time token:
```
        public void pushLiveHelpToken(String helpSessionId) {
                this.liveHelpToken = helpSessionId;
        }

        public String popLiveHelpToken() {
                String oneTimeToken = liveHelpToken;
                liveHelpToken = null;
                return oneTimeToken;
        }
```

(3) index.jsp

(3a) The new code should be moved down some lines into the "else" branch of "if(data.isBot())" since it is not required for web crawlers.

(3b) With (2) and to discard the one-time token in all cases where the page is requested with the "token" parameter, it can be simplified to:
```
        // Live help token, to make sure that only the help opened by the
application can execute commands
                String token = request.getParameter("token"); //$NON-NLS-1$
```

```
                if (token != null &&
token.equals(BaseHelpSystem.getInstance().popLiveHelpToken())
                                && request.getSession().getAttribute("XSESSION") ==
null) { //$NON-NLS-1$
                        String token2 = UUID.randomUUID().toString();
                        request.getSession().setAttribute("XSESSION", token2);
//$NON-NLS-1$
                        int port = request.getLocalPort();
                        response.addHeader("Set-Cookie", "XSESSION-" + port + "=" +
token2 + "; HttpOnly; SameSite=Strict"); //$NON-NLS-1$ //$NON-NLS-2$ //$NON-NLS-3$
//$NON-NLS-4$
                        String token3 = UUID.randomUUID().toString();
                        request.getSession().setAttribute("LSESSION", token3);
//$NON-NLS-1$
                }
```

(4) LiveHelpServlet.java
In the new/modernized help UI prototype, livehelp does not work yet, because the livehelp JavaScript is bound to the HTML 4 frameset structure of the legacy UI. But as soon this is fixed, it should work the same way. So please, remove the experimental UI check, by replacing

```
                // @FIXME - is this needed for the new UI
                String experimentalUi =
System.getProperty("org.eclipse.help.webapp.experimental.ui"); //$NON-NLS-1$
                if (!lsessOK && experimentalUi == null) {
```

with:

```
                if (!lsessOK) {
```

(5) 2x MANIFEST.MF
The version number of "org.eclipse.help.base" has to be increased and in "org.eclipse.help.webapp" the version range of this dependency adapted accordingly.

Eclipse Genie ECA 2021-02-16 15:57:42 EST Comment 21

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/176366

Eclipse Genie ECA 2021-02-16 16:41:11 EST Comment 22

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/176367

Andrew Johnson ECA 2021-02-17 05:28:33 EST Comment 23

I've seen the approval in Gerrit.
Here are my responses to from comment 20

1. Done
2. push/pop not done as could lead to a denial of service if a supplied token was wrong - instead the matchOnceLiveHelpToken method clears the stored token if successful. We could store a hash and use a constant time check to be safer.
3. Code moved down - I left the regex check to avoid garbage data getting further into the system before being rejected.
4. Done
5. Done - the method addition would normally increment the minor version, but as they are internal packages and method just used by friend plugins no one externally should be relying on this, so I just made it a fix level change.

I have not made the following changes, which would be nice to have but not essential:

1. store and compare hashes in BaseHelpSystem
2. Return 400 (or 422) for live help where the bundle or class name is wrong.
3. Use and enforce POST not GET for live help

They can be done later if required.

Eclipse Genie ECA 2021-02-22 07:24:00 EST Comment 24

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/176367 was merged to [master].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=90661664f30d4e1225e4a68b179c6ee110cd7f75

Lakshmi P Shanmugam ECA 2021-02-23 06:31:58 EST Comment 25

@Andrew, @Holger,
Please verify the fix in the latest build -
https://download.eclipse.org/eclipse/downloads/drops4/I20210222-1800/

Andrew Johnson ECA 2021-02-23 11:07:10 EST Comment 26

Tested using https://download.eclipse.org/eclipse/downloads/drops4/I20210222-1800/download.php?dropFile=eclipse-SDK-I20210222-1800-win32-x86_64.zip

Active help still works using external browser (Firefox) and internal.
[I get a cheat sheet not found message inside Eclipse, but that is a problem with the missing CVS cheat sheet].

Copying the page link to another browser (Chrome) and clicking on the link gives a "HTTP ERROR 403 token". I don't see any way of driving a valid live help request from another browser.

Trying to access a live help URL link directly from another page loaded from a file URL in the same browser gives a similar 403 error.

Adding the token query from the live help JavaScript to the above link gives a "HTTP ERROR 403 XSESSION-62017"

Copying the help page link to another tab in the same browser (Firefox) and clicking on the example cheat sheet link does work but I think that is in the same browser session. I can't see how though a page loaded from elsewhere in the same browser could read the live help JavaScript from the help server and drive a valid request - but perhaps someone else could check this.

Does the ?token=3f48a3c1-17d2-4278-b587-943f75aa8c1e in the main URL bar of the browser look too ugly? [It is not the token supplied as part of the live help request].

Should we update https://www.eclipse.org/eclipse/platform-ua/testing/test_plan.html A7 with an additional test to confirm pasting the help URL into another browser then clicking on the link does not work?

I think this is now working as it should.

Lakshmi P Shanmugam ECA 2021-02-24 04:36:56 EST Comment 27

Thanks for the fix and verification, Andrew!

Lakshmi P Shanmugam ECA 2021-02-24 04:53:55 EST Comment 28

(In reply to Andrew Johnson from comment #26)
>
> Does the ?token=3f48a3c1-17d2-4278-b587-943f75aa8c1e in the main URL bar of
> the browser look too ugly? [It is not the token supplied as part of the live
> help request].
>

The url is much longer than before, but looks fine to me.
@Holger, WDYT?

> Should we update
> https://www.eclipse.org/eclipse/platform-ua/testing/test_plan.html A7 with
> an additional test to confirm pasting the help URL into another browser then
> clicking on the link does not work?
>
> I think this is now working as it should.

I'm not sure if the test plan is used by the team, but we could update it.

Holger Voormann ECA 2021-02-24 08:08:07 EST Comment 29

(In reply to Lakshmi P Shanmugam from comment #28)
> (In reply to Andrew Johnson from comment #26)
> >
> > Does the ?token=3f48a3c1-17d2-4278-b587-943f75aa8c1e in the main URL bar of
> > the browser look too ugly? [It is not the token supplied as part of the live
> > help request].
> >
> The url is much longer than before, but looks fine to me.
> @Holger, WDYT?
>

Yes, I'm fine with the token too.

The token is not used in the Infocenter where shorter URLs are nicer for sharing them. In contrast, the token is only visible to the user in the Eclipse IDE when an external browser instead of the help browser is used (e.g. for "Help > Help Contents", in the preferences "Help" the option "Open help contents" have to be set to "In an external browser") where sharing of URLs does not make sense.

And it does not affect security since it's a one-time token.

Thanks for reporting and fixing this vulnerability, Andrew! Great job.
Thanks also to Lakshmi for approving the fix in time for 4.19.

Wayne Beaton ECA 2021-02-24 11:52:24 EST Comment 30

The committer-only flag needs to be turned off. According to the policy, we need to do this within three months of having received the report. Since the matter has been resolved, my strong preference is to turn it off now.

I believe that this is worthy of escalating as a CVE. For that, I need the information described in the handbook.

https://www.eclipse.org/projects/handbook/#vulnerability-cve

Andrew Johnson ECA 2021-02-24 13:26:58 EST Comment 31

Here is a draft version, ready for a committer to review, edit and rewrite:

project: Eclipse Help

version: [3.0, 4.18]

cwe: CWE-306: Missing Authentication for Critical Function

summary: The Eclipse Help subsystem, version 4.18 and earlier,
does not authenticate Active help requests to the local help
web server, so an unauthenticated local attacker
can issue active help commands to the associated Eclipse platform
process or Eclipse Rich Client Platform process.

Lakshmi P Shanmugam ECA 2021-02-25 05:07:50 EST Comment 32

(In reply to Andrew Johnson from comment #31)
> Here is a draft version, ready for a committer to review, edit and rewrite:
>
> project: Eclipse Help
>
> version: [3.0, 4.18]
>
> cwe: CWE-306: Missing Authentication for Critical Function
>
> summary: The Eclipse Help subsystem, version 4.18 and earlier,
> does not authenticate Active help requests to the local help
> web server, so an unauthenticated local attacker
> can issue active help commands to the associated Eclipse platform
> process or Eclipse Rich Client Platform process.

Thanks Andrew! Looks good to me.

@Wayne,
Should we also mention the product as Eclipse SDK?
I've turned-off the committer-only flag.

Andrew Johnson ECA 2021-02-25 09:20:52 EST Comment 33

We may need to be careful explaining what is affected - by giving the Eclipse Platform version or the Eclipse help feature version?

so these would be affected
  Eclipse Platform 4.18.0.v20201202-1800 org.eclipse.platform.feature.group Eclipse.org
    Eclipse Help System 2.3.400.v20201202-1800 org.eclipse.help.feature.group Eclipse.org
but the fix should be available with the release of Eclipse Platform 4.19 and Eclipse Help System feature 2.3.500

What is the smallest unit that adopters use? How will be it clear to adopters or end-users if their Eclipse-based application
is affected?

Andrew Johnson ECA 2021-02-25 12:50:58 EST Comment 34

It is better to use a newer version of Eclipse help without the vulnerability, but a work-around for vulnerable versions is to disable active help.

Choose one of these:

A. When building the RCP application, in the product plugin_customization.ini add the lines:

Disable active help
org.eclipse.help.base/activeHelp=false

B. or possibly if the product is installed with an unpacked plug-in jars, modify plugin_customization.ini as above

C. 1. create a file pc.ini containing the lines
Disable active help
org.eclipse.help.base/activeHelp=false
   2. then start eclipse including command line options
-pluginCustomization pc.ini

D. 1. create a file pc.ini containing the lines
Disable active help
org.eclipse.help.base/activeHelp=false
   2. modify the application startup options file (for example eclipse.ini) to include the lines (before any -vmargs line):

-pluginCustomization
<change this to the full path to>/pc.ini

Test that active help has been disabled by launching help from the Eclipse application, then click on an active help link in the existing help. If the existing help does not have an active help link that does not mean that the installation is safe. It may be possible to check by adding some help with an active help link.

For example:
Window > Preferences > Help > Content
Add new information center
http://help.eclipse.org/2020-12/

Then start the help and go to:
Workbench User Guide
Reference
Preferences

Click on the 'Use the Preferences dialog pages' link.

See if a preference page comes up - if so then active help has not been disabled.

If active help has been disabled then this message appears:
"Active help is not enabled in your installation."

Wayne Beaton ECA 2021-02-26 17:03:20 EST Comment 35

I've assigned CVE-2020-27225 (Don't use this until after we push the report)

(In reply to Lakshmi P Shanmugam from comment #32)
> Should we also mention the product as Eclipse SDK?

(In reply to Andrew Johnson from comment #33)
> What is the smallest unit that adopters use? How will be it clear to
> adopters or end-users if their Eclipse-based application
> is affected?

Good questions. I'm thinking "Eclipse Platform" is the unit that the most people will understand.

Perhaps something like this:

--
In versions 4.18 and earlier of the Eclipse Platform, the Help Subsystem does not authenticate active help requests to the local help web server, allowing an unauthenticated local attacker to issue active help commands to the associated Eclipse Platform process or Eclipse Rich Client Platform process.
--

Wayne Beaton ECA 2021-03-09 13:06:08 EST Comment 36

> --
> In versions 4.18 and earlier of the Eclipse Platform, the Help Subsystem
> does not authenticate active help requests to the local help web server,
> allowing an unauthenticated local attacker to issue active help commands to
> the associated Eclipse Platform process or Eclipse Rich Client Platform
> process.
> --

Can I get a +1 to promote from a committer?

Wayne Beaton ECA 2021-03-09 13:14:46 EST Comment 37

I went ahead and promoted it. I can adjust as necessary.

Congratulation on your first CVE!

Lakshmi P Shanmugam ECA 2021-03-10 08:30:54 EST Comment 38

(In reply to Wayne Beaton from comment #36)
> > --
> > In versions 4.18 and earlier of the Eclipse Platform, the Help Subsystem
> > does not authenticate active help requests to the local help web server,
> > allowing an unauthenticated local attacker to issue active help commands to
> > the associated Eclipse Platform process or Eclipse Rich Client Platform
> > process.
> > --
>
> Can I get a +1 to promote from a committer?

+1

Lakshmi P Shanmugam ECA 2021-03-10 08:33:06 EST Comment 39

(In reply to Wayne Beaton from comment #37)
> I went ahead and promoted it. I can adjust as necessary.
>
> Congratulation on your first CVE!

Thanks Wayne for taking care of this!

Eclipse Genie ECA 2021-03-12 04:51:09 EST Comment 40

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177620

Eclipse Genie ECA 2021-03-12 06:14:28 EST Comment 41

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177621

Eclipse Genie ECA 2021-03-15 10:14:13 EDT Comment 42

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177758

Eclipse Genie ECA 2021-03-15 10:22:30 EDT Comment 43

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177759

Eclipse Genie ECA 2021-03-15 10:24:45 EDT Comment 44

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177760

Eclipse Genie ECA 2021-03-15 10:27:03 EDT Comment 45

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177761

Eclipse Genie ECA 2021-03-17 11:54:25 EDT Comment 46

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177621 was merged to [R4_11_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=28aac2514656c669ffa16acb996c77def3d4a8d4

Eclipse Genie ECA 2021-03-17 11:54:30 EDT Comment 47

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177758 was merged to [R4_8_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=57b4a4d9d52f0fab96ceed30431c0df20df1c5f6

Eclipse Genie ECA 2021-03-17 11:54:35 EDT Comment 48

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177759 was merged to [R4_7_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=719c17330c09a665201b23beb24a4e14172afbd5

Eclipse Genie ECA 2021-03-17 11:54:56 EDT Comment 49

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177760 was merged to [R4_6_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=32bdd38390b3c35d865199988aef60a24a7ebbf9

Eclipse Genie ECA 2021-03-17 11:55:06 EDT Comment 50

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/177761 was merged to [R4_5_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=d671d742159d0c15a6c69acfb334f11299e2aea2

Niraj Modi ECA 2021-03-17 12:01:37 EDT Comment 51

Done with back-porting this bug fix to below branches:
- R4_15_maintenance
- R4_11_maintenance
- R4_8_maintenance
- R4_7_maintenance
- R4_6_maintenance
- R4_5_maintenance

Eclipse Genie ECA 2021-03-22 06:12:58 EDT Comment 52

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/178200

Eclipse Genie ECA 2021-03-22 06:49:01 EDT Comment 53

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/178200 was merged to [R4_5_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=258a3e5840366137ce87e1e63c2933364980ff62

Kit Lo ECA 2021-04-12 14:27:53 EDT Comment 54

Created attachment 286087 [details]
dbg_aix_cmui_msg.jpg

Andrew, we backported the fix to Eclipse R4_6_maintenance. We received a report that an application on AIX received some "GLib-GObject-CRITICAL **: g_signal_connect_closure_by_id: assertion 'signal_id > 0' failed" messages and application was hung after applying the patch. Do you think it's related to the fix? Please investigate.

Kit Lo ECA 2021-04-12 14:28:54 EDT Comment 55

Reopen to investigate.

Andrey Loskutov ECA 2021-04-12 15:14:27 EDT Comment 56

(In reply to Kit Lo from comment #55)
> Reopen to investigate.

Please create new bug for investigation. This was delivered in 4.19 and shouldn't be used to track any possible new patches in 4.20 or other releases.

Beside this, the crash seem to be GTK related and the patch doesn't look like touching any GTK related code. I would rather assume the concrete SDK patch build was done on wrong environment and that again would speak for a different bug.

Kit Lo ECA 2021-04-12 15:24:56 EDT Comment 57

Sure! We are performing more investigation. Will open a separate bug if we have more information.

Andrew Johnson ECA 2021-04-12 16:40:57 EDT Comment 58

I don't know anything about the AIX problem though did hear about a problem with a back port to a Linux 64 version of an Eclipse RCP application. The following messages appeared on the console after clicking on help:
Error sending IPC message: Broken pipe
Error sending IPC message: Broken pipe
Error sending IPC message: Broken pipe

The message stopped after a restart.

Eclipse Genie ECA 2021-04-15 07:17:16 EDT Comment 59

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179364

Eclipse Genie ECA 2021-04-15 07:25:32 EDT Comment 60

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179365

Eclipse Genie ECA 2021-04-15 07:26:47 EDT Comment 61

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179366

Eclipse Genie ECA 2021-04-15 07:27:13 EDT Comment 62

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179367

Eclipse Genie ECA 2021-04-15 07:28:09 EDT Comment 63

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179369

Eclipse Genie ECA 2021-04-15 07:47:52 EDT Comment 64

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179364 was merged to [R4_15_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=c93f4ce3663e41bd8cc1726ca50a4638c8601565

Eclipse Genie ECA 2021-04-15 07:47:57 EDT Comment 65

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179365 was merged to [R4_11_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=24d230e592b412d7f4a76eb1693b83b9184e7b6c

Eclipse Genie ECA 2021-04-15 07:48:12 EDT Comment 66

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179366 was merged to [R4_8_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=6374397a8bb5c279ff16b362a47d196420f5659a

Eclipse Genie ECA 2021-04-15 07:49:01 EDT Comment 67

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179367 was merged to [R4_7_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=bc33dee350e157c088ddb461f37e1fae60b3d251

Eclipse Genie ECA 2021-04-15 07:49:21 EDT Comment 68

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179369 was merged to [R4_6_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=db286bbebb48eb23d26823362d7b69105e24e053

Eclipse Genie ECA 2021-04-15 09:54:23 EDT Comment 69

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179378

Eclipse Genie ECA 2021-04-15 09:56:38 EDT Comment 70

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179379

Eclipse Genie ECA 2021-04-15 09:58:01 EDT Comment 71

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179380

Eclipse Genie ECA 2021-04-15 10:00:16 EDT Comment 72

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179381

Eclipse Genie ECA 2021-04-15 10:02:31 EDT Comment 73

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179382

Eclipse Genie ECA 2021-04-15 10:04:48 EDT Comment 74

New Gerrit change created:
https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179383

Eclipse Genie ECA 2021-04-15 10:07:17 EDT Comment 75

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179378 was merged to [R4_15_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=40386324b4e86df9d0de8ff660d771104ec3767e

Eclipse Genie ECA 2021-04-15 10:07:37 EDT Comment 76

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179379 was merged to [R4_11_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=06778b3594ba1df3baebbf43467e2f026c6cb802

Eclipse Genie ECA 2021-04-15 10:07:42 EDT Comment 77

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179380 was merged to [R4_8_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=ae2c2e9a231c00579655148e1c142058a0037319

Eclipse Genie ECA 2021-04-15 10:08:06 EDT Comment 78

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179381 was merged to [R4_7_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=3d6c117324b33cbcc1d06207ba844b601289453e

Eclipse Genie ECA 2021-04-15 10:08:11 EDT Comment 79

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179382 was merged to [R4_6_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=3cbb668d6083fd87efb4fa71b8a786950717609a

Eclipse Genie ECA 2021-04-15 10:08:26 EDT Comment 80

Gerrit change https://git.eclipse.org/r/c/platform/eclipse.platform.ua/+/179383 was merged to [R4_5_maintenance].
Commit: http://git.eclipse.org/c/platform/eclipse.platform.ua.git/commit/?id=819a728b98c23e350f8bff52c009df993723f4ca

Georg Breitschopf ECA 2021-06-21 10:36:25 EDT Comment 81

(In reply to Wayne Beaton from comment #35)
> I've assigned CVE-2020-27225 (Don't use this until after we push the report)
>
> (In reply to Lakshmi P Shanmugam from comment #32)
> > Should we also mention the product as Eclipse SDK?

>
> (In reply to Andrew Johnson from comment #33)
> > What is the smallest unit that adopters use? How will be it clear to
> > adopters or end-users if their Eclipse-based application
> > is affected?
>
> Good questions. I'm thinking "Eclipse Platform" is the unit that the most
> people will understand.
>
> Perhaps something like this:
>
> --
> In versions 4.18 and earlier of the Eclipse Platform, the Help Subsystem
> does not authenticate active help requests to the local help web server,
> allowing an unauthenticated local attacker to issue active help commands to
> the associated Eclipse Platform process or Eclipse Rich Client Platform
> process.
> --

We recently upgraded our RAP-based application to RAP 3.16 which is based on Eclipse 4.19. When checking the product (WAR file) using the OWASP dependency check command line tool in version 6.2.2 (https://owasp.org/www-project-dependency-check/) the report still shows vulnerabilites related to CVE-2020-27225, e.g., for org.eclipse.equinox.http.servlet_1.7.0.v20210202-1229.jar (cpe:2.3:a:eclipse:platform:1.7.0:*:*:*:*:*:*:*, pkg:maven/org.eclipse.platform/org.eclipse.equinox.http.servlet@1.7.0). However, since the related platform plugins have been upgraded along with RAP, I expected that the report would not show any vulnerabilites related to CVE-2020-27225. In order to make this CVE disappear, it must be explicitly suppressed. An upgrade alone is not sufficcient.

Can you give me any advice on how to handle this without suppressing the CVE? Maybe it would be better to assign the CVE only to the affected components (plugins).

Andrew Johnson ECA 2021-06-22 04:44:10 EDT Comment 82

I don't know the right way to fix this but I note the following.

If I run https://github.com/jeremylong/DependencyCheck on Eclipse Memory Analyzer, it reports rows such as

org.eclipse.core.databinding.beans_1.7.200.v20210111-0759.jar
cpe:2.3:a:eclipse:platform:1.7.200:*:*:*:*:*:*:*
pkg:maven/org.eclipse.platform/org.eclipse.core.databinding.beans@1.7.200
HIGH 1 Highest 45

org.eclipse.e4.ui.workbench3_0.15.500.v20201021-1339.jar
cpe:2.3:a:eclipse:platform:0.15.500:*:*:*:*:*:*:*
pkg:maven/org.eclipse.platform/org.eclipse.e4.ui.workbench3@0.15.500 HIGH 1
Highest 40

so it seems to map every Eclipse plugin version to the Eclipse platform of the same version number. That's not right as all sorts of plugins make up the Eclipse Platform, and results in lots of warnings about this CVE. The problem has been reported as an issue: https://github.com/jeremylong/DependencyCheck/issues/3255

Dependency Check uses https://nvd.nist.gov/products/cpe
"Official Common Platform Enumeration (CPE) Dictionary

CPE is a structured naming scheme for information technology systems, software, and packages. Based upon the generic syntax for Uniform Resource Identifiers (URI), CPE includes a formal name format, a method for checking names against a system, and a description format for binding text and tests to a name."

So somehow there should be a better mapping of plugin and versions to the Eclipse platform.
Also, perhaps this should have been reported as a CVE against the Eclipse Help subsytem as that might be a separately upgradable item, but I think even this is going to have problems.
So for Memory Analyzer 1.12
features include:

Eclipse.org Eclipse RCP 4.20.0.v20210611-1600 org.eclipse.rcp
Eclipse.org Help System Base 2.3.600.v20210611-1600 org.eclipse.help
Eclipse Memory Analyzer Memory Analyzer RCP 1.12.0.202106190923
org.eclipse.mat.ui.rcp.feature
Eclipse Modeling Project EMF Common 2.22.0.v20210319-0732
org.eclipse.emf.common
Eclipse Modeling Project EMF Ecore 2.24.0.v20210405-0628
org.eclipse.emf.ecore

but the help plugins are:

Eclipse.org Help System Base 4.3.300.v20210611-1600
org.eclipse.help.base
Eclipse.org Help System Core 3.9.0.v20210507-0822
org.eclipse.help
Eclipse.org Help System UI 4.3.0.v20210409-1726 org.eclipse.help.ui
Eclipse.org Help System Webapp 3.10.300.v20210507-0822
org.eclipse.help.webapp

However in the CPE database are only the following for "eclipse help".

Vendor
Product
Version
Update
Edition
Language
cpe:2.3:a:ibm:eclipse_help_system:3.4.3:*:*:*:*:*:*:*
ibm
eclipse_help_system
3.4.3

cpe:2.3:a:ibm:eclipse_help_system:3.6.2:*:*:*:*:*:*:*
ibm
eclipse_help_system
3.6.2

and clicking on the associated CVEs (nothing to do with this bug, but for illustration: CVE 2013 0464
"Multiple cross-site scripting (XSS) vulnerabilities in IBM Eclipse Help System (IEHS) 3.4.3 and 3.6.2, as used in IBM SPSS Data Collection 6.0, 6.0.1, and 7.0, allow remote attackers to inject arbitrary web script or HTML via a crafted URL."
so the CPE versions above seem to be the Eclipse Platform versions.