

Bug 567921 (CVE-2020-27216) - Jetty vulnerable to temporary directory hijacking

Status: RESOLVED FIXED

Alias: CVE-2020-27216

Product: Community

Component: Vulnerability Reports (show other bugs)

Version: unspecified

Hardware: All Unix All

Importance: P3 major (vote)

Target Milestone: ---

Assignee: Security vulnerabilitied reported against Eclipse projects

QA Contact:

URL: https://cve.mitre.org/cgi-bin/cvename...

Whiteboard:

Keywords: security

Depends on:

Blocks:

Reported: 2020-10-16 04:49 EDT by Greg Wilkins

Modified: 2020-10-22 21:48 EDT (History)

CC List: 2 users (show)

See Also:

Attachments

Add an attachment (proposed patch, testcase, etc.)

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

Greg Wilkins 2020-10-16 04:49:53 EDT Description

A vulnerability in jetty has been reported where it's creation of a temporary directory can be hijacked by a local attacker with access to /tmp

The full description of the vulnerability and its fix is at

https://github.com/eclipse/jetty.project/security/advisories/GHSA-g3wg-6mcf-8116#advisory-comment-63053

The CVE score for the vulnerability has been assessed as 7.8/10 (AV:L/AC:L/FR:L/UI:N/S:U/C:H/I:H/A:H) and a CVE should be created for the vulnerability.

Wayne Beaton 2020-10-16 18:22:09 EDT Comment 1

I need the information for the CVE, please.

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

Joakim Erdfelt 2020-10-17 06:43:42 EDT Comment 2

(In reply to Wayne Beaton from comment #1)  
> I need the information for the CVE, please.  
>  
> https://www.eclipse.org/projects/handbook/#vulnerability-cve

The most up to date information about this is present at the advisory management page on github for this. (a currently private page which eclipse foundation has access to)

https://github.com/eclipse/jetty.project/security/advisories/GHSA-g3wg-6mcf-8116

It's a pretty big issue in scope, the handbook ... "A one or two sentence summary of the issue which clearly identifies the Eclipse project/product and impacted versions." ... falls apart here.

This impacts \*ALL\* released versions of Jetty, from Eclipse and prior, even releases that predate the move to Eclipse. So the advisory needs to indicate that.

The range of versions:

1.0 thru 9.4.32.v20200930  
10.0.0.alpha1 thru 10.0.0.beta2  
11.0.0.alpha1 thru 11.0.0.beta2

This includes eclipse and non-eclipse releases.

The short description:

On Unix like systems, the system's temporary directory is shared between all users on that system. A collocated user can observe the process of creating a temporary sub directory in the shared temporary directory and race to complete the creation of the temporary subdirectory. If the attacker wins the race then they will have read and write permission to the subdirectory used to unpack web applications, including their WEB-INF/lib jar files and JSP files. If any code is ever executed out of this temporary directory, this can lead to a local privilege escalation vulnerability.

Additionally, any user code uses of WebApplicationContext::getTempDirectory would similarly be vulnerable.

Additionally, any user application code using the ServletContext attribute for the javax.servlet.ServletContext.TEMPDIR will also be impacted.

Suggestion for Eclipse Foundation:

The Eclipse handbook for vulnerabilities should be a bit more flexible here. Also updated to integrate the entire github advisory process, which is frankly incredibly useful for coordinating between an open source project, the security researchers, and other impacted projects in a private way.

Wayne Beaton 2020-10-17 12:57:54 EDT Comment 3

> The most up to date information about this is present at the advisory management page on github for this. (a currently private page which eclipse foundation has access to)

Great. But we need to report this to the central authority in the format that they'll accept. They expect a paragraph description. We can (and will) provide pointers to more information. Note that those pointers need to be publicly accessible at the time we push the report.

> It's a pretty big issue in scope, the handbook ... "A one or two sentence summary of the issue which clearly identifies the Eclipse project/product and impacted versions." ... falls apart here.

For the CVE report, this is all that we have space to provide. Think of this as the

thesis statement that the reader will use to determine whether or not they will pursue it further.

> This impacts \*ALL\* released versions of Jetty, from Eclipse and prior, even  
> releases that predate the move to Eclipse. So the advisory needs to  
> indicate that.

Yes. This is why we ask for the versions affected. Multiple ranges are supported by the reporting format.

> The Eclipse handbook for vulnerabilities should be a bit more flexible here.  
> Also updated to integrate the entire github advisory process, which is  
> frankly incredibly useful for coordinating between an open source project,  
> the security researchers, and other impacted projects in a private way.

There is nothing in the handbook that says that you can't use the GitHub process for dealing with your vulnerability. I don't believe that I've given any indication to the contrary. There is a bit in the handbook that recommends the use of Bugzilla to resolve the matter, but that is not a requirement.

The handbook specifically states:

--

To request a CVE Number assignment, the vulnerability must be captured in a Eclipse Bugzilla record. The project team can track work on a vulnerability elsewhere, but the vulnerability reporting is tracked via Bugzilla.

--

So... do what you feel is best to resolve the vulnerability and open a bug to request the assignment of a CVE and track the reporting process.

I'm not sure specifically what you think needs to change.

I'll point out that the vulnerability policy was developed under the supervision of the Eclipse Foundation's Security Team which includes an Eclipse Jetty committer. If there is some gap in our policy or process, then he is in an excellent position to drive necessary changes. Our implementation of the policy documented in the handbook, specifically with regard to reporting CVEs is informed primarily by my understanding of the API by which we interact with the central authority. If I've made an error, I will happily correct it.

Finally, the handbook is a component of the Eclipse Dash project. If you'd like to see specific changes to it, then you can open an issue and start a discussion there. Contributions are, of course, always welcome.

By way of expectation management, I'm currently waiting for a block of new CVE ids to be assigned to the Eclipse Foundation. The central authority's process requires that we use up the complete block that we'd been previously assigned before requesting an additional block (and I used the last one in our previous block a couple of days ago). Based on past behaviour, this should be resolved in a day or so. I cannot push the report until I get that block of ids.

Greg Wilkins  2020-10-19 07:15:24 EDT [Comment 4](#)


Joakim,

my understanding (albeit second hand and fuzzy) of the CVE process is that Github will only do a CVE on behalf of projects that don't already have an authority that can do them. So, even if we had clicked the CVE button in the advisory, I think Github would have bounced us back to Eclipse.

Having said that, even without the CVE, the github advisory process appears really good and useful and should dovetail with the eclipse process.

Wayne,

I did include your ID on the github security advisory, so you should be able to see it, including all the details, which I believe are in a format suitable for the CVE declarations. Either way, I will repost them in my next comment so they are on the eclipse record... hopefully I can get the formatting as well.

Greg Wilkins  2020-10-19 07:18:55 EDT [Comment 5](#)

Affected versions: <= 9.4.32.v20200930, <= 10.0.0.beta2, <= 11.0.0.beta2  
Patched versions: 9.4.33.v202010??, 10.0.0.beta3, 11.0.0.beta3  
Packages: org.eclipse.jetty:jetty-webapp, org.mortbay.jetty:jetty-webapp  
Package ecosystem: maven  
Severity: High  
CVSSv3.1 Score: 7.8/10 (AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)

### Impact

On Unix like systems, the system's temporary directory is shared between all users on that system. A collocated user can observe the process of creating a temporary sub directory in the shared temporary directory and race to complete the creation of the temporary subdirectory. If the attacker wins the race then they will have read and write permission to the subdirectory used to unpack web applications, including their WEB-INF/lib jar files and JSP files. If any code is ever executed out of this temporary directory, this can lead to a local privilege escalation vulnerability.

Additionally, any user code uses of [WebAppContext:getTempDirectory] ([https://www.eclipse.org/jetty/javadoc/9.4.31.v20200723/org/eclipse/jetty/webapp/WebAppContext.html#getTempDirectory\(\)](https://www.eclipse.org/jetty/javadoc/9.4.31.v20200723/org/eclipse/jetty/webapp/WebAppContext.html#getTempDirectory())) would similarly be vulnerable.

Additionally, any user application code using the 'ServletContext' attribute for the tempdir will also be impacted.

See: <https://javaee.github.io/javaee-spec/javadocs/javax/servlet/ServletContext.html#TEMPDIR>

For example:

```
java
import java.io.File;
import java.io.IOException;
import javax.servlet.ServletContext;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class ExampleServlet extends HttpServlet {
    @Override
    protected void doGet(HttpServletRequest req, HttpServletResponse resp) throws
ServletException, IOException {
        File tempDir =
(File)getContext().getAttribute(ServletContext.TEMPDIR); // Potentially
compromised
        // do something with that temp dir
    }
}
```

Example: The JSP library itself will use the container temp directory for compiling the JSP source into Java classes before executing them.

### CVSSv3.1 Evaluation

This vulnerability has been calculated to have a [CVSSv3.1 score of 7.8/10 (AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H)] (<https://nvd.nist.gov/vuln-metrics/cvss/v3-calculator?vector=AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H&version=3.1>)

### Patches

Fixes were applied to the 9.4.x branch with:

-

<https://github.com/eclipse/jetty.project/commit/53a0a0e9b25a6309bf24ee3b10984f4145701edeb>  
-  
<https://github.com/eclipse/jetty.project/commit/9ad6beb80543b392c91653f6bfce233fc75b9d5f>

These will be included in releases: 9.4.33, 10.0.0.beta3, 11.0.0.beta3

### Workarounds

A work around is to set a temporary directory, either for the server or the context, to a directory outside of the shared temporary file system.  
For recent releases, a temporary directory can be created simple by creating a directory called 'work' in the \${jetty.base} directory (the parent directory of the 'webapps' directory).  
Alternately the java temporary directory can be set with the System Property 'java.io.tmpdir'. A more detailed description of how jetty selects a temporary directory is below.

The Jetty search order for finding a temporary directory is as follows:

1. If the 'WebAppContext' has a temp directory specified ([https://www.eclipse.org/jetty/javadoc/current/org/eclipse/jetty/webapp/WebAppContext.html#setTempDirectory\(java.io.File\)](https://www.eclipse.org/jetty/javadoc/current/org/eclipse/jetty/webapp/WebAppContext.html#setTempDirectory(java.io.File))), use it.
2. If the 'ServletContext' has the 'javax.servlet.context.tmpdir' attribute set, and if directory exists, use it.
3. If a '\${jetty.base}/work' directory exists, use it (since Jetty 9.1)
4. If a 'ServletContext' has the 'org.eclipse.jetty.webapp.basetempdir' attribute set, and if the directory exists, use it.
5. Use 'System.getProperty("java.io.tmpdir")' and use it.

Jetty will end traversal at the first successful step.

To mitigate this vulnerability the directory must be set to one that is not writable by an attacker. To avoid information leakage, the directory should also not be readable by an attacker.

### Setting a Jetty server temporary directory.

Choices 3 and 5 apply to the server level, and will impact all deployed webapps on the server.

For choice 3 just create that work directory underneath your '\${jetty.base}' and restart Jetty.

For choice 5, just specify your own 'java.io.tmpdir' when you start the JVM for Jetty.

```
``` shell
[jetty-distribution]$ java -Djava.io.tmpdir=/var/web/work -jar start.jar
```
```

### Setting a Context specific temporary directory.

The rest of the choices require you to configure the context for that deployed webapp (seen as '\${jetty.base}/webapps/<context>.xml')

Example (excluding the DTD which is version specific):

```
``` xml
<Configure class="org.eclipse.jetty.webapp.WebAppContext">
  <Set name="contextPath"><Property name="foo"/></Set>
  <Set name="war">/var/web/webapps/foo.war</Set>
  <Set name="tempDirectory">/var/web/work/foo</Set>
</Configure>
```
```

### References

- <https://github.com/eclipse/jetty.project/issues/5451>
- [CWE-378: Creation of Temporary File With Insecure Permissions] (<https://cwe.mitre.org/data/definitions/378.html>)
- [CWE-379: Creation of Temporary File in Directory with Insecure Permissions] (<https://cwe.mitre.org/data/definitions/379.html>)
- [CodeQL Query PR To Detect Similar Vulnerabilities] (<https://github.com/github/codeql/pull/4473>)

### Similar Vulnerabilities

Similar, but not the same.

- JUnit 4 - <https://github.com/junit-team/junit4/security/advisories/GHSA-269g-pwp5-87pp>
- Google Guava - <https://github.com/google/guava/issues/4011>
- Apache Ant - <https://nvd.nist.gov/vuln/detail/CVE-2020-1945>
- JetBrains Kotlin Compiler - <https://nvd.nist.gov/vuln/detail/CVE-2020-15824>

### For more information

The original report of this vulnerability is below:

> On Thu, 15 Oct 2020 at 21:14, Jonathan Leitschuh <[jonathan.leitschuh@gmail.com](mailto:jonathan.leitschuh@gmail.com)> w  
> Hi WebTide Security Team,



>  
> I'm a security researcher writing some custom CodeQL queries to find Local Tempora



>  
> <https://lgtm.com/query/5615014766184643449/>

>  
> I've recently been looking into security vulnerabilities involving the temporary c  
> There exists a race condition between the deletion of the temporary file and the c




```
> ```java
> // ensure file will always be unique by appending random digits
> tmpDir = File.createTempFile(temp, ".dir", parent); // Attacker knows the full pat
> // delete the file that was created
> tmpDir.delete(); // Attacker sees file is deleted and begins a race to create thei
> // and make a directory of the same name
> // SECURITY VULNERABILITY: Race Condition! - Attacker beats Jetty and now owns thi
> tmpDir.mkdirs();
> ```
```



>  
> <https://github.com/eclipse/jetty.project/blob/1b59672b7f668b8a421690154b98b4b2b03f>



>  
> In several cases the 'parent' parameter will not be the system temporary directory,  
>  
>  
> <https://github.com/eclipse/jetty.project/blob/1b59672b7f668b8a421690154b98b4b2b03f>  
>  
>  
> If any code is ever executed out of this temporary directory, this can lead to a  
>  
> Would your team be willing to open a GitHub security advisory to continue the disc  
>  
> \*\*This vulnerability disclosure follows Google's [90-day vulnerability disclosure  
>  
> Cheers,  
> Jonathan Leitschuh


Wayne Beaton  2020-10-20 06:34:21 EDT [Comment 6](#)

I've assigned CVE-2020-27216.

I have the report ready to be pushed to the central authority. Per their requirements, the links that we provide (which include this bug and the GitHub link that you provided) need to be publicly accessible when I push.

Are you ready for me to push?

For future reference, I don't require (and didn't ask) that you copy all of the details of the issue to the bug. All I need is the information that you want me to relay to the central authority. I'll reiterate that the information that I ask for (and can accept) is entirely to satisfy the requirements of the central authority.

Joakim Erdfelt  2020-10-20 11:37:19 EDT [Comment 7](#)

(In reply to Wayne Beaton from [comment #6](#))

> I've assigned CVE-2020-27216.  
>  
> I have the report ready to be pushed to the central authority. Per their  
> requirements, the links that we provide (which include this bug and the  
> GitHub link that you provided) need to be publicly accessible when I push.  
>  
> Are you ready for me to push?  
>  
> For future reference, I don't require (and didn't ask) that you copy all of  
> the details of the issue to the bug. All I need is the information that you  
> want me to relay to the central authority. I'll reiterate that the  
> information that I ask for (and can accept) is entirely to satisfy the  
> requirements of the central authority.

Thank you.

We are working through a checklist of tasks before the GH advisory page goes public.

See: <https://github.com/eclipse/jetty.project/security/advisories/GHSA-g3wq-6mcf-8jj6#advisory-comment-63124>


Once we get through that checklist we'll be able to notify this issue to publish the CVE details through the normal processes.

Greg Wilkins  2020-10-22 15:58:34 EDT [Comment 8](#)

Wayne,

We are ready to publish the github advisory and the CVE. However something has changed on github and I've lost the permission to publish the advisory (see [https://bugs.eclipse.org/bugs/show\\_bug.cgi?id=56866](https://bugs.eclipse.org/bugs/show_bug.cgi?id=56866))

Are you able to publish both?

Wayne Beaton  2020-10-22 20:03:47 EDT [Comment 9](#)

> Are you able to publish both?

Apparently I have the necessary superpowers. I've published the advisory.

I've taken the committers-only flag off of this issue and have pushed the CVE report to the central authority.

<https://github.com/CVEProject/cvelist/pull/5158>

I'll leave this bug open until after I've verified that they've merged the commit.