CVE-2019-0232：Apache Tomcat RCE漏洞分析

lyle / 2019-04-23 10:30:00 / 浏览数 8694 安全技术 漏洞分析 顶(1) 踩(0)

简介

利用前提

该漏洞是由于Tomcat CGI将命令行参数传递给Windows程序的方式存在错误，使得CGIServlet被命令注入影响。

该漏洞只影响Windows平台，要求启用了CGIServlet和enableCmdLineArguments参数。但是CGIServlet和enableCmdLineArguments参数默认情况下都不启用。

时间线

- 报告漏洞 2019-3-3
- 漏洞公开 2019-4-10

漏洞影响范围

- Apache Tomcat 9.0.0.M1 to 9.0.17
- Apache Tomcat 8.5.0 to 8.5.39
- Apache Tomcat 7.0.0 to 7.0.93

复现

笔者使用的复现环境为9.0.12 + JRE 1.8.0。

首先进行CGI相关的配置，在 `conf/web.xml` 中启用CGIServlet：

```
<servlet>
   <servlet-name>cgi</servlet-name>
   <servlet-class>org.apache.catalina.servlets.CGIServlet</servlet-class>
   <init-param>
     <param-name>cgiPathPrefix</param-name>
     <param-value>WEB-INF/cgi-bin</param-value>
   </init-param>
   <init-param>
     <param-name>enableCmdLineArguments</param-name>
     <param-value>true</param-value>
   </init-param>
   <init-param>
     <param-name>executable</param-name>
     <param-value></param-value>
   </init-param>
   <load-on-startup>5</load-on-startup>
</servlet>
```

这里主要的设置是 `enableCmdLineArguments` 和 `executable` 两个选项。 `enableCmdLineArguments` 启用后才会将Url中的参数传递到命令行，`executable` 指定了执行的二进制文件，默认是 `perl`，需要置为空才会执行文件本身。

同样在 `conf/web.xml` 中启用cgi的servlet-mapping

```
<servlet-mapping>
   <servlet-name>cgi</servlet-name>
   <url-pattern>/cgi-bin/*</url-pattern>
</servlet-mapping>
```

之后修改 `conf/context.xml` 的 `<Context>` 添加 `privileged="true"`属性，否则会没有权限

```
<Context privileged="true">

   <!-- Default set of monitored resources. If one of these changes, the    -->
   <!-- web application will be reloaded.                                   -->
   <WatchedResource>WEB-INF/web.xml</WatchedResource>
   <WatchedResource>WEB-INF/tomcat-web.xml</WatchedResource>
   <WatchedResource>${catalina.base}/conf/web.xml</WatchedResource>
```

```
   <!-- Uncomment this to disable session persistence across Tomcat restarts -->
   <!--
   <Manager pathname="" />
   -->
</Context>
```

然后在 `ROOT\WEB-INF` 下创建 `cgi-bin` 目录, 并在该目录下创建一个内容为 `echo Content-type: text/html` 的 `e.bat` 文件。

配置完成后，启动tomcat，访问 `http://127.0.0.1:8080/cgi-bin/e.bat?&ver` ，可以看到命令执行成功。

原理

漏洞相关的代码在 `tomcat\java\org\apache\catalina\servlets\CGIServlet.java` 中，CGIServlet提供了一个cgi的调用接口，在启用 `enableCmdLineArguments` 参数时，会根据RFC 3875来从Url参数中生成命令行参数，并把参数传递至Java的 `Runtime` 执行。 这个漏洞是因为 `Runtime.getRuntime().exec` 在Windows中和Linux中底层实现不同导致的。下面以一个简单的case来说明这个问题，在Windows下创建arg.bat：

```
rem arg.bat
echo %*
```

并执行如下的Java代码

```
String [] cmd={"arg.bat", "arg", "&", "dir"};
Runtime.getRuntime().exec(cmd);
```

在Windows下会输出 `arg` 和 `dir` 命令运行后的结果。同样的，用类似的脚本在Linux环境下测试：

```
# arg.sh
for key in "$@"
do
   echo '$@' $key
done
```

```
String [] cmd={"arg.sh", "arg", "&", "dir"};
Runtime.getRuntime().exec(cmd);
```

此时的输出为

```
$@ arg
$@ &
$@ dir
```

导致这种输出的原因是在JDK的实现中 `Runtime.getRuntime().exec` 实际调用了 `ProcessBuilder` ，而后 `ProcessBuilder` 调用 `ProcessImpl`使用系统调用 `vfork` ，把所有参数直接传递至 `execve`。

用 `strace -F -e vfork,execve java Main` 跟踪可以看到上面的Java代码在Linux中调用为

```
execve("arg.sh", ["arg.sh", "arg", "&", "dir"], [/* 23 vars */])
```

而如果跟踪类似的PHP代码 `system('a.sh arg & dir');` ，得到的结果为

```
execve("/bin/sh",  ["sh", "-c", "a.sh arg & dir"], [/* 23 vars */])
```

所以Java的 `Runtime.getRuntime().exec` 在CGI调用这种情况下很难有命令注入。而Windows中创建进程使用的是 `CreateProcess` ，会将参数合并成字符串，作为 `lpComandLine` 传入 `CreateProcess` 。程序启动后调用 `GetCommandLine` 获取参数，并调用 `CommandLineToArgvW` 传至 argv。在Windows中，当 `CreateProcess` 中的参数为 bat 文件或是 cmd 文件时，会调用 `cmd.exe`，故最后会变成 `cmd.exe /c "arg.bat & dir"`，而Java的调用过程并没有做任何的转义，所以在Windows下会存在漏洞。

除此之外，Windows在处理参数方面还有一个特性，如果这里只加上简单的转义还是可能被绕过， 例如 `dir "\"&whoami"` 在Linux中是安全的，而在Windows会执行命令。

这是因为Windows在处理命令行参数时，会将 `"` 中的内容拷贝为下一个参数，直到命令行结束或者遇到下一个 `"` ，但是对 `\"` 的处理有误。同样用 `arg.bat` 做测试，可以发现这里只输出了 `\` 。因此在Java中调用批处理或者cmd文件时，需要做合适的参数检查才能避免漏洞出现。

修复方式

开发者在 patch 中增加了 `cmdLineArgumentsDecoded` 参数，这个参数用来校验传入的命令行参数，如果传入的命令行参数不符合规定的模式，则不执行。

校验写在 `setupFromRequest` 函数中：

```
String decodedArgument = URLDecoder.decode(encodedArgument, parameterEncoding);
if (cmdLineArgumentsDecodedPattern != null &&
       !cmdLineArgumentsDecodedPattern.matcher(decodedArgument).matches()) {
```

```
   if (log.isDebugEnabled()) {
       log.debug(sm.getString("cgiServlet.invalidArgumentDecoded",
               decodedArgument, cmdLineArgumentsDecodedPattern.toString()));
   }
   return false;
}
```

不通过时，会将 `CGIEnvironment` 的 `valid` 参数设为 `false` ，在之后的处理函数中会直接跳过执行。

```
if (cgiEnv.isValid()) {
   CGIRunner cgi = new CGIRunner(cgiEnv.getCommand(),
                                 cgiEnv.getEnvironment(),
                                 cgiEnv.getWorkingDirectory(),
                                 cgiEnv.getParameters());

   if ("POST".equals(req.getMethod())) {
       cgi.setInput(req.getInputStream());
   }
   cgi.setResponse(res);
   cgi.run();
} else {
   res.sendError(404);
}
```

修复建议

1. 使用更新版本的Apache
   Tomcat。这里需要注意的是，虽然在9.0.18就修复了这个漏洞，但这个更新是并没有通过候选版本的投票，所以虽然9.0.18没有在被影响的列表中，用户仍需要下载9.0.
2. 关闭enableCmdLineArguments参数

参考链接

- https://tomcat.apache.org/security-9.html
- https://tomcat.apache.org/tomcat-9.0-doc/cgi-howto.html
- https://github.com/apache/tomcat/commit/4b244d8
- https://github.com/pyn3rd/CVE-2019-0232/
- https://tools.ietf.org/html/rfc3875
- https://blogs.msdn.microsoft.com/twistylittlepassagesallalike/2011/04/23/everyone-quotes-command-line-arguments-the-wrong-way/
- https://codewhitesec.blogspot.com/2016/02/java-and-command-line-injections-in-windows.html

上一篇：POSCMS v3.2.0漏洞复现 下一篇：Linux权限维持之影子SUID的利用

1. 0 条回复
   - 动动手指，沙发就是你的了！