Tomcat CVE-2017-12616 □信息泄露漏洞分析

hades / 2017-09-21 05:35:48 / 浏览数 4123 安全技术 漏洞分析 顶(0) 踩(0)

Author■n1nty
■■■http://mp.weixin.qq.com/s/sulJSg0Ru138oASiI5cYAA

最近的几个 Tomcat 的 CVE

1. CVE-2017-5664 Tomcat Security Constraint Bypass
2. CVE-2017-12615 远程代码执行漏洞
3. CVE-2017-12616 信息泄露漏洞

共性

1. 都是鸡肋
2. 都跟 JspServlet 和 DefaultServlet 有关系

CVE-2017-12615 这个远程代码执行遍地都是了，好像没有人看 CVE-2017-12616 造成的 JSP 源代码泄露的问题。这里简单写一下。

CVE-2017-12616

要求

目标利用 VirtualDirContext 来挂载虚拟目录。挂载虚拟目录的需求应该还是有一些的，所以应该比开 PUT 的概率要大一些，不过也是鸡肋。

简要分析

要造成 Jsp 源代码泄露，肯定需要让 DefaultServlet 来处理 jsp 的请求。Tomcat 利用类似 JNDI 的方式来管理 Web 资源（JSP，静态文件，Class 等）。默认情况下，资源由 FileDirContext 来进行管理。 而利用 VirtualDirContext 挂载的虚拟目录，是由 VirtualDirContext 来管理的。

通过类似 CVE-2017-12615 的利用方式访问虚拟目录中的资源，让请求由 DefaultServlet 处理，Tomcat 从 VirtualDirContext 管理的资源中获取访问的 jsp 文件（通过 doLookup 方法），直接将内容返回，造成源代码泄露。

为什么只有虚拟目录存在这个漏洞？因为非虚拟目录默认由 FileDirContext 管理的。FileDirContext 中有存在有一个名为 file 的检查方法。

```
protected File file(String name) {

   File file = new File(base, name);
   if (file.exists() && file.canRead()) {

       if (allowLinking)
           return file;

       // Check that this file belongs to our root path
       String canPath = null;
       try {
           canPath = file.getCanonicalPath();
       } catch (IOException e) {
           // Ignore
       }
       if (canPath == null)
           return null;

       // Check to see if going outside of the web application root
       if (!canPath.startsWith(absoluteBase)) {
           return null;
       }

       // Case sensitivity check - this is now always done
       String fileAbsPath = file.getAbsolutePath();
       if (fileAbsPath.endsWith("."))
           fileAbsPath = fileAbsPath + "/";
       String absPath = normalize(fileAbsPath);
       canPath = normalize(canPath);
       if ((absoluteBase.length() < absPath.length())
           && (absoluteBase.length() < canPath.length())) {
           absPath = absPath.substring(absoluteBase.length() + 1);
```

```
            if (absPath.equals(""))
                absPath = "/";
            canPath = canPath.substring(absoluteBase.length() + 1);
            if (canPath.equals(""))
                canPath = "/";
            if (!canPath.equals(absPath))
                return null;
        }

    } else {
        return null;
    }
    return file;

}
```

该方法不能防止 /a.jsp/ 这样的 URL，但是 DefaultServlet 随后有检查末尾的 /，导致 / 不能被使用。

而新版本的修复方式也是对代码进行了小范围的重构，将上面的检查方法拆到了名为 validate 的方法中，并重写了 VirtualDirContext 中大量的方法，调用 validate 来访问的文件进行检查。

利用

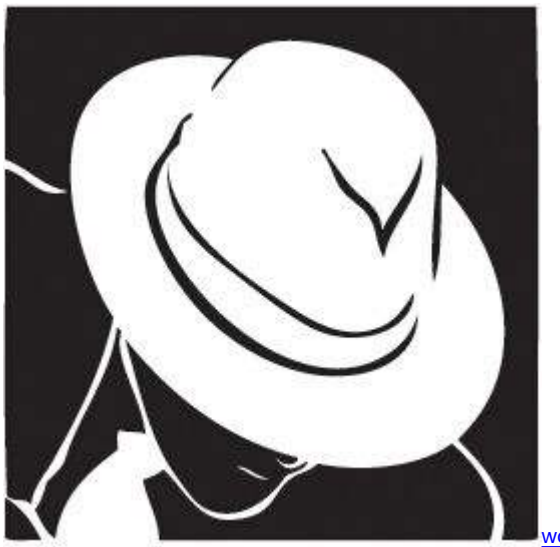
与 CVE-2017-12615 类似，达到查看 Jsp 文件源代码的效果。

修复

升级

上一篇：Python沙箱逃逸的n种姿势 下一篇：请问能burpsuite的插件中直...

1. 2 条回复



wooyun 2017-09-21 07:11:16

大王叫我来巡山.....

0 回复Ta

shaoibngmm 2017-09-24 05:50:02

这里也有个同学再分析该漏洞
https://zhuanlan.zhihu.com/p/29620375

0 回复Ta

登录 后跟帖

先知社区

现在登录

热门节点

技术文章

社区小黑板

目录

RSS 关于社区 友情链接 社区小黑板