



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

CORSO DI PENETRATION TESTING  
AND ETHICAL HACKING

# Metodologie utilizzate

## *Momentum: 1*

STUDENTE

**Vincenzo Emanuele Martone**

DOCENTE

**Prof. Arcangelo Castiglione**

Anno Accademico 2022-2023

<b>Indice</b>	<b>i</b>
<b>Elenco delle figure</b>	<b>ii</b>
<b>1 Pre-Exploitation</b>	<b>1</b>
1.1 Target Scoping . . . . .	1
1.2 Information Gathering . . . . .	1
1.3 Target Discovery . . . . .	2
1.3.1 Informazioni preliminari . . . . .	2
1.3.2 Scansione con <i>nmap</i> . . . . .	3
1.3.3 Determinazione <i>MAC Address</i> con <i>arping</i> . . . . .	3
1.3.4 Scansione con <i>arp-scan</i> . . . . .	4
1.3.5 OS Fingerprinting attivo con <i>nmap</i> . . . . .	4
1.3.6 OS Fingerprinting passivo con <i>p0f</i> . . . . .	5
1.4 Target Enumeration . . . . .	6

---

## Elenco delle figure

---

1.1	Output del comando <i>ifconfig</i> . . . . .	2
1.2	Output del comando <i>nmap</i> ( <i>ping scan</i> ) . . . . .	3
1.3	Output del comando <i>arping</i> . . . . .	3
1.4	Output del comando <i>arp-scan</i> . . . . .	4
1.5	Output del comando <i>nmap</i> ( <i>SO Fingerprinting</i> ) . . . . .	5
1.6	Output del comando <i>p0f</i> . . . . .	5

### 1.1 Target Scoping

Il processo di *Penetration Testing*, come evidenziato nella fase introduttiva, ha uno scopo puramente didattico, per cui non è prevista una fase di accordo tra le parti coinvolte in quanto l'asset da analizzare è una macchina virtuale vulnerabile *by design*. Non vi è, infatti, un cliente dal quale raccogliere requisiti e con il quale definire obiettivi di business e modelli dei costi. Il processo verrà svolto senza particolari vincoli formali relativi all'asset.

### 1.2 Information Gathering

La caratterizzazione dell'asset da analizzare può generalmente avvenire mediante molteplici *tool* e coinvolgere diversi aspetti dell'asset stesso. Dal momento che si sta trattando una macchina virtuale vulnerabile *by-design* contestualizzata in un'attività progettuale avente uno scopo didattico non risulta utile ricorrere a particolari tecniche *OSINT* (*Open Source INTelligence*), né a tecniche volte all'ottenimento di informazioni di routing e record DNS. Sono state, tuttavia, consultate le informazioni di base dell'asset disponibili sulla piattaforma *VulnHub* che mette a disposizione la macchina virtuale. Le informazioni fornite sono le seguenti:

- **Nome della macchina:** *Momentum: 1*;
- **Sistema Operativo:** *Linux*;

- **DHCP Server:** abilitato;
- **Indirizzo IP:** assegnato in automatico.

Non risultano, dunque, note le informazioni relative all'indirizzo *IP* della macchina né le credenziali di accesso alla stessa.

## 1.3 Target Discovery

L'individuazione della macchina *Momentum: 1* all'interno della rete è stata effettuata, in accordo con quanto descritto nel capitolo introduttivo, utilizzando una macchina virtuale con *Kali Linux* connessa alla medesima rete.

### 1.3.1 Informazioni preliminari

Prima di procedere alla trattazione delle metodologie di individuazione della macchina target è necessario considerare alcuni aspetti dell'architettura di rete virtuale nell'ambito della quale è stata svolta l'attività di *Penetration Testing*. La gestione del *NAT* e del *DHCP* da parte di *VirtualBox* fa sì che risultino connessi alla rete degli host aventi *IP* 10.0.2.1, 10.0.2.2 e 10.0.2.3. Alla rete risulterà altresì connessa la macchina virtuale con *Kali Linux* della quale è stato rilevato l'indirizzo *IP* (10.0.2.15) mediante il comando *ifconfig* il cui output è illustrato nella figura 1.1.

```
(kali@kali)-[~]
$ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::7525:725e:c670:3d88 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:c7:e1:36 txqueuelen 1000 (Ethernet)
    RX packets 3719 bytes 238023 (232.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6349 bytes 385837 (376.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 2856 bytes 159140 (155.4 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 2856 bytes 159140 (155.4 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

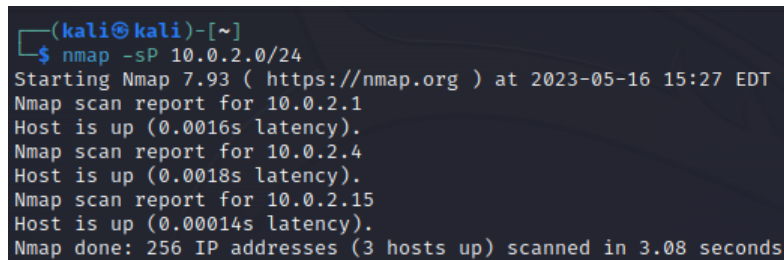
Figura 1.1: Output del comando *ifconfig*

### 1.3.2 Scansione con *nmap*

Come evidenziato in precedenza, l'indirizzo IP di *Momentum: 1* non è noto in quanto viene assegnato mediante il servizio di *DHCP* di *VirtualBox*. Per tale ragione è stata eseguita una scansione volta alla rilevazione della macchina target sulla rete *10.0.2.0/24* mediante il comando:

```
$ nmap -sP 10.0.2.0/24
```

Questo comando effettua un *ping scan* di tutti gli host della rete specificata in input [1], nell'ambito della quale vengono rilevati 3 host attivi, come mostrato nella figura 1.2. Dal momento che, come specificato in precedenza, l'indirizzo *10.0.2.1* fa riferimento ad un host di *VirtualBox* e l'indirizzo *10.0.2.15* è relativo alla macchina virtuale con *Kali*, risulta immediato stabilire che l'indirizzo IP di *Momentum: 1* è *10.0.2.4*.



```
(kali㉿kali)-[~]  
$ nmap -sP 10.0.2.0/24  
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-16 15:27 EDT  
Nmap scan report for 10.0.2.1  
Host is up (0.0016s latency).  
Nmap scan report for 10.0.2.4  
Host is up (0.0018s latency).  
Nmap scan report for 10.0.2.15  
Host is up (0.00014s latency).  
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.08 seconds
```

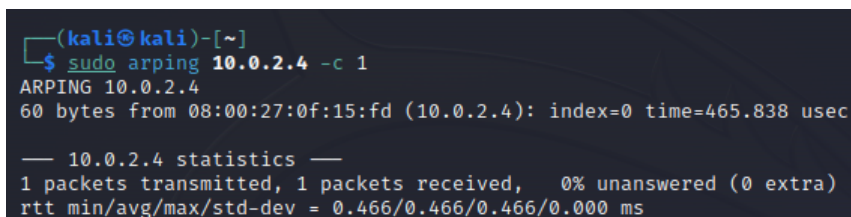
Figura 1.2: Output del comando *nmap* (*ping scan*)

### 1.3.3 Determinazione MAC Address con *arping*

A partire dall'indirizzo IP di *Momentum: 1*, risulta possibile arricchire la conoscenza della macchina target individuandone il MAC Address. Ciò è possibile mediante il comando:

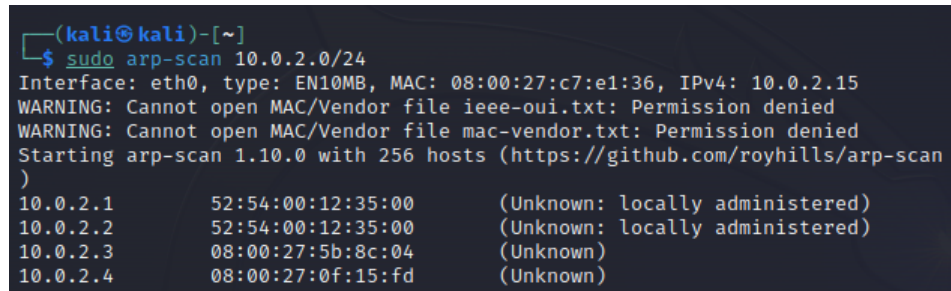
```
$ sudo arping 10.0.2.4 -c 1
```

Tale comando invia un'ARP Request al dispositivo indicato in input [2]. Mediante l'output, illustrato nella figura 1.3, si stabilisce che il MAC Address di *Momentum: 1* è *08:00:27:0f:15:fd*.



```
(kali㉿kali)-[~]  
$ sudo arping 10.0.2.4 -c 1  
ARPING 10.0.2.4  
60 bytes from 08:00:27:0f:15:fd (10.0.2.4): index=0 time=465.838 usec  
  
— 10.0.2.4 statistics —  
1 packets transmitted, 1 packets received, 0% unanswered (0 extra)  
rtt min/avg/max/std-dev = 0.466/0.466/0.466/0.000 ms
```

Figura 1.3: Output del comando *arping*



```
(kali㉿kali)-[~]
└─$ sudo arp-scan 10.0.2.0/24
Interface: eth0, type: EN10MB, MAC: 08:00:27:c7:e1:36, 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      08:00:27:5b:8c:04      (Unknown)
10.0.2.4      08:00:27:0f:15:fd      (Unknown)
```

Figura 1.4: Output del comando *arp-scan*

### 1.3.4 Scansione con *arp-scan*

Durante il processo di *Penetration Testing* è stato utilizzato un approccio volto all’ottenimento delle medesime informazioni mediante molteplici tool al fine di confrontarne i risultati per massimizzare il quantitativo di informazioni ottenute nell’ambito di una determinata fase. A tale scopo ci si è serviti del tool *arp-scan* per effettuare una scansione sulla rete *10.0.2.0/24* mediante il comando:

```
$ sudo arp-scan 10.0.2.0/24
```

L’output del comando, illustrato nella figura 1.4, mostra che la scansione ha rilevato 4 host, per ciascuno dei quali ha fornito il relativo indirizzo *IP* ed il relativo *MAC Address*. La precedente scansione (effettuata con il tool *nmap*) non ha rilevato gli host *192.168.1.2* e *192.168.1.3*, tuttavia, come evidenziato in precedenza, questi host sono gestiti da *VirtualBox* per cui non hanno rilevanza nel processo di *Penetration Testing* effettuato. L’host avente indirizzo *IP 10.0.2.4* e *MAC Address 08:00:27:0f:15:fd* è relativo alla macchina target.

### 1.3.5 OS Fingerprinting attivo con *nmap*

Al fine di arricchire la conoscenza relativa alla macchina target è stata effettuata un’operazione di *OS detection* mediante il comando:

```
$ sudo nmap -O 10.0.2.4
```

L’output ottenuto (figura 1.5) fornisce diverse informazioni relative al sistema operativo in esecuzione sulla macchina target. È possibile stabilire che si tratta di un sistema *Linux* presumibilmente ad una versione 4.15 o 5.6 (quando *nmap* non riesce a stabilirlo con precisione mostra tutte i possibili match [3]); il tool fornisce, infine, la *CPE*<sup>1</sup> di riferimento (*cpe:/o:linux:linux\_kernel:4 cpe:/o:linux\_kernel:5*). Sono, altresì, presenti informazioni relative

<sup>1</sup>CPE (*Common Platform Enumeration*) è un sistema di naming strutturato per sistemi operativi, software e packages [4]

```
(kali㉿kali)-[~]
$ sudo nmap -O 10.0.2.4
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-16 15:38 EDT
Nmap scan report for 10.0.2.4
Host is up (0.00051s latency).
Not shown: 998 closed tcp ports (reset)
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:0F:15:FD (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 - 5.6
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 2.88 seconds
```

**Figura 1.5:** Output del comando *nmap* (SO Fingerprinting)

alle porte aperte ed ai servizi attivi sulla macchina target, che nell’ambito della fase di *Target Discovery*, sono state sfruttate unicamente per effettuare OS Fingerprinting passivo; sulla macchina target risultano aperte la porta 80 (servizio *HTTP*) e la porta 22 (servizio *SSH*).

### 1.3.6 OS Fingerprinting passivo con *p0f*

La fase di OS Fingerprinting attivo non ha portato all’individuazione dell’esatta versione del sistema operativo in esecuzione sulla macchina target: è stata, dunque, svolta una fase di OS Fingerprinting passivo finalizzata all’ottenimento di ulteriori informazioni in merito, utilizzando il tool *p0f*. La tecnica di fingerprinting passivo adottata consiste nel porsi in ascolto su una specifica interfaccia di rete ed ispezionare i pacchetti *TCP/IP* intercettati al fine di individuare le informazioni desiderate. Tale operazione è stata svolta mediante il comando (figura 1.6):

```
$ sudo p0f -i eth0
```

```
(kali㉿kali)-[~]
$ nmap -sP 10.0.2.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-05-16 15:27 EDT
Nmap scan report for 10.0.2.1
Host is up (0.0016s latency).
Nmap scan report for 10.0.2.4
Host is up (0.0018s latency).
Nmap scan report for 10.0.2.15
Host is up (0.00014s latency).
Nmap done: 256 IP addresses (3 hosts up) scanned in 3.08 seconds
```

**Figura 1.6:** Output del comando *p0f*

Il passo successivo consiste nel fare in modo che la macchina target invii pacchetti sull’interfaccia di rete *eth0*; a tale scopo sono state effettuate delle richieste ai servizi esposti dalla



macchina, ossia *HTTP* e *SSH*, mediante i comandi:

```
$ curl -X GET http://10.0.2.4/  
$ ssh user@10.0.2.4
```

A tali richieste corrispondono delle risposte, intercettate dal tool *p0f* e dalle quali sono state ottenute le informazioni riportate nelle figure

## 1.4 Target Enumeration

---

## Bibliografia

---

- [1] *nmap(1) - Linux man page*, Maggio 2023. (Citato a pagina 3)
- [2] *arping(8) - Linux man page*, Maggio 2023. (Citato a pagina 3)
- [3] "Nmap manual." (Citato a pagina 4)
- [4] "Cpe dictionary - nist." (Citato a pagina 4)