Relazione esercizio di penetration test con metodo Black box

A cura di Canole Iris

Data: 02/11/2025

Corso: Cybersecurty specialist

Introduzione

L'obiettivo dell'esercizio era ottenere i permessi di `root` sulla virtual machine (VM) target BsidesVancouver2018 (indirizzo IP 192.168.50.8), partendo da zero. L'accesso iniziale, è stato ottenuto tramite un errore di configurazione dell'autenticazione SSH. L'escalation dei privilegi è stata ottenuta sfruttando una configurazione errata del file `sudoers`.

- Obiettivo ➜ ottenere la shell di `root`.
- Vettore di Accesso Iniziale ➜ SSH (utente `anne`, password `princess`).
- Vettore di Privilege Escalation ➜ Configurazione errata di SUDO (`(ALL : ALL) ALL`).
- Stato Finale ➜ Compromissione Completa (Shell di `root` ottenuta, *flag* finale trovata).

Azioni preliminari

Su Virtualbox, ho fatto partire la macchina Kali, mantenendo tutte le altre macchine spente.
Sul terminale della Kali, ho fatto partire il comando `nmap -sn 192.168.50.0/24` (con notazione CIDR), usando la subnet-mask come argomento, per verificare gli host attivi.

Facendo ciò, come risultato della scansione, Kali si presenta l'unico host attivo.
Dopo aver verificato ciò, ritorno sulla home della VirtualBox e accendo la macchina BsidesVancouver2018.

Ritornando sulla Kali, rilancio lo stesso comando di prima, e controllo il risultato della scansione: ora vedo due host attivi, un host con un indirizzo IP non riconducibile alle altre macchine installate della VirtualBox.

Sicuramente questo sarà l'indirizzo IP della nuova macchina importata su VirtualBox.

Provo un ping veloce con `ping 192.168.50.8` per verificare la connessione, ed effettivamente la connessione è riuscita.

Ricognizione ed Enumerazione (Fase di Scansione)

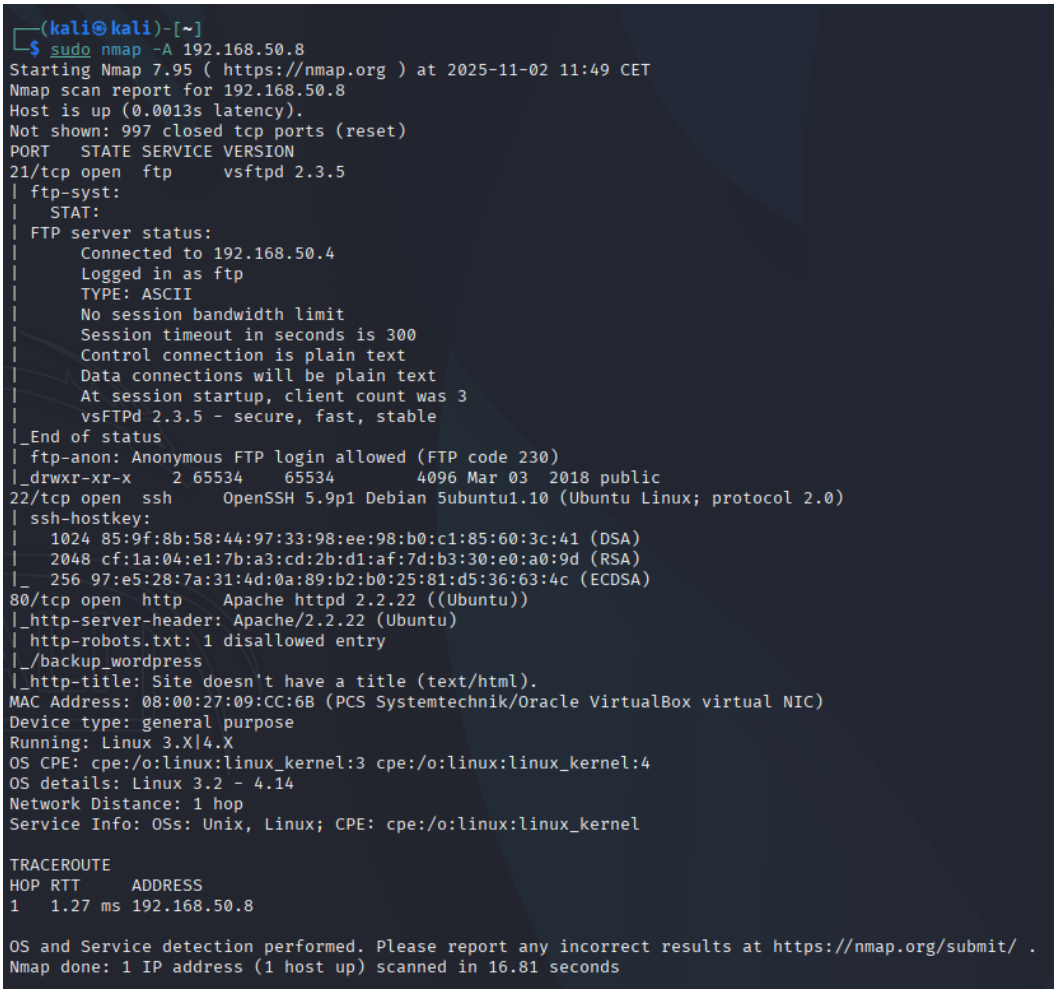
Per verificare i servizi attivi sulla BsidesVancouver2018, ho lanciato il comando `nmap -O 192.168.50.8`. La scansione ha identificato i seguenti servizi aperti:

PORT	STATE	SERVICE
21/tcp	OPEN	ftp
22/tcp	OPEN	ssh
80/tcp	OPEN	http

Cosa significa? Sappiamo che abbiamo 3 porte aperte (21,22,80) e conosciamo i rispettivi servizi che girano su quelle porte.

Abbiamo informazioni anche sul tipo di sistema operativo installato sulla macchina, Linux.

Questo non mi basta, provo a fare una scansione più aggressiva, per cercare di raccogliere più informazioni possibili. Eseguo da teminale il comando `sudo nmap -A 192.168.50.8`.
Quello che ricavo è prezioso per il PT che sto eseguendo, come mostra la seguente figura:



Facciamo un'analisi dell'output di Nmap:

- Sistema Operativo (OS) ➜ Linux, Debian Subuntu1.10 (probabilmente Ubuntu 18.04 LTS o simile) con kernel 3.x - 4.x
- Porta 21 (FTP) ➜ Servizio `vsftpd 2.3.5`
  - CRITICO: il servizio è configurato per consentire l'accesso Anonymous FTP (`ftp-anon: Anonymous FTP login allowed`).
- Porta 22 (SSH) ➜ Servizio `OpenSSH 5.9p1` (su Debian).
- Porta 80 (HTTP) ➜ Servizio `Apache httpd 2.2.22` (Su Ubuntu).
  - INDICAZIONE: il sito web ha una directory chiamata `_backup_wordpress`.

Prove di penetrazione

Ora che abbiamo questo tipo di informazioni, possiamo sfruttare le vulnerabilità ottenute con le scansioni.

Il vettore di attacco più probabile e semplice per l'accesso è il servizio FTP sulla porta 21.

> **Versione vsftpd 2.3.5**
> La versione vsftpd 2.3.5, è famosa nel mondo del penetration testing perché contiene una vulnerabilità di tipo backdoor che consente, in determinate configurazioni, di ottenere immediatamente una shell di comando remota (Remote Command Execution) come utente con privilegi elevati.

Provo a collegarmi via FTP:

```
ftp 192.168.50.8
User: anonymous

...
connected
```

Una volta instaurata la connessione, comincio a navigare nella shell e cercare:

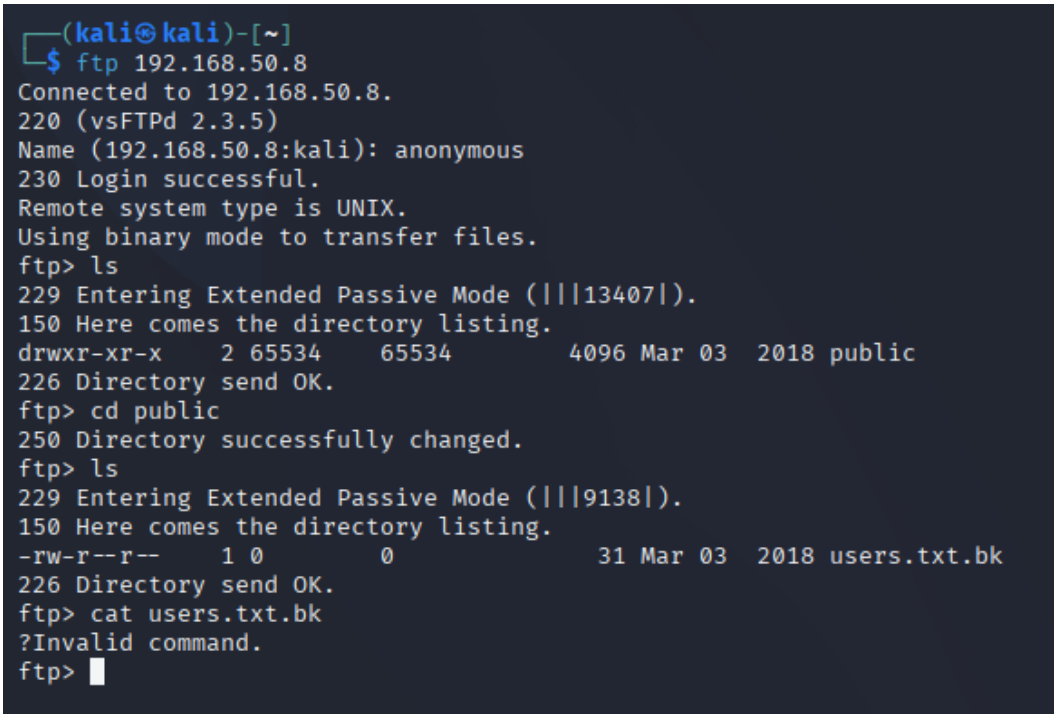
- file di configurazione (.conf, .php)
- file di backup (.zip, .tgz)
- file di testo (.txt, readme.txt)

che possono contenere credenziali o informazioni nascoste.

Facendo un semplice `ls` trovo:

- una directory dal nome `public`
- con i permessi `rwxr-xr-x` (read, write, execute)
- `65534 65534` ➜ sono gli ID utente e di gruppo del proprietario

Mi sposto, a questo punto, dentro la direcory `public` con `cd` e rifaccio un `ls`, per vedere i contenuti della directory:



Trovo il file chiamato `users.txt.bk`, che a giudicare dall'estensione, sembrerebbe un file di backup.

Oggetto molto utile ed interessante per il mio PT, motivo per il quale con il comando `get users.txt.bk` lo scarico all'interno della mia Kali, non potendo fare un semplice `cat` da terminale.

Il file scaricato, contiene una lista di utenti:

- `abatchy`
- `john`
- `mai`
- `anne`
- `doomguy`

Ho così identificato alcuni potenziali validi utenti per la BsidesVancouver2018

Tentativi falliti

Ora l'obiettivo è associare una password a uno di questi utenti, per accedere alla macchina tramite shell.
Bisogna avere a disposizione un tool che permetta di fare dei tentativi di login con più password (un brute-force). Lo strumento adatto a fare questo tipo di lavoro è *hydra*.

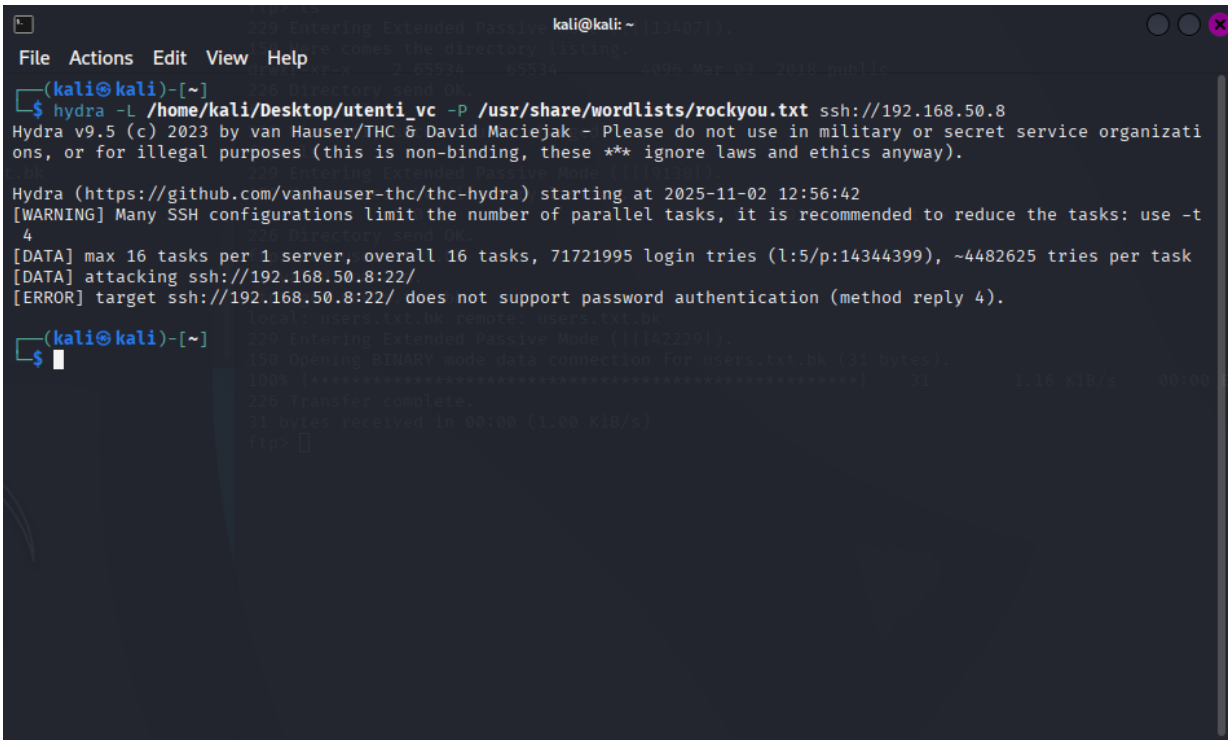
Cosa fare?

1. Creo un file con l'elenco degli utenti precedentemente trovato

2. Uso una lista di password comuni, una lista come ad esempio `/usr/share/wordlists/rockyou.txt`
3. Lancio il comando hydra così: `hydra -L /home/kali/Desktop/users.txt -P /usr/share/wordlists/rockyou.txt ssh://192.168.50.8`

Se Hydra ha successo, restituirà la combinazione utente/password vincente

In questo caso, purtroppo hydra non ha dato esito positivo, ma l'output mi dice grandi cose su come funziona l'autenticazione per gli utenti:



Hydra mi sta dicendo che il target `ssh://192.168.50.8` (che è la BsidesVancouver2018) non supporta password per l'autenticazione. Questo mi fa pensare che se l'autenticazione non avviene con le password, avverà probabilemnte con delle chiavi SSH.

Via di Successo

A questo punto provo a fare una scansione nome per nome, un utente alla volta:

- per `john`, `mai`, `abatchy` e `doomguy` l'output era sempre lo stesso ➜ confermo che il metodo di autenticazione di questi utenti non avviene tramite password.
- per `anne`, la scansione funziona, questo vuol dire che `anne` è **l'unico** tra tutti gli utenti ad utilizzare un metodo di autenticazione basato sulla password.

Hydra mi trova in pochissimo tempo la combinazione nome utente-password, come mostra la seguente figura:

```
┌──(kali㉿kali)-[~]
└─$ hydra -l anne -P /usr/share/wordlists/rockyou.txt ssh://192.168.50.8
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizati
ons, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2025-11-02 15:25:43
[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://192.168.50.8:22/
[22][ssh] host: 192.168.50.8   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 complete 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 2025-11-02 15:25:57
```

Sono a un'ottimo punto! Adesso posso eseguire il login con la connessione SSH.
Lancio il comando `ssh anne@192.168.50.8`, immetto `anne` come username e `princess` come password e mi autentico.

A questo punto si aprirà la shell di `anne`: ora sono dentro e può iniziare la fase di escalation dei permessi `root`.

Fase di escalation dei permessi di `root`

A questo punto, mi avvalgo del tool `linpeas` già preinstallato sulla mia Kali.
Lancio il programma e con un serverino in python, faccio partire in ascolto la porta `8000`.

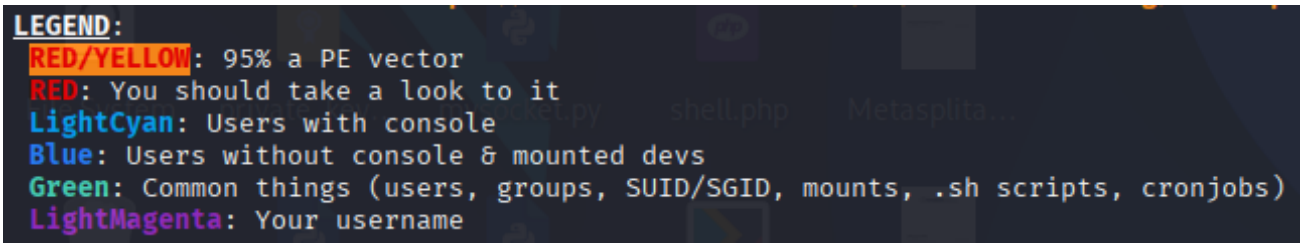
```
kali@kali: /usr/share/peass/linpeas
File  Actions  Edit  View  Help
$ linpeas

> peass ~ Privilege Escalation Awesome Scripts SUITE

/usr/share/peass/linpeas
├── linpeas_darwin_amd64
├── linpeas_darwin_arm64
├── linpeas_fat.sh
├── linpeas_linux_386
├── linpeas_linux_amd64
├── linpeas_linux_arm
├── linpeas_linux_arm64
├── linpeas.sh
└── linpeas_small.sh
┌──(kali㉿kali)-[/usr/share/peass/linpeas]
└─$ python3 -m http.server 8000
Serving HTTP on 0.0.0.0 port 8000 (http://0.0.0.0:8000/) ...
192.168.50.8 - - [02/Nov/2025 15:43:08] "GET /linpeas.sh HTTP/1.1" 200 -
```

Trasferisco ora sulla macchina della vittima lo script .sh di `linpeas` e lo rendo un eseguibile da poter lanciare.

Una volta lanciato, `linpeas` mostra una legenda, come quella nella figura sottostante, nella quale mi elenca cosa andare a vedere nello specifico:



Essendo che l'output di `linpeas`, una volta lanciato l'eseguibile dalla macchina della vittima, è molto lungo, mi concentro sui risultati evidenziati in rosso o giallo che indicano potenziali vettori di attacco.

Noto diverse informazioni utili come, per esempio, la sezione *Analizing Wordpress files*: file `/var/www/backup_wordpress/wp-config.php` cui contenuto:

- `define ('DB_USER', john@localhost)`
- `define ('DB_PASSWORD', thiscannotbeit)`

Credenziali del database MySQL/MariaDB.
Questo espone i dati del database e fornisce una potenziale nuova password per l'utente `john`.

Dò un occhiata anche ai log si sistema (`/var/log/bootstrap/dpkg.log`) che contengono riferimenti a `passwd` e al pacchetto `base-passwd`, indicando attività relative a password e potenziali file di log non epurati che potrebbero contenere dati sensibili.

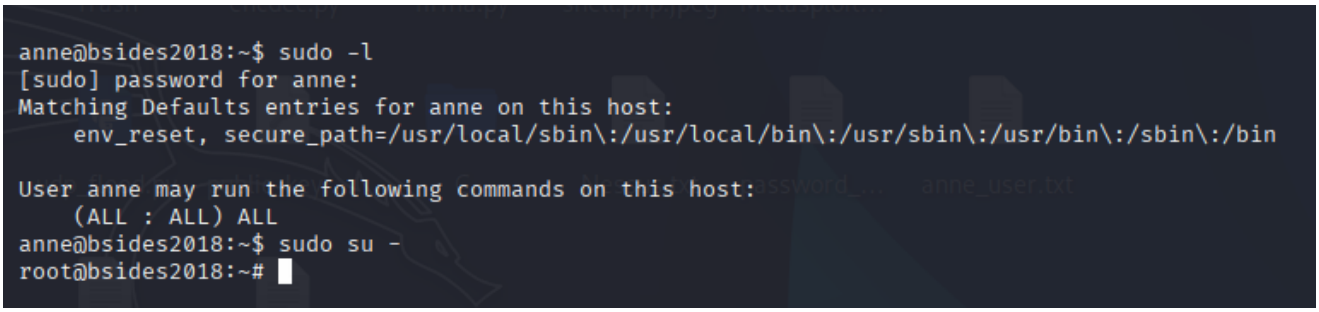
Usando il comando `sudo -l` sull'utente `anne` , per vedere cosa `anne` può eseguire come `root` e trovo come output:
`user anne may run the following command on this host: (ALL : ALL) ALL`

Ho trovato il vettore privilage escalation.
Significa che l'utente `anne` è configurato per esguire qualsiasi comando `(ALL)` come qualsiasi utente `(ALL)`, inclusi `root` e il sistema non ha specificato `NOPASSWD`, il che significa che l'utente deve fornire la sua password.

Poiché la password di `anne` è `princess` la soluzione è semplice:

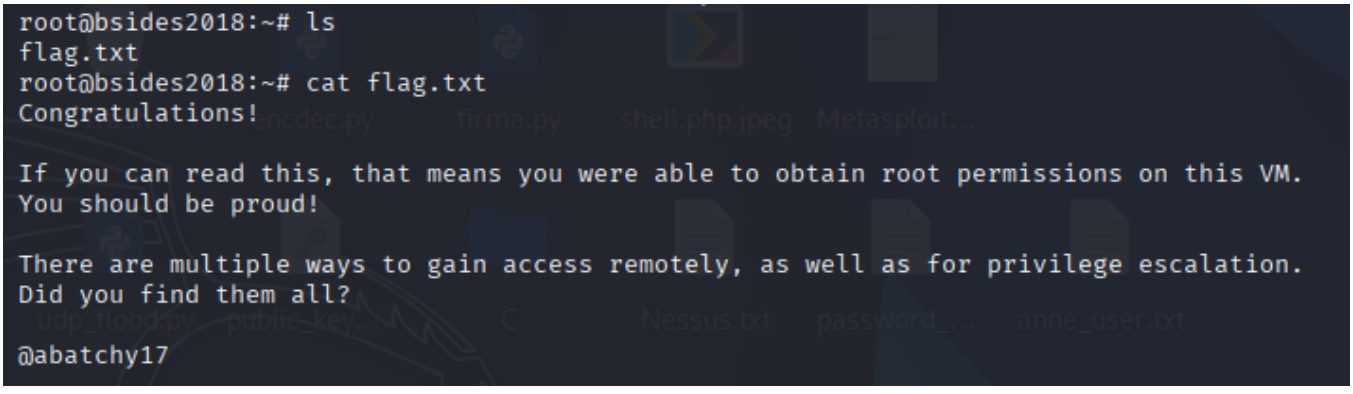
1. Eseguire il comando di escalation sulla shell di `anne`: `sudo su -`
2. Inserire la password `princess`
3. Se la configurazione è corretta, questo fornirà immediatamente una shell con i massimi privilegi: `root@bsides2018:~#`

L'escalation dei privilegi era dovuta a un errore di configurazione nel file `sudoers`



Ultimo passaggio: trovare la flag finale

Sul terminale con i privilegi di root, faccio un `ls` e vedo uscire fuori un `file.txt` ➜ `flag.txt`.
Faccio un `cat` sul file e quello che vedo è:



L'esercizio è completato! Sono riuscita ad arrivare ad ottenere i privilegi di root, come richiesto nella consegna dell'esercizio.

Conclusioni

L'obiettivo di ottenere l'accesso amministrativo (shell di `root`) sulla macchina target, è stato pienamente raggiunto. La compromisisone è avvenuta in fasi distinte sfruttando lacune critiche nella configurazione di sistema.

Questa sfida dimostra che l'intera catena di sicurezza del sistema è fallita a causa di errori banali di configurazione:

- **Massima Priorità** ➜ rimuovere l'autorizzazione `(ALL : ALL) ALL` per l'utente `anne` e disabilitare l'autenticazione tramite password per tutti gli utenti, rendendo obbligatorie le chiavi SSH.
- Rischio Dati ➜ rimuovere immediatamente le password nel database `john@localhost: thiscannotbeit` che era esposta in chiaro nel file di backup di WordPress.

Ho recuperato, infine, il flag finale nella directory `/root/`, confermando che ora il sistema è completamente sotto il mio controllo.