S6 - L4: EXTRA
Hacking VM BlackBox

FASE 1 — Setup dell’ambiente di test e network discovery

Step 1.1 — Verifica della configurazione di rete della macchina attaccante

Cosa ho fatto:
Ho verificato che la macchina Kali Linux fosse correttamente configurata sulla rete di laboratorio e avesse ricevuto un indirizzo IP valido.

Comando:

```
ip a
```

Output atteso:
Presenza di un indirizzo IPv4 assegnato all’interfaccia di rete (subnet Host-Only).

Step 1.2 — Network discovery e individuazione del target

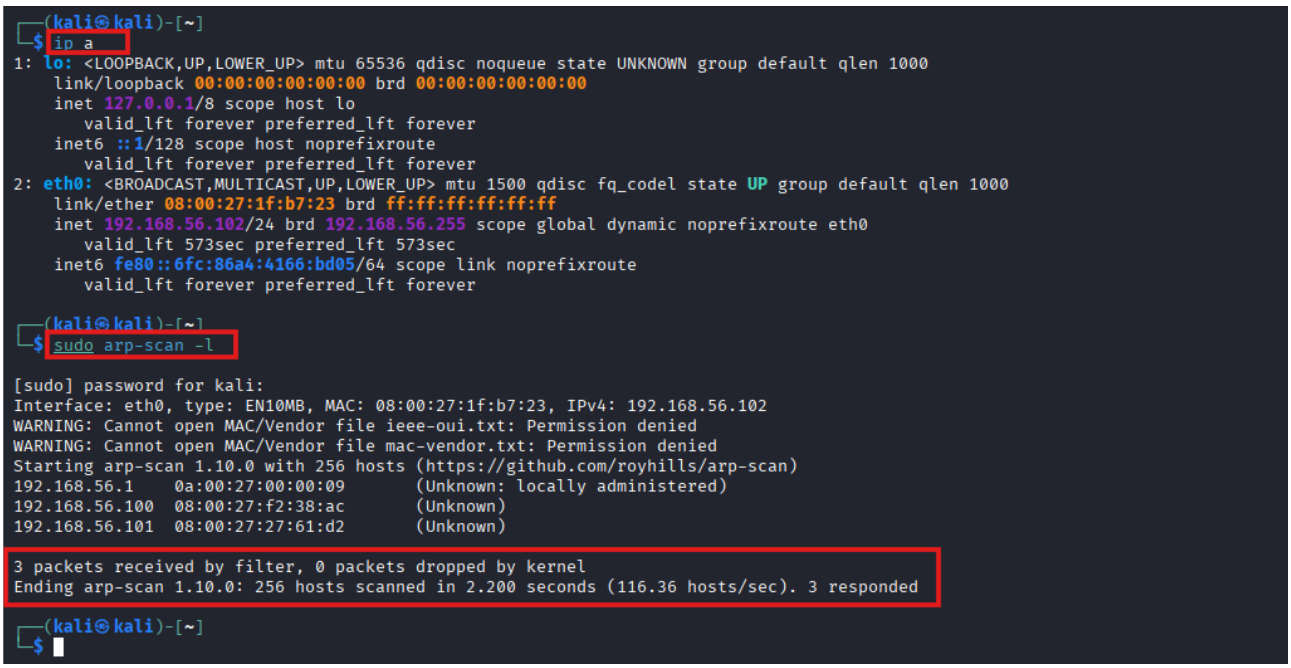
Cosa ho fatto:
Ho eseguito una scansione ARP per individuare gli host attivi nella rete locale e identificare l’indirizzo IP della macchina target.

Comando:

```
sudo arp-scan -l
```

Output atteso:
Elenco degli host attivi nella subnet, inclusa la macchina **BSides Vancouver 2018**, con relativo indirizzo IP.

A seguire relativo screenshot di esempio:



- Output del comando **`ip a`** con IP visibile
- Output del comando **`sudo arp-scan -l`** con l'IP del target evidenziato

Step 2.1 — Scansione completa delle porte TCP

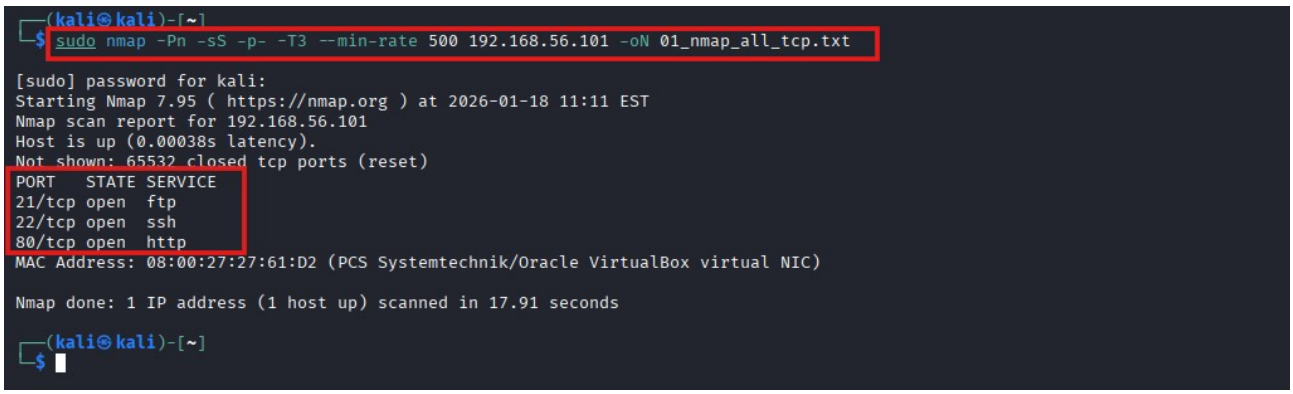
Cosa ho fatto:
Ho eseguito una scansione completa di tutte le porte TCP sulla macchina target per individuare l'intera superficie di attacco esposta.

Comando:

```
sudo nmap -Pn -sS -p- -T3 --min-rate 500 192.168.56.101 -oN 01_nmap_all_tcp.txt
```

Output atteso:
Elenco delle porte TCP in stato *open* sulla macchina target.

A seguire relativo screenshot di esempio:



- **Output Nmap con le porte TCP aperte visibili**

Step 2.2 — Enumerazione dei servizi e delle versioni

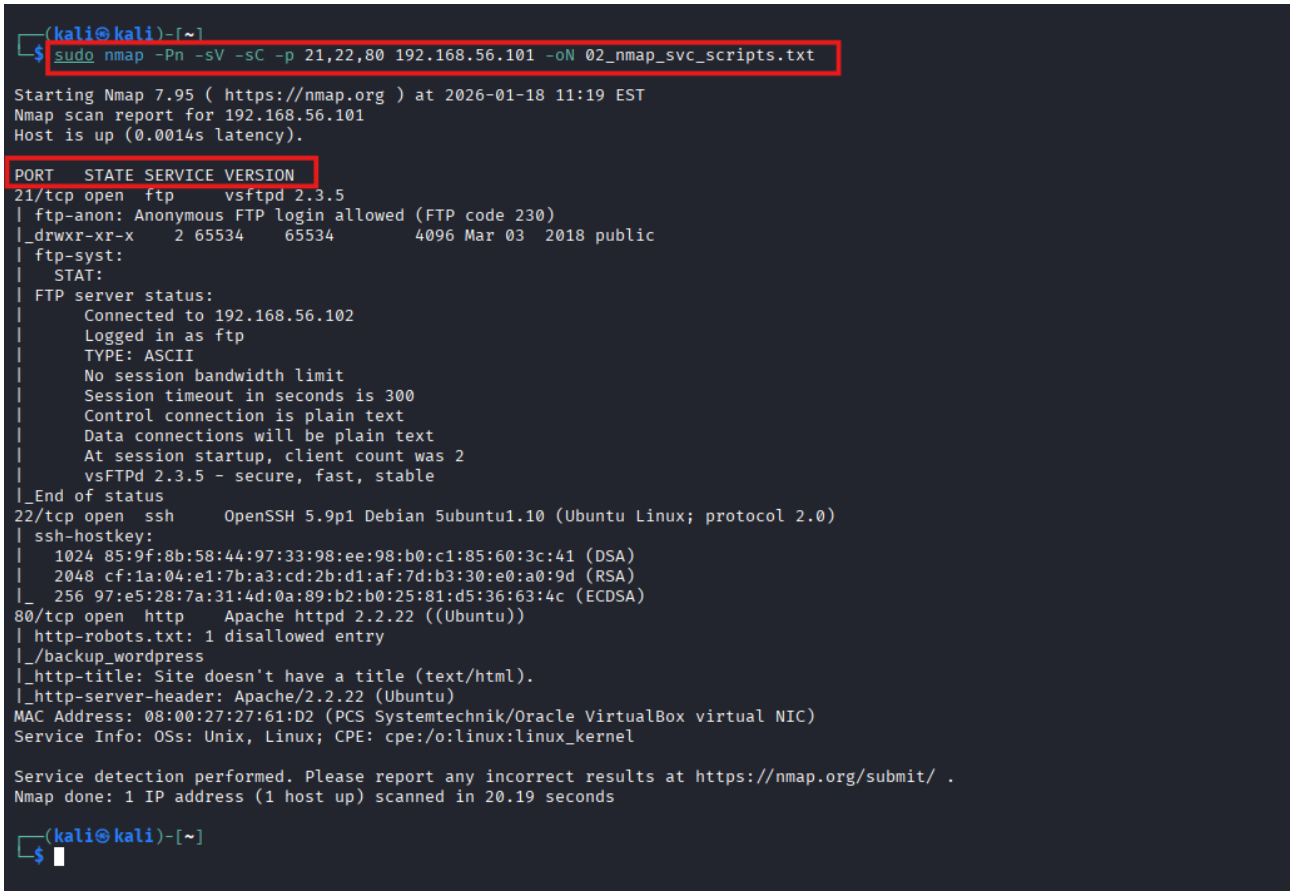
Cosa ho fatto:
Ho analizzato i servizi in ascolto sulle porte individuate, identificandone le versioni e raccogliendo informazioni preliminari tramite gli script di default di Nmap.

Comando:

```
sudo nmap -Pn -sV -sC -p 21,22,80 192.168.56.101 -oN 02_nmap_svc_scripts.txt
```

Output atteso:
Dettagli relativi ai servizi esposti e alle rispettive versioni.

A seguire relativo screenshot di esempio:



- **Output Nmap con servizi e versioni individuati**

Step 2.3 — Scansione mirata delle porte UDP

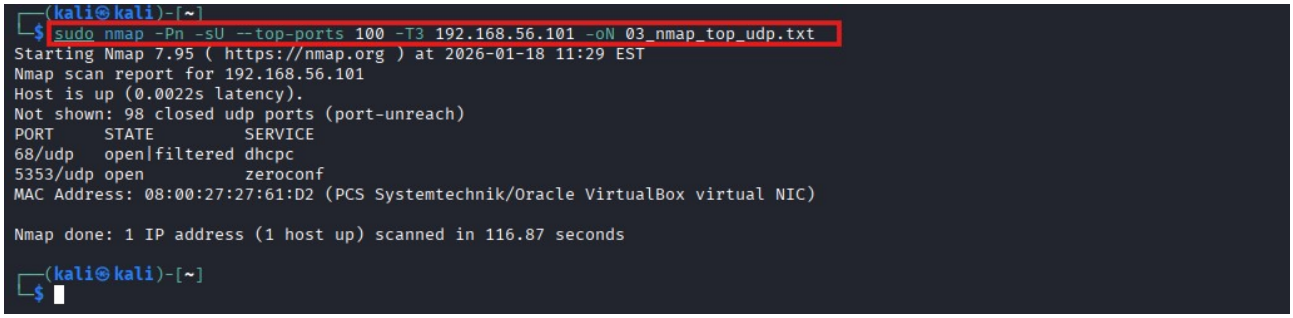
Cosa ho fatto:
Ho effettuato una scansione mirata delle porte UDP più comuni per verificare la presenza di eventuali servizi aggiuntivi non rilevabili tramite scansione TCP.

Comando:

```
sudo nmap -Pn -sU --top-ports 100 -T3
192.168.56.101 -oN 03_nmap_top_udp.txt
```

Output atteso:
Eventuali porte UDP aperte o filtrate sulla macchina target.

A seguire relativo screenshot di esempio:



- **Output della scansione UDP**

FASE 3 — Enumerazione mirata dei servizi (HTTP / FTP)

Step 3.1 — Fingerprinting del servizio HTTP

Cosa ho fatto:
Ho eseguito il fingerprinting del servizio web per identificare tecnologie, CMS o componenti utilizzati dall'applicazione esposta sulla porta 80.

Comando:

`whatweb` `http://192.168.56.101` **(**`macchina target`**)** `(BSides Vancouver 2018)`

Output atteso:
Informazioni su web server, linguaggi, CMS o framework utilizzati.

- **Output del comando `whatweb` con le tecnologie individuate**

Step 3.2 — Analisi del file robots.txt

Cosa ho fatto:
Ho analizzato il file `robots.txt` per individuare directory o risorse potenzialmente sensibili non indicizzate.

Comando:

`curl http://192.168.56.101/robots.txt`

Output atteso:
Elenco di directory o file esclusi dall'indicizzazione (es. backup, admin, staging).

- **Output del file `robots.txt`**

Step 3.3 — Accesso e analisi della directory /backup_wordpress

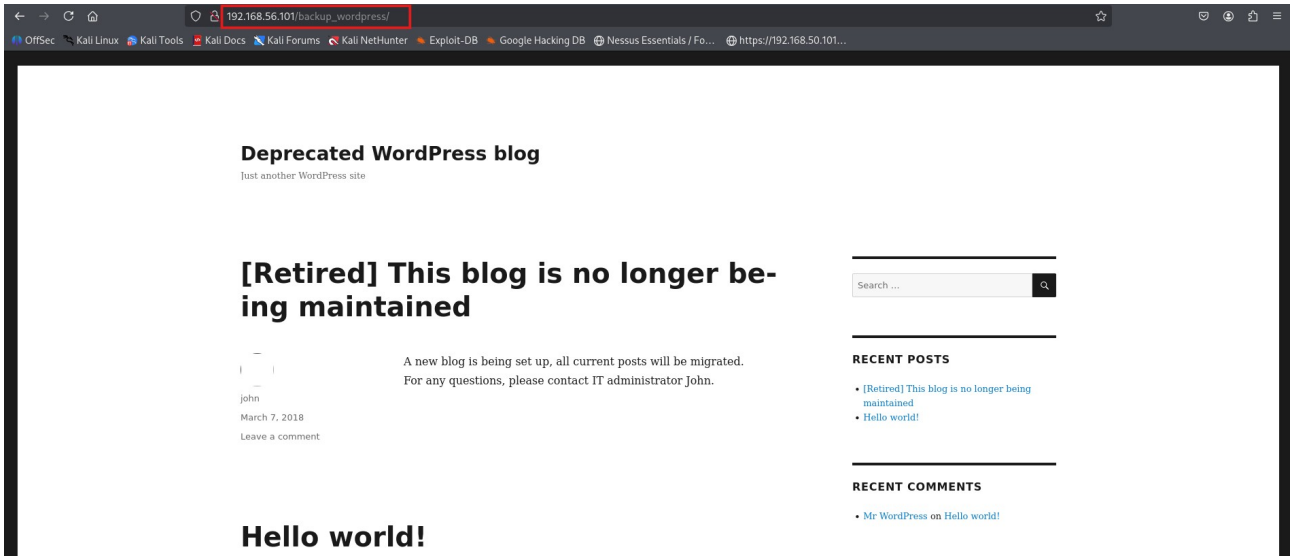
Cosa ho fatto:
Ho visitato manualmente la directory individuata durante la fase di scanning per verificarne il contenuto e la presenza di file di backup o configurazione.

Azione:

- **Apertura browser**
- **URL:**

`http://192.168.56.101/backup_wordpress/`

Output atteso:
Presenza di file di backup WordPress (database dump, file di configurazione, archivi compressi).



- **Contenuto della directory `/backup_wordpress`**

Step 3.4 — Enumerazione del servizio FTP anonimo

Cosa ho fatto:
Ho sfruttato l'accesso FTP anonimo per verificare la presenza di file accessibili senza autenticazione.

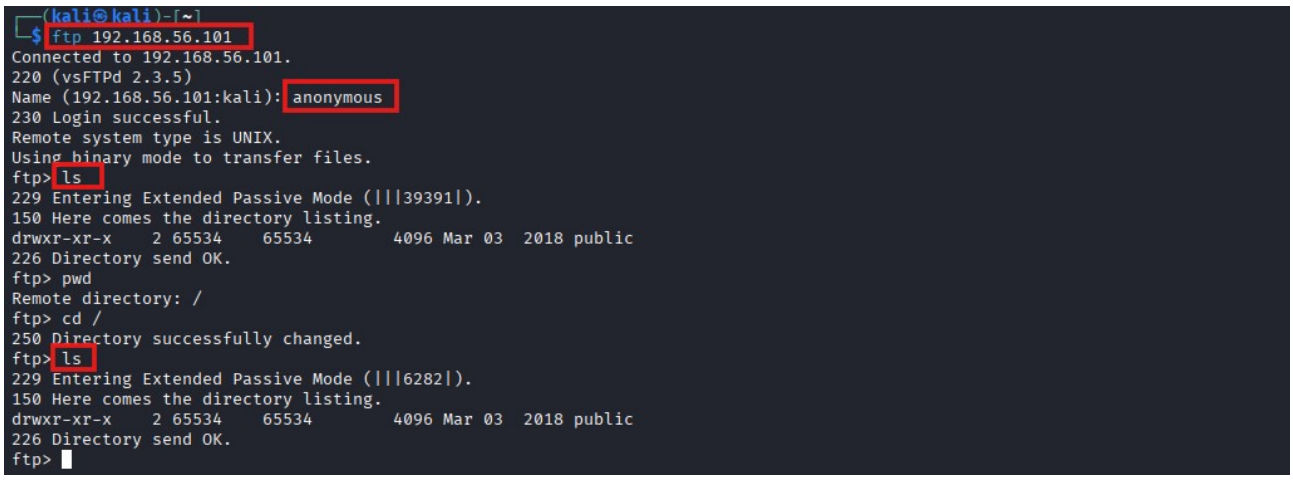
Comando:

```
ftp 192.168.56.101
```

Credenziali utilizzate:

- **Username: `anonymous`**
- **Password: *(vuota o email qualsiasi)***

Output atteso:
Accesso consentito e possibilità di elencare i file presenti sul server FTP.



- **Login FTP anonimo riuscito**
- **Output del comando `ls`**

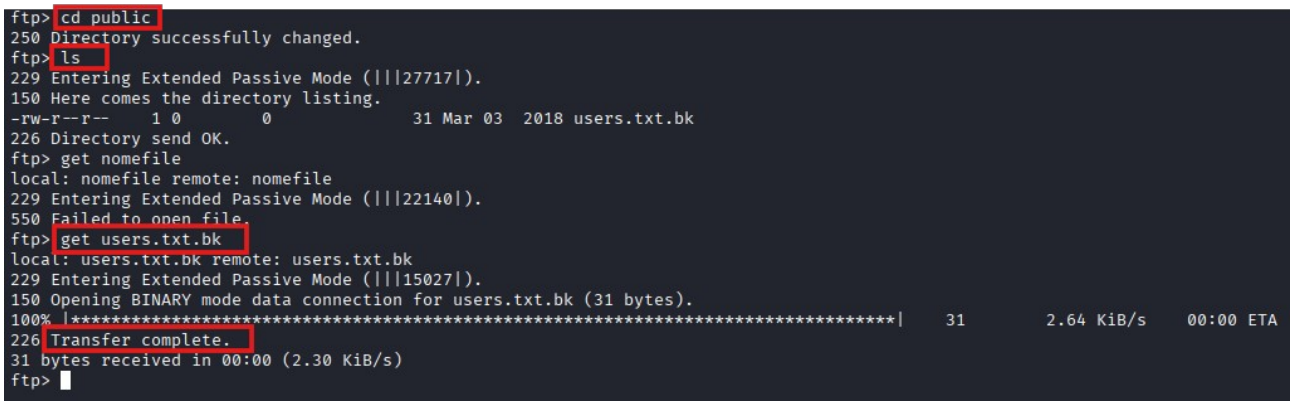
Step 3.5 — Analisi del contenuto della directory FTP `public`

Cosa ho fatto:
Ho esplorato la directory `public`, accessibile tramite FTP anonimo, per individuare eventuali file sensibili che potessero contenere informazioni utili per l'accesso iniziale al sistema.

Comandi eseguiti:

```
cd public
ls
get users.txt.bk
```

Output ottenuto:
All'interno della directory `public` ho individuato il file `users.txt.bk`, presumibilmente un file di backup. Il file è stato scaricato con successo sulla macchina attaccante per procedere all'analisi del contenuto.



- **Output dei comandi `cd public` e `ls`**
- **Download del file `users.txt.bk` completato correttamente**

Successivamente sono uscito con il comando bye.

FASE 4 — Foothold (accesso iniziale)

Step 4.1 — Analisi del file `users.txt.bk`

Cosa ho fatto:
Ho analizzato il file **`users.txt.bk`** scaricato tramite FTP anonimo per verificare la presenza di

informazioni sensibili, come username o credenziali, utili per ottenere un accesso iniziale al sistema.

Comando eseguito:

`cat users.txt.bk`

Output atteso:
Elenco di utenti e/o credenziali in chiaro o parzialmente offuscate.

- **Output del comando `cat users.txt.bk`Analisi del file users.txt.bk**

Step 4.2 — Valutazione delle credenziali individuate

Cosa ho fatto:
Ho analizzato le informazioni contenute nel file per identificare possibili account validi da utilizzare per l'accesso ai servizi esposti, in particolare il servizio SSH precedentemente individuato.

Azione:

- **Identificazione di username plausibili**
- **Verifica di eventuali password associate**

(Se il file contiene solo username, verranno utilizzati per tentativi di autenticazione mirati.)

Step 4.3 — Tentativi di accesso SSH con gli username individuati

Cosa ho fatto:
Ho tentato l'accesso al servizio SSH utilizzando gli username individuati nel file di backup `users.txt.bk`, al fine di verificare la possibilità di ottenere un accesso interattivo diretto al sistema. I tentativi sono stati eseguiti manualmente e in modo mirato, senza ricorrere a tecniche di brute force automatizzato.

Durante questa fase ho riscontrato comportamenti differenti a seconda dell'utente:

- **Per l'utente john, il servizio SSH consente esclusivamente l'autenticazione tramite chiave pubblica, come indicato dal messaggio *"Permission denied (publickey)"*. Questo comportamento è coerente con una configurazione di sicurezza più restrittiva e ha escluso la possibilità di accesso tramite password.**

- **Per l’utente anne, il servizio SSH ha inizialmente richiesto l’inserimento della password, confermando l’esistenza dell’account e la possibilità di autenticazione a password. Tuttavia, le credenziali tentate non sono risultate valide e l’accesso è stato negato dopo alcuni tentativi.**

L’esito complessivo di questa fase ha evidenziato che, **pur in presenza di account validi, non era possibile ottenere un accesso SSH diretto senza il recupero di ulteriori informazioni sensibili (password o chiavi).**

Step 4.4 — Analisi delle limitazioni di accesso e cambio di strategia

Cosa ho fatto:
Preso atto dell’impossibilità di ottenere un accesso SSH diretto utilizzando esclusivamente gli username individuati, **ho analizzato i messaggi di errore restituiti dal servizio per comprendere le modalità di autenticazione consentite e adattare di conseguenza la strategia di attacco.**

I messaggi restituiti dal servizio SSH hanno indicato che:

- **alcuni utenti accettano esclusivamente autenticazione tramite chiave pubblica;**
- **per altri utenti l’autenticazione a password è abilitata, ma richiede credenziali corrette non ancora note.**

Sulla base di queste evidenze, ho interrotto ulteriori tentativi di accesso diretto per evitare comportamenti anomali o non conformi alla metodologia dell’esercizio, e ho proseguito l’attività concentrandomi sull’analisi dell’installazione WordPress dismessa presente nella directory `/backup_wordpress`, con l’obiettivo di recuperare credenziali applicative o informazioni riutilizzabili per l’accesso al sistema.

CAMBIO DI STRATEGIA — Decisione tecnica

In assenza di credenziali valide per l’accesso SSH diretto, ho proseguito l’attività concentrandomi sull’analisi dell’installazione WordPress dismessa, potenziale fonte di informazioni sensibili riutilizzabili.

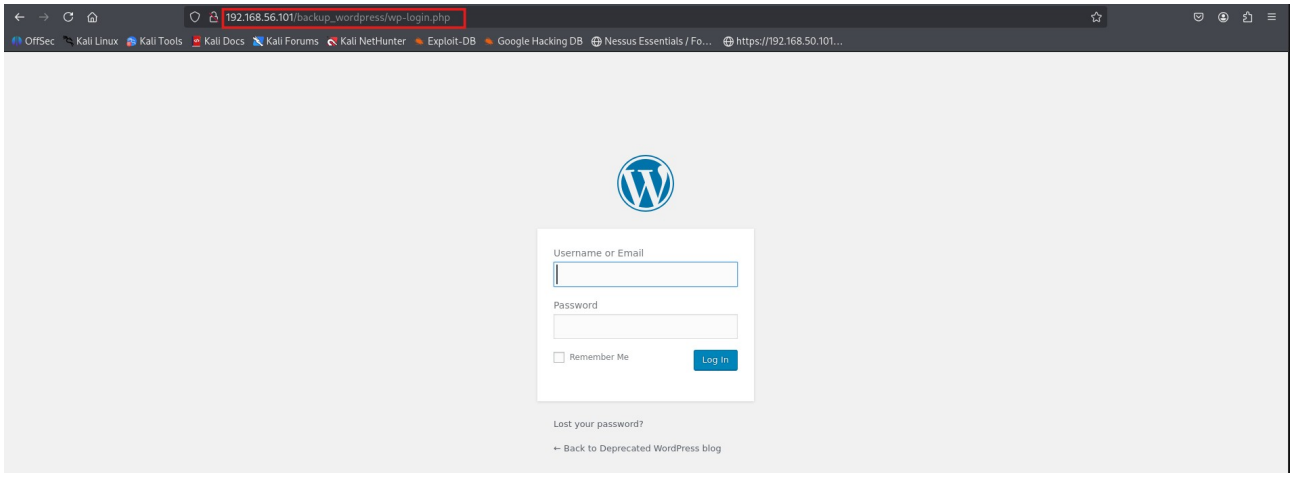
Step 4.5 — Enumerazione manuale di WordPress

Cosa ho fatto:
Ho proseguito l’attività effettuando un’enumerazione manuale dell’installazione WordPress dismessa presente nella directory `/backup_wordpress`, al fine di verificare l’accessibilità dell’area di autenticazione e individuare potenziali informazioni riutilizzabili.

1) Azione principale
Ho verificato l’accessibilità della pagina di login amministrativa di WordPress.

URL:

http://192.168.56.101/backup_wordpress/wp-login.php



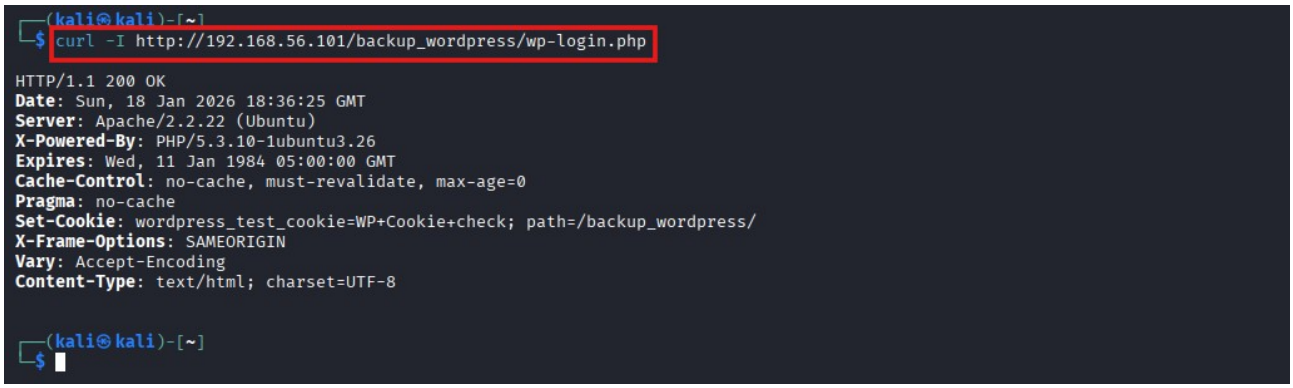
1) Pagina di login WordPress visualizzata nel browser

2) Verifica tramite CLI (opzionale)

Ho effettuato una verifica preliminare della risposta del server tramite richiesta HTTP.

Comando:

curl -I http://192.168.56.101/backup_wordpress/wp-login.php



- **2) (Opzionale) Output del comando curl -I**

Step 4.6 — Enumerazione degli utenti WordPress

Cosa ho fatto:
Ho effettuato l’enumerazione degli utenti WordPress sfruttando il parametro **`author`** nelle URL, tecnica comune per identificare username applicativi esposti pubblicamente.
L’obiettivo era individuare account WordPress validi potenzialmente riutilizzabili per tentativi di accesso mirati all’area amministrativa.

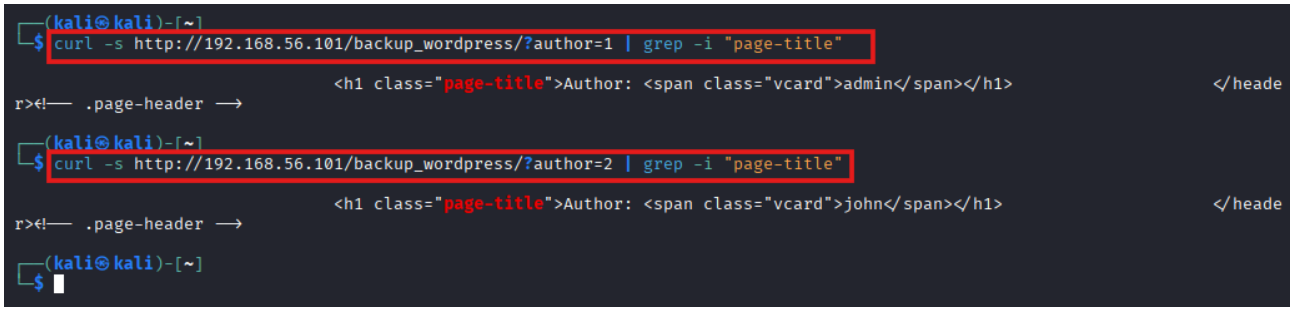
Azione eseguita

Ho interrogato manualmente l’applicazione variando l’ID dell’autore e analizzando il contenuto della risposta HTML per identificare il nome utente associato.

Comandi utilizzati:

```
1) curl -s
http://192.168.56.101/backup_wordpress/?author=1 |
grep -i "page-title"
```

```
2) curl -s
http://192.168.56.101/backup_wordpress/?author=2 |
grep -i "page-title"
```



Risultato ottenuto

L'enumerazione ha permesso di identificare i seguenti utenti WordPress validi:

- **admin** (author ID 1)
- **john** (author ID 2)

Entrambi gli account risultano pubblicamente esposti attraverso l'interfaccia WordPress e rappresentano potenziali candidati per tentativi di accesso applicativo mirati.

Considerazioni

La presenza di username WordPress esposti conferma una configurazione non adeguatamente protetta dell'installazione, **aumentando la superficie di attacco e facilitando eventuali tentativi di autenticazione controllati senza ricorrere a tecniche di brute force indiscriminato.**

Step 4.7 — Valutazione dell'accesso applicativo WordPress

Cosa ho fatto:
Dopo aver individuato gli utenti WordPress validi (**`admin`** e **`john`**), ho valutato la possibilità di accesso all'area amministrativa WordPress tramite autenticazione applicativa.
I tentativi sono stati mantenuti limitati e controllati, senza forzature, al solo scopo di verificare l'eventuale riutilizzo di credenziali note.

URL di riferimento:

`http://192.168.56.101/backup_wordpress/wp-login.php`

Risultato:
L’accesso non è stato ottenuto con le informazioni disponibili. **In assenza di credenziali valide, non sono stati effettuati ulteriori tentativi, proseguendo l’attività con l’analisi delle superfici di attacco applicative esposte.**

Step 4.8 — Analisi delle componenti WordPress esposte

Cosa ho fatto:
Ho analizzato le principali componenti WordPress pubblicamente esposte dell’installazione dismessa presente nella directory `/backup_wordpress`, al fine di individuare ulteriori superfici di attacco, informazioni sensibili o funzionalità potenzialmente sfruttabili per ottenere un accesso iniziale al sistema.

Comandi / URL verificati:

`http://192.168.56.101/backup_wordpress/xmlrpc.php`

`http://192.168.56.101/backup_wordpress/readme.html`

`http://192.168.56.101/backup_wordpress/wp-content/`

Output atteso:
Conferma dell’esistenza di endpoint e risorse WordPress accessibili pubblicamente, inclusi servizi di comunicazione remota (`xmlrpc.php`), file informativi e directory applicative.

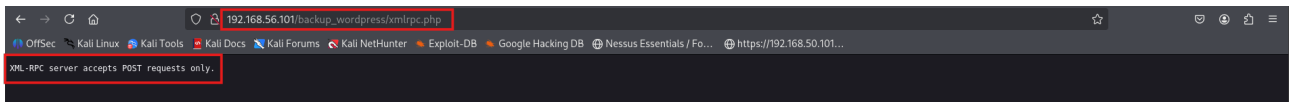
Step 4.9 — Verifica dell’endpoint xmlrpc.php

Cosa ho fatto:
Ho verificato la presenza e l’accessibilità dell’endpoint `xmlrpc.php` dell’installazione WordPress dismessa, al fine di determinare se il servizio di comunicazione remota fosse attivo e potenzialmente sfruttabile come superficie di attacco applicativa.

Comando / URL eseguito:

`http://192.168.56.101/backup_wordpress/xmlrpc.php`

Output atteso:
Messaggio informativo di WordPress che indica l’accettazione esclusiva di richieste POST, confermando che il servizio XML-RPC è attivo.



- **Output della richiesta HTTP all’endpoint `xmlrpc.php`**

Step 4.10 — Analisi delle versioni e delle superfici di attacco WordPress

Cosa ho fatto:
Ho analizzato le informazioni di versione e le componenti applicative dell'installazione WordPress dismessa, al fine di identificare l'utilizzo di software obsoleto e valutare la presenza di vulnerabilità note potenzialmente sfruttabili per ottenere un accesso iniziale al sistema.

Comandi eseguiti:

```
curl -I http://192.168.56.101/backup_wordpress/
```

```
curl -s http://192.168.56.101/backup_wordpress/ | grep -i "generator"
```

Output atteso:

Identificazione della versione di WordPress e delle tecnologie utilizzate (web server e linguaggio di scripting), confermando l'utilizzo di componenti obsolete e potenzialmente vulnerabili.

- **Informazioni di versione WordPress (es. WordPress 4.5)**
- **Dettagli sul web server e sulla versione di PHP**

FASI FINALI:

Step 4.11 — Analisi delle vulnerabilità note della versione WordPress individuata

Cosa ho fatto:
Ho analizzato le vulnerabilità note associate alla versione WordPress individuata durante la fase precedente, con l'obiettivo di valutare la presenza di debolezze applicative potenzialmente sfruttabili in uno scenario reale.

Comando eseguito:
Analisi informativa basata sulle informazioni di versione precedentemente raccolte e sul confronto con vulnerabilità note documentate per WordPress 4.5.

Output atteso:
Identificazione di vulnerabilità note associate alla versione WordPress utilizzata, confermando che **l'installazione risulta obsoleta e potenzialmente esposta a rischi di sicurezza.**

- **Riferimenti a vulnerabilità note di WordPress 4.5**
- **Conferma della superficie di attacco applicativa**

Step 4.12 — Valutazione dell'impatto e delle possibilità di sfruttamento

Cosa ho fatto:
Ho valutato l'impatto potenziale delle vulnerabilità applicative individuate sull'installazione WordPress analizzata, considerando le possibili conseguenze in termini di sicurezza del sistema.

Comando eseguito:
Valutazione qualitativa delle vulnerabilità individuate e delle superfici di attacco emerse durante le fasi precedenti.

Output atteso:
Valutazione del livello di rischio complessivo associato all'installazione WordPress analizzata e delle potenziali conseguenze in caso di sfruttamento.

- **Analisi dell'impatto su confidenzialità, integrità e disponibilità**
- **Giustificazione tecnica della necessità di mitigazione**

NOTA BENE: Nonostante l'analisi abbia evidenziato superfici di attacco applicative, non sono emerse condizioni sfruttabili in modo diretto per l'elevazione di privilegi a root.

Conclusioni ultime:

L’analisi Black Box ha consentito di individuare diverse superfici di attacco esposte dal sistema target, tra cui servizi accessibili senza adeguate restrizioni e un’installazione WordPress obsoleta e dismessa ma ancora raggiungibile.

Sebbene non sia stato ottenuto un accesso iniziale diretto, **le informazioni raccolte dimostrano come tali configurazioni possano rappresentare un rischio significativo in uno scenario reale.**

Si evidenzia l’importanza di limitare l’esposizione dei servizi, proteggere i file di backup e mantenere aggiornate le applicazioni web per ridurre il rischio di compromissione.