FINAL PENETRATION TEST REPORT: BSIDES VANCOUVER 2018

Studente: Caccamo Rocco Paolo

Corso: Cyber Security & Ethical Hacking (Epicode)

Target: BSides Vancouver 2018 (IP: 192.168.20.14) **Data:** 18 Gennaio 2026

---

1. EXECUTIVE SUMMARY

Il presente report documenta l'attività di **BlackBox Penetration Testing** condotta contro l'host `192.168.20.14`. L'attività ha portato alla compromissione totale del sistema (Root) attraverso una metodologia strutturata: dalla ricognizione iniziale (che ha evidenziato file di configurazione esposti come `robots.txt`) fino allo sfruttamento di software obsoleti e vulnerabilità del kernel.

---

2. FASE 1: INFORMATION GATHERING & NETWORK DISCOVERY

2.1 Host Discovery

Non conoscendo l'indirizzo della macchina target, è stata eseguita una scansione della sottorete locale per identificare gli host attivi.

- **Target Identificato:** `192.168.20.14`

2.2 Port Scanning (Nmap)

La scansione dei servizi TCP ha mappato la superficie di attacco iniziale.

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

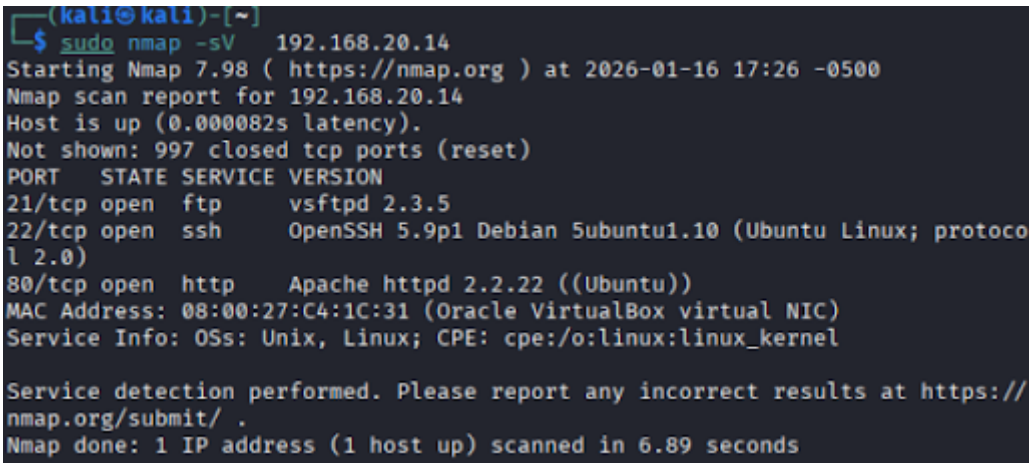

Fig 1: Output della scansione Nmap (Service Discovery).

2.3 Web Enumeration (Gobuster & Robots.txt)

Per investigare il servizio Web sulla porta 80, è stato utilizzato il tool **Gobuster**. La scansione ha rilevato due risorse fondamentali:

1. `/backup_wordpress` (Status 301): Una directory non visibile dalla home page.
2. `/robots.txt` (Status 200): File di configurazione per i crawler dei motori di ricerca.

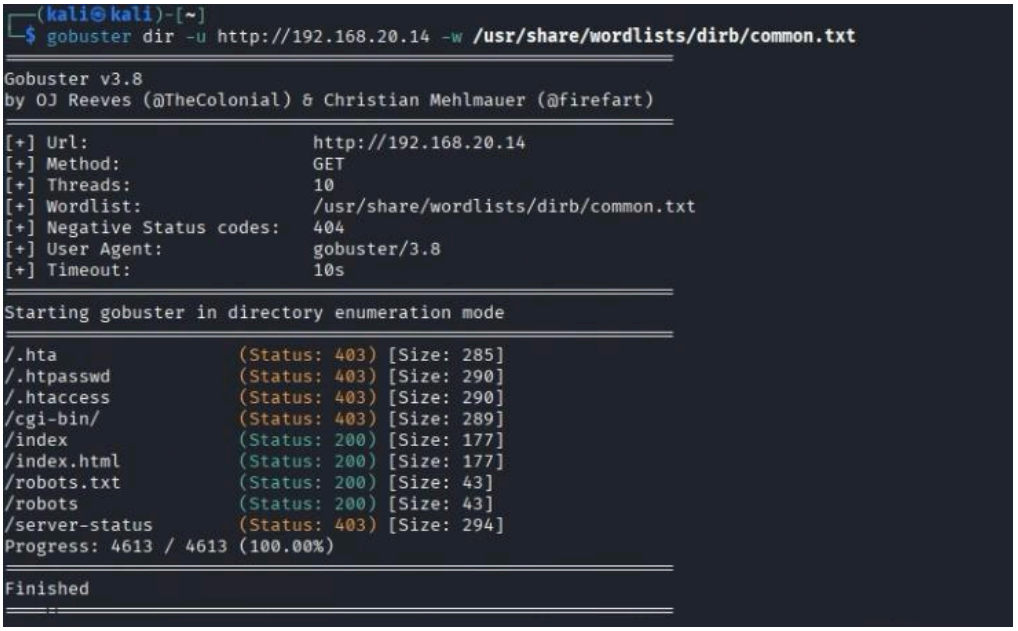

Fig 2: Gobuster individua il file robots.txt e la directory di backup.

Analizzando manualmente il contenuto del file `robots.txt` via browser, ho trovato la regola `Disallow: /backup_wordpress`. Questo ha confermato che l'amministratore intendeva nascondere deliberatamente quella cartella, rendendola un target prioritario per l'attacco.

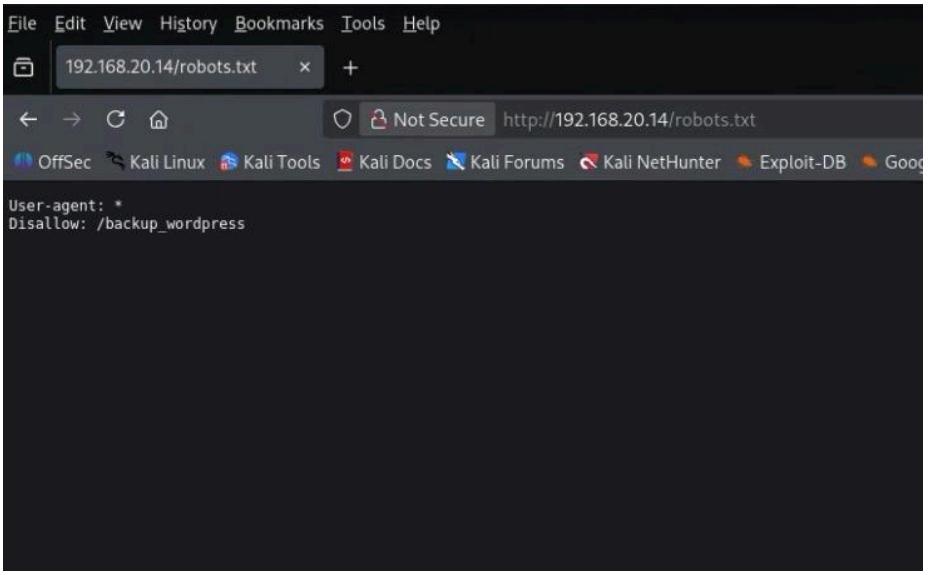

Fig 3: Il file robots.txt svela esplicitamente il percorso nascosto.

2.4 OSINT & Content Analysis

Visitando la pagina principale, un messaggio indicava che il sito era "Retired" e invitava a contattare l'amministratore **"John"**. Questa informazione (Information Leakage) ha permesso di focalizzare i successivi attacchi di brute-force su questo specifico username.

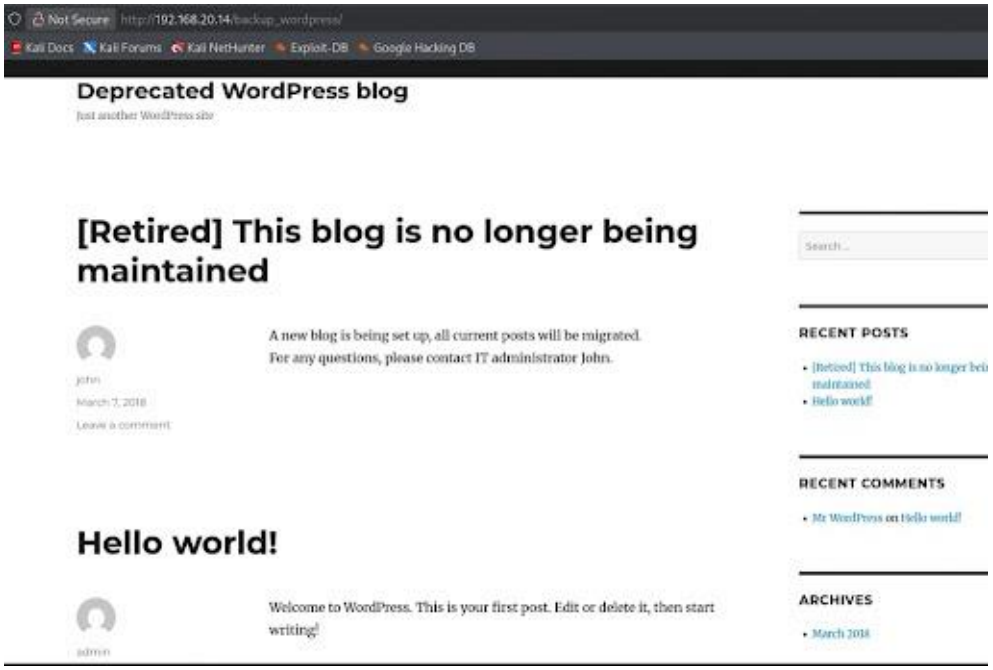

Fig 4: La pagina web rivela il nome dell'admin "John".

---

3. FASE 2: VULNERABILITY ASSESSMENT

3.1 FTP Enumeration

Il servizio FTP (Porta 21) permetteva l'accesso anonimo. All'interno è stato individuato e scaricato il file `users.txt.bk`. L'analisi del contenuto ha confermato la lista degli utenti di sistema, validando l'esistenza dell'utente `john`.

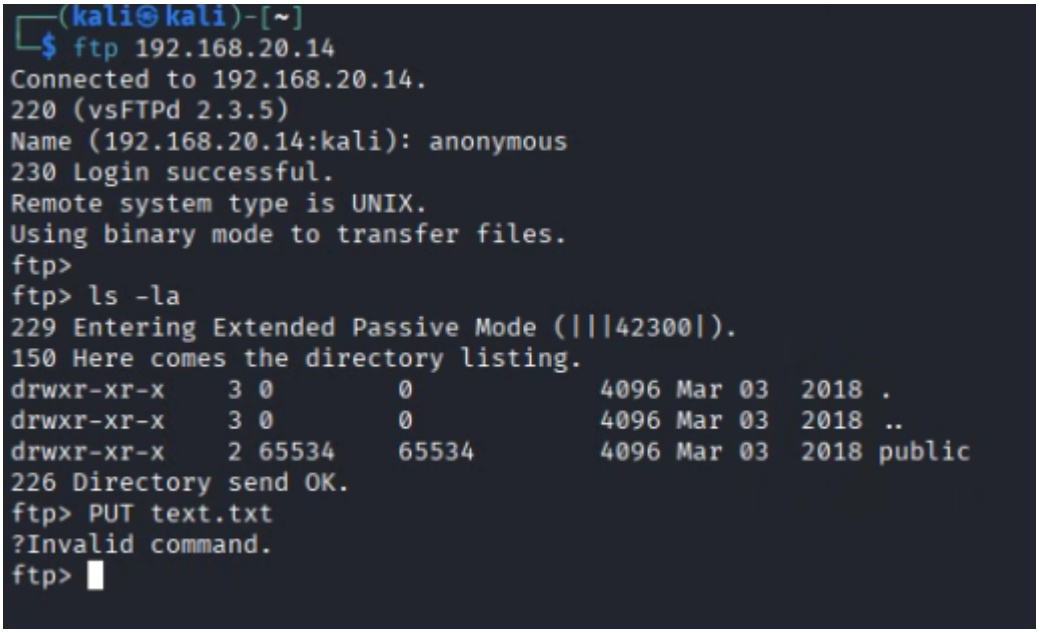

Fig 5: Accesso FTP Anonimo e listing dei file.

---

4. FASE 3: EXPLOITATION (INITIAL ACCESS)

4.1 Tentativo Fallito: SSH Brute-Force (Dead End)

Un primo tentativo di attacco a dizionario contro il servizio SSH è fallito.

- **Errore:** `ssh target does not support password auth`.
- **Analisi:** Il server SSH accetta solo chiavi pubbliche, rendendo inutile il brute-force delle password.

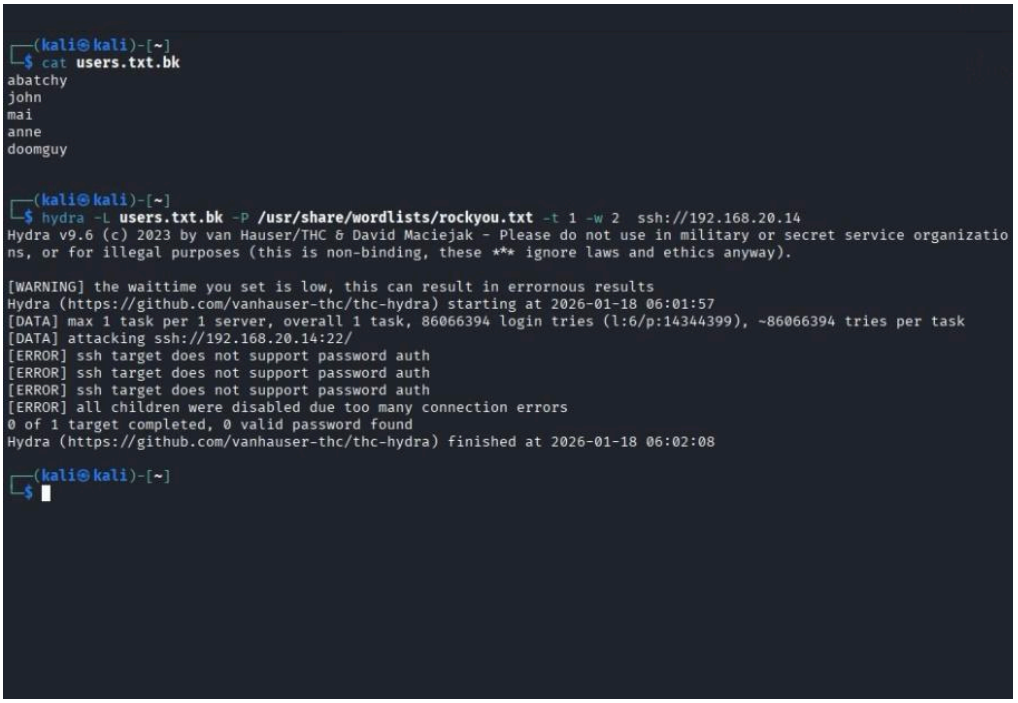

Fig 6: Fallimento dell'attacco SSH (Configurazione Key-Only).

4.2 Attacco Riuscito: WordPress Login

Sfruttando la directory scoperta tramite `robots.txt` (`/backup_wordpress`) e l'utente `john`, è stato lanciato un attacco Hydra contro il pannello di login.

- **Target:** `wp-login.php`
- **Credenziali Trovate:** `john` / `enigma`

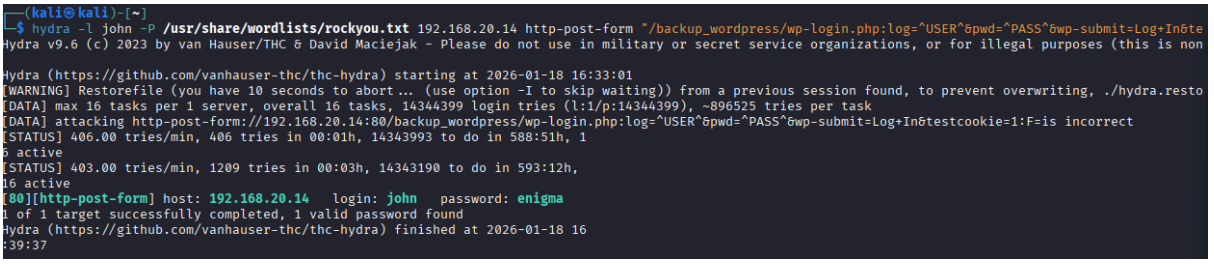

Fig 7: Successo del cracking su WordPress.

4.3 Remote Code Execution (RCE)

Dalla dashboard di amministrazione, ho iniettato una **Reverse Shell PHP** modificando il file `404.php` tramite l'editor dei temi. Visitando la pagina infetta, ho ottenuto una shell remota come utente `www-data`.

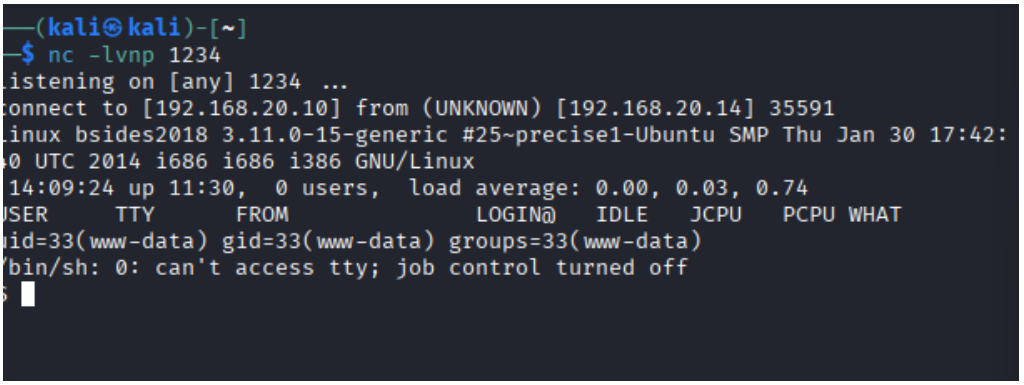

Fig 8: Accesso al sistema ottenuto (Shell www-data).

---

5. FASE 4: PRIVILEGE ESCALATION (ROOT)

Per l'elevazione dei privilegi, è stata identificata una vulnerabilità nel componente di sistema `pkexec`. È stato utilizzato l'exploit locale **PwnKit (CVE-2021-4034)**:

1. Creazione del sorgente `pwnkit.c` in `/tmp`.
2. Compilazione con `gcc`.
3. Esecuzione e ottenimento immediato dei privilegi di **ROOT**.

Fig 9: Esecuzione exploit PwnKit.

Proof of Concept: Lettura del file `/root/flag.txt`.

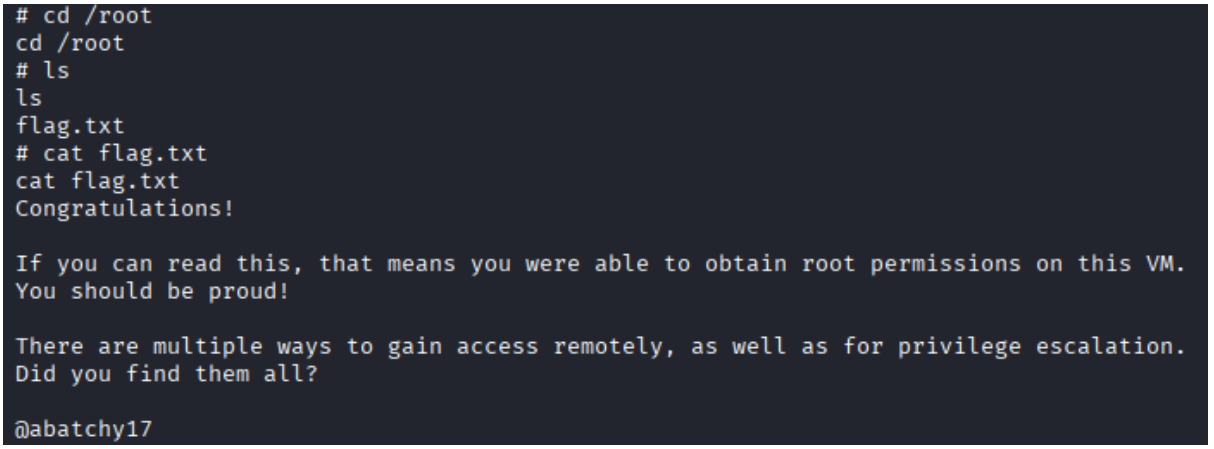

Fig 10: Missione Compiuta - Cattura della Flag.

5.1 Cleanup

Al termine delle operazioni, sono state rimosse le tracce forensi (file `pwnkit.c` ed eseguibile `exploit`) dalla directory temporanea.

```
# rm -rf /tmp/pwnkit.c /tmp/exploit /tmp/GCONV_PATH*
rm -rf /tmp/pwnkit.c /tmp/exploit /tmp/GCONV_PATH*
#
```

Fig 11: Pulizia dell'ambiente.

---

6. CONCLUSIONI E REMEDIATION

L'attività di test ha evidenziato diverse criticità di livello Alto e Critico che, concatenate, hanno permesso la compromissione totale del sistema.

Analisi delle Criticità (Root Cause Analysis)

1. **Security by Obscurity Fallita:** L'uso del file `robots.txt` per nascondere la cartella `/backup_wordpress` ha avuto l'effetto opposto, rivelandone l'esistenza.
2. **Credenziali Deboli:** La password dell'amministratore WordPress ("enigma") era presente nei dizionari comuni (`rockyou.txt`), rendendo triviale l'attacco brute-force.
3. **Configurazioni FTP Insicure:** Lasciare l'accesso anonimo abilitato e file di backup (`users.txt.bk`) nelle directory pubbliche ha esposto la lista degli utenti validi.
4. **Patch Management Carente:** Il sistema operativo non era aggiornato, esponendolo a vulnerabilità note del kernel (CVE-2021-4034) che hanno permesso la scalata a Root.

Piano di Rimedio (Consigli per il Blue Team)

Per mettere in sicurezza l'infrastruttura, si raccomanda di:

- **FTP:** Disabilitare immediatamente l'accesso anonimo e rimuovere qualsiasi file sensibile dalle directory pubbliche.
- **WordPress:**
    - Rimuovere le installazioni di test o di backup (`/backup_wordpress`).
    - Imporre policy per password complesse (minimo 12 caratteri, alfanumerici e simboli).
    - Disabilitare l'editor dei file da dashboard aggiungendo `define('DISALLOW_FILE_EDIT', true);` nel file `wp-config.php`.
- **Sistema Operativo:** Applicare urgentemente le patch di sicurezza per correggere la vulnerabilità di `polkit` (`pkexec`).

- **SSH:** Mantenere l'attuale configurazione "Key-Only" (solo chiavi pubbliche), in quanto si è rivelata l'unica difesa efficace durante il test.