Relazione esercizio S6-L4

Introduzione

L'esercizio di oggi prevede l'utilizzo di John the ripper per prendere le password hashate e renderle in chiaro.

Primo passo

Come prima cosa dobbiamo fare un SQL Injection per risalire al db di DVWA.

Usiamo il comando: `sqlmap -u "http://192.168.50.5/dvwa/vulnerabilities/sqli/?id='&Submit=Submit" --cookie="security=low; PHPSESSID=b1155feab6e7a965ad3931bf498e1c90" --dbs`

Dove:

- `192.168.50.5`: è l'indirizzo della *metasploitable*
- `cookie`: sono i cookie di sessione
- `security=low`: è il livello di sicurezza della DVWA
- `--dbs`: è il comando per mostrare i database su DVWA

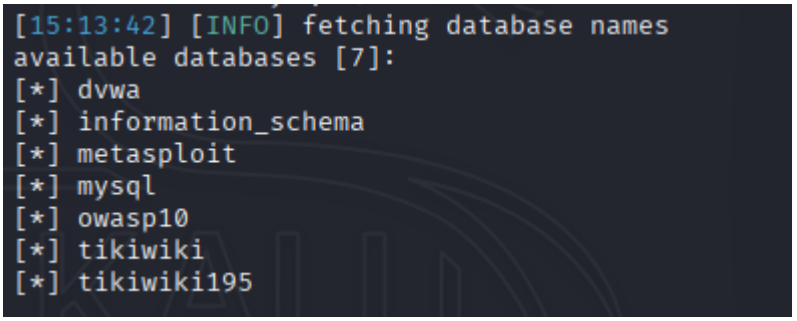
Come si ottengono i cookie di sessione?

1. Dobbiamo andare su `http://192.168.50.5/` e andare sulla DWA
2. Andiamo su DVWA security e impostiamo a `low`
3. Andiamo su SQL Injection e facciamo un'ispezione di pagina
4. Apriamo la console e usiamo il comando `document.cookie` per ricevere i cookie di sessione. Copiamo tutto (anche i doppi apici).

Secondo passo

Una volta lanciato il primo comando si avvierà una scansione che può durare diversi minuti. Facciamo tutto yes e andiamo avanti.

Una volta terminata la scansione, vediamo su output i database disponibili, come in figura:



Andiamo avanti eseguendo il comando `sqlmap -u "http://192.168.50.5/dvwa/vulnerabilities/sqli/?id='&Submit=Submit" --cookie="security=low; PHPSESSID=b1155feab6e7a965ad3931bf498e1c90" -D dvwa --tables`

dove:

- `-D dvwa --tables` : è il comando che permette di prendere il database chiamato `dvwa` e mostrare le sue tabelle

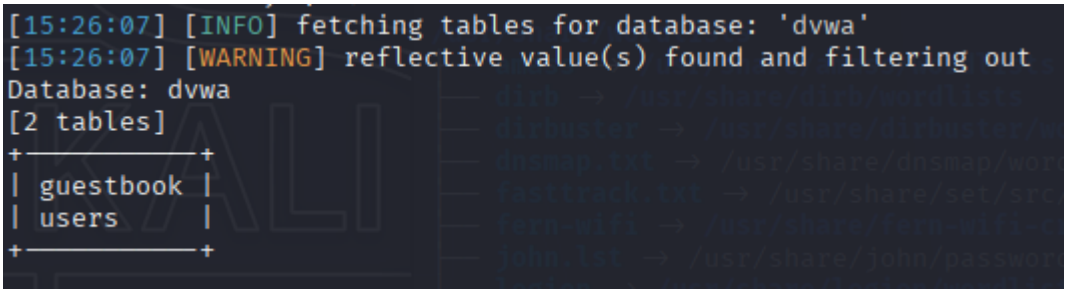
Dopo aver lanciato questo comando, apparirà un *legal disclaimer*: accettiamo tutto e andiamo avanti.

> **Tip**
>
> Questo disclaimer apparirà per tutti i prossimi comandi. Accettiamo sempre tutto e andiamo avanti.

Apparirà poi il risultato. In questo caso, all'interno del database dvwa abbiamo due tabelle:

- guestbook
- users

Come possiamo vedere nell'immagine sottostante:



Terzo passo

Andiamo avanti e lanciamo il seguente comando: `sqlmap -u "http://192.168.50.5/dvwa/vulnerabilities/sqli/?id='&Submit=Submit" --cookie="security=low; PHPSESSID=b1155feab6e7a965ad3931bf498e1c90" -D dvwa -T users --columns`

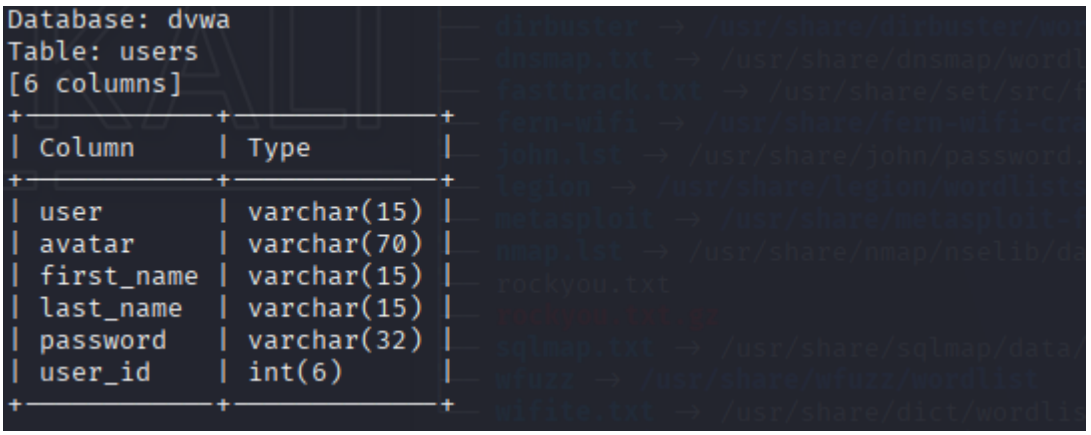
dove:

- `-D dvwa -T users --columns`: è il comando che ci permette di vedere le colonne della tabella che stiamo prendendo in considerazione (in questo caso `users`).

Accettiamo il disclaimer e andiamo avanti.

A questo punto dovremmo trovare le colonne della tabella `users` con i suoi rispettivi `Type`.

Di seguito la figura:



Quarto passo

Andiamo avanti eseguendo il comando `sqlmap -u "http://192.168.50.5/dvwa/vulnerabilities/sqli/?id='&Submit=Submit" --cookie="security=low; PHPSESSID=b1155feab6e7a965ad3931bf498e1c90" -D dvwa -T users --dump`

dove:

- `-D dvwa -T users --dump`: è il comando che prende la tabella users dal database dvwa e mostra il suo contenuto.

Accettiamo il disclaimer e andiamo avanti.

Dovrebbe uscire il contenuto della tabella come nella seguente immagine:

Vediamo come si possono vedere i contenuti della tabella users.

Notiamo anche che le password sono hashate con metodo MD5.

Notiamo come la prima e l'ultima password abbiano lo stesso hash, di conseguenza quelle password saranno uguali.

Quinto passo

A questo punto, prendiamo tutte le password hashate e le mettiamo in un file `.txt`.

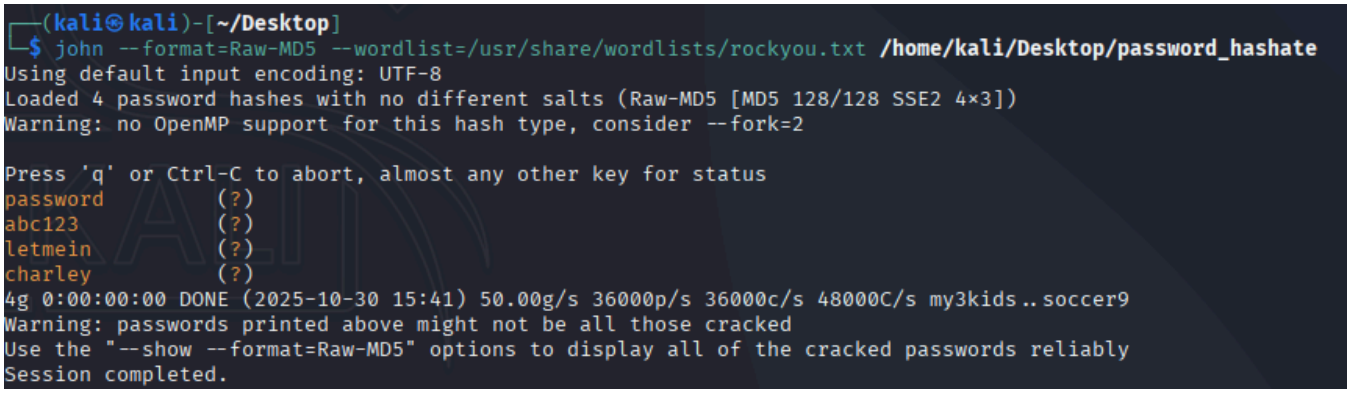
Prima di usare John, facciamo partire `Wordlist` per estrarre il file `rockyou.txt` che ci servirà quando lanceremo John.

Una volta fatto questo passaggio, ci spostiamo nella directory in cui abbiamo creato il file `.txt` con le password hashte e apriamo un terminale.

Usiamo e lanciamo il comando `john --format=Raw-MD5 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Desktop/password_hashate`

che ti apre il file `.txt`, legge le password hashte e le mette in chiaro.

Di seguito l'immagine mostra come John the ripper mette le password in chiaro in pochi secondi:



Conlusioni

John the Ripper si conferma uno strumento essenziale nel workflow di analisi e verifica delle credenziali all'interno di un laboratorio di sicurezza informatica. Grazie alla sua semplicità d'uso, al supporto per numerosi formati di hash e all'integrazione con wordlist e regole di trasformazione, John permette di dimostrare in modo chiaro e ripetibile la vulnerabilità derivante dall'impiego di algoritmi di hashing obsoleti o privi di sale (es. MD5 non salato). Nell'esercizio svolto su DVWA, l'utilizzo di John ha messo in evidenza quanto rapidamente password deboli o presenti in dizionari pubblici possano essere recuperate, fornendo una prova pratica dell'efficacia di attacchi offline contro hash insufficientemente protetti.

I principali punti di forza emersi sono la flessibilità operativa (modalità dizionario, regole, maschere), la possibilità di riprendere sessioni interrotte e la facilità di integrazione in workflow automatizzati per la raccolta e l'analisi dei risultati. Tuttavia, lo strumento presenta limiti intrinseci: contro hash robusti, salati e derivati da funzioni intenzionalmente lente (bcrypt, scrypt, Argon2) il successo del cracking diminuisce drasticamente; in questi scenari è preferibile l'adozione di piattaforme GPU (es. Hashcat) o attacchi più sofisticati, ma spesso con costi computazionali elevati.

Le evidenze sperimentali ottenute con John dovrebbero quindi essere usate come leva per raccomandare miglioramenti reali nella gestione delle credenziali:

- migrazione verso algoritmi di hashing sicuri e salati (bcrypt, Argon2id) con parametri di costo aggiornati;
- adozione di policy che includano controllo delle password contro blacklist e strumenti di password manager;

- implementazione di contromisure lato applicazione come rate-limiting, blocco temporaneo dopo tentativi ripetuti e monitoraggio degli accessi;
- protezione e minimizzazione dell'accesso alle risorse che contengono hash (principio del privilegio minimo, cifratura a riposo, gestione sicura delle credenziali del DB).

Infine, è fondamentale sottolineare l'aspetto etico e legale: l'uso di John the Ripper deve essere limitato a ambienti autorizzati e controllati (lab, test di penetrazione con consenso). Ogni attività di cracking eseguita su sistemi di produzione o su dati di terzi senza autorizzazione è illegale e potenzialmente dannosa. Nella documentazione della verifica è opportuno includere sempre l'ambito del test, le autorizzazioni ricevute e le misure adottate per evitare impatti non autorizzati, in modo da garantire trasparenza e responsabilità nell'attività di sicurezza.