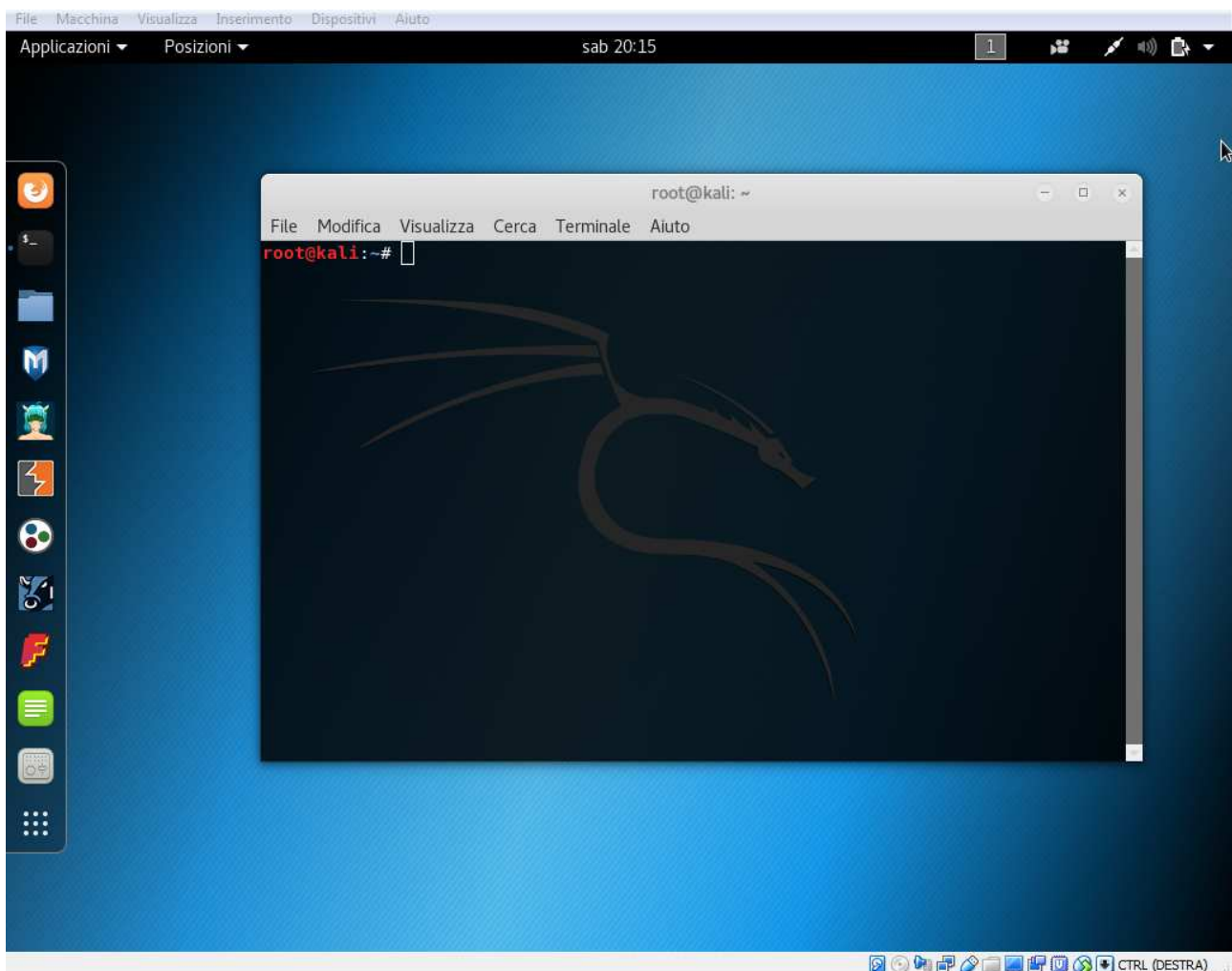


## KALI LINUX in VIRTUALBOX

Kali Linux è una distribuzione basata su Debian GNU/Linux, pensata per l'informatica forense e la sicurezza informatica, in particolare per effettuare penetration test.

Kali offre agli utenti un semplice accesso ad una larga collezione di tools per la sicurezza dal port scanning ai password cracker.

Utilizzeremo una macchina virtuale con la distribuzione Kali Linux installata, in ambiente Virtualbox e ci loggeremo con le credenziali di root con password Cor\$oRet!



## ESERCIZIO – CRACK DI PASSWORD CON HASCAT

Hashcat è un programma che permette di risalire alla password degli utenti Linux partendo dal suo hash memorizzato nel file `/etc/shadow`,

Linux utilizza come algoritmo di hashing lo SHA-512.

Per effettuare il crack delle password dobbiamo entrare in possesso del file shadow memorizzato sul pc da violare

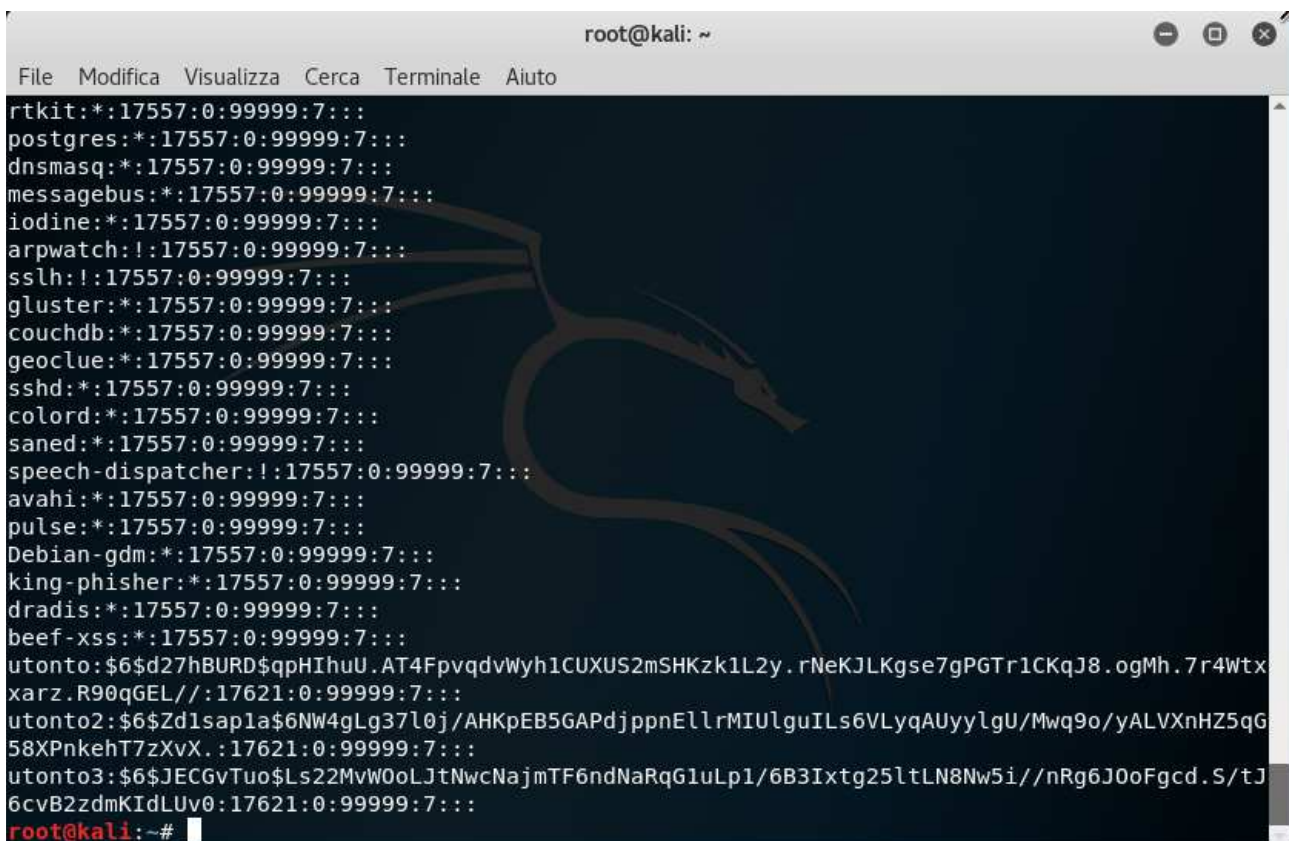
Ci sono diverse tecniche per ottenerlo ma, non essendo l'argomento del presente corso, supponiamo di esserne in possesso.

Per l'esercizio utilizziamo il file shadow presente nella macchina virtuale.

Per visualizzare il file shadow, apriamo un terminale e scriviamo:

```
cat /etc/shadow
```

Otteniamo il seguente risultato:



```
root@kali: ~
File Modifica Visualizza Cerca Terminale Aiuto
rtkit:*:17557:0:99999:7:::
postgres:*:17557:0:99999:7:::
dnsmasq:*:17557:0:99999:7:::
messagebus:*:17557:0:99999:7:::
iodine:*:17557:0:99999:7:::
arpwatch:!:17557:0:99999:7:::
sslh:!:17557:0:99999:7:::
gluster:*:17557:0:99999:7:::
couchdb:*:17557:0:99999:7:::
geoclue:*:17557:0:99999:7:::
sshd:*:17557:0:99999:7:::
colord:*:17557:0:99999:7:::
saned:*:17557:0:99999:7:::
speech-dispatcher:!:17557:0:99999:7:::
avahi:*:17557:0:99999:7:::
pulse:*:17557:0:99999:7:::
Debian-gdm:*:17557:0:99999:7:::
king-phisher:*:17557:0:99999:7:::
dradis:*:17557:0:99999:7:::
beef-xss:*:17557:0:99999:7:::
utonto:$6$d27hBURD$qpHIhuU.AT4Fpvqdvwyh1CUXUS2mSHKzk1L2y.rNeKJLKgse7gPGTr1CKqJ8.ogMh.7r4Wtx
xarz.R90qGEL//:17621:0:99999:7:::
utonto2:$6$Zd1sap1a$6NW4gLG37l0j/AHKpEB5GAPdjppnEllrMIUlgUILs6VlyqAUyylgU/Mwq9o/yALVXnHZ5qG
58XPnkehT7zXvX.:17621:0:99999:7:::
utonto3:$6$JECGvTuo$LS22MvW0oLJtNwcNajmTF6ndNaRqG1uLp1/6B3Ixtg25ltLN8Nw5i//nRg6J0oFgcd.S/tJ
6cvB2zdmKIDLuv0:17621:0:99999:7:::
root@kali:~#
```

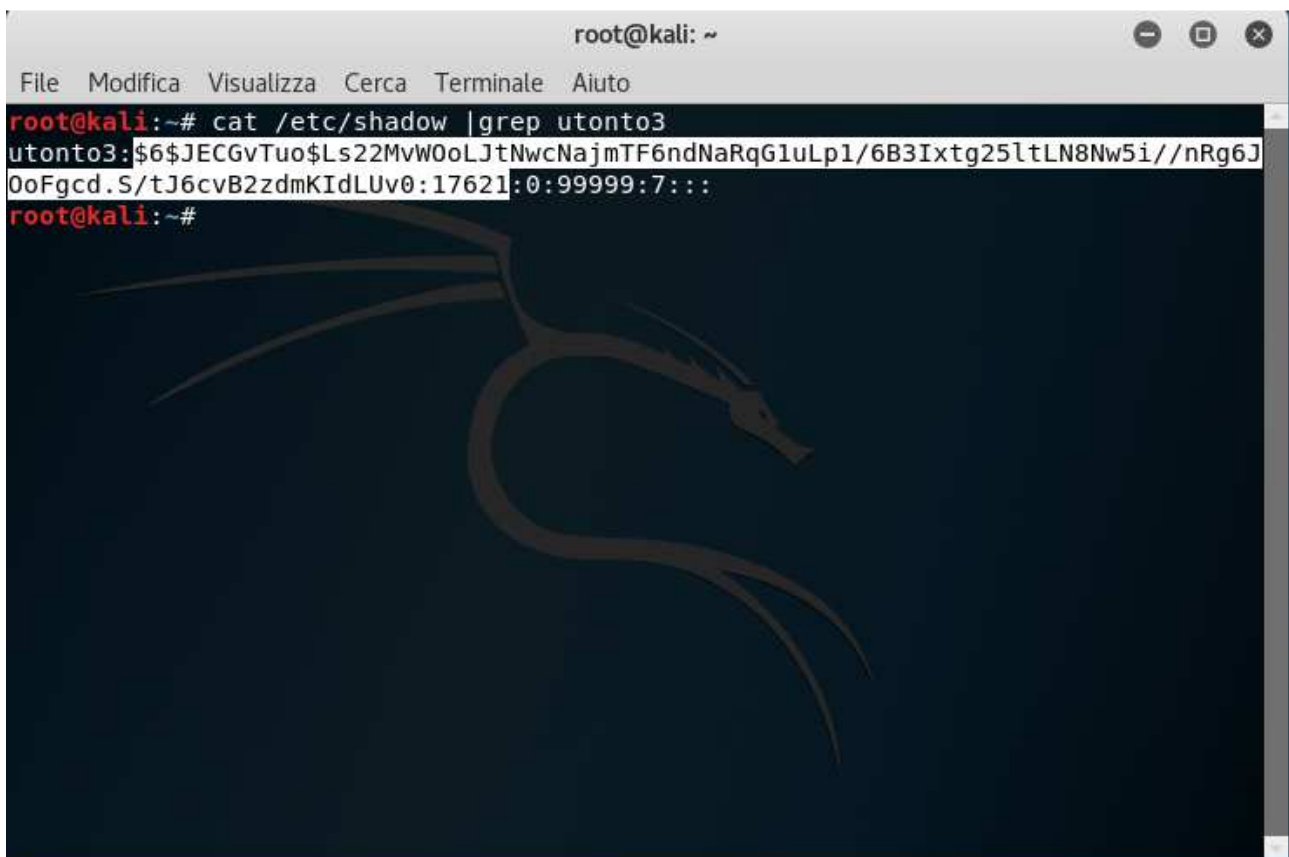
Notiamo che abbiamo 3 utenti alla fine del file: utonto, utonto2, utonto3

Proviamo a crackare la password di utonto3

Copiamo l'hash in un file chiamato hash.txt

```
cat /etc/shadow |grep utonto3
```

copiamo la stringa dell'hash, evidenziamo l'hash , andiamo sul menù modifica e selezioniamo copia.



```
root@kali: ~  
File Modifica Visualizza Cerca Terminale Aiuto  
root@kali:~# cat /etc/shadow |grep utonto3  
utonto3:$6$JECGvTuo$Ls22MvW0oLJtNwcNajmTF6ndNaRqG1uLp1/6B3Ixtg25ltLN8Nw5i//nRg6J  
0oFgcd.S/tJ6cvB2zdmKIdLUv0:17621:0:99999:7:::  
root@kali:~#
```

Apriamo un editor di testo , per esempio vim, digitiamo quindi:

```
vim hash.txt
```

E incolliamo l'hash, andando sul menù e selezionando modifica , incolla.

Premiamo **ESC** , e poi digitiamo: **:wq!**

Per salvare e uscire.

## GENERAZIONE DI UN DIZIONARIO

Un attacco del tipo “Brute Force” consiste nel provare tutte le parole o le stringhe di caratteri contenute in un file chiamato dizionario.

Esistono svariati dizionari disponibili online, qui ne trovate qualcuno:

<https://wiki.skullsecurity.org/Passwords>

ma ora vediamo come crearne uno nostro utilizzando il programma Crunch contenuto in Kali Linux.

L'utilizzo di questo tool da linea di comando, consente la creazione di dizionari contenenti anche milioni di password, mediante l'input di pochi caratteri.

La sua sintassi è la seguente:

```
crunch [min] [max] [charset] [options] -o [output]
```

Per esempio, se volessimo creare un file, chiamato dizionario.txt, contenente password lunghe da 4 a 6 caratteri, con cifre da 0 a 9, digitiamo:

```
crunch 4 6 0123456789 -o dizionario.txt
```

L'opzione **-t**, ci permette di definire un pattern, un set di caratteri dove alcuni di essi sono fissi mentre altri cambiano in ogni password scritta.

Mettiamo di volere un dizionario contenente la parola **pippo** seguita da tutte le lettere dell'alfabeto, sarà sufficiente digitare:

```
crunch 6 6 -t pippo@ -o dizionariopippo.txt
```

In questo caso, il carattere “@” viene sostituito da tutte le lettere minuscole dell'alfabeto.

Possiamo sostituire “@” con il carattere “,” per le lettere maiuscole, “%” per i numeri e “^” per i simboli.

Un'altra opzione molto interessante è la **-p**, che permette di generare permutazioni di parole, utilizzandola sarà comunque necessario dichiarare il numero minimo e

massimo di caratteri da utilizzare, sebbene questi verranno ignorati durante la creazione del dizionario, deve sempre essere scritta come ultima opzione.

```
crunch 1 1 -o dizionarioabc.txt -p abc
```

Questo comando genererà un file contenente le seguenti parole: abc acb bac bca cab cba.

Può anche essere utilizzata per generare permutazioni di parole, ovvero parole composte da altre parole in tutte le possibili combinazioni:

```
crunch 1 1 -o dizionario.txt -p mela pera arancia
```

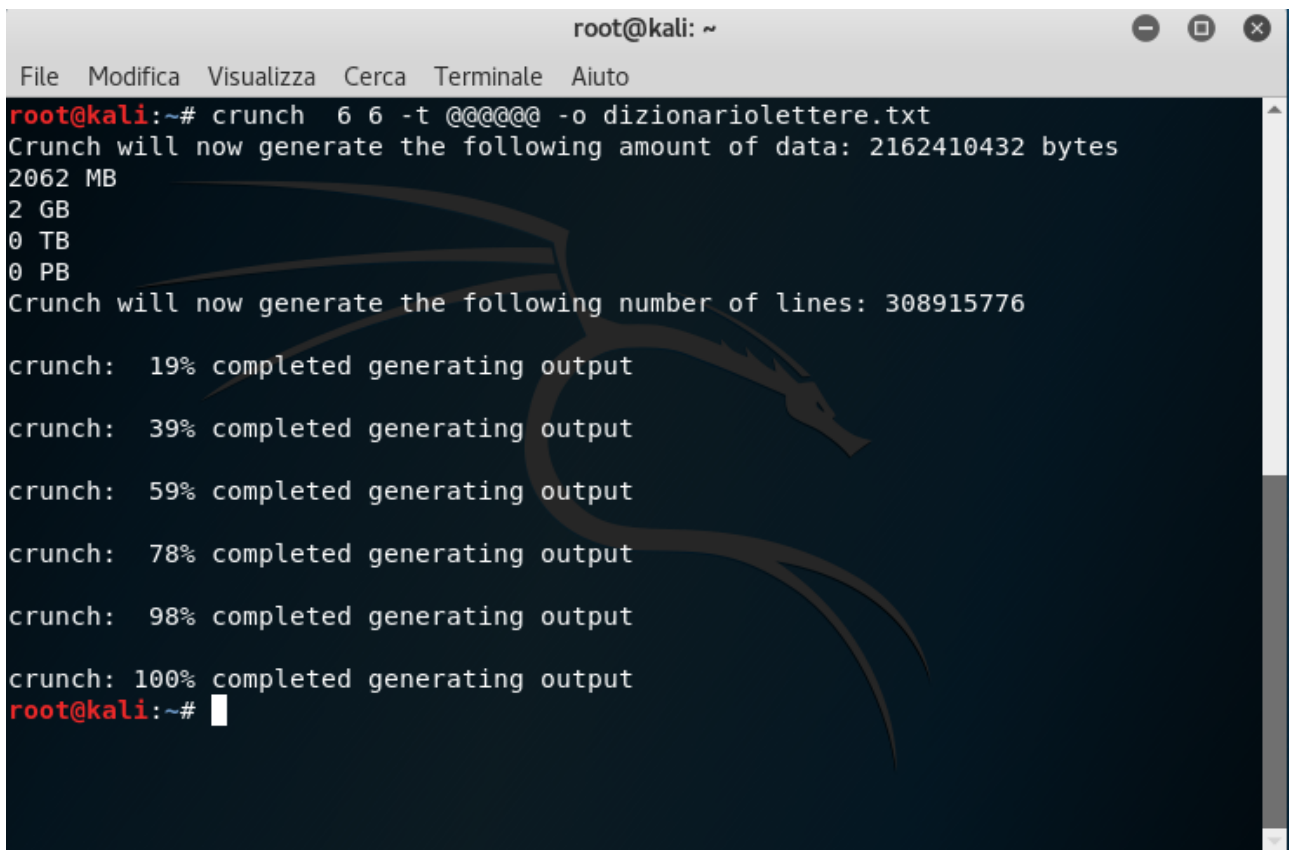
Con questo comando otterremo la seguente combinazione: aranciamelapera aranciaperamela melaaranciapera melaperaarancia peraaranciamela peramelaarancia.

## CRACKING DI UNA PASSWORD

Ora che abbiamo visto come creare un dizionario, procediamo con il crack della password dell'utente utonto3

Creiamoci un ulteriore dizionario contenente password lunghe 6 caratteri, con lettere minuscole dell'alfabeto

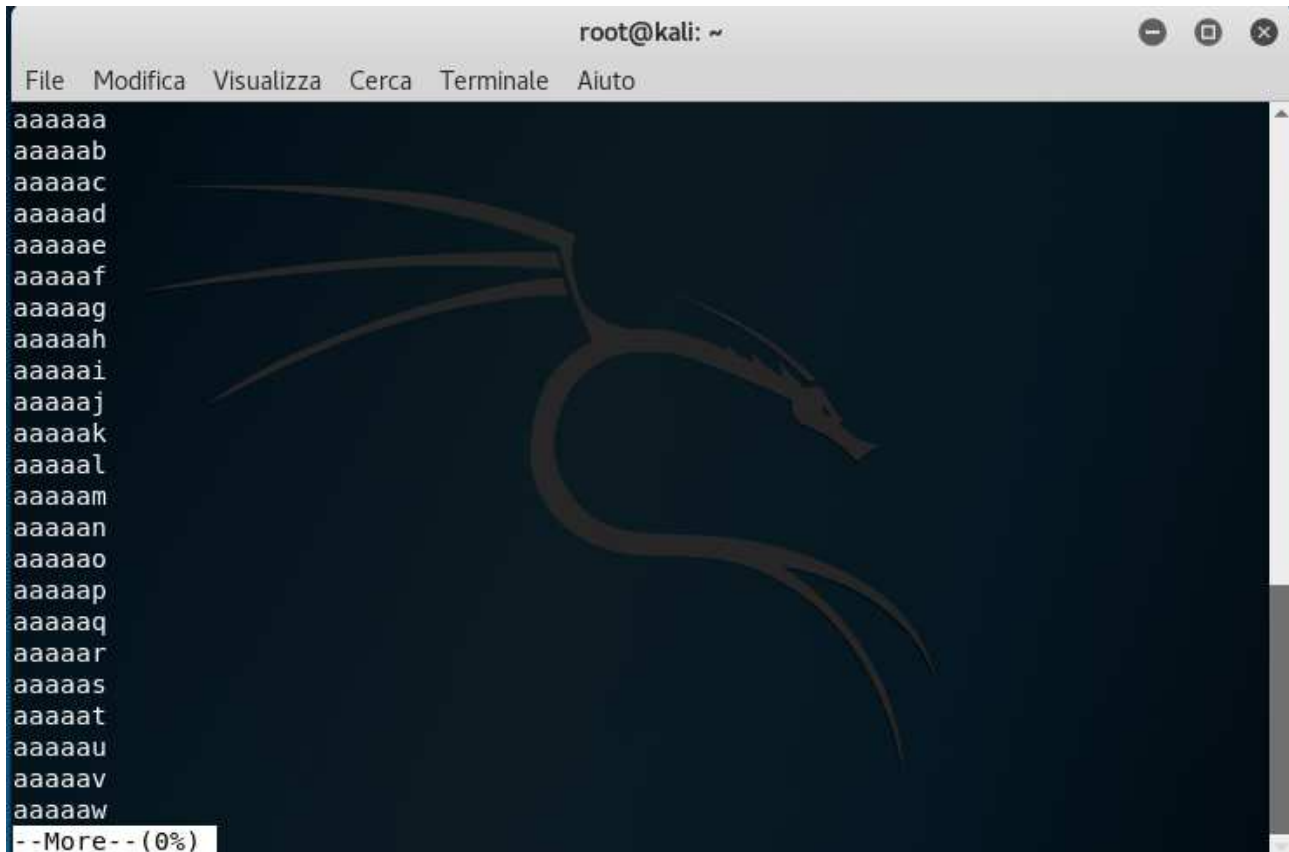
```
crunch 6 6 -t @@@@#@ -o dizionariolettere.txt
```

A screenshot of a terminal window titled 'root@kali: ~'. The window has a menu bar with 'File', 'Modifica', 'Visualizza', 'Cerca', 'Terminale', and 'Aiuto'. The terminal shows the command 'crunch 6 6 -t @@@@#@ -o dizionariolettere.txt' being executed. The output indicates that 2162410432 bytes (2062 MB) of data will be generated, consisting of 308915776 lines. Progress updates show the process reaching 100% completion. The terminal ends with the prompt 'root@kali:~#'.

```
root@kali: ~
File Modifica Visualizza Cerca Terminale Aiuto
root@kali:~# crunch 6 6 -t @@@@#@ -o dizionariolettere.txt
Crunch will now generate the following amount of data: 2162410432 bytes
2062 MB
2 GB
0 TB
0 PB
Crunch will now generate the following number of lines: 308915776
crunch: 19% completed generating output
crunch: 39% completed generating output
crunch: 59% completed generating output
crunch: 78% completed generating output
crunch: 98% completed generating output
crunch: 100% completed generating output
root@kali:~#
```

Diamo un'occhiata al file, digitiamo

```
more dizionariolettere.txt
```



```
root@kali: ~
File Modifica Visualizza Cerca Terminale Aiuto
aaaaa
aaaaab
aaaaac
aaaaad
aaaaae
aaaaaf
aaaaag
aaaaah
aaaaai
aaaaaj
aaaaak
aaaaal
aaaaam
aaaaan
aaaaao
aaaaap
aaaaaq
aaaaar
aaaaas
aaaaat
aaaaau
aaaaav
aaaaaw
--More-- (0%)
```

Usiamo ora, questo dizionario con il file hash dell'utente utonto3, digitiamo (tutto su una riga):

```
hashcat -force -m 1800 -a 0 hash.txt dizionariolettere.txt
```

```

root@kali:~# hashcat --force -m 1800 -a 0 hash.txt dizionariolettere.txt
hashcat (pull/1273/head) starting...

OpenCL Platform #1: The pocl project
=====
* Device #1: pthread-Intel(R) Core(TM) i7-4810MQ CPU @ 2.80GHz, 1498/1498 MB all
ocatable, 1MCU

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

Applicable optimizers:
* Zero-Byte
* Single-Hash
* Single-Salt
* Uses-64-Bit

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

* Device #1: build_opts '-I /usr/share/hashcat/OpenCL -D VENDOR_ID=64 -D CUDA_AR
CH=0 -D VECT_SIZE=2 -D DEVICE_TYPE=2 -D DGST_R0=0 -D DGST_R1=1 -D DGST_R2=2 -D D
GST_R3=3 -D DGST_ELEM=16 -D KERN_TYPE=1800 -D _unroll -cl-std=CL1.2'
Dictionary cache built:
    
```

Dopo un pò di minuti il programma riesce a crackare la password

```

Dictionary cache built:
* Filename..: dizionariolettere.txt
* Passwords.: 308915776
* Bytes.....: 2162410432
* Keyspace..: 308915776
* Runtime...: 23 secs

- Device #1: autotuned kernel-accel to 176
- Device #1: autotuned kernel-loops to 46
[s]tatus [p]ause [r]esume [b]ypass [c]heckpoint [q]uit => [s]tatus [p]ause [r]es
$6$JECGvTuo$Ls22MvW0oLJtNwcNajmTF6ndNaRqG1uLp1/6B3Ixtg25ltLN8Nw5i//nRg6J0oFgcd.S
/tJ6cvB2zdmKIdLUv0:aaaflg

Session.....: hashcat
Status.....: Cracked
Hash.Type.....: sha512crypt $6$, SHA512 (Unix)
Hash.Target.....: $6$JECGvTuo$Ls22MvW0oLJtNwcNajmTF6ndNaRqG1uLp1/6B3I...IdLUv0
Time.Started.....: Sat Mar 31 21:16:50 2018 (31 secs)
Time.Estimated...: Sat Mar 31 21:17:21 2018 (0 secs)
Guess.Base.....: File (dizionariolettere.txt)
Guess.Queue.....: 1/1 (100.00%)
Speed.Dev.#1.....: 115 H/s (14.04ms)
Recovered.....: 1/1 (100.00%) Digests, 1/1 (100.00%) Salts
Progress.....: 3696/308915776 (0.00%)
Rejected.....: 0/3696 (0.00%)
Restore.Point....: 3520/308915776 (0.00%)
Candidates.#1....: aaaffk -> aaafmd
HWMon.Dev.#1.....: N/A

Started: Sat Mar 31 21:16:27 2018
Stopped: Sat Mar 31 21:17:22 2018
    
```



Analizziamo le opzioni di hascat che abbiamo inserito:

- force serve esclusivamente se si utilizza Kali Linux su una macchina virtuale;
- m 1800 indica il tipo di hash da attaccare, nel nostro caso SHA-512;
- a 0 specifica che vogliamo eseguire un attacco standard;

un'altra opzione utile è la seguente;

-potfile-disable , per disabilitare la lettura delle hash già crackate, serve per riprovare a ricrackare password già crackate, visto che hashcat salva l'elenco delle password già trovate nel file ./hascat/hashcat.potfile

```
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~#  
root@kali:~# cat ./hascat/hashcat.potfile  
$6$JECGvTuo$Ls22MvW0oLJtNwcNajmTF6ndNaRqG1uLp1/6B3Ixtg25ltLN8Nw5i//nRg6J0oFgcd.S  
/tJ6cvB2zdmKIdLUV0:aaaf1g  
root@kali:~#
```

Infatti, se volete riprovare a ricrackare la password di utonto3, magari con un altro dizionario hashcat vi risponde che la password è già stata trovata.

```
root@kali:~# hashcat --force -m 1800 -a 0 hash.txt dizionariolettere.txt  
hashcat (pull/1273/head) starting...  
  
OpenCL Platform #1: The pocl project  
=====
```

\* Device #1: pthread-Intel(R) Core(TM) i7-4810MQ CPU @ 2.80GHz, 1498/1498 MB all  
ocatable, 1MCU

```
INFO: All hashes found in potfile! Use --show to display them.  
Started: Sat Mar 31 21:33:23 2018  
Stopped: Sat Mar 31 21:33:24 2018  
root@kali:~#
```

Pertanto occorre digitare il seguente comando:

```
hashcat -potfile-disable -force -m 1800 -a 0 hash.txt dizionariolettere.txt
```

oppure si può cancellare direttamente il potfile con il comando:

```
rm .hashcat/hashcat.potfile
```

**Il problema principale nel Brute Force è che se la password è corta, allora sarà trovata in un breve lasso di tempo.**

**Se la password è lunga, allora questo metodo impiegherà da qualche ora a svariate settimane, se non anni, per poter generare tutte le possibili combinazioni fino ad arrivare alla lunghezza desiderata.**

**Agli utenti, si consiglia di usare password più lunghe e complesse per scongiurare attacchi di questo tipo.**

Ora provate a crackare la password di utonto1 e utonto2

utilizzando i dizionari contenuti in /usr/share/wordlists/

Infine potete ora inserire nuovi utenti tramite i comandi:

```
useradd -m <utente>  
passwd <utente>
```

e provare a crackare la password con i dizionari trovati in rete

