

## Password Hashing and Cracking

Password is an important user authentication mechanism. Passwords are stored in the database as hashes rather than plain text. If users' passwords are stored in a database as plain text, malicious hackers can get immediate access to them if they break into the system. In this lab, you will understand what password hash is and how password hash be cracked.

### Password Hashing

A hash function turns any amount of data into a fixed-length “fingerprint” that cannot be reversed. The fixed-length “fingerprint” is called a hash value or hash code. The unique feature of hash functions is that any change in input data will result in a completely different hash code. Cryptographic hash functions are used to implement password hashing. There are multiple cryptographic hash functions, such as MD5, SHA-2.

For example, the following are the MD5 hash codes of different strings.

MD5(“hello”) = “5D41402ABC4B2A76B9719D911017C592”

MD5(“hollo”) = “181D1F65FC3EDFC75945B24F22CD7E22”

MD5(“helloo”) = “B373870B9139BBAD8E3396A49B1AFC9A”

```
victim@a8d078387d02:~$ echo -n "hello" | md5sum
5d41402abc4b2a76b9719d911017c592 -
victim@a8d078387d02:~$ echo -n "hollo" | md5sum
181d1f65fc3edfc75945b24f22cd7e22 -
victim@a8d078387d02:~$ echo -n "helloo" | md5sum
b373870b9139bbade83396a49b1afc9a -
```

### Task 1: Exploring Linux Password Hashes

The users' information and passwords in Linux and UNIX operating system are stored in files. /etc/passwd stores user accounts, and the /etc/shadow stores the information about user accounts and the encrypted password hashes. /etc/shadow has more restrictive permissions than the /etc/passwd file. On most Linux systems, only the root account has the ability to read the contents of the shadow file.

**Step 1: View the /etc/passwd file. Open a terminal and type the following command**

```
cat /etc/passwd
```

Each entry is the password information for each user (or user account) of the system. For example:

<b>root:</b>	<b>x:</b>	<b>0:</b>	<b>0:</b>	<b>root:</b>	<b>/root:</b>	<b>/bin/bash</b>
Username	password	User ID	Group ID	User ID info	Home directory	Command/shell

- **Username:** it should be between 1 and 32 characters in length.
- **Password:** an 'x' character indicates that encrypted password is stored in /etc/shadow file.
- **User ID (UID):** Each user must be assigned a user ID (UID). UID 0 (zero) is reserved for root and UIDs 1-99 are reserved for other predefined accounts. Further UID 100-999 are reserved by system for administrative and system accounts/groups.
- **Group ID (GID):** The primary group ID (stored in /etc/group file)
- **User ID Info:** The comment field. It allows you to add extra information about the users such as user's full name, phone number etc.
- **Home directory:** The absolute path to the directory the user will be in when they log in.
- **Command/shell:** The absolute path of a command or shell (/bin/bash).

Please take a screenshot of the result.

```
victim@a8d078387d02:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin)/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/:/nonexistent:/usr/sbin/nologin
messagebus:x:101:101:/:/nonexistent:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
victim:x:1000:1000:/:/home/victim:/bin/bash
```

**Step 2: View the permission of the passwd file using the command:**

```
ls -l /etc/passwd
```

The permissions are broken into 4 sections:

-	rw-	r--	r--
'-' indicates a file	Read, write and execute permissions for the owner of the file	Read, write and execute permissions for the members of the group owning the file	Read, write and execute permissions for other users
'd' indicates a directory			
'l' indicates a link			

Please take a screenshot of the results.

```
victim@a8d078387d02:~$ ls -l /etc/passwd
-rw-r--r-- 1 root root 1200 May 28 19:34 /etc/passwd
```

**Step 3: View the /etc/shadow file using the command:**

```
sudo cat /etc/shadow
```

Each entry is the hashed password for each user (or user account) of the system. As with the passwd file, each field in the shadow file is also separated with ":" colon characters, and are as follows:

```
root:$6$OCTu.M/v$fpnhbjkpA4S29IKZ2TzRsl6ArWyvu9elfWfC0H98t8OoLPokE8.d7q54
cynb0BTtLgN.lolE72npACz7Dr2p.:16983:0:99999:7:::
```

- **Username:** a direct match to the username in the /etc/passwd file.
- **Password:** encrypted password. A blank entry (eg. ::) indicates a password is not required to log in (usually a bad idea), and a '\*' entry (eg. :\*) indicates the account has been disabled. The "!" symbol (often called a bang) represents that fact the password has not been set. Usually password format is set to **\$id\$salt\$hash**. The \$id is the algorithm used On GNU/Linux as follows: **\$1\$** is MD5, **\$2a\$** is Blowfish, **\$2y\$** is Blowfish, **\$5\$** is SHA-256, **\$6\$** is SHA-512
- **Last change:** the number of days (since January 1, 1970) since the password was last changed.
- **Min:** the minimum number of days required between password changes. 0 indicates it may be changed at any time.
- **Max:** the number of days after which password must be changed. 99999 indicates user can keep his or her password unchanged for many, many years.
- **Warn:** the number of days to warn user of an expiring password (7 for a full week)
- **Inactive:** the number of days after password expires that account is disabled
- **Expire:** the number of days since January 1, 1970 that an account has been disabled
- **A reserved field for possible future use**

Please take a screenshot of the result.

```
victim@a8d078387d02:~$ sudo cat /etc/shadow
root:*:17962:0:99999:7:::
daemon:*:17962:0:99999:7:::
bin:*:17962:0:99999:7:::
sys:*:17962:0:99999:7:::
sync:*:17962:0:99999:7:::
games:*:17962:0:99999:7:::
man:*:17962:0:99999:7:::
lp:*:17962:0:99999:7:::
mail:*:17962:0:99999:7:::
news:*:17962:0:99999:7:::
uucp:*:17962:0:99999:7:::
proxy:*:17962:0:99999:7:::
www-data:*:17962:0:99999:7:::
backup:*:17962:0:99999:7:::
list:*:17962:0:99999:7:::
irc:*:17962:0:99999:7:::
gnats:*:17962:0:99999:7:::
nobody:*:17962:0:99999:7:::
_apt:*:17962:0:99999:7:::
messagebus:*:17975:0:99999:7:::
systemd-network:*:17975:0:99999:7:::
systemd-resolve:*:17975:0:99999:7:::
victim:$6$/WJBjJhx$Xhv6wpe9QgHUCFCblookeeHr4gmLoxEGu4daPhsG.TlKEDGSi76FcPsYnsQxJbTlKS5nu3tURFZQ0wyRbGT5M1:18044:0:99999:7:::
```

**Step 4: Create a new user named alice using the command:**

```
sudo useradd alice
```

**Step 5: Create a new user named bob using the command:**

```
sudo useradd bob
```

```
victim@a8d078387d02:~$ sudo useradd alice
victim@a8d078387d02:~$ sudo useradd bob
```

**Step 6: Now, view the changes made to the passwd file:**

```
cat /etc/passwd
```

Please take a screenshot of the result.

```
victim@a8d078387d02:~$ cat /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534:/:/nonexistent:/usr/sbin/nologin
messagebus:x:101:101:/:/nonexistent:/usr/sbin/nologin
systemd-network:x:102:103:systemd Network Management,,,:/run/systemd/netif:/usr/sbin/nologin
systemd-resolve:x:103:104:systemd Resolver,,,:/run/systemd/resolve:/usr/sbin/nologin
victim:x:1000:1000:/:/home/victim:/bin/bash
alice:x:1001:1001:/:/home/alice:/bin/sh
bob:x:1002:1002:/:/home/bob:/bin/sh
```

**Step 7: Set alice's password to apw123 using the following command:**

```
sudo passwd alice
Enter new UNIX password: (type in apw123)
Retype new UNIX password: (type in apw123)
```

**Step 8: Set bob's password to bpw123 using the command:**

```
sudo passwd bob
Enter new UNIX password: (type in bpw123)
Retype new UNIX password: (type in bpw123)
```

```
victim@a8d078387d02:~$ sudo passwd alice
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
victim@a8d078387d02:~$ sudo passwd bob
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

**Step 9: Examine the alterations to the shadow file by typing the following:**

```
cat tail /etc/shadow
```

Please take a screenshot of the result.

```
victim@a0d078387d02:~$ sudo cat tail /etc/shadow
cat: tail: No such file or directory
root:*:17962:0:99999:7:::
daemon:*:17962:0:99999:7:::
bin:*:17962:0:99999:7:::
sys:*:17962:0:99999:7:::
sync:*:17962:0:99999:7:::
games:*:17962:0:99999:7:::
man:*:17962:0:99999:7:::
lp:*:17962:0:99999:7:::
mail:*:17962:0:99999:7:::
news:*:17962:0:99999:7:::
uucp:*:17962:0:99999:7:::
proxy:*:17962:0:99999:7:::
www-data:*:17962:0:99999:7:::
backup:*:17962:0:99999:7:::
list:*:17962:0:99999:7:::
irc:*:17962:0:99999:7:::
gnats:*:17962:0:99999:7:::
nobody:*:17962:0:99999:7:::
_apt:*:17962:0:99999:7:::
messagebus:*:17975:0:99999:7:::
systemd-network:*:17975:0:99999:7:::
systemd-resolve:*:17975:0:99999:7:::
victim:$6$WJByJhx$Xhv6wpe9QgHUCFCblookeeHr4gmLoxEGu4daPhsG.TlKEDGSi76FcPsYnsQxJbTLKS5nu3tURFZQ0wyRbGT5M1:18044:0:99999:7:::
alice:$6$AA166nC9$gmpCdaRiCzcL1FVmVvAWJ0BoWgg8HQqn0gjBktjJcgjM5ySbSUR.cN8bQ8mDPErjTpJR.McqnyxN3rLUdMTbE0:18044:0:99999:7:::
bob:$6$0qXcYh4x$ExCiqySn3mXgVQcuX3gMmwLUp/SjX7yfe7Ufj/nSVbjATBwpoIjL020QkGzS5Fn.WGxcRBfC2utLKVu.3U3bd/:18044:0:99999:7:::
```

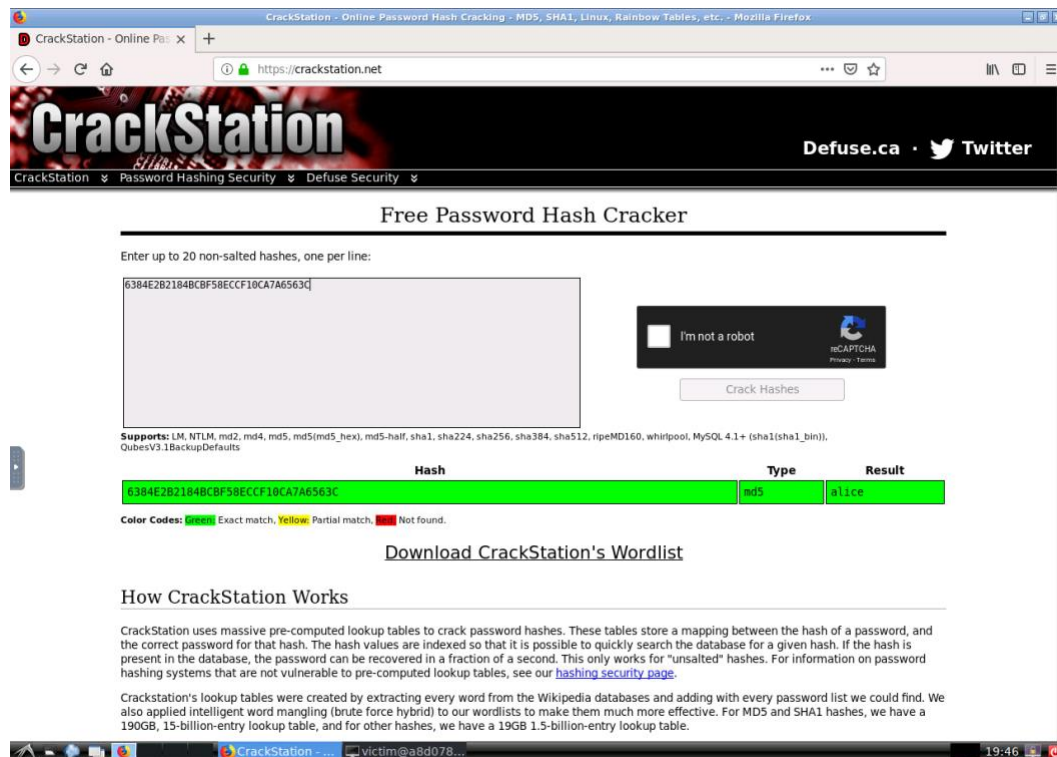
## Cracking Hashes and Rainbow Table

Although the hashing algorithms cannot be reversed, password hashes could be cracked. Hackers can generate hashes from a dictionary of strings that are commonly used as passwords. If hackers gain access to a database of hashed passwords, they can calculate the hash code for each string in the database and match it with the current hash code. If one in the database matches, the plaintext password of that hash is known. This is so-called brute force dictionary attack.

The brute force cracking described above is very time-consuming for calculating the hash code for every string in the database. The opposite way is to pre-calculate all the hash codes for the strings in the database and store the mapping of the hash codes to the strings. Then, hackers just need to look up a hash in the mapping table to find the password. However, this method requires too much space considering the large volume of strings that could be used as passwords. Considering the time-memory tradeoff, “rainbow table” is a better method that takes a place in between. It is a pre-commutated lookup table, but sacrifice hash cracking speed to make the lookup tables smaller.

## Task 2: Simple Password Cracking

There are some online cracking tools. For example, CrackStation (<https://crackstation.net/>) is a online websites for cracking simple password hashes.



Please use the CrackStation to crack the following password hashes:

**Hash 1: 6384E2B2184BCBF58ECF10CA7A6563C**

What is the password? What hash algorithm is used?

### Algorithm: md5

Result: alice



**Hash 2:**

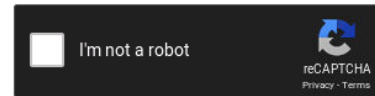
**4E40E8FFE0EE32FA53E139147ED559229A5930F89C2204706FC174BEB36210B3**

What is the password? What hash algorithm is used?

Algorithm: sha256

Result: alice123

Enter up to 20 non-salted hashes, one per line:



Crack Hashes

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
4E40E8FFE0EE32FA53E139147ED559229A5930F89C2204706FC174BEB36210B3	sha256	alice123

### Hash 3: 5994F091C5CBC05EE2DF38DA2C54EA5BE663D54E

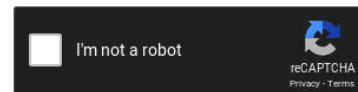
What is the password? What hash algorithm is used?

Algorithm: sha1

Result: aliceindiana

### Free Password Hash Cracker

Enter up to 20 non-salted hashes, one per line:



Crack Hashes

**Supports:** LM, NTLM, md2, md4, md5, md5(md5\_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1 sha1\_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
5994F091C5CBC05EE2DF38DA2C54EA5BE663D54E	sha1	aliceindiana

**Color Codes:** Green Exact match, Yellow Partial match, Red Not found.

## Salt

Rainbow tables rely on the assumption that each password is hashed in the exact the same way. To defeat rainbow tables, “salt” is invented to randomize the hash for each password. Using salt, when the same password is hashed twice, two different hash codes will be generated. Salt is usually stored together with hash code in the user account file.



## Task 3: Using John the Ripper for Password Cracking

John the Ripper is one of the well-known fast password cracking tool that can crack passwords through a dictionary attack or through the use of brute force. It can be downloaded free at [www.openwall.com/john/](http://www.openwall.com/john/).

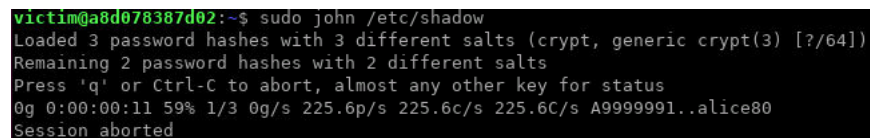
**Step 1: Install John the Ripper using the following command:**

```
sudo apt-get install john
```

**Step 2: Type the following command to attempt to crack the passwords with john:**

```
sudo john /etc/shadow
```

Please take a screenshot of the result.



```
victim@a8d078387d02:~$ sudo john /etc/shadow
Loaded 3 password hashes with 3 different salts (crypt, generic crypt(3) [?/64])
Remaining 2 password hashes with 2 different salts
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:11 59% 1/3 0g/s 225.6p/s 225.6c/s 225.6C/s A9999991..alice80
Session aborted
```

Password.lst did not contain the password *victim*, *apw123*, *bpw123* therefore, brute force was not successful.

**Step 3: Can you crack the following passwords?**

emiller:3e05v.ztZ8LNE:15652:0:99999:7:::

tanderson:\$1\$AqW8SRi1\$Dd0m3hFyOI276/IHinecr0:15652:0:99999:7:::

awilliams:mQK2Y4hWq0SvY:15652:0:99999:7:::

mdavis:\$5\$i3uY6Gfp\$ywsyCNRs7kbKbN7Ad0SnGR7P6bVmMQ8iJ7008mrGHC:15652:0:99999:7:::

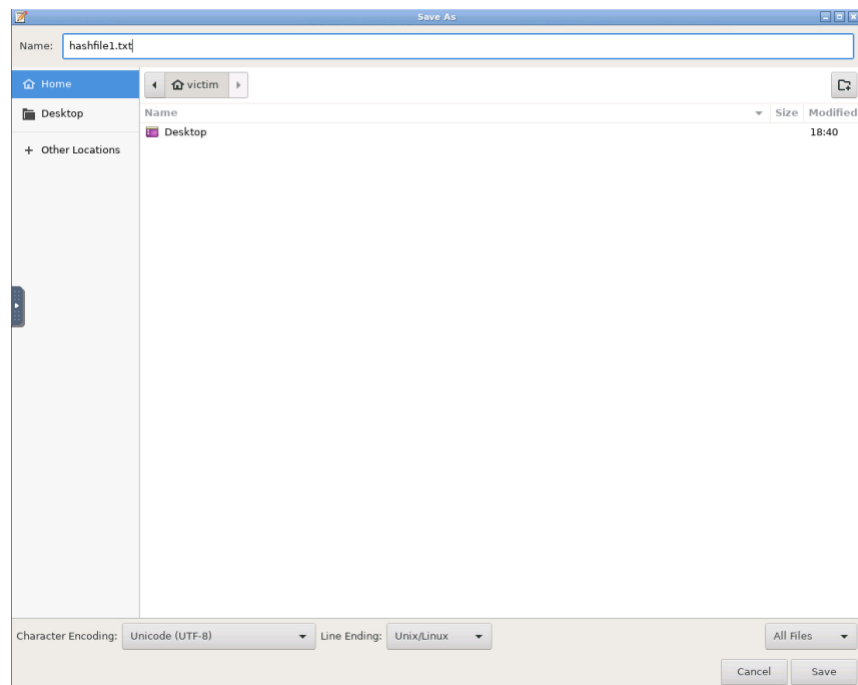
djameson:\$6\$iimflwnL\$T/0zG89BxF.qKzMyX7BZJCSye5x7wIQxox5dMMwWPdvpzFMOs2YkknqHdMbbdxyBN7NNNBnAh/d7YY2fRRV3k0:15652:0:99999:7:::

**Open a terminal and type**

```
gedit
```

**Paste the above information in the text editor and click the Save button in the menu to save**

it to Desktop as “hashfile1.txt”



**Step 4: Open a terminal, go to the desktop directory, and use john to crack the passwords**

```
sudo john hashfile1.txt
```

```
victim@a8d078387d02:~$ sudo john hashfile1.txt  
Warning: only loading hashes of type "descrypt", but also saw type "md5crypt"  
Use the "--format=md5crypt" option to force loading hashes of that type instead  
Warning: only loading hashes of type "descrypt", but also saw type "crypt"  
Use the "--format=crypt" option to force loading hashes of that type instead  
Loaded 2 password hashes with 2 different salts (descrypt, traditional crypt(3) [DES 128/128 SSE2-16])  
Press 'q' or Ctrl-C to abort, almost any other key for status  
qwerty (awilliams)  
iloveyou (emiller)  
2g 0:00:00.00 100% 2/3 14.28g/s 18900p/s 19814c/s 19814C/s 123456..marley  
Use the "--show" option to display all of the cracked passwords reliably  
Session completed
```

**Tip: After running john, you can check out the cracked password at any time using the command:**

```
sudo john --show hashfile1.txt
```

```
victim@a8d078387d02:~$ sudo john --show hashfile1.txt
emiller:iloveyou:15652:0:99999:7:::
awilliams:qwerty:15652:0:99999:7:::
```

**Step 5: Are there any remaining password to be cracked? You may get the following warnings:**

Warning: only loading hashes of type "des", but also saw type "md5"

Use the "--format=md5" option to force loading hashes of that type instead

Warning: only loading hashes of type "des", but also saw type "crypt"

Use the "--format=crypt" option to force loading hashes of that type instead

**It suggests us to specify the hash format.**

**We can use the following commands to specify the format option and run john again:**

```
sudo john --format=md5 --wordlist=/usr/share/john/password.lst hashfile1.txt
```

**(if the password.lst can not be found, you can use the command locate password.lst to get the correct directory)**

**You can use the command to check the cracked passwords at this moment:**

```
sudo john --show hashfile1.txt
```

**Step 6: Use the following commands to specify the format option "crypt" and run john again:**

```
sudo john --format=crypt hashfile1.txt
```

```
victim@a8d078387d02:~$ sudo john --format=crypt hashfile1.txt
Loaded 5 password hashes with 5 different salts (crypt, generic crypt(3) [?/64])
Remaining 2 password hashes with 2 different salts
Press 'q' or Ctrl-C to abort, almost any other key for status
whatever      (djameson)
blink182      (mdavis)
2g 0:00:00:36 100% 2/3 0.05408g/s 217.9p/s 223.1c/s 223.1C/s pretty..celtic
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

```
victim@a8d078387d02:~$ sudo john --format=md5crypt hashfile1.txt
Loaded 1 password hash (md5crypt [MD5 32/64 X2])
Press 'q' or Ctrl-C to abort, almost any other key for status
password          (tanderson)
lg 0:00:00:00 100% 2/3 2.702g/s 6918p/s 6918c/s 6918C/s password..password1
Use the "--show" option to display all of the cracked passwords reliably
Session completed
```

**You can use the command to check the cracked passwords again:**

```
sudo john --show hashfile1.txt
```

```
victim@a8d078387d02:~$ sudo john --show hashfile1.txt
emiller:iloveyou:15652:0:99999:7:::
tanderson:password:15652:0:99999:7:::
awilliams:qwerty:15652:0:99999:7:::
mdavis:blink182:15652:0:99999:7:::
djameson:whatever:15652:0:99999:7:::
5 password hashes cracked, 0 left
```

What are the passwords cracked? Please take a screenshot of the result.

## Post-Task Questions

Question 1 (4 points): What file in Linux store the names of the user accounts? What file stores the users' password?

Question 2 (3 points): We observed that salts are stored together with hash codes, and are not kept as secrets. Why?

Question 3 (3 points): Is there any risk if the same salt is reused for several hashes on a user?