Assignment 5

Task 1: Created 6 users with passwords meeting requirements.
User 1 password: four
User 2 password: 1234
User 3 password: four1234
User 4 password: four1234!@
User 5 password: five12345
User 6 password: f0uR1@3$

```
┌──(student㉿kali.example.com)-[~]
└─$ sudo useradd user1

┌──(student㉿kali.example.com)-[~]
└─$ sudo passwd user1
New password:
Retype new password:
passwd: password updated successfully

┌──(student㉿kali.example.com)-[~]
└─$ sudo useradd user2

┌──(student㉿kali.example.com)-[~]
└─$ sudo passwd user2
New password:
Retype new password:
passwd: password updated successfully

┌──(student㉿kali.example.com)-[~]
└─$ sudo useradd user3

┌──(student㉿kali.example.com)-[~]
└─$ sudo passwd user3
New password:
Retype new password:
passwd: password updated successfully

┌──(student㉿kali.example.com)-[~]
└─$ sudo useradd user4

┌──(student㉿kali.example.com)-[~]
└─$ sudo passwd user4
New password:
Retype new password:
passwd: password updated successfully

┌──(student㉿kali.example.com)-[~]
└─$ sudo useradd user5

┌──(student㉿kali.example.com)-[~]
└─$ sudo passwd user5
New password:
Retype new password:
passwd: password updated successfully

┌──(student㉿kali.example.com)-[~]
└─$ sudo useradd user6

┌──(student㉿kali.example.com)-[~]
└─$ sudo passwd user6
New password:
Retype new password:
passwd: password updated successfully

┌──(student㉿kali.example.com)-[~]
└─$
```

Task 2: Exported users hashes to a file named jsamo006.hash.

```
┌──(student㉿kali.example.com)-[~]
└─$ sudo cat shadow > jsamo006.hash
```

Task 3: Use John the ripper and let it run for 10 minutes.

```
┌──(student㉿kali.example.com)-[~]
└─$ sudo jogn --format=crypt jsamo006.hash --wordlist=/home/student/rockyou.txt
sudo: jogn: command not found

┌──(student㉿kali.example.com)-[~]
└─$ sudo john --format=crypt jsamo006.hash --wordlist=/home/student/rockyou.txt
Using default input encoding: UTF-8
Loaded 14 password hashes with 14 different salts (crypt, generic crypt(3) [?/64])
Remaining 10 password hashes with 10 different salts
Cost 1 (algorithm [1:descrypt 2:md5crypt 3:sunmd5 4:bcrypt 5:sha256crypt 6:sha512crypt]) is 0 for all loaded hashes
Cost 2 (algorithm specific iterations) is 1 for all loaded hashes
Will run 2 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
1234             (user2)
```

It only cracked the password: 1234 in 10 minutes. Which surprised me. I thought that it would be able to crack the simpler passwords like user1 and even users 3 and 5 pretty easily.