Secure System Design: Threats and Countermeasures
CS392
Assignment 2

Name: Chandrawanshi Mangesh Shivaji
Roll No.: 1801cs16
Filename: assign2.pdf
Date: 11 April 2021

Submitted Files and their description in short

Makefile => to compile all .c source codes into executables at once

.c Source code files

initialize.c => to initialize F1.txt, F2.txt, F3.txt with honeypot accounts
registration.c => to register a new user
regenerate_honeyindexsets.c => to regenerate honeyindexsets
login.c => to login into user account
honeychecker.c => to verify user login process

generated files

F1.txt => contains (Username, Honeyindexset) pairs for all users
F2.txt => contains (Sugarindex, Hash(Password)) for all users
F3.txt => contains (Username, Sugarindex) for all users

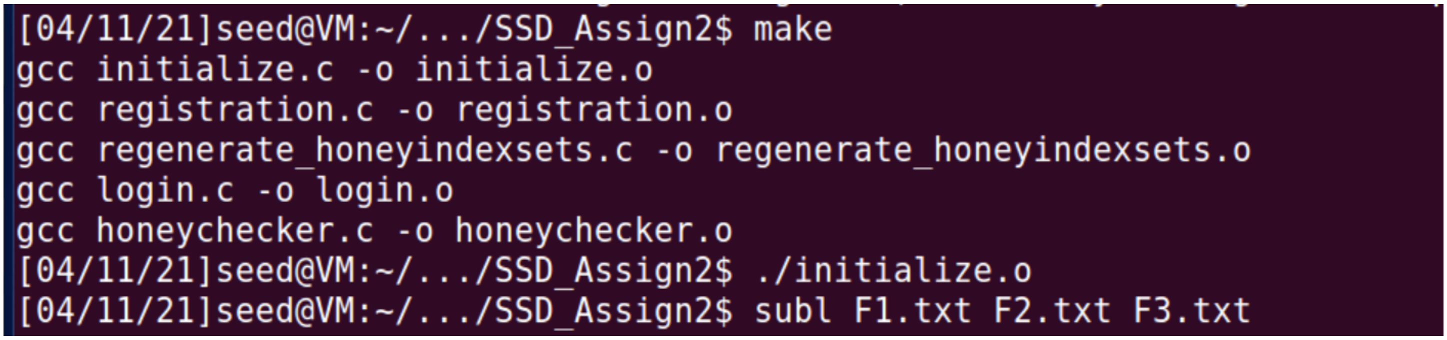
Other attached files:

passwordFile => contains 10 md5 hashes of new users added by me into created system, generated manually can be used for analysis using john the ripper tool.
Wordlist File => rockyou.txt, size is too large to upload (133 MB)

1.1 Implement a Honeyword Based System

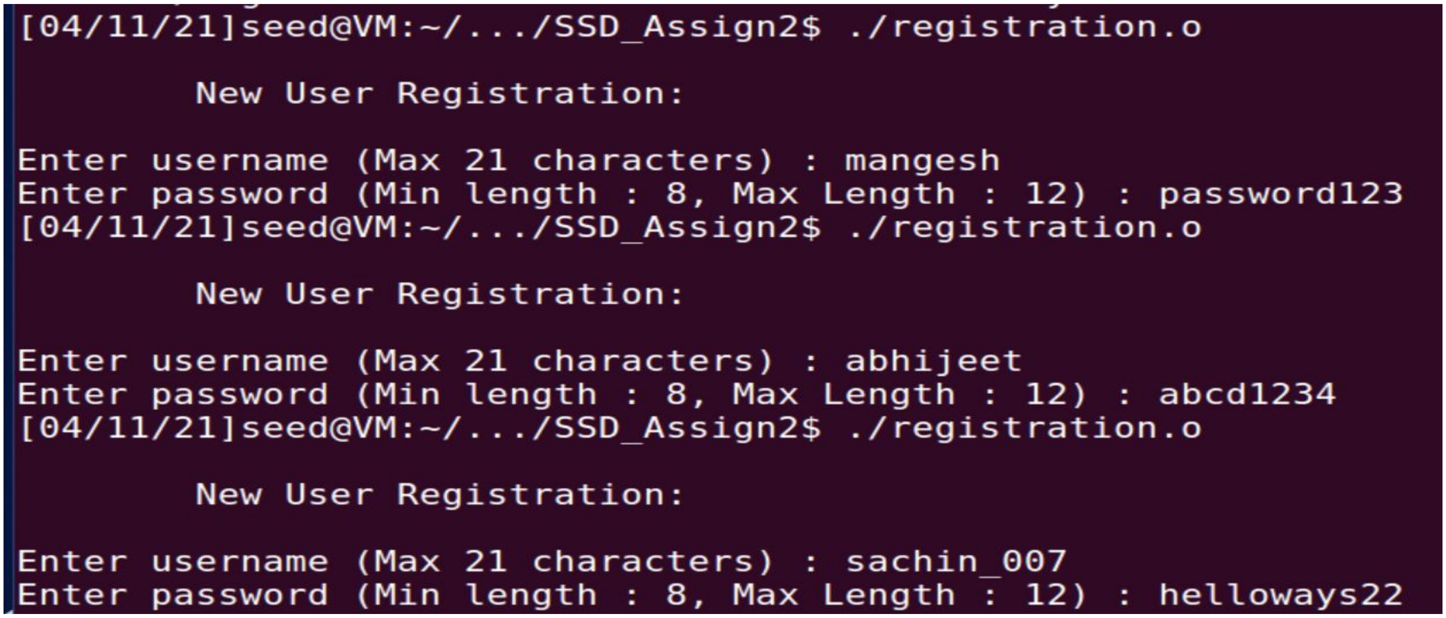
make: generate all executables

./initialize.o : initialize honeypot accounts

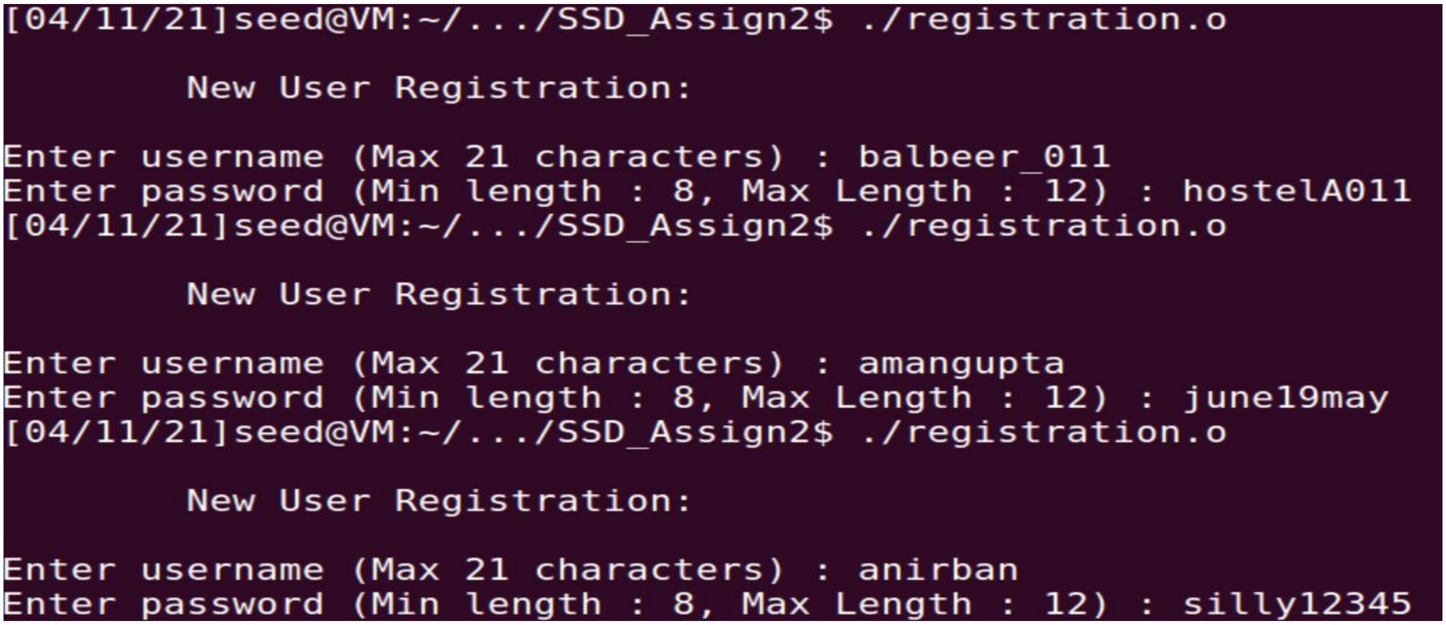


```
[04/11/21]seed@VM:~/.../SSD_Assign2$ make
gcc initialize.c -o initialize.o
gcc registration.c -o registration.o
gcc regenerate_honeyindexsets.c -o regenerate_honeyindexsets.o
gcc login.c -o login.o
gcc honeychecker.c -o honeychecker.o
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./initialize.o
[04/11/21]seed@VM:~/.../SSD_Assign2$ subl F1.txt F2.txt F3.txt
```

Register Users on system : ./registration.o

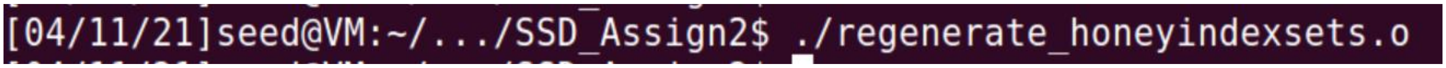


```
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./registration.o

        New User Registration:

Enter username (Max 21 characters) : mangesh
Enter password (Min length : 8, Max Length : 12) : password123
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./registration.o

        New User Registration:

Enter username (Max 21 characters) : abhijeet
Enter password (Min length : 8, Max Length : 12) : abcd1234
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./registration.o

        New User Registration:

Enter username (Max 21 characters) : sachin_007
Enter password (Min length : 8, Max Length : 12) : helloways22
```



```
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./registration.o

        New User Registration:

Enter username (Max 21 characters) : balbeer_011
Enter password (Min length : 8, Max Length : 12) : hostelA011
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./registration.o

        New User Registration:

Enter username (Max 21 characters) : amangupta
Enter password (Min length : 8, Max Length : 12) : june19may
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./registration.o

        New User Registration:

Enter username (Max 21 characters) : anirban
Enter password (Min length : 8, Max Length : 12) : silly12345
```

Regenerate honeyindexsets: this will ensure sugarindex of newly added users is not easily distinguishable (./regenerate_honeyindexsets.o)



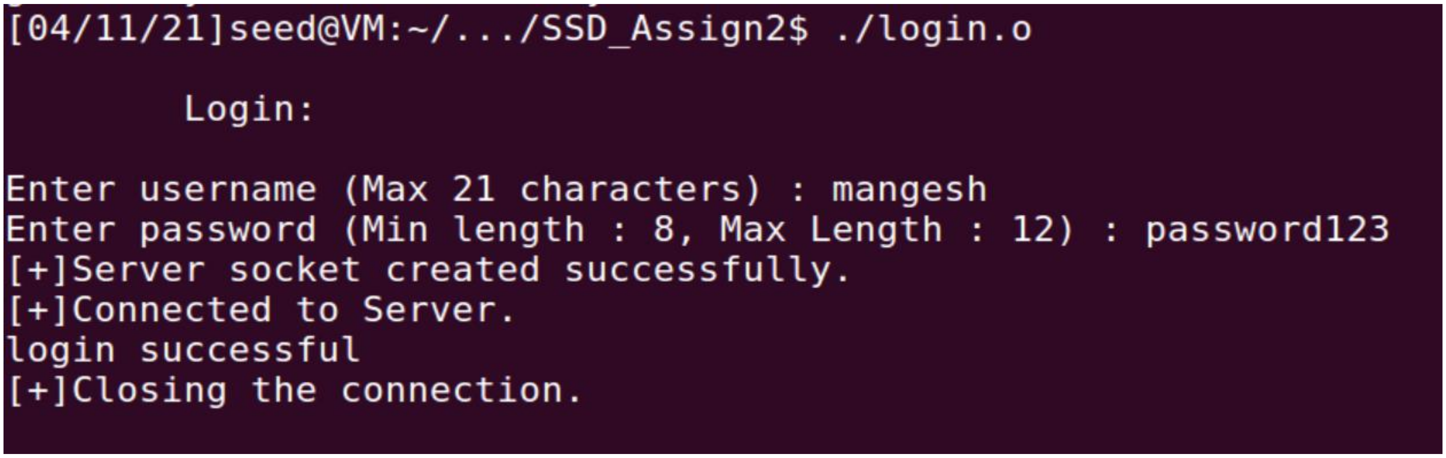
```
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./regenerate_honeyindexsets.o
```

Login into system along with verification by honeychecker

- First run ./honeychecker.o server and then ./login.o
- TCP connection will be established between them
- honeychecker receives (username, match_index) pair from main server
- honeychecker will respond accordingly to main server
- reponse from honeychecker printed on main server side

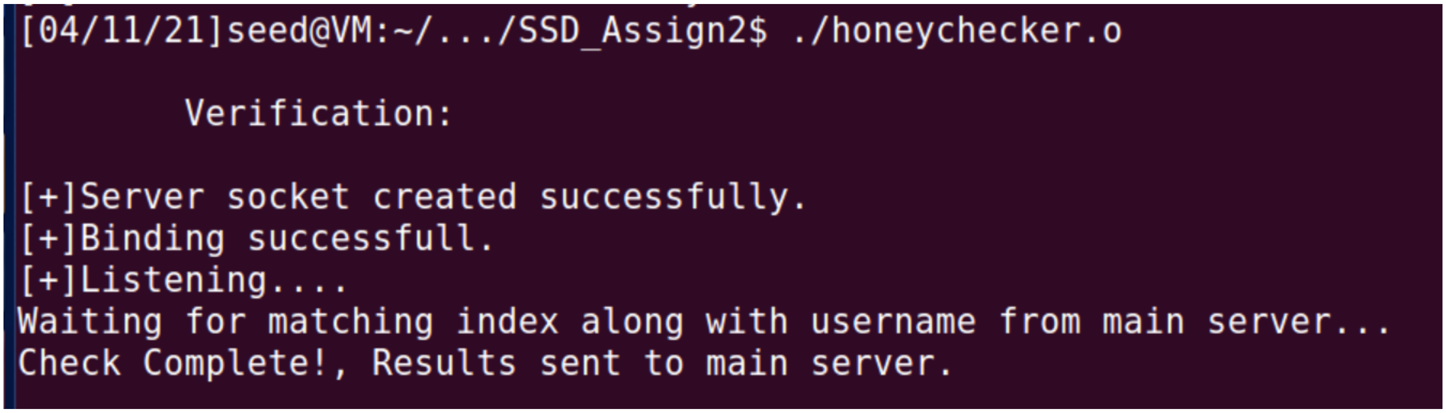
Successful Login Example

./login.o side screenshot



```
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./login.o

        Login:

Enter username (Max 21 characters) : mangesh
Enter password (Min length : 8, Max Length : 12) : password123
[+]Server socket created successfully.
[+]Connected to Server.
login successful
[+]Closing the connection.
```

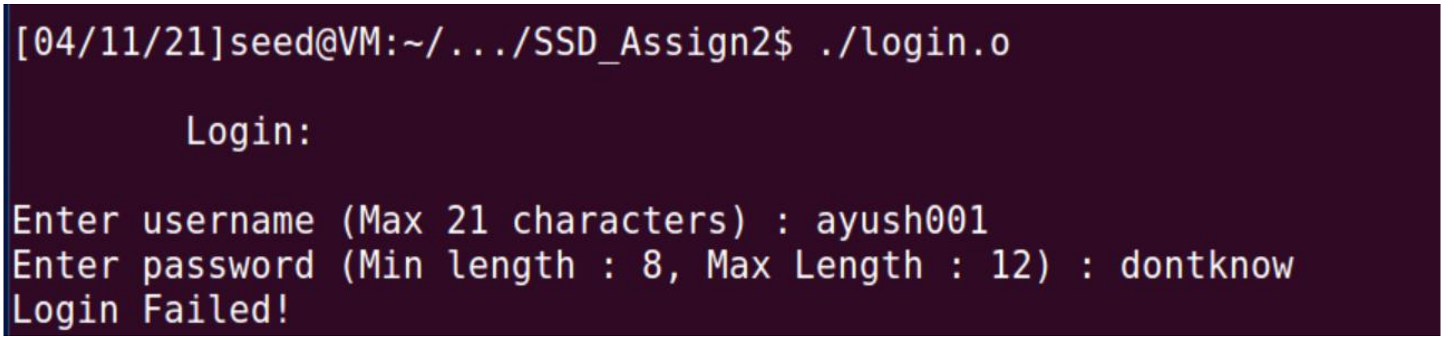
./honeychecker.o side screenshot



```
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./honeychecker.o

        Verification:

[+]Server socket created successfully.
[+]Binding successfull.
[+]Listening....
Waiting for matching index along with username from main server...
Check Complete!, Results sent to main server.
```

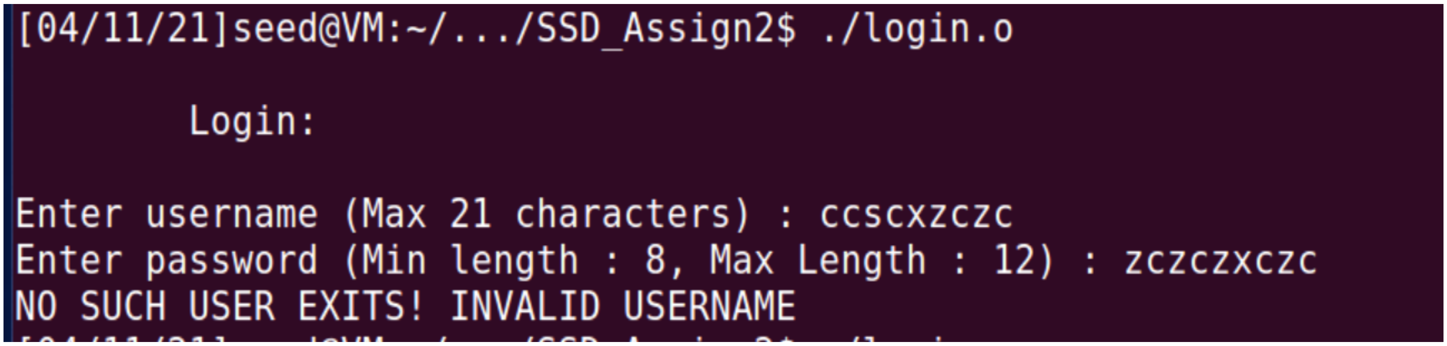
Failed Login Examples (No honeychecker required as no match index is found)

Due to wrong password



```
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./login.o

        Login:

Enter username (Max 21 characters) : ayush001
Enter password (Min length : 8, Max Length : 12) : dontknow
Login Failed!
```

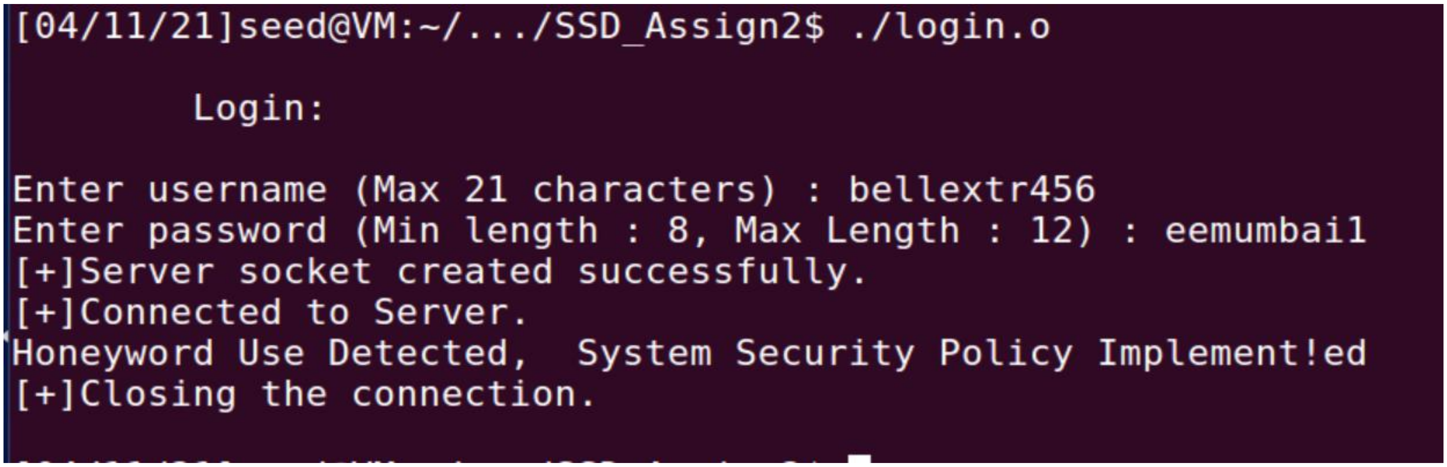
Due to wrong username



```
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./login.o

        Login:

Enter username (Max 21 characters) : ccscxzczc
Enter password (Min length : 8, Max Length : 12) : zczczxczc
NO SUCH USER EXITS! INVALID USERNAME
```

PASSWORD FILE COMPROMISE EXAMPLE (HONEYWORD)

./login.o side screenshot



```
[04/11/21]seed@VM:~/.../SSD_Assign2$ ./login.o

        Login:

Enter username (Max 21 characters) : bellextr456
Enter password (Min length : 8, Max Length : 12) : eemumbai1
[+]Server socket created successfully.
[+]Connected to Server.
Honeyword Use Detected,  System Security Policy Implement!ed
[+]Closing the connection.
```

In this case, entered password matches with one of the index's hash value in F2.txt but that index is honeyindex not sugarindex, this results in scenario similar to password file compromise. System Admin should execute some security policy in this case. (Note: ./honeychecker side will be same as successful login, only message sent to main server will differ, which is shown in above screenshot)

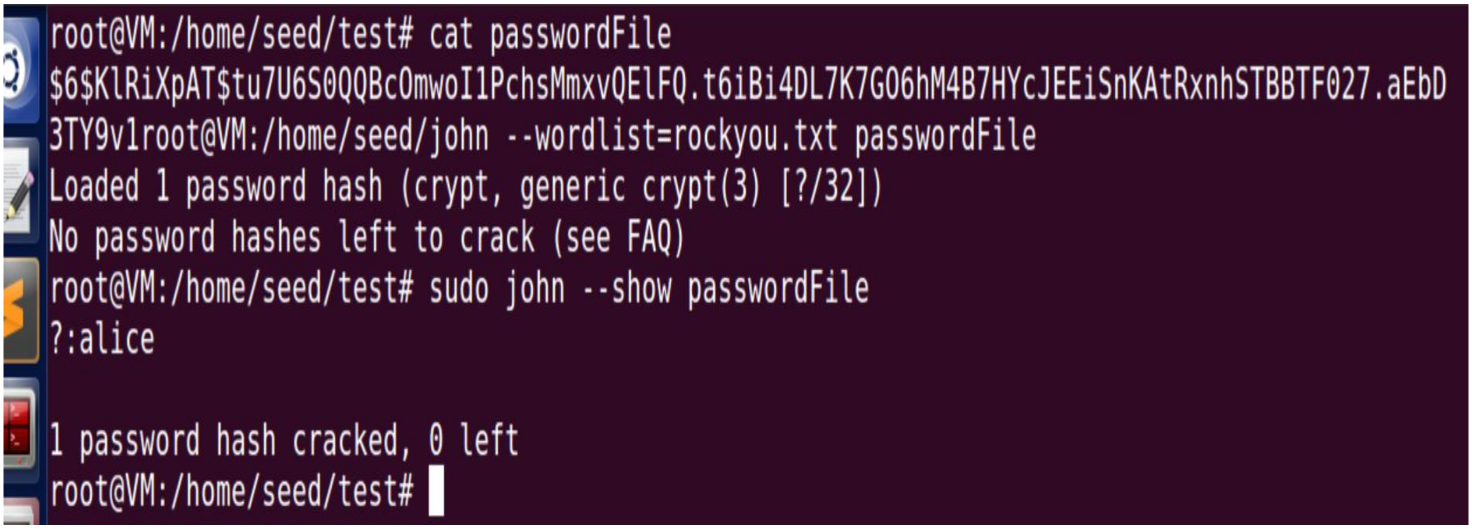
1.2 Use of Password Cracking Tool

Result on following instructions in assignment related to user alice

Username : alice

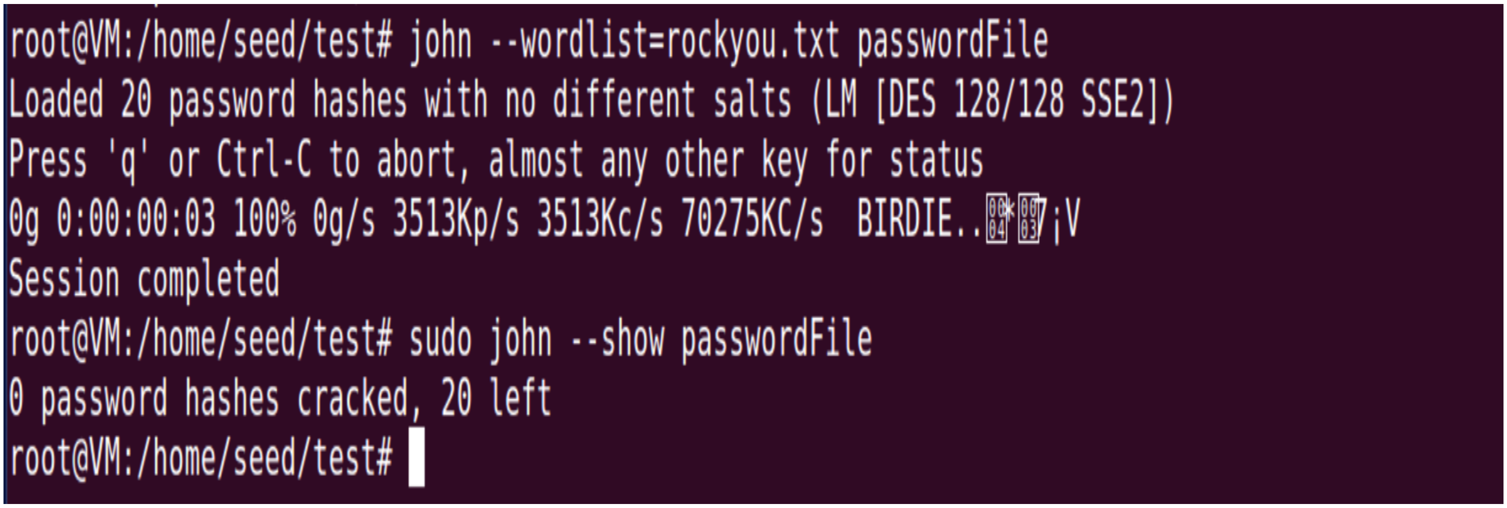
Password alice

John the ripper tool successfully cracks alice with rockyou.txt wordlist file



```
root@VM:/home/seed/test# cat passwordFile
$6$KlRiXpAT$tu7U6S0QQBcOmwoI1PchsMmxvQElFQ.t6iBi4DL7K7G06hM4B7HYcJEEiSnKAtRxnhSTBBTF027.aEbD
3TY9v1root@VM:/home/seed/john --wordlist=rockyou.txt passwordFile
Loaded 1 password hash (crypt, generic crypt(3) [?/32])
No password hashes left to crack (see FAQ)
root@VM:/home/seed/test# sudo john --show passwordFile
?:alice

1 password hash cracked, 0 left
root@VM:/home/seed/test# 
```

Result on passwordFile containing md5 hashes of newly added users in created system



```
root@VM:/home/seed/test# john --wordlist=rockyou.txt passwordFile
Loaded 20 password hashes with no different salts (LM [DES 128/128 SSE2])
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:03 100% 0g/s 3513Kp/s 3513Kc/s 70275KC/s  BIRDIE..␄*␃7¡V
Session completed
root@VM:/home/seed/test# sudo john --show passwordFile
0 password hashes cracked, 20 left
root@VM:/home/seed/test# 
```

John the ripper tool not able to crack hashes of given passwordFile with rockyou.txt wordlist file