PROJECT 3 REPORT

CS4371

Instructor: Dr. Randy Klepetko

May 2, 2023

PROJECT 3

BY

Lauren Taylor

Dillon Hughes

Stanley Nwajiaku

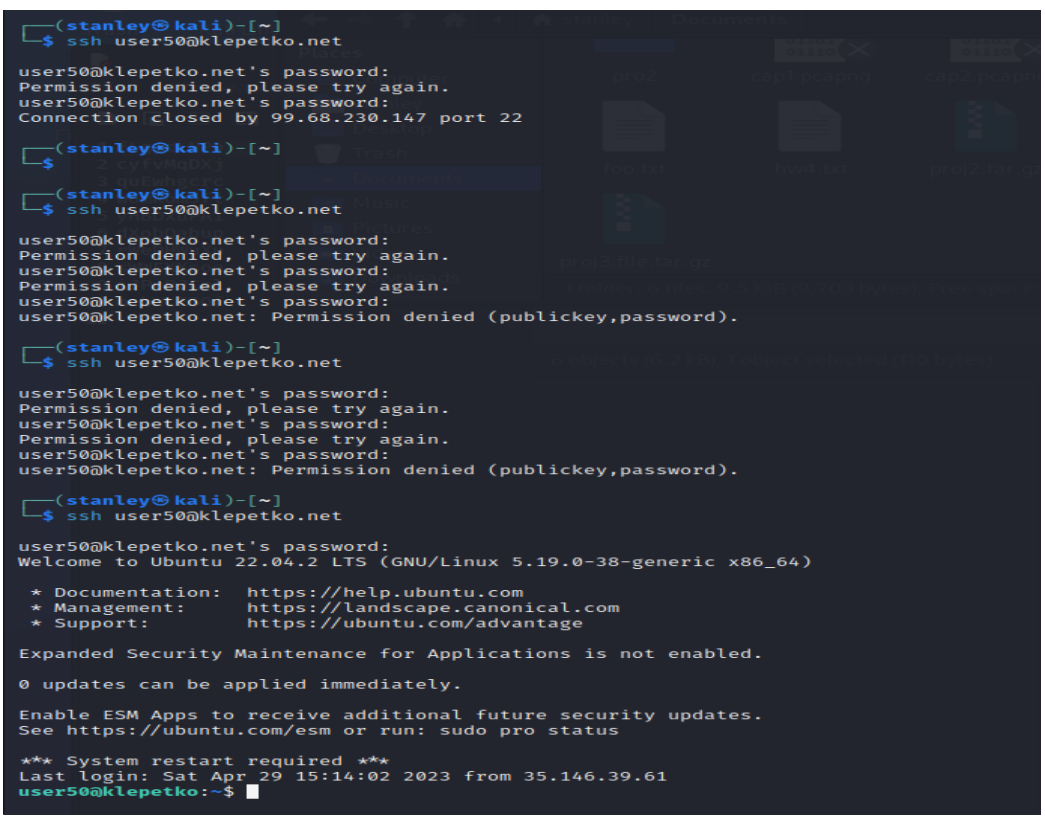
Section I (Introduction):

The protection of digital systems and data is highly dependent on cryptographic algorithms and keys. The objective of this project is to learn about the use and implementation of cryptographic protocols involving passwords and keys. Additionally, this project aims to show many methods of cracking passwords and keys, as well as developing and utilizing security tools. The project is divided into four tasks, each focused on different aspects of password and key cracking. Task one involves creating a program to crack a password using a dictionary attack. Task two utilizes the Metasploit Framework to crack an unknown username and password combination. Task three involves cryptanalysis cracking of an encrypted PDF file using a flawed encryption algorithm. Finally, task four requires the use of the brute force method to crack the key of an encrypted PDF file using the DES-ECB encryption algorithm.

Each group member communicated through discord. We worked around each of our schedules to meet and solve the tasks. Each group member did not have specific tasks because the sandbox is set up on Stanley's machine but we easily coordinated times to meet through discord and we were able to work through the project.

Section II (Task I):

a) Show the screenshot of your program in A.2 when you are testing each password and obtaining the password to ssh klepetko.net as “user50”.



```
┌──(stanley㉿kali)-[~]
└─$ ssh user50@klepetko.net

user50@klepetko.net's password:
Permission denied, please try again.
user50@klepetko.net's password:
Connection closed by 99.68.230.147 port 22

┌──(stanley㉿kali)-[~]
└─$

┌──(stanley㉿kali)-[~]
└─$ ssh user50@klepetko.net

user50@klepetko.net's password:
Permission denied, please try again.
user50@klepetko.net's password:
Permission denied, please try again.
user50@klepetko.net's password:
user50@klepetko.net: Permission denied (publickey,password).

┌──(stanley㉿kali)-[~]
└─$ ssh user50@klepetko.net

user50@klepetko.net's password:
Permission denied, please try again.
user50@klepetko.net's password:
Permission denied, please try again.
user50@klepetko.net's password:
user50@klepetko.net: Permission denied (publickey,password).

┌──(stanley㉿kali)-[~]
└─$ ssh user50@klepetko.net

user50@klepetko.net's password:
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.19.0-38-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

*** System restart required ***
Last login: Sat Apr 29 15:14:02 2023 from 35.146.39.61
user50@klepetko:~$
```

b) Report how long it takes to test each password on average.

- 4 minutes / 10 passwords = 0.4 minutes or 24 seconds.

c) If the dictionary has 1 million passwords, estimate how long it will take to find the password with your program.

- 1,000,000 passwords * 0.4 minutes = 400000 minutes or 6666.67 hours.

Section III (Task II):

For cracking "user50" to klepetko.net,

a) Show the screen shot of the parameters of the ssh login module. Use the "info" command in the MSF console console.

```
msf6 auxiliary(scanner/ssh/ssh_login) > info

       Name: SSH Login Check Scanner
     Module: auxiliary/scanner/ssh/ssh_login
    License: Metasploit Framework License (BSD)
       Rank: Normal

Provided by:
  todb <todb@metasploit.com>

Check supported:
  No

Basic options:
  Name                 Current Setting  Required  Description
  ----                 ---------------  --------  -----------
  BLANK_PASSWORDS      false            no        Try blank passwords for all use
                                                  rs
  BRUTEFORCE_SPEED     5                yes       How fast to bruteforce, from 0
                                                  to 5
  DB_ALL_CREDS         false            no        Try each user/password couple s
                                                  tored in the current database
  DB_ALL_PASS          false            no        Add all passwords in the curren
                                                  t database to the list
  DB_ALL_USERS         false            no        Add all users in the current da
                                                  tabase to the list
  DB_SKIP_EXISTING     none             no        Skip existing credentials store
                                                  d in the current database (Acce
                                                  pted: none, user, user&realm)
  PASSWORD                              no        A specific password to authenti
                                                  cate with
  PASS_FILE            dictionary.txt   no        File containing passwords, one
                                                  per line
  RHOSTS               klepetko.net     yes       The target host(s), see https:/
                                                  /github.com/rapid7/metasploit-f
                                                  ramework/wiki/Using-Metasploit
  RPORT                22               yes       The target port
  STOP_ON_SUCCESS      true             yes       Stop guessing when a credential
                                                   works for a host
  THREADS              1                yes       The number of concurrent thread
                                                  s (max one per host)
  USERNAME             user50           no        A specific username to authenti
                                                  cate as
  USERPASS_FILE                         no        File containing users and passw
                                                  ords separated by space, one pa
                                                  ir per line
  USER_AS_PASS         false            no        Try the username as the passwor
                                                  d for all users
  USER_FILE                             no        File containing usernames, one
                                                  per line
  VERBOSE              true             yes       Whether to print output for all
                                                   attempts
```

b) Show the screenshot of finding the correct password in the MSF console.

```
msf6 auxiliary(scanner/ssh/ssh_login) > run

[*] 99.68.230.147:22 - Starting bruteforce
[-] 99.68.230.147:22 - Failed: 'user50:fLqjcLNPo'
[!] No active DB -- Credential data will not be saved!
[-] 99.68.230.147:22 - Failed: 'user50:cyfvMqDXj'
[-] 99.68.230.147:22 - Failed: 'user50:quEwhgcrc'
[-] 99.68.230.147:22 - Failed: 'user50:womRomJft'
[-] 99.68.230.147:22 - Failed: 'user50:yHBDxuPAi'
[-] 99.68.230.147:22 - Failed: 'user50:dXobQabup'
[-] 99.68.230.147:22 - Failed: 'user50:rWeDHWuXu'
[-] 99.68.230.147:22 - Failed: 'user50:sWWFXXsoe'
[+] 99.68.230.147:22 - Success: 'user50:iJPvPJCel' 'uid=1001(user50) gid=1001(us
er50) groups=1001(user50) Linux klepetko 5.19.0-38-generic #39~22.04.1-Ubuntu SM
P PREEMPT_DYNAMIC Fri Mar 17 21:16:15 UTC 2 x86_64 x86_64 x86_64 GNU/Linux '
[*] SSH session 1 opened (12.45.23.100:34637 → 99.68.230.147:22) at 2023-04-29
20:51:19 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

c) Report how long it takes to test each password on average.

- 35.83 seconds / 9 passwords = 3.98 seconds on average to test each password.

For cracking ssh to B.2,

d) Show the screen shot of the parameters of the ssh login module. Use the "info" command in the MSF console console.

```
Basic options:
  Name                 Current Setting      Required  Description
  ----                 ---------------      --------  -----------
  BLANK_PASSWORDS      false                no        Try blank passwords for all
                                                      users
  BRUTEFORCE_SPEED     5                    yes       How fast to bruteforce, from
                                                       0 to 5
  DB_ALL_CREDS         false                no        Try each user/password coupl
                                                      e stored in the current data
                                                      base
  DB_ALL_PASS          false                no        Add all passwords in the cur
                                                      rent database to the list
  DB_ALL_USERS         false                no        Add all users in the current
                                                       database to the list
  DB_SKIP_EXISTING     none                 no        Skip existing credentials st
                                                      ored in the current database
                                                       (Accepted: none, user, user
                                                      &realm)
  PASSWORD                                  no        A specific password to authe
                                                      nticate with
  PASS_FILE            /usr/share/metaspl   no        File containing passwords, o
                       oit-framework/data              ne per line
                       /wordlists/http_de
                       fault_pass.txt
  RHOSTS               klepetko.net         yes       The target host(s), see http
                                                      s://github.com/rapid7/metasp
                                                      loit-framework/wiki/Using-Me
                                                      tasploit
  RPORT                22                   yes       The target port
  STOP_ON_SUCCESS      true                 yes       Stop guessing when a credent
                                                      ial works for a host
  THREADS              1                    yes       The number of concurrent thr
                                                      eads (max one per host)
  USERNAME             user50               no        A specific username to authe
                                                      nticate as
  USERPASS_FILE                             no        File containing users and pa
                                                      sswords separated by space,
                                                      one pair per line
  USER_AS_PASS         false                no        Try the username as the pass
                                                      word for all users
  USER_FILE            /usr/share/metaspl   no        File containing usernames, o
                       oit-framework/data              ne per line
                       /wordlists/http_de
                       fault_users.txt
  VERBOSE              true                 yes       Whether to print output for
                                                      all attempts

Description:
  This module will test ssh logins on a range of machines and report
  successful logins. If you have loaded a database plugin and
  connected to a database this module will record successful logins
```

e) Show the screenshot of finding the correct username and password in the MSF console.

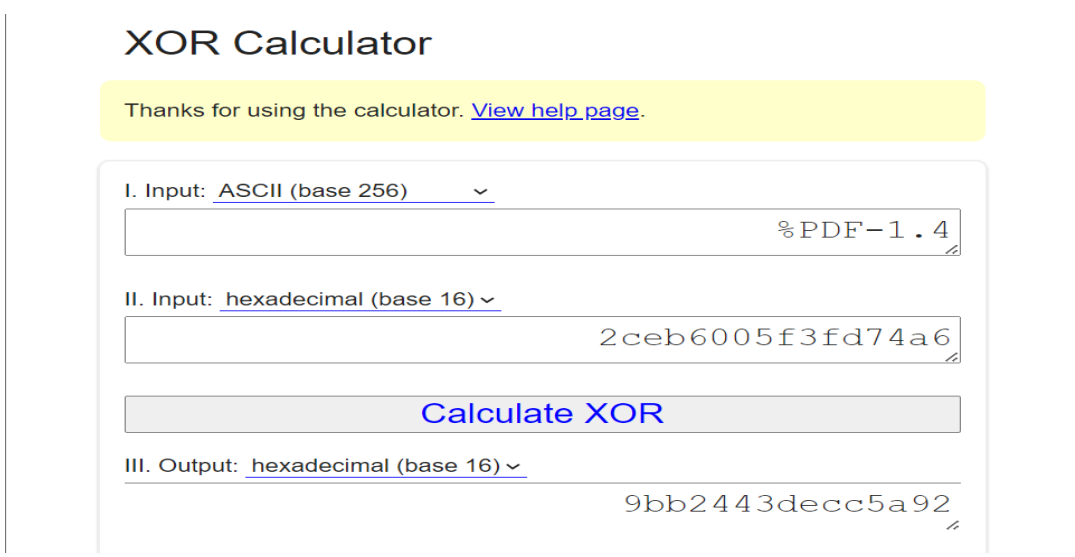
```
[-] 99.68.230.147:22 - Failed: 'xampp-dav-unsecure:wampp'
[-] 99.68.230.147:22 - Failed: 'xampp-dav-unsecure:ppmax2011'
[-] 99.68.230.147:22 - Failed: 'xampp-dav-unsecure:turnkey'
[-] 99.68.230.147:22 - Failed: 'xampp-dav-unsecure:vagrant'
[-] 99.68.230.147:22 - Failed: 'vagrant:admin'
[-] 99.68.230.147:22 - Failed: 'vagrant:password'
[-] 99.68.230.147:22 - Failed: 'vagrant:manager'
[-] 99.68.230.147:22 - Failed: 'vagrant:letmein'
[-] 99.68.230.147:22 - Failed: 'vagrant:cisco'
[-] 99.68.230.147:22 - Failed: 'vagrant:default'
[-] 99.68.230.147:22 - Failed: 'vagrant:root'
[-] 99.68.230.147:22 - Failed: 'vagrant:apc'
[-] 99.68.230.147:22 - Failed: 'vagrant:pass'
[-] 99.68.230.147:22 - Failed: 'vagrant:security'
[-] 99.68.230.147:22 - Failed: 'vagrant:user'
[-] 99.68.230.147:22 - Failed: 'vagrant:system'
[-] 99.68.230.147:22 - Failed: 'vagrant:sys'
[-] 99.68.230.147:22 - Failed: 'vagrant:none'
[-] 99.68.230.147:22 - Failed: 'vagrant:xampp'
[-] 99.68.230.147:22 - Failed: 'vagrant:wampp'
[-] 99.68.230.147:22 - Failed: 'vagrant:ppmax2011'
[-] 99.68.230.147:22 - Failed: 'vagrant:turnkey'
[+] 99.68.230.147:22 - Success: 'vagrant:vagrant' 'uid=1002(vagrant) gid=1002(va
grant) groups=1002(vagrant) Linux klepetko 5.19.0-38-generic #39~22.04.1-Ubuntu
SMP PREEMPT_DYNAMIC Fri Mar 17 21:16:15 UTC 2 x86_64 x86_64 x86_64 GNU/Linux '
[*] SSH session 1 opened (12.45.23.100:35059 → 99.68.230.147:22) at 2023-04-30
11:16:54 -0500
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

f) Report how long it takes to test each password on average.

- 3.41 minutes / 266 passwords = 0.060965 minutes or 3.657895 seconds.

Section IV (Task III):

a) Show the screenshot of your cryptoanalysis program when you get the key



b) Show the key

- Key in little Endian is 09bb2443decc5a92
- In big Endian the key is 925accde4324bbo9

```
┌──(stanley㉿kali)-[~/Documents/pro3]
└─$ ./enc1 secret.pdf.enc1 925accde4324bb09

┌──(stanley㉿kali)-[~/Documents/pro3]
└─$ nano secret.pdf.enc1.enc1
```

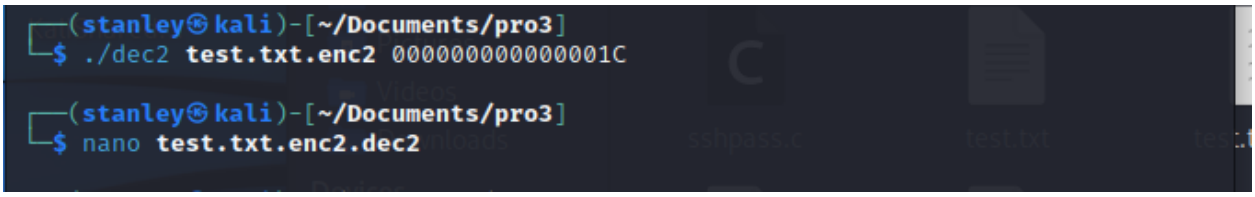
c) Show the content of the encrypted file secret.pdf.enc1.

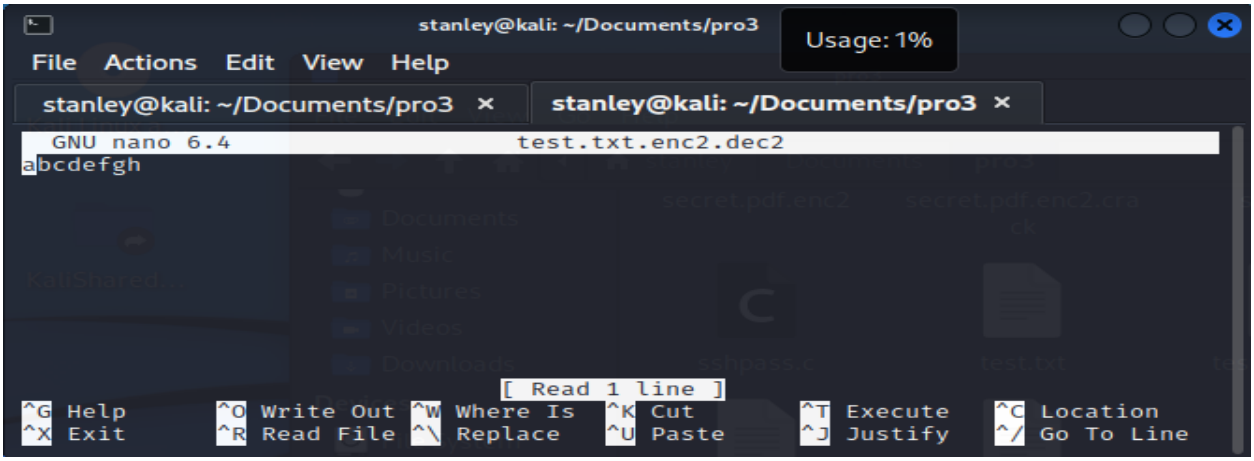
```
stanley@kali: ~/Documents/pro3 ×   stanley@kali: ~/Documents/pro3 ×
  GNU nano 6.4                                secret.pdf.enc1.enc1
%PDF-1.4
%◆◆◆◆
3 0 obj
<< /Length 4 0 R
   /Filter /FlateDecode
>>
stream
x◆mR◆j◆0^P◆◆+◆^Cqve=,^H>^TJ!◆C[◆B^NI^\^GBsh{◆◆w◆Z◆◆^T3◆◆◆◆◆Ð-◆◆Ů^Od/_◆a0◆^R◆^}M◆mR◆◆^G;◆◆zZ◆
[v◆◆nCD◆'WV^G◆◆^G^B^P◆^Dt@^F^N◆WbG=^?^BF◆◆Y7◆z◆◆◆^O[◆Ĉ◆bl◆e^X◆◆YK◆◆Ůf◆8◆◆K6◆^Y◆◆◆j◆^W◆%/◆◆◆◆^O◆◆◆;◆T◆◆◆^WTE◆◆:j^F)E
endstream
endobj
4 0 obj
   342
endobj
2 0 obj
<<
   /ExtGState <<
      /a0 << /CA 1 /ca 1 >>
   >>
   /Font <<
      /f-0-0 5 0 R
      /f-1-0 6 0 R
   >>
>>
endobj
7 0 obj
<< /Type /Page
   /Parent 1 0 R
   /MediaBox [ 0 0 611.999983 791.999983 ]
   /Contents 3 0 R
   /Group <<
      /Type /Group
      /S /Transparency
      /CS /DeviceRGB
   >>
   /Resources 2 0 R
>>
endobj
8 0 obj
<< /Length 9 0 R
   /Filter /FlateDecode
   /Length1 8052
>>
stream
x◆◆Y{x^TE◆◆◆◆◆3◆=3◆◆◆◆◆◆◆d^RB◆Ą^@^A4-◆F1@@◆◆◆@◆Ī          ★^Fn@%^A^E◆<◆C◆,/!◆^X^Pa^B^Ae◆
^H◆X◆oY◆◆^XD◆^Q◆◆◆PrO◆$◆◆◆◆◆◆nw◆◆◆◆◆◆◆◆yO^H%◆(d!^A◆◆:◆◆◆f~◆^U!◆◆8◆p◆^S◆{◆#◆◆◆◆^[!◆^W◆><sV◆'f^P^R◆}◆◆◆G◆^U◆.◆◆^Q◆◆^P

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location    M-U Undo
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^/ Go To Line  M-E Redo
```

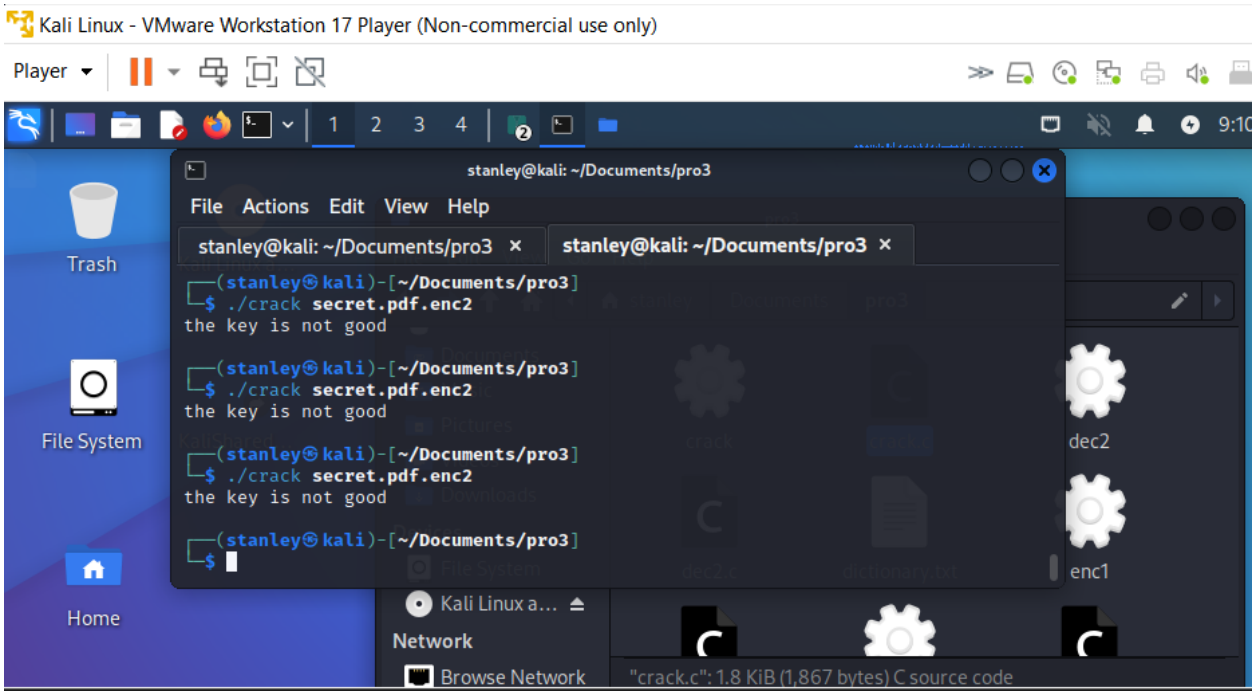
Section V (Task IV):

a) Show the screenshot of your DES program when it deciphers the testing file.





b) Show the screenshot of your DES program when you are brute force cracking the key of secret.pdf.enc2.



c) Report how many keys are tested in 10 minutes.

- To estimate how many keys can be tested in 10 minutes, we can multiply the number of seconds in 10 minutes (600) by the number of keys tested per second:
- keys tested in 10 minutes = 600 * (1ULL << 56) / 5 ≈ 5.166 * 10^18

d) Estimate how long it will take to find the key.

- To estimate how long it will take to find the key, we can assume that the program tests keys sequentially and that each key has an equal chance of being the correct key. This means that the expected number of keys that need to be tested before finding the correct key is half the total number of possible keys, or 2^55 = 36,028,797,018,963,968.

Note that you may not be able to find the key given the current hardware.

- Yes i was unable to find the key when doing the brute force cracking to find the key. Overall, the difficulty of finding an encryption key depends on a number of factors, including the key length, the strength of the encryption algorithm, the power of the hardware being used, and any additional security measures that may be in place.