Name: Syed Abdul Samad Ahsan

Subject: Operating System

Seat no: Ep1849102

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S.No | Date | Lab | Mark | sign |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |

Lab no 1

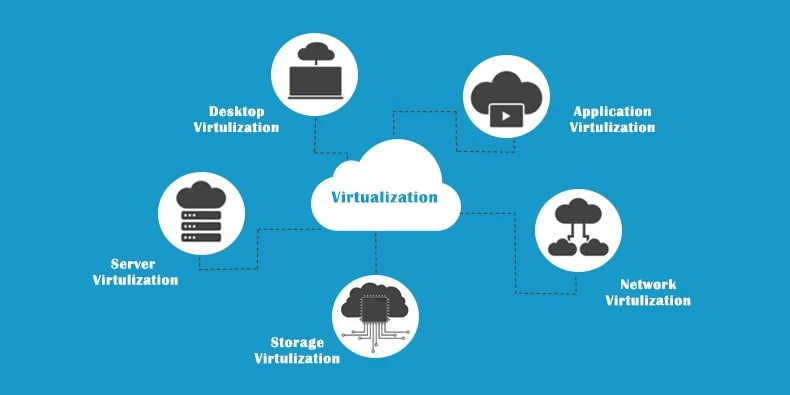
Virtualization:

Virtualization = Virtual + Realization. In other words, virtualization means to realize something virtually. A simple definition of virtualization is the formation of virtual computer inside the real computer. It is an abstraction layer above the hardware layer. It is hardware reducing , memory saving and cost & energy saving technology that is rapidly transforming the IT landscape. In computing, ***virtualization refers to the creation of a virtual (rather than a real) version of something, including virtual computer hardware platforms, storage devices, and computer resources.*** In other words, virtualization can also be perceived as a technique which allows us to share a physical instance of a resource or an application among multiple customers and organizations.

The idea of virtualization is not new. It was introduced by IBM in 1960 when mainframe computers were in use. Mainframe computers were underutilized most of the time, hence to amplify the resource utilization of these mainframe computers, the virtualization technology was introduced which allows to run multiple OS (Operating Systems) simultaneously.

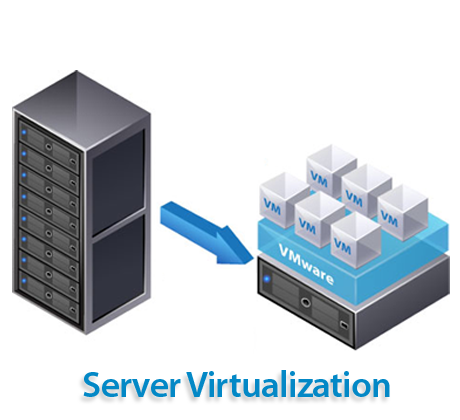
Due to invention of desktop and reduction in hardware prices this technology become dormant for long time.

But over the past few decades with the invention of new technologies like utility computing and cloud computing, virtualization has regained its importance.



**Server Virtualization**

Server virtualization allows multiple servers to be installed on one or more existing servers. This saves floor space and money since you don’t have to purchase new servers or expand the square footage of your server room.



## **What is a hypervisor?**

A **hypervisor**, also known as a virtual machine monitor or VMM, is software that creates and runs virtual machines (VMs). A hypervisor allows one host computer to support multiple guest VMs by virtually sharing its resources, such as memory and processing.

## **Why use a hypervisor?**

Hypervisors make it possible to use more of a system’s available resources and provide greater IT mobility since the guest VMs are independent of the host hardware. This means they can be easily moved between different servers. Because multiple virtual machines can run off of one physical server with a hypervisor, a hypervisor reduces:

* Space
* Energy
* Maintenance requirements

## **How does a hypervisor work?**

Hypervisors support the creation and management of virtual machines (VMs) by abstracting a computer’s software from its hardware. Hypervisors make virtualization possible by translating requests between the physical and virtual resources. Bare-metal hypervisors are sometimes embedded into the firmware at the same level as the motherboard basic input/output system (BIOS) to enable the operating system on a computer to access and use virtualization software.

**Benefits of hypervisors**

There are several benefits to using a hypervisor that hosts multiple virtual machines:

* **Speed:**Hypervisors allow virtual machines to be created instantly, unlike bare-metal servers. This makes it easier to provision resources as needed for dynamic workloads.
* **Efficiency:** Hypervisors that run several virtual machines on one physical machine’s resources also allow for more efficient utilization of one physical server. It is more cost- and energy-efficient to run several virtual machines on one physical machine than to run multiple underutilized physical machines for the same task.
* **Flexibility:**Bare-metal hypervisors allow operating systems and their associated applications to run on a variety of hardware types because the hypervisor separates the OS from the underlying hardware, so the software no longer relies on specific hardware devices or drivers.
* **Portability:** Hypervisors allow multiple operating systems to reside on the same physical server (host machine). Because the virtual machines that the hypervisor runs are independent from the physical machine, they are portable. IT teams can shift workloads and allocate networking, memory, storage and processing resources across multiple servers as needed, moving from machine to machine or platform to platform. When an application needs more processing power, the virtualization software allows it to seamlessly access additional machines.

## 

Lab no 2

# **Architecture of Linux**

Let's first start with the basic knowledge of the Linux operating system.

## **Linux operating system**

An operating system can be described as an interface among the computer hardware and the user of any computer. It is a group of software that handles the resources of the computer hardware and facilitates basic services for computer programs.

An operating system is an essential component of system software within a computer system. The primary aim of an operating system is to provide a platform where a user can run any program conveniently or efficiently.

On the other hand, [Linux](https://www.javatpoint.com/linux-tutorial)

OS is one of the famous versions of the UNIX OS. It is developed to provide a low-cost or free OS for several personal computer system users. Remarkably, it is a complete OS Including an **X Window System, Emacs editor, [IP/TCP](https://www.javatpoint.com/computer-network-tcp-ip-model)**

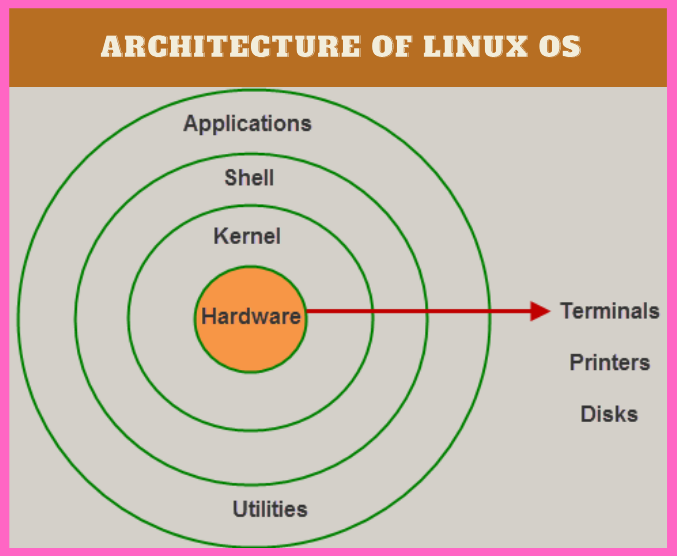
**, GUI** (graphical user interface), etc.

## **Linux operating system history**

In 1991, the Linux history started with the starting of a particular project by the Finland student **Linus Torvalds** for creating a new free **OS kernel**. The final Linux Kernel was remarked by continuous development throughout the history since then.

* Linux was proposed by the Finland student Linus Torvalds in 1991.
* HP-UX (**Hewlett Packard**) 8.0 version was published.
* Hewlett Packard 9.0 version was published in 1992.
* FreeBSD 1.0 version and **NetBSD**8 version was released in 1993.
* Red Hat Linux was proposed in 1994. Caldera was detected by Ransom love and Bryan Sparks and NetBSD 1.0 version published.
* HP-UX 10.0 version and FreeBSD 2.0 version was released in 1995.
* K Desktop Environment was established by **Matthias Ettrich** in 1996.
* HP-UX 11.0 version was released in 1997.
* The IRIX 6.5 version, i.e., the fifth SGI UNIX generation, Free BSD 3.0 version, and Sun Solaris 7 OS was released in 1998.
* The **Caldera System** agreement with professional services division and SCO server software division was released in 2000.
* **Linus Torvalds** published the Linux version 2.4 source code in 2001.
* **Microsoft** filed the Trademark collection against Lindows.com in 2001.
* Lindows name was modified to Linspire in 2004.
* The first publication of **Ubuntu** was published in 2004.
* The openSUSE project started a free distribution from the community of **Novell** In 2005.
* **Oracle** published its Red Hat distribution in 2006.
* **Dell** begun laptop distribution with Ubuntu which was pre-installed on it in 2007.
* Linux kernel version 3.0 was released in 2011.
* Linux-based android of Google insisted 75% of the market share of the Smartphone, based on the number of phones exported in 2013.
* Ubuntu insisted on 20000000+ users in 2014.

## **Architecture of Linux system**



The Linux operating system's architecture mainly contains some of the components: **the Kernel, System Library, Hardware layer, System,** and **Shell utility**.

**1. Kernel:-** The kernel is one of the core section of an operating system. It is responsible for each of the major actions of the Linux OS. This operating system contains distinct types of modules and cooperates with underlying hardware directly. The kernel facilitates required abstraction for hiding details of low-level hardware or application programs to the system. There are some of the important kernel types which are mentioned below:

* Monolithic Kernel
* Micro kernels
* Exo kernels
* Hybrid kernels

**2. System Libraries:-** These libraries can be specified as some special functions. These are applied for implementing the operating system's functionality and don't need code access rights of the modules of kernel.

**3. System Utility Programs:-** It is responsible for doing specialized level and individual activities.

**4. Hardware layer:-** Linux operating system contains a hardware layer that consists of several peripheral devices like CPU,HDD,RAM

.

**5. Shell: -** It is an interface among the kernel and user. It can afford the services of kernel. It can take commands through the user and runs the functions of the kernel. The shell is available in distinct types of OSes. These operating systems are categorized into two different types, which are the **graphical shells** and **command-line shells**.

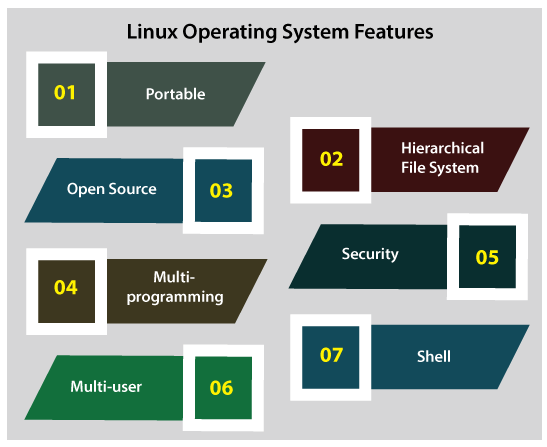
The graphical line shells facilitate the graphical user interface, while the command line shells facilitate the command line interface. Thus, both of these shells implement operations. However, the graphical user interface shells work slower as compared to the command-line interface shells.

There are a few types of these shells which are categorized as follows:

* Korn shell
* Bourne shell
* C shell
* POSIX shell

## **Linux Operating System Features**

Some of the primary features of Linux OS are as follows:



* **Portable:** Linux OS can perform different types of hardware and the kernel of Linux supports the installation of any type of hardware environment.
* **Open source:** Linux operating system source code is available freely and for enhancing the capability of the Linux OS, several teams are performing in collaboration.
* **Multiprogramming:** Linux OS can be defined as a multiprogramming system. It means more than one application can be executed at the same time.
* **Multi-user:** Linux OS can also be defined as a multi-user system. It means more than one user can use the resources of the system such as **application programs, memory,** or **RAM** at the same time.
* **Hierarchical file system:** Linux OS affords a typical file structure where user files or system files are arranged.
* **Security:** Linux OS facilitates user security systems with the help of various features of authentication such as controlled access to specific files, password protection, or data encryption.
* **Shell:** Linux operating system facilitates a unique interpreter program. This type of program can be applied for executing commands of the operating system. It can be applied to perform various types of tasks such as call application programs and others.

Lab no 3

Basic linux commands:

**1. pwd Command**

The [pwd](https://www.javatpoint.com/linux-pwd) command is used to display the location of the current working directory.

**Syntax:**

1. pwd

**Output:**

Linux Commands with Examples

**2. mkdir Command**

The [mkdir](https://www.javatpoint.com/linux-mkdir) command is used to create a new directory under any directory.

**Syntax:**

1. mkdir **<directory** name**>**

**Output:**

Linux Commands with Examples

**3. rmdir Command**

The [rmdir](https://www.javatpoint.com/linux-rmdir) command is used to delete a directory.

**Syntax:**

1. rmdir **<directory** name**>**

**Output:**

Linux Commands with Examples

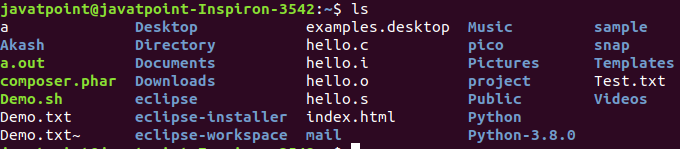
**4. ls Command**

The [ls](https://www.javatpoint.com/linux-ls) command is used to display a list of content of a directory.

**Syntax:**

1. ls

**Output:**



**5.cd Command**

The [cd](https://www.javatpoint.com/linux-cd) command is used to change the current directory.

**Syntax:**

1. cd **<directory** name**>**

**Output:**

Linux Commands with Examples

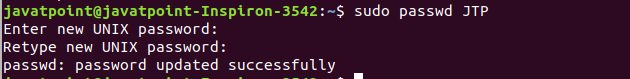
**6. passwd Command**

The [passwd](https://www.javatpoint.com/linux-user-password) command is used to create and change the password for a user.

**Syntax:**

1. passwd **<username>**

**Output:**



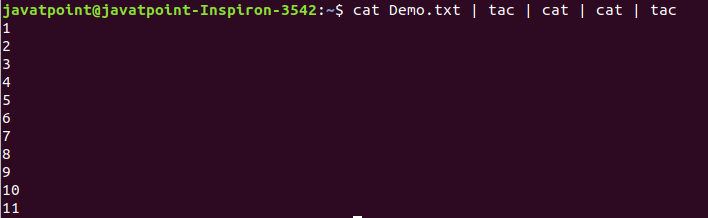
**7. cat Command**

The [cat](https://www.javatpoint.com/linux-cat-filters) command is also used as a filter. To filter a file, it is used inside pipes.

**Syntax:**

1. cat **<fileName>** | cat or tac | cat or tac |. . .

**Output:**



Lab no 4

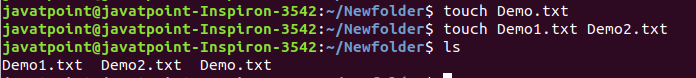
Linux file commands:

**1. touch Command**

The [touch](https://www.javatpoint.com/linux-touch) command is used to create empty files. We can create multiple empty files by executing it once.

**Syntax:**

1. touch **<file** name**>**
2. touch **<file1>**  **<file2>** ....



**2.cat Command**

The [cat](https://www.javatpoint.com/linux-cat) command is a multi-purpose utility in the Linux system. It can be used to create a file, display content of the file, copy the content of one file to another file, and more.

**Syntax:**

1. cat [OPTION]... [FILE]..

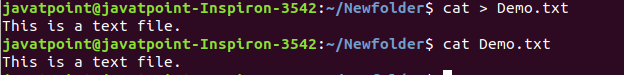
To create a file, execute it as follows:

1. cat **>** **<file** name**>**
2. // Enter file content

Press "**CTRL+ D**" keys to save the file. To display the content of the file, execute it as follows:

1. cat **<file** name**>**

**Output:**



**3. rm Command**

The [rm](https://www.javatpoint.com/linux-rm) command is used to remove a file.

**Syntax:**

rm <file name>

**Output:**

Linux Commands with Examples

**4. cp Command**

The [cp](https://www.javatpoint.com/linux-cp) command is used to copy a file or directory.

**Syntax:**

To copy in the same directory:

1. cp **<existing** file name**>** **<new** file name**>**

To copy in a different directory:

**Output:**

Linux Commands with Examples

**5. mv Command**

The [mv](https://www.javatpoint.com/linux-mv) command is used to move a file or a directory form one location to another location.

**Syntax:**

1. mv **<file** name**>** **<directory** path**>**

**Output:**

Linux Commands with Examples

**6. rename Command**

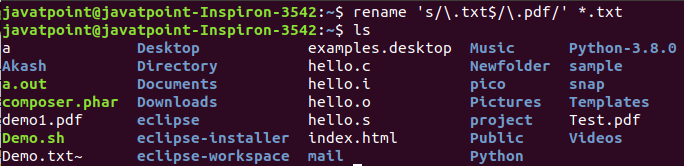
The [rename](https://www.javatpoint.com/linux-rename) command is used to rename files. It is useful for renaming a large group of files.

**Syntax:**

1. rename 's/old-name/new-name/' files

For example, to convert all the text files into pdf files, execute the below command:

1. rename 's/\.txt$/\.pdf/' \*.txt



Lab no 5

Introduction to security

Cyber kill chain:

The cyber kill chain is a series of steps that trace stages of a cyberattack from the early reconnaissance stages to the exfiltration of data. The kill chain helps us understand and combat ransomware, security breaches, and advanced persistent attacks (APTs).

Lockheed Martin derived the kill chain framework from a military model – originally established to identify, prepare to attack, engage, and destroy the target. Since its inception, the kill chain has evolved to better anticipate and recognize insider threats, social engineering, advanced ransomware and innovative attacks.

## **How the Cyber Kill Chain Works**

There are several core stages in the cyber kill chain. They range from reconnaissance (often the first stage in a malware attack) to lateral movement (moving laterally throughout the network to get access to more data) to data exfiltration (getting the data out).  All of your common attack vectors – whether phishing or brute force or the latest strain of malware – trigger activity on the cyber kill chain.

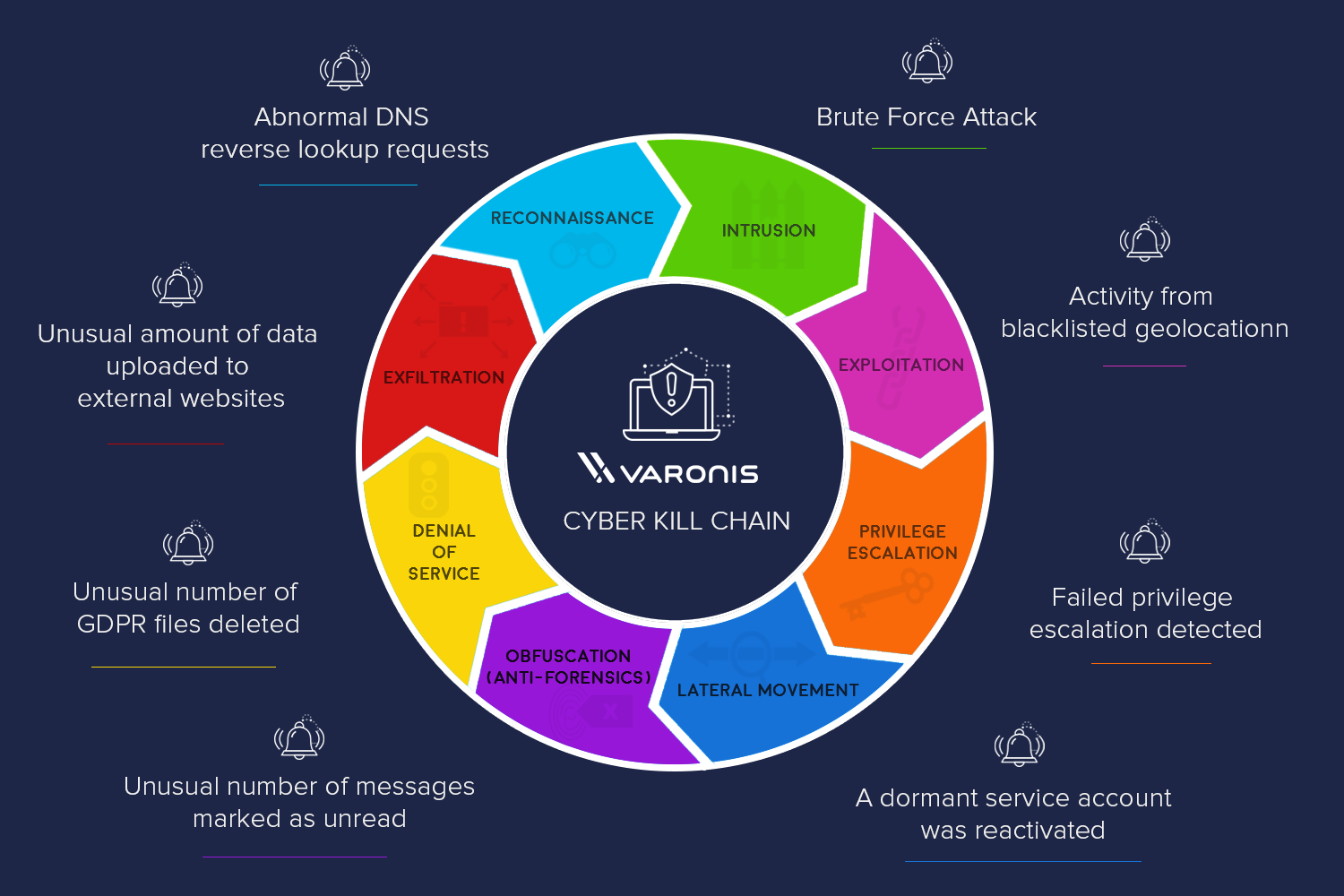
[](https://blogvaronis2.wpengine.com/wp-content/uploads/2016/06/cyber-kill-chain-phases-2@2x.png)

Each stage is related to a certain type of activity in a cyber attack, regardless of whether it’s an internal or external attack:

* **Reconnaissance**  
  The observation stage: attackers typically assess the situation from the outside-in, in order to identify both targets and tactics for the attack.
* **Intrusion**  
  Based on what the attackers discovered in the reconnaissance phase, they’re able to get into your systems: often leveraging malware or security vulnerabilities.
* **Exploitation**  
  The act of exploiting vulnerabilities, and delivering malicious code onto the system, in order to get a better foothold.
* **Privilege Escalation**  
  Attackers often need more privileges on a system to get access to more data and permissions: for this, they need to escalate their privileges often to an Admin.
* **Lateral Movement**  
  Once they’re in the system, attackers can move laterally to other systems and accounts in order to gain more leverage: whether that’s higher permissions, more data, or greater access to systems.
* **Obfuscation / Anti-forensics**  
  In order to successfully pull off a cyberattack, attackers need to cover their tracks, and in this stage they often lay false trails, compromise data, and clear logs to confuse and/or slow down any forensics team.
* **Denial of Service**  
  Disruption of normal access for users and systems, in order to stop the attack from being monitored, tracked, or blocked
* **Exfiltration**  
  The extraction stage: getting data out of the compromised system.

Below, we’ll explore each phase of the cyber kill chain in more detail.

## **8 Phases of The Cyber Kill Chain**

[](https://blogvaronis2.wpengine.com/wp-content/uploads/2016/06/KC-bgadd.png)

Each phase of the kill chain is an opportunity to stop a cyberattack in progress: with the right tools to detect and recognize the behavior of each stage, you’re able to better defend against a systems or data breach.

### **Reconnaissance**

In every heist, you’ve got to scope the joint first. Same principle applies in a cyber-heist: it’s the preliminary step of an attack, the information gathering mission. During reconnaissance, an attacker is seeking information that might reveal vulnerabilities and weak points in the system. Firewalls, intrusion prevention systems, perimeter security – these days, even social media accounts – get ID’d and investigated. Reconnaissance tools scan corporate networks to search for points of entry and vulnerabilities to be exploited.

### **Intrusion**

Once you’ve got the intel, it’s time to break in. Intrusion is when the attack becomes active: attackers can send malware – including ransomware, spyware, and adware – to the system to gain entry. This is the delivery phase: it could be delivered by phishing email, it might be a compromised website or that really great coffee shop down the street with free, hacker-prone wifi. Intrusion is the point of entry for an attack, getting the attackers inside.

### **Exploitation**

You’re inside the door, and the perimeter is breached. The exploitation stage of the attack…well, exploits the system, for lack of a better term. Attackers can now get into the system and install additional tools, modify security certificates and create new script files for nefarious purposes.

### **Privilege Escalation**

What’s the point of getting in the building, if you’re stuck in the lobby? Attackers use privilege escalation to get elevated access to resources. Privilege escalation techniques often include brute force attacks, preying on password vulnerabilities, and exploiting zero day vulnerabilities. They’ll modify GPO security settings, configuration files, change permissions, and try to extract credentials.

### **Lateral Movement**

You’ve got the run of the place, but you still need to find the vault. Attackers will move from system to system, in a lateral movement, to gain more access and find more assets. It’s also an advanced data discovery mission, where attackers seek out critical data and sensitive information, admin access and email servers – often using the same resources as IT and leveraging built-in tools like PowerShell – and position themselves to do the most damage.

### **Obfuscation (anti-forensics)**

Put the security cameras on a loop and show an empty elevator so nobody sees what’s happening behind the scenes. Cyber-attackers do the same thing: conceal their presence and mask activity to avoid detection and thwart the inevitable investigation. This might mean wiping files and metadata, overwriting data with false timestamps (timestomping) and misleading information, or modifying critical information so that it looks like the data was never touched.

### **Denial of Service**

Jam the phone lines and shut down the power grid. Here’s where the attackers target the network and data infrastructure, so that the legitimate users can’t get what they need. The denial of service (DoS) attack disrupts and suspends access, and could crash systems and flood services.

### **Exfiltration**

Always have an exit strategy. The attackers get the data: they’ll copy, transfer, or move sensitive data to a controlled location, where they do with the data what they will. Ransom it, sell it on ebay, send it to wikileaks. It can take days to get all of the data out, but once it’s out, it’s in their control.

## **The Takeaway**

Different security techniques bring forward different approaches to the cyber kill chain – everyone from Gartner to Lockheed Martin defines the stages slightly differently. Alternative models of the cyber kill chain combine several of the above steps into a C&C stage (command and control, or C2) and others into an ‘Actions on Objective’ stage. Some combine lateral movement and privilege escalation into an exploration stage; others combine intrusion and exploitation into a ‘point of entry’ stage.

It’s a model often criticized for focusing on perimeter security and limited to malware prevention. When combined with advanced analytics and predictive modeling, however, the cyber kill chain becomes critical to data security.

With the above breakdown, the kill chain is structured to reveal the active state of a data breach. Each stage of the kill chain requires specific instrumentation to detect cyber attacks , and Varonis has out-of-the-box threat models to detect those attacks at every stage of the kill chain.

Varonis monitors attacks at the entry, exit, and everywhere in between. By monitoring outside activity – like VPN, DNS, and Proxy, Varonis helps guard the primary ways to get in and out of an organization.  By monitoring file activity and user behavior, Varonis can detect attack activity on every stage of the kill chain – from kerberos attacks to malware behavior.

Want to see it in action? See how Varonis addresses each stage of the kill chain– and learn how you can prevent and stop ongoing attacks before the damage is done.

Lab no 6

Nmap

# **What is Nmap?**

**Nmap** is an open-source utility for network discovery. Network Mapper is a security auditing and network scanning independent tool developed by **Gordon Lyon**. It is used by network administrators to detect the devices currently running on the system and the port number by which the devices are connected.

Many systems and network administrators are used for managing **network inventory, service upgrade schedules, monitoring hosts** and **service uptime**.

### **Nmap Definition**

At the top-level, [Nmap](https://www.educba.com/nmap-commands/" \t "_blank)

is defined as a tool that can detect or diagnose services that are running on an **Internet-connected system** by a network administrator in their networked system used to identify potential security flaws. It is used to automate redundant tasks, such as monitoring the service.

### **Working of Nmap**

Nmap is convenient during penetration testing

of networked systems. Nmap provides the network details, and also helps to determine the security flaws present in the system. Nmap is **platform-independent** and runs on popular **operating systems** such as **Linux**

**, Windows**

and **Mac**.

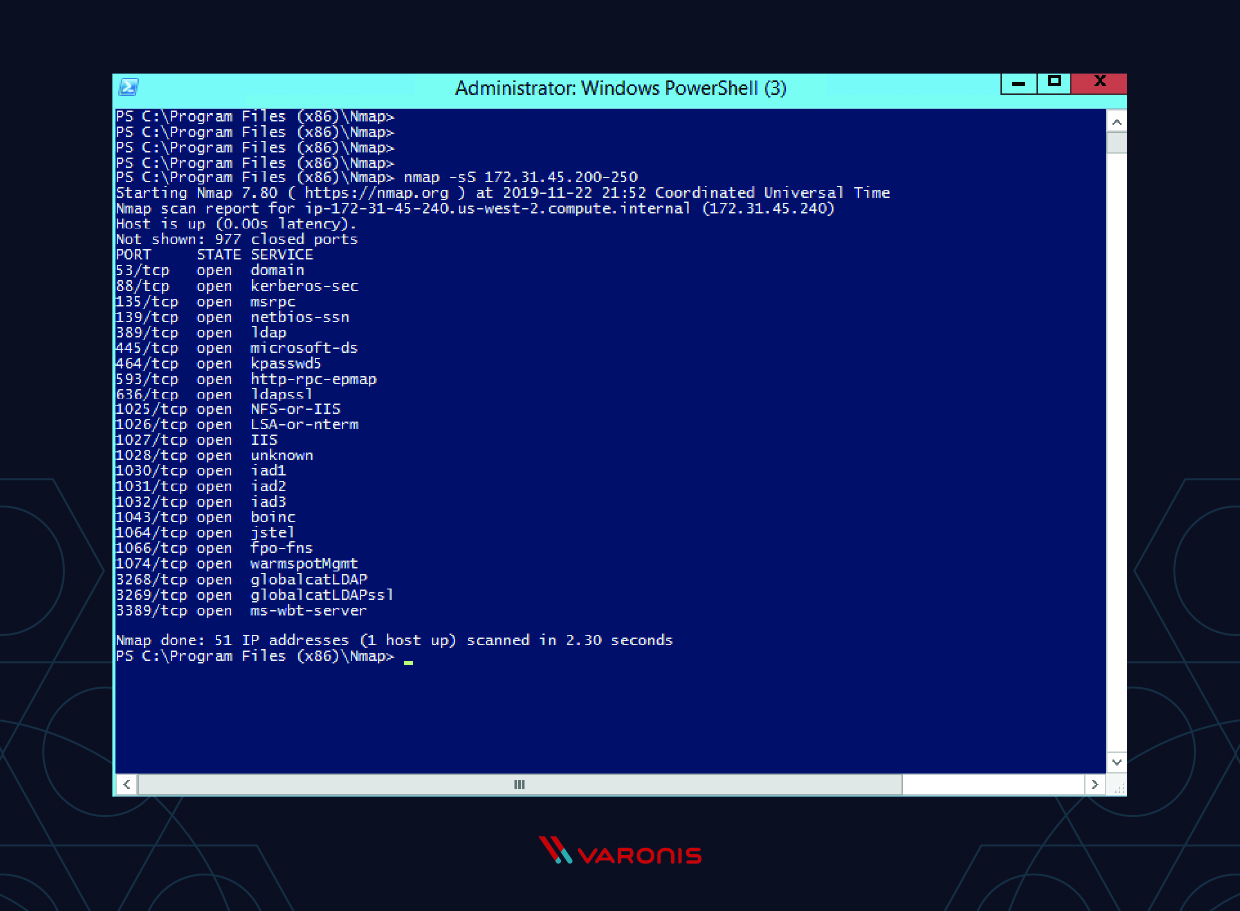
To run a ping scan, run the following command:

# nmap -sp 192.100.1.1/24

To run a host scan, use the following command:

1. # nmap -sp <target IP range>
2. If you see anything unusual in this list, you can then run a DNS query on a specific host, by using:
3. # namp -sL <IP address>

### **Port Scanning**



Lab no 7

Email security and encryption

## **Spoofing**

It is basically an attack on a computer device in which the attacker stoles the identity of a user to steal the user's information or to breach the system's security. In spoofing, the attacker acts as another person. It is identity theft. It can be part of phishing, but it is not exactly phishing.



Spoofing is pretending that you are from a trusted source. There are various types of spoofing such as IP spoofing, Email spoofing, MAC spoofing, DNS spoofing, and URL spoofing. The DNS spoofing attacks can go on for a long period of time without being detected and can cause serious security issues. Attackers generally target large enterprises or organizations to steal the information and then connect with the target group to hack their system.

## **Phishing**

"Phish" is pronounced like the word "fish" - the analogy is that anyone who throws a backed hook out (phishing email) and expects you to bite. Phishing is a crime where people share their confidential information like passwords and credit card numbers with hackers.

It starts with a fraudulent email or communication which is designed to attract a victim. The attackers embedded a link in an email that redirects employees to an unsafe web that requests sensitive information.

The email looks like it is from a company that you trust easily. A normal greeting in the email is, "Hi Dear." If we have an account with the business, it won't use a common greeting in this way. It invites you to click on a link to update the payment details and look as genuine as a bank website.

The methods of phishing include -

* An email that requires the user's personal data verification with the text on the email like "we couldn't verify the data you entered - click the below link for the same". There could be an email that is designed to attract the user by saying the things such as tax refunds.
* Some other methods of phishing are - Infected attachments, like .exe files PDF documents, and Microsoft Office files can install **ransomware** or various other **malware**.

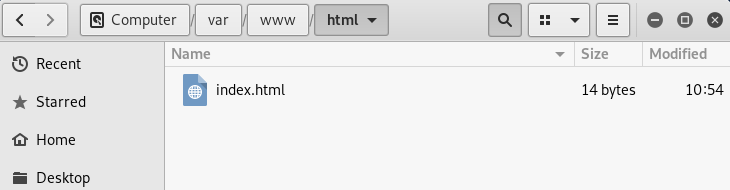
# **DNS Spoofing**

In this section, we will learn about DNS server. DNS is basically a server that converts the domain name to the IP address of the device. We can convert the domain name like www.google.com to the IP addresses of the device where the Google website is stored. Since we are the MITM, we can have a DNS server running on our computer and resolve DNS requests the way we want. For example, whenever a person requests to Google.com, we can actually take them to another website, because we are in the middle. So, when someone requests it, we will actually give them an IP that we want, and then they will see a completely different website than what they are expecting. So, we can have a fake website running on our server and get requests, for example, from xyz.com to that website.

To do this attack, the first thing we will do is redirect people to our web server. The web server is going to be run on our local Kali machine. We can redirect the people anywhere we want. But in this section, we are going to redirect them to our local web server. To do this, we will start Apache web server. It comes preinstalled with the Kali machine, so all we have to do is run the following command, and after this, the web server will start:

DNS Spoofing

The file for the web server is stored in the **/var/www/html** directory. We are going to open the file manager, and we are going to go to the **/var/www/html** directory. Now, if we browse our web server, the following page will be displayed as shown in the given screenshot:



In the above image, we can see a whole complete website installed here, and it will be displayed whenever a person visits our web server. If we go to the browser and browse **10.0.0.11**, which is our IP address, we will see **index.html** page there.

HTML Tutorial

Now let's configure the DNS server that comes in with MITMf. To do that we are going to use the leafpad which is the text editor. Then we are going to run the following command:

DNS Spoofing

After executing this command, we are going to scroll down to where the **A** records are, as seen in the following screenshot. A records are basically the records that are responsible for transforming or translating domain names to IP addresses:

DNS Spoofing

We are going to be targeting **xyz.com** and using the \* as a wildcard. So, basically, we are saying any subdomain to **xyz.com** should be redirected to our IP address which is **10.0.0.11**. If we want to replace this, we can do this with any IP address, for example, we can redirect it to Google by putting the IP of Google. Any IP we put here will redirect **xyz.com**. Now save the file and close it, and we are going to run our command. The following command is very similar to the command that we were running before in the previous sections. The only difference is we are going to add one extra option which is **--dns**. The command is as follows:



In the above screenshot, we can see that DNS spoofing enabled. Now let's go to the target and try to go **xyz.com** and see what happens. In the following screenshot, we can see that **xyz.com** is redirected to our website, which displays some simple text. But if we want, we can install anything. We can ask them to download something, or we can have a fake page, steal stuff, and steal credentials:

## LAB 8

## **What is Shell?**

**Shell** is a UNIX term for an interface between a user and an operating system service. Shell provides users with an interface and accepts human-readable commands into the system and executes those commands which can run automatically and give the program’s output in a shell script.

An Operating is made of many components, but its two prime components are

* Kernel
* Shell

Components of Shell Program

A Kernel is at the nucleus of a computer. It makes the communication between the hardware and software possible. While the Kernel is the innermost part of an operating system, a shell is the outermost one

A shell in a Linux operating system takes input from you in the form of commands, processes it, and then gives an output. It is the interface through which a user works on the programs, commands, and scripts. A shell is accessed by a terminal which runs it.

When you run the terminal, the Shell issues **a command prompt (usually $),** where you can type your input, which is then executed when you hit the Enter key. The output or the result is thereafter displayed on the terminal.

The Shell wraps around the delicate interior of an Operating system protecting it from accidental damage. Hence the name **Shell**

## Types of Shell

There are two main shells in Linux:

**1**. The **Bourne Shell**: The prompt for this shell is $ and its derivatives are listed below:

* POSIX shell also is known as sh
* Korn Shell also knew as sh
* **B**ourne **A**gain **SH**ell also knew as bash (most popular)

**2.** **The C shell**: The prompt for this shell is %, and its subcategories are:

1) C shell also is known as csh

2) Tops C shell also is known as tcsh

## How to Write Shell Script in Linux/Unix

**Shell Scripts** are written using text editors. On your Linux system, open a text editor program, open a new file to begin typing a shell script or shell programming, then give the shell permission to execute your shell script and put your script at the location from where the shell can find it.

Let us understand the steps in creating a Shell Script:

1. **Create a file** **using** a **vi** editor (or any other editor). Name script file with **extension .sh**
2. **Start** the script with **#! /bin/sh**
3. Write some code.
4. Save the script file as filename.sh
5. For **executing** the script type **bash filename.sh**

“#!” is an operator called shebang which directs the script to the interpreter locate on. So, if we use”#! /bin/sh” the script gets directed to the bourne-shell.

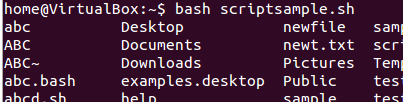
Create new script file scriptexample.sh



Adding the command ‘ls’ after #l/bin/sh



Executing the script file

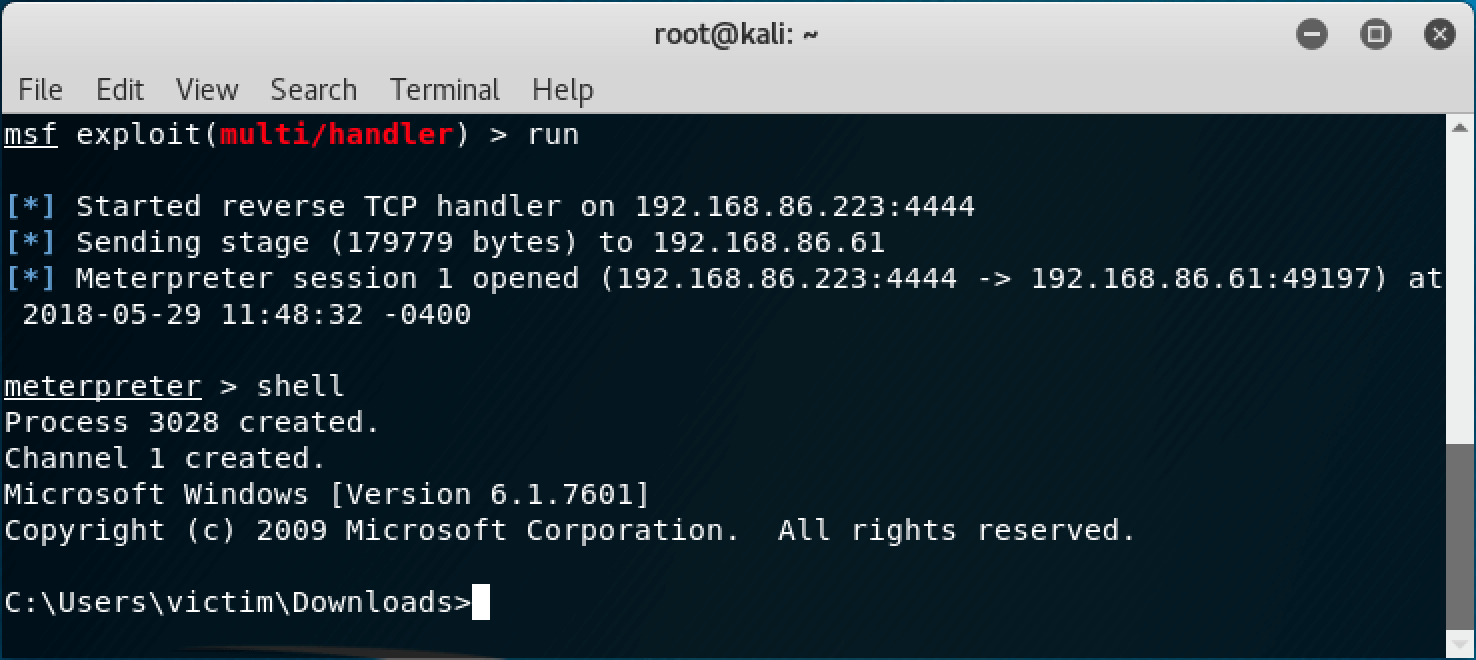


## What are Shell Variables?

As discussed earlier, Variables store data in the form of characters and numbers. Similarly, Shell variables are used to store information and they can by the shell only

MetaSPloit

Metasploit is one of the **most commonly used penetration testing tools** and comes built-in to Kali Linux. ... Modules are standalone pieces of code or software that provide functionality to Metasploit. There are six total modules: exploits, payloads, auxiliary, nops, posts, and encoders



## MSFCONSOLE CORE COMMANDS

back Move back from the current context

banner Display an awesome metasploit banner

cd Change the current working directory

color Toggle color

connect Communicate with a host

edit Edit the current module with $VISUAL or $EDITOR

exit Exit the console

get Gets the value of a context-specific variable

getg Gets the value of a global variable

go\_pro Launch Metasploit web GUI

grep Grep the output of another command

help Help menu

info Displays information about one or more module

irb Drop into irb scripting mode

jobs Displays and manages jobs

kill Kill a job

load Load a framework plugin

loadpath Searches for and loads modules from a path

makerc Save commands entered since start to a file

popmmodulestack Pops the latest module off the stack and makes it active

quit Exit the console

reload\_all Reloads all modules from all defined module paths

rename\_job Rename a job

resource Run the commands stored in a file

route Route traffic through a session

save Saves the active datastores

search Searches module names and descriptions

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

show Displays modules of a given type, or all modules

sleep Do nothing for the specified number of seconds

spool Write console output into a file as well the screen

threads View and manipulate background threads

unload Unload a framework plugin

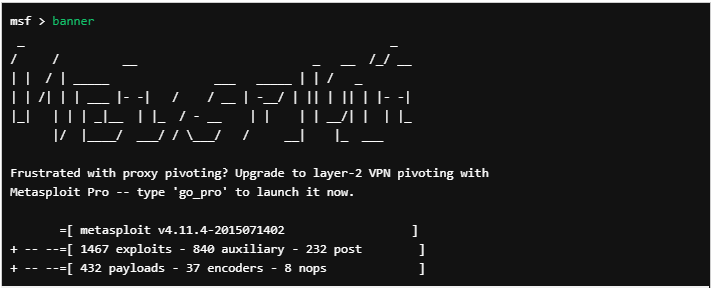
unset Unsets one or more context-specific variables

unsetg Unsets one or more global variables

use Selects a module by name

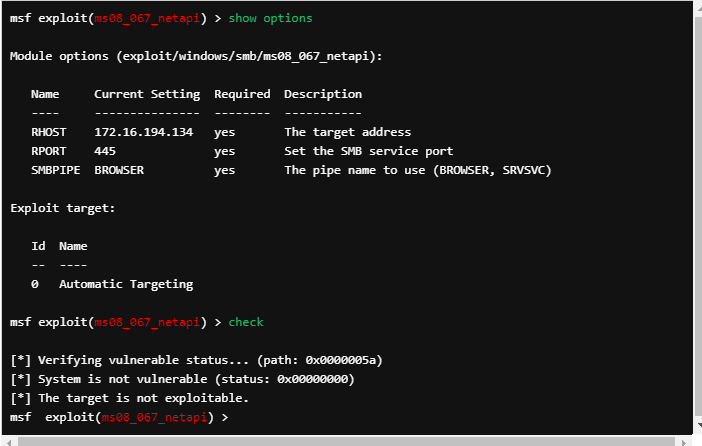
version Show the framework and console library version numbers

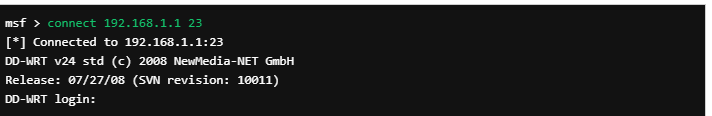
Banner

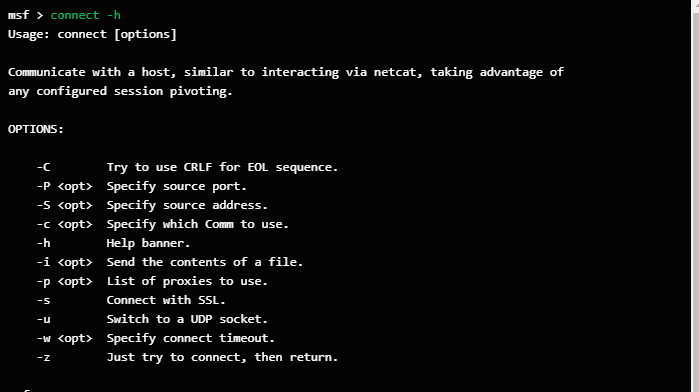


Check

There aren’t many exploits that support it, but there is also a **check** option that will check to see if a target is vulnerable to a particular exploit instead of actually exploiting it.



Connect

Use -h for additional informatio

Use