Chapter 3.2

Linux Security and Physical Security

Aim

To familiarize the students with the various vulnerabilities that exists in Linux environment and find the ways to prevent them

Instructional Objectives

After completing this chapter, you should be able to:

- Explain how to compile a Linux kernel
- Demonstrate the use of a GCC command for compilation
- Explain how to install Linux kernel Modules (LKM)
- Describe Linux hardening methods
- Outline the types of intrusion detection systems and evasion techniques
- Explain what a firewall is and the different types of firewalls

Learning Outcomes

At the end of this chapter, you are expected to:

- List some of the Linux commands, along with a description
- Identify the reasons behind recompilation of the Linux kernel
- Summarise the steps required to improve the security of a Linux Server
- Categorise the intrusion detection system
- Discuss how firewalls protect the system from intruders

Linux Hacking

Introduction to LINUX Hacking

Installing Firewalls is a good way of enhancing security

Installing intrusion detection systems, encrypting data and using strong passwords

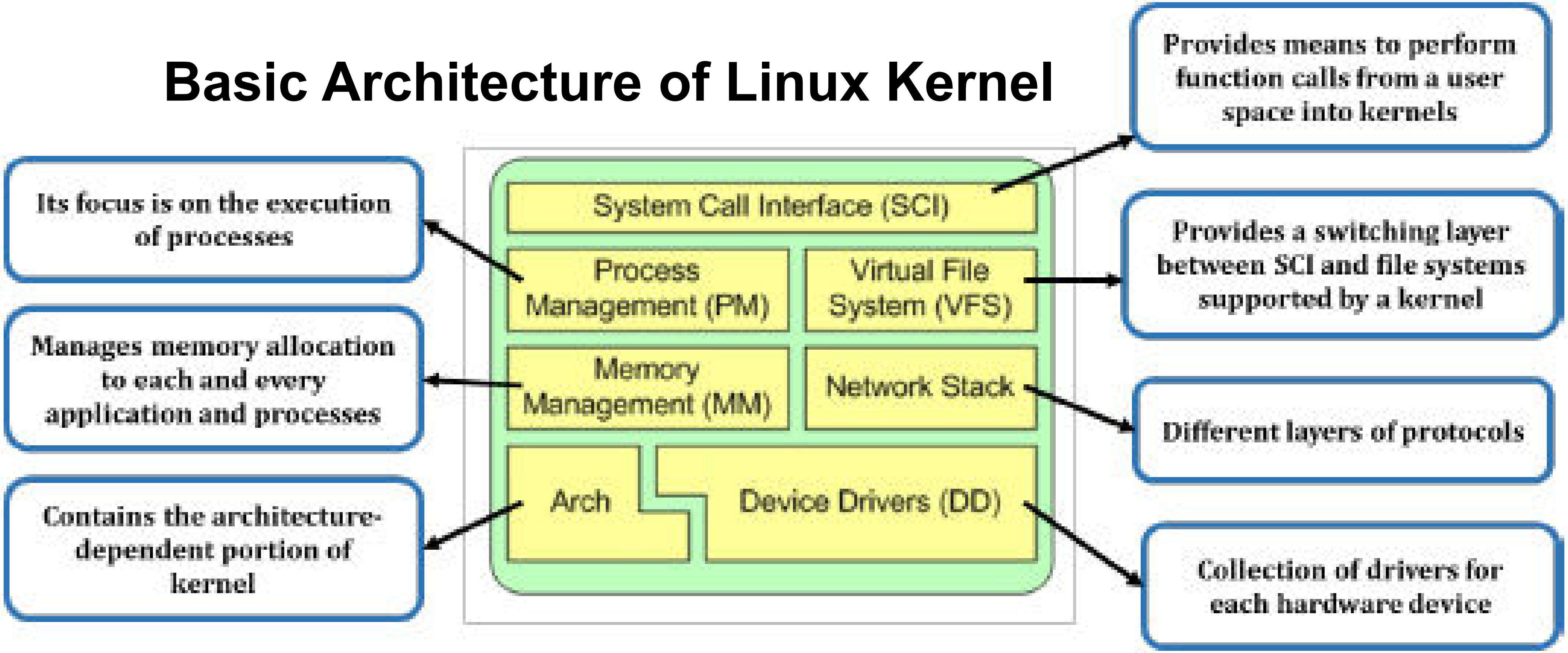
Linux is the most loved and open-source operating system, trusted by millions of people across the world

Examples of Linux commands

COMMAND	DESCRIPTION
locate	Fast searching command that displays directory and files on a new line
cd directory_name	Executing the command Cd songs will take the user to directory songs
cp file1 file2	Copies files and directories. In this case, file1 is copied to file2
pwd	Will display your current working directory
export	Converts the file to ad different format
man command	Shows the user the “manual” of the command. For example, typing man cp will show the manual of the command cp, with the parameters to be used
ls	List the files and directories, within a directory
rm file_name	Used to delete files
mv	Used to move a file or directory to specified location. mv /home1/ras/Desktop/bang /home/bas/Desktop/games will move the file namely bang from desktop to a directory called games, present on the desktop

Compiling a Linux kernel

Basic Architecture of Linux Kernel



Its focus is on the execution of processes

Manages memory allocation to each and every application and processes

Contains the architecture-dependent portion of kernel

Provides means to perform function calls from a user space into kernels

Provides a switching layer between SCI and file systems supported by a kernel

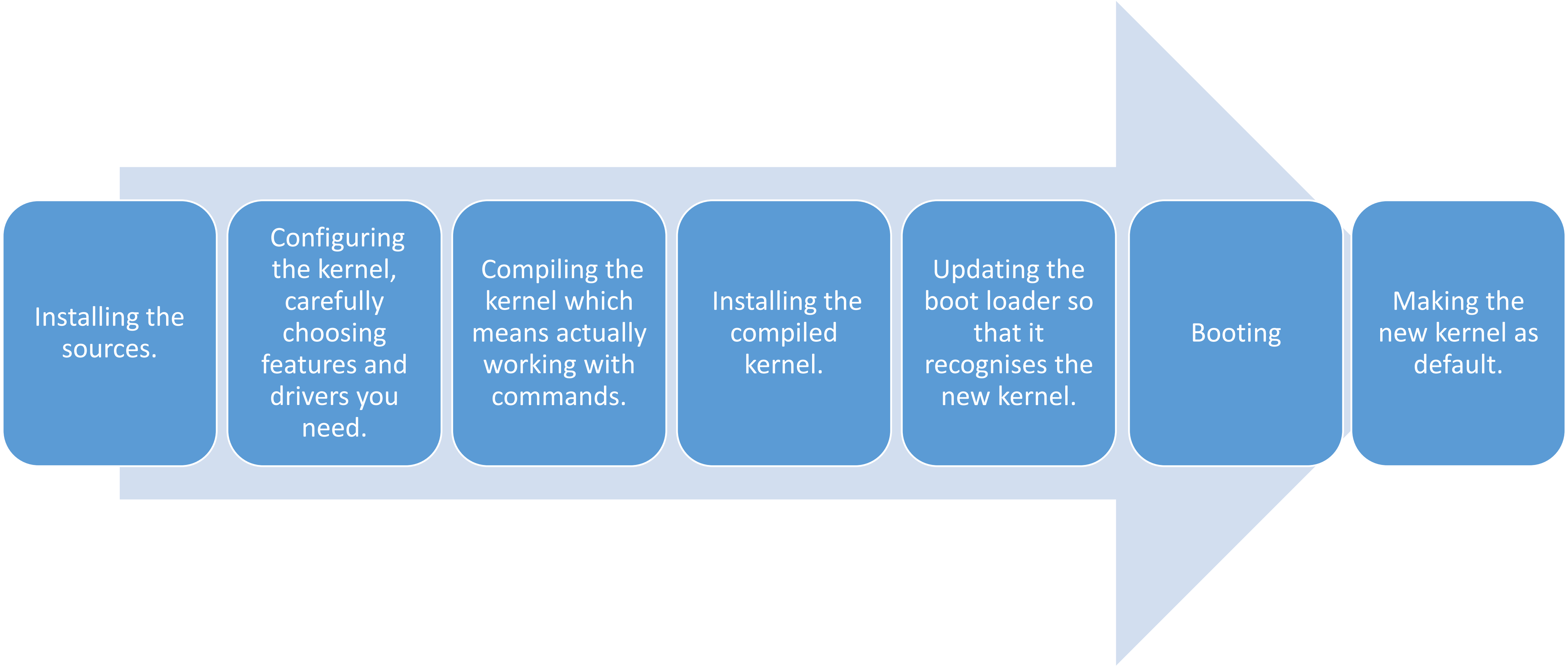
Different layers of protocols

Collection of drivers for each hardware device

Why should you recompile your Linux kernel?

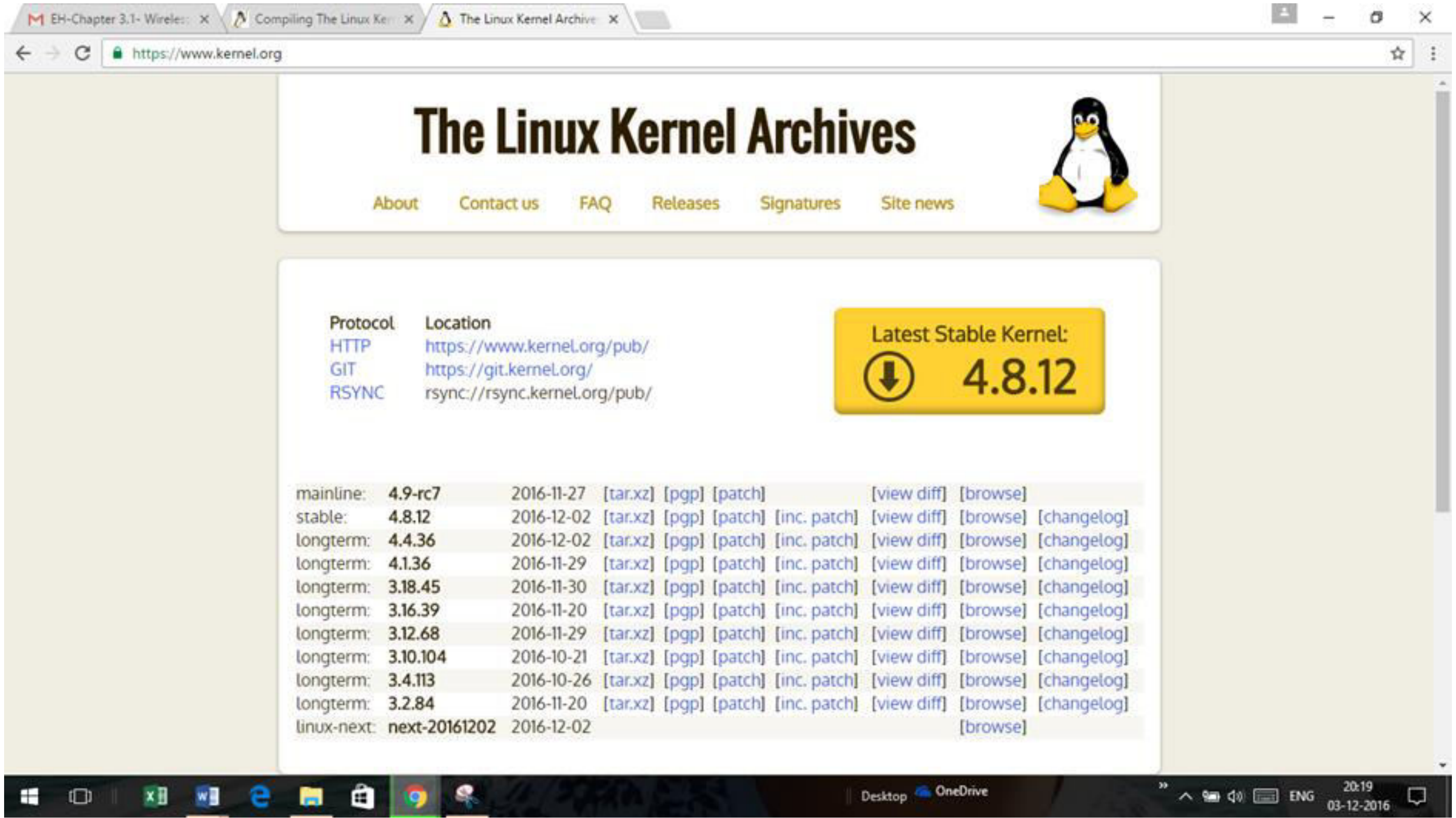
1. To fix any issues in the drivers of the current kernel
2. To customize your kernel
3. To optimize your kernel

Step-by-step process for compiling a Linux kernel



1. Installing the sources.
2. Configuring the kernel, carefully choosing features and drivers you need.
3. Compiling the kernel which means actually working with commands.
4. Installing the compiled kernel.
5. Updating the boot loader so that it recognises the new kernel.
6. Booting
7. Making the new kernel as default.

Installing the sources



The Linux Kernel Archives

Protocol	Location
HTTP	https://www.kernel.org/pub/
GIT	https://git.kernel.org/
RSYNC	rsync://rsync.kernel.org/pub/

Latest Stable Kernel:
4.8.12

mainline:	4.9-rc7	2016-11-27	[tar.xz]	[pgp]	[patch]		[view diff]	[browse]	
stable:	4.8.12	2016-12-02	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.4.36	2016-12-02	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	4.1.36	2016-11-29	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	3.18.45	2016-11-30	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	3.16.39	2016-11-20	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	3.12.68	2016-11-29	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	3.10.104	2016-10-21	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	3.4.113	2016-10-26	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
longterm:	3.2.84	2016-11-20	[tar.xz]	[pgp]	[patch]	[inc. patch]	[view diff]	[browse]	[changelog]
linux-next:	next-20161202	2016-12-02						[browse]	

Compiling the kernel

a. To make sure that we start with a clean slate, *make clean* command will help.
b. Execute *make depend* to prepare the dependency list.
c. To compile the main part of the kernel, execute *make bzImage.*
d. Now, to compile the kernel modules, *make modules*

Installing the new kernel

a. Installing the kernel: Normally all the runnable kernels are found in the */boot* directory
b. Installing the kernel modules

The command to install the new kernel is

```
$ sudo dpkg –i linux-headers-4.8.12*.deb linux-image-4.8.12*.deb
```

Reboot the system by executing the command

```
sudo reboot
```

Installing the kernel modules

Kernel modules are usually located in **/lib/modules/<version>**

To install the modules , run the command

make modules_install

Updating the Boot directory

a. Kernel image path file
b. Partition of root directory
c. Kernel parameters
d. Label

Booting the new kernel

This is the first time you are booting your computer with the new kernel.

a. Reboot your computer
b. Choose the new kernel when you see the prompt at the boot loader
c. Quickly read the messages that appear, informing you about any errors that occurred and proceed accordingly to fix them
d. If new kernel hasn’t installed successfully, load the existing kernel and follow the above mentioned process again.

Compiling C Program using GCC Command line in Linux

Yum install gcc gcc-c++ autoconf automake

To perform the same task in Ubuntu/Debian,

Sudo apt-get install build-essential

If you want to know whether the compiler has been installed on your computer and if so, what is its version, the below command will help

gcc - - version

If this command is not found, it means that compiler is not installed and you can install the latest version from the internet.

Compiling C Program using GCC Command line in Linux

Let us consider the famous “Hello world” program code in C, which is given below

```
#include<stdio.h>
void main()
{
Printf(“Hello World!”);
}
```

Installing Linux kernel modules

lsmod – will list all loaded modules

insmod - installs a specific module

modprobe –loads a module and any dependency

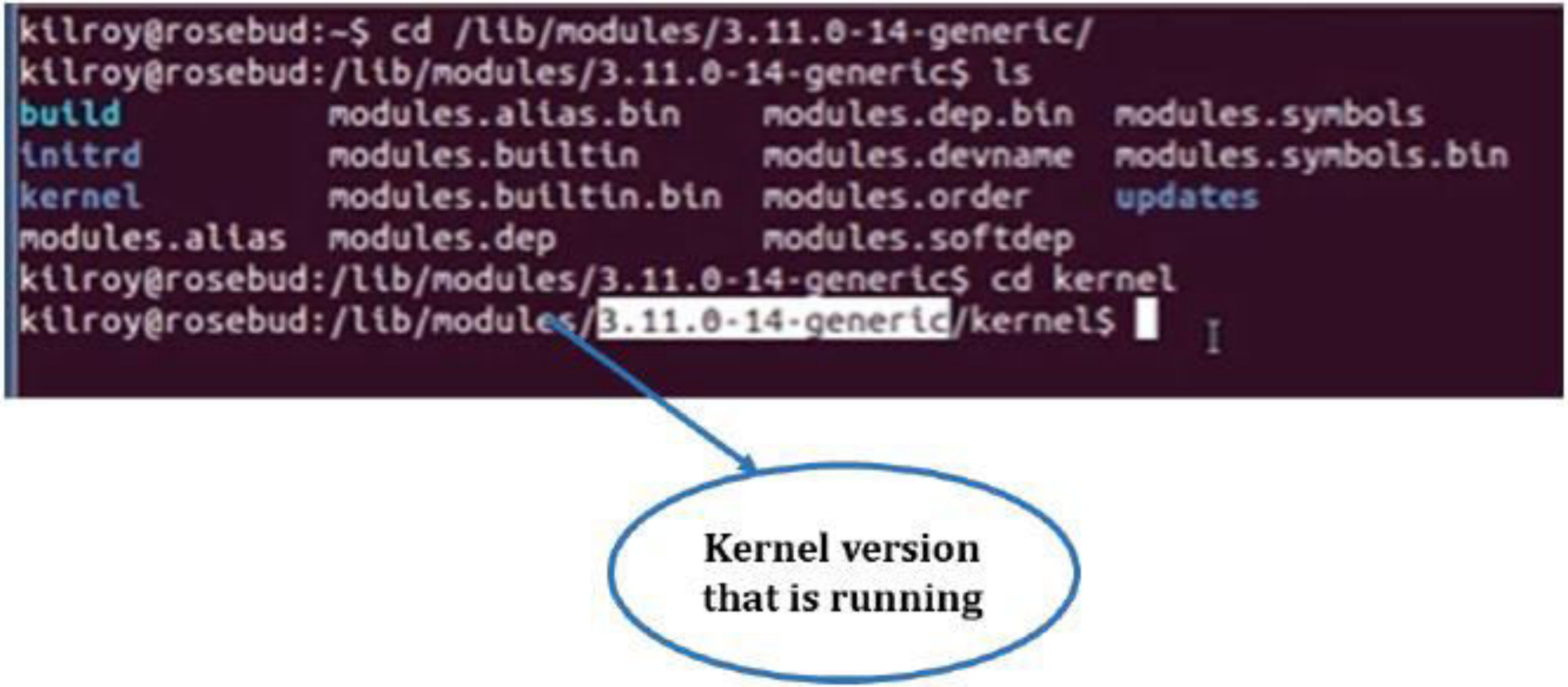
rmmod – removes a module

The step-by step procedure to install a Linux kernel module are given below

1. Open a terminal

2. To view the list of modules that are available for us to build into the kernel, type

cd /lib/modules/3.11.0.14.generic

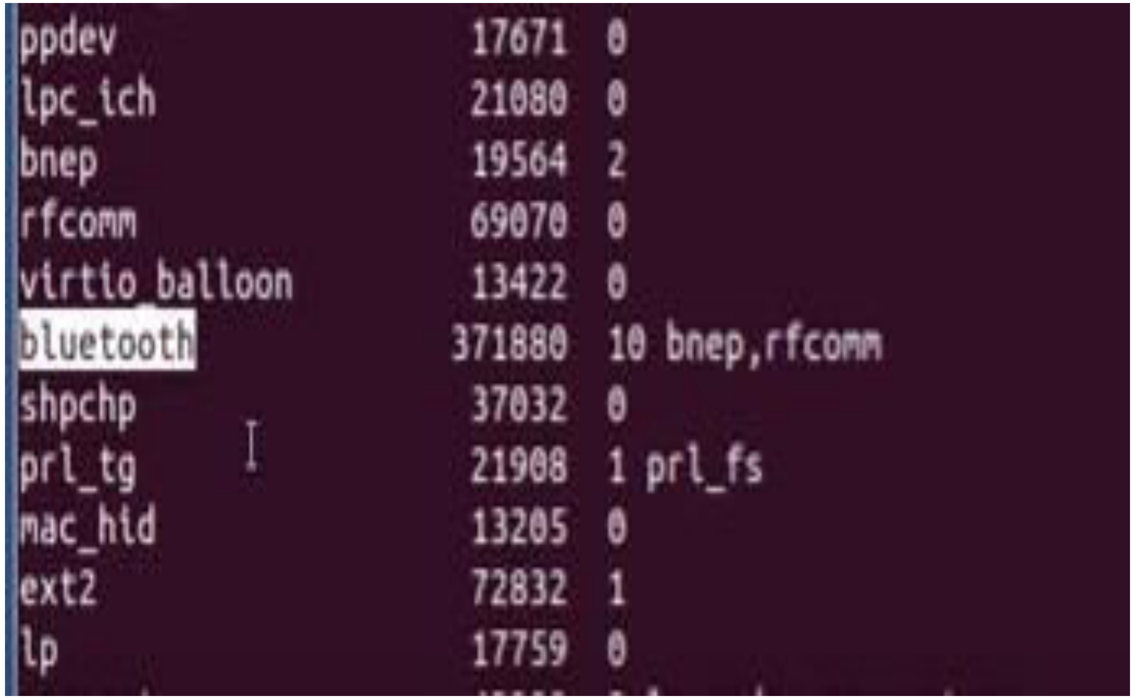


```
kilroy@rosebud:~$ cd /lib/modules/3.11.0-14-generic/
kilroy@rosebud:/lib/modules/3.11.0-14-generic$ ls
build           modules.alias.bin    modules.dep.bin  modules.symbols
initrd          modules.builtin      modules.devname  modules.symbols.bin
kernel          modules.builtin.bin  modules.order    updates
modules.alias   modules.dep          modules.softdep
kilroy@rosebud:/lib/modules/3.11.0-14-generic$ cd kernel
kilroy@rosebud:/lib/modules/3.11.0-14-generic/kernel$ 
```

Kernel version
that is running

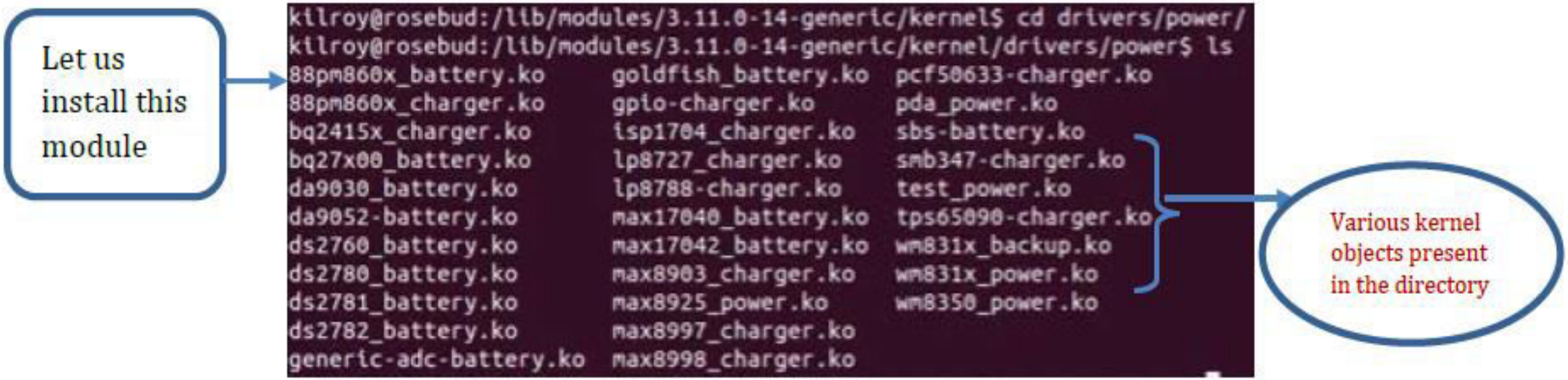
3. Typing command *ls* at this stage will display all the directories present in the kernel.
4. Let us get inside a specific directory for example *drivers* and sub directory *power* within it. Refer the below screen shot

A part of the screenshot is shown below. The highlighted one in the figure is the module for Bluetooth functionality.



```
ppdev                 17671  0
lpc_ich               21080  0
bnep                  19564  2
rfcomm                69070  0
virtio_balloon        13422  0
bluetooth            371880  10 bnep,rfcomm
shpchp                37032  0
prl_tg                21908  1 prl_fs
mac_hid               13205  0
ext2                  72832  1
lp                    17759  0
```

5. Now, to install a module, let us see a list of modules that are available for us to install, from the *power* directory. To display the list, type *ls.*



```
kilroy@rosebud:/lib/modules/3.11.0-14-generic/kernel$ cd drivers/power/
kilroy@rosebud:/lib/modules/3.11.0-14-generic/kernel/drivers/power$ ls
88pm860x_battery.ko     goldfish_battery.ko  pcf50633-charger.ko
88pm860x_charger.ko     gpio-charger.ko      pda_power.ko
bq2415x_charger.ko      isp1704_charger.ko   sbs-battery.ko
bq27x00_battery.ko      lp8727_charger.ko    smb347-charger.ko
da9030_battery.ko       lp8788-charger.ko    test_power.ko
da9052-battery.ko       max17040_battery.ko  tps65090-charger.ko
ds2760_battery.ko       max17042_battery.ko  wm831x_backup.ko
ds2780_battery.ko       max8903_charger.ko   wm831x_power.ko
ds2781_battery.ko       max8925_power.ko     wm8350_power.ko
ds2782_battery.ko       max8997_charger.ko
generic-adc-battery.ko  max8998_charger.ko
```

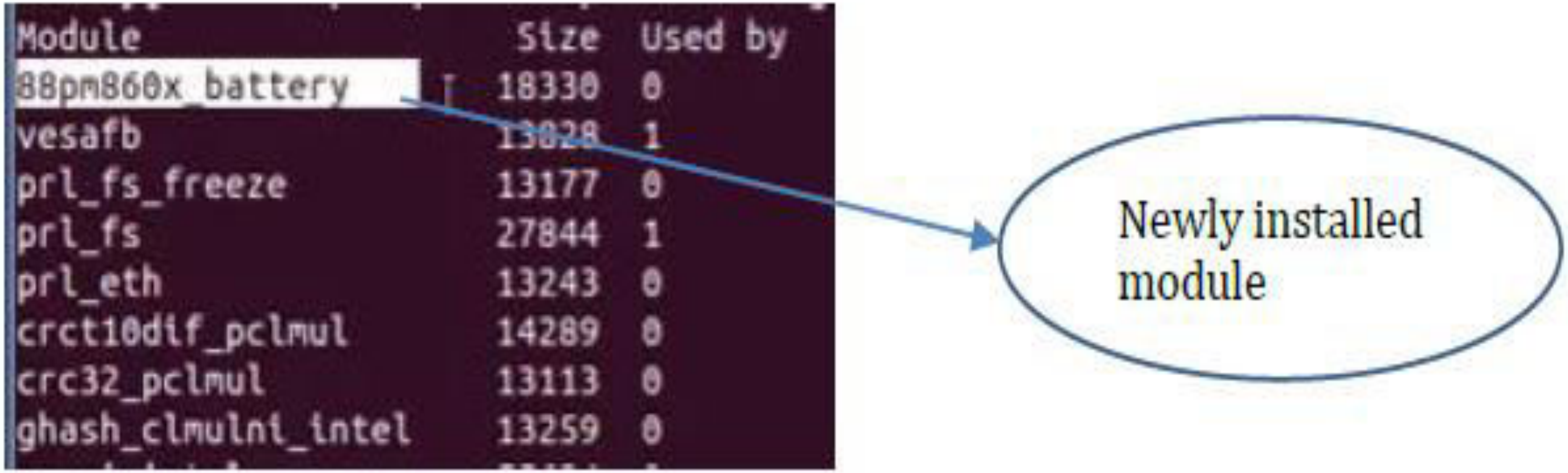
Let us install this module

Various kernel objects present in the directory

6. Let us assume that we want to install the module 88pm860x_battery.ko.

7. If you are not logged to the terminal as root, do so by typing the *sudo* command.

8. Now, type *insmod 88pm860x_battery.ko.*
9. The particular module will be inserted and to verify this, let us execute *lsmod and* we will find the module we just inserted in the list as shown below:



```
Module                  Size  Used by
88pm860x_battery       18330  0
vesafb                 13828  1
prl_fs_freeze          13177  0
prl_fs                 27844  1
prl_eth                13243  0
crct10dif_pclmul       14289  0
crc32_pclmul           13113  0
ghash_clmulni_intel    13259  0
```

Linux hardening methods

The term hardening refers to the method of securing a system by minimising the scope of its vulnerability or the area of the system which is exposed to vulnerabilities.

Linux, like any other operating system, suffers from vulnerabilities, which are often exploited by hackers to compromise your data. Let us know some of them.

1. Remote Procedure Calls or RPC
2. Clear Text Services
3. SNMP
4. Open Secure Sockets Layer
5. Vulnerable ports

Quiz / Assessment

1) Linux command that displays the present working directory is ___________			
a) cd	b) pwd	c) export	d) None of the above
2) “Vanilla” and “Distribution kernel” are two types of			
a) Drivers	b) Windows Operating systems	c) Kernels	d) Configuration programs

Intrusion Detection Systems (IDS)

It can be defined as *“a security system that monitors computer systems and network traffic and analyses that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside the organization”*.

- Businesses to expand globally, reaching a large customer base
- Attracting a lot of risks, as they are exposing a significant area of their business to public users over the Internet
- Most organizations are successful in protecting their internal network from attacks by installing firewalls on their networks.
- Both by external and internal users.

Components of Intrusion Detection Systems

Central data processing and analysing engine

- Consists of a database of attack signatures with details of previous attacks.

Sensors

- Their job is to monitor hosts or networks real time.

Response generating mechanism

- Acting quickly, they generate events to countermeasure the attack like resetting the TCP connection, or notifying the network administrators or modifying the Access Control List on the router, to stop the intrusion.

Storage capacity to log events

- Every movement or change in the system will be logged, including the attack events, which can be very helpful to analyse the implications at a later stage.

Types of Intrusion Detection Systems

Host-based IDS (HIDS)

- It is a system-level IDS that monitors a host computer and not the network
- Most effective in identifying and preventing attacks from internal users
- A **system integrity verifier** (SIV) is an example of such a tool which is used to detect changes made in the system files.
- **Tripwire** is a popular SIV tool that is open-source.
- A **Log file monitor or LFM** is used to scan log files to learn more about attacks that have already occurred

Network-based IDS (NIDS)

- these are meant to monitor and analyse network traffic on a particular network segment
- Management and Monitoring Interfaces are used.

Types of Intrusion Detection Systems

IDS Evasion

- Hackers always seem to find a way to gain access to a system, no matter the security implemented on it.
- When IDS seemed as an effective means towards this cause, attackers have found ways to evade or bypass them, to gain unauthorized access to systems
- This itself is a great vulnerability and some of the attacks that occur due to this are:
    - Obfuscation
    - Fragmentation
    - Encryption
    - Denial of service

IDS Evasion tools

Stick

- Can produce around 250 alarms per second.
- It is used to test the strength of IDS by sending a large number of different attacks that causes a DoS

Mendax

- TCP client software package that injects overlapping packet segments in random orders in the form of attack signatures or single typed lines

Fragroute

- Attackers will be able to fragment packets before transmission. Works with NIDS.

Snot

- DOS-based tool that can generate floods of packets using snort rules

Nmap

- Some of the methods used by Nmap to bypass IDS are packet fragmentation, spoof source IP address or source port

Quiz / Assessment

3) A well-known system integrity verifier is?			
a) Log file monitor	b) Tripwire	c) Nmap	d) None of the above
4) An example of an IDS evasion tool that injects overlapping packet segments in a random order in the form of attack signatures is known as			
a) Stick	b) Mendax	c) Fragroute	d) Nmap

Viruses and Worms

- Viruses and worms are a serious threat to information systems.
- They come in different forms and cause varying levels of damage to computers and networks.
- While some attacks only slow down a system’s performance or deny access to a specific website or network resource,
- Some are more serious, bringing an entire system or network to a complete halt.
- A virus inflicts damage on the operating system by either replacing a program or by associating itself with a program
- Thereby modifying the way that program functions.
- On the other hand, worms are self-replicating, malicious computer programs or codes that spread to computers in a network by exploiting a weakness on the target system

Firewalls

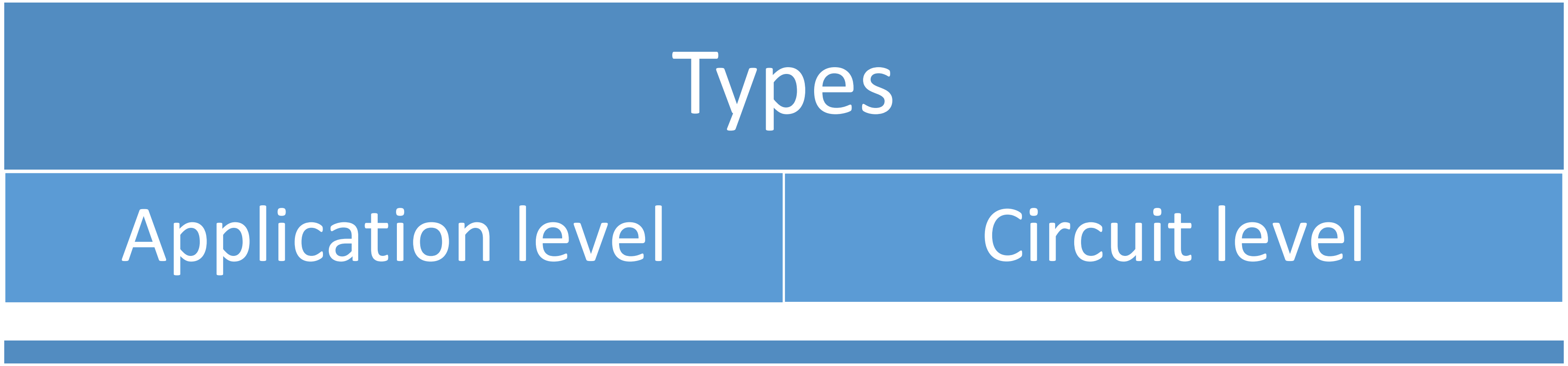
- Firewalls are an essential element of what we call ‘network protectors’ that protect our networks from hackers, malicious codes and intruders.
- They filter traffic coming to your computer or network and depending on the manner in which they have been configured to treat good traffic and bad, they raise an alarm when they sense a troubled situation.
- This gives good control for network administrators to monitor traffic going in and out of your network, thus minimising the probability of possible attacks.

Firewall types

- Proxy Firewalls
- Packet-filtering Firewall
- Stateful Inspection Firewalls

Proxy Firewall

- In a proxy firewall type, both incoming and outgoing transmission is stopped at the firewall.
- If the connection is allowed, the firewall initiates a connection with the destination host on behalf of the originating source host.

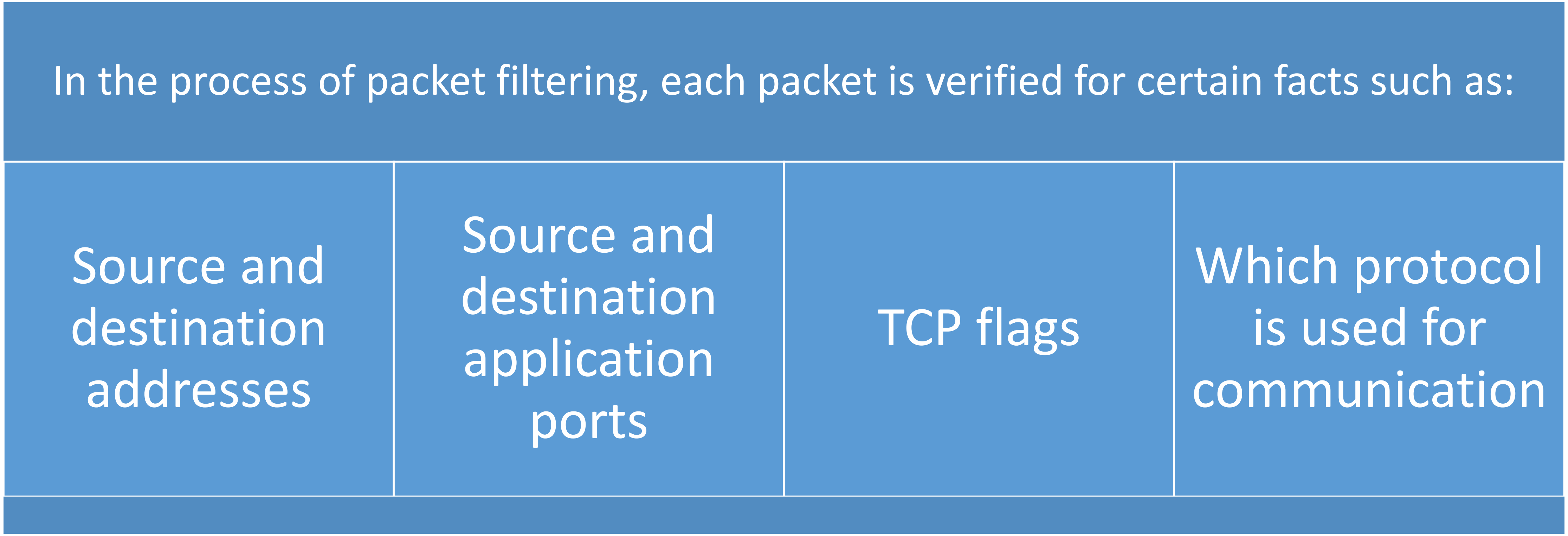


Types

Application level

Circuit level

Packet filtering firewall



In the process of packet filtering, each packet is verified for certain facts such as:

- Source and destination addresses
- Source and destination application ports
- TCP flags
- Which protocol is used for communication

Stateful Inspection Firewalls

- From our discussion on packet filtering firewall, we know that it doesn’t maintain the state information about active sessions.
- This issue is addressed by Stateful Inspection Firewalls, which store this information in what is called the ‘*dynamic state tables*’.
- This will prevent an attacker trying to pose like a part of an already existing communication session, from passing through a firewall.

Quiz / Assessment

5) Which of the below is not a type of firewall?

a) Proxy Firewalls	b) Packet-filtering Firewall	c) Stateful Inspection Firewalls	d) TCP flags

6) Stateful inspection firewalls store state information about communication sessions in a table called the ___________.

a) Access Control List	b) Dynamic state tables	c) MAC address	d) None of the above

e-References & External Resources

- How to install a Linux kernel in Ubuntu, http://ubuntuhandbook.org/index.php/2016/05/install-linux-kernel-4-6-ubuntu-16-04/
- How to compile a C Program using GCC, http://cs-fundamentals.com/c-programming/how-to-compile-c-program-using-gcc.php
- Linux Hardening methods, http://www.softpanorama.org/Commercial_linuxes/Security/hardening.shtml
- Tips on Linux Security https://www.cyberciti.biz/tips/linux-security.html
- Intrusion Detection Systems, https://www.sans.org/reading-room/whitepapers/detection/intrusion-detection-evasion-attackers-burglar-alarm-1284

External Resources

1. Ronald L.Krutz and Russel Dean Vines (2007), *The CEH Prep Guide,* Wiley Publications

Activities

Online Activity
(30min)

Description 1:
Write an assignment on how to download, configure and compile a Linux kernel.

Description 2:
Prepare a power point presentation on the different types of firewalls and how they work.

Thank You