FUNDAMENTAL OF CYBER SECURITY

TCS-492

UNIT-2

Cyber Security G2

WhatsApp group

KNOW YOUR MENTOR

SIDDHANT THAPLIYAL

B.Tech(C.S.E.),M.Tech(C.S.E.)

Ph.D.(Pursuing), Cyber Security in IoT Devices with Machine Learning

Profile Link : https://sites.google.com/view/siddhant-thapliyal/home

COURSE OUTCOME

After completion of the course the students will be able to:

CO1: Explain the three pillars of cyber security, types of hackers and penetration testing.

CO2: . Implement the scripting concepts used in cyber security.

CO3: Use the netcat, ping and wireshark tools to analyze the security of network.

CO4: Use the Javascript, php, sql to analyze the web security.

CO5: Explain the use of cyber security protocols for cyber threats.

CO6: Analyze the security level of web applications.

What is Linux?

Linux distribution is an operating system that is made up of a collection of software based on Linux kernel or you can say distribution contains the Linux kernel and supporting libraries and software. And you can get Linux based operating system by downloading one of the Linux distributions and these distributions are available for different types of devices like embedded devices, personal computers, etc. Around 600 + Linux Distributions are available and some of the popular Linux distributions are:

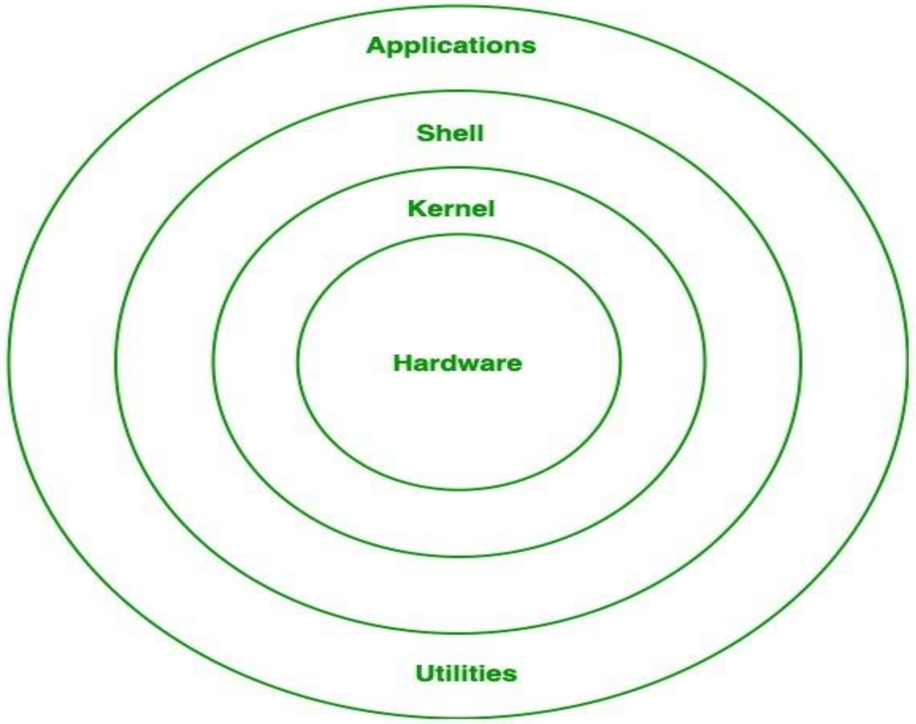
- MX Linux
- Manjaro
- Linux Mint
- elementary
- Ubuntu
- Debian
- Solus
- Fedora
- openSUSE
- Deepina

UNIT – II TOPICS

Unit 2: Linux Basics and Scripting for Ethical Hacking Bash, Linux commands, man page, Adding and deleting, users and adding them to sudo group, switching users, creating, copying, moving and removing file, Writing and appending text to a file, File permissions, working with editors, grep, cut command, Starting and stopping services, Automating tasks with cron jobs Introduction to Bash Scripting - Basics of Bash or Shell Scripting, conditional statements, loops, Manipulating files

Introduction to Python - Basics of Python, conditional statements, loops,list, tuple, dictionary, functions

Architecture of Linux



Kernel

Kernel is the core of the Linux based operating system. It virtualizes the common hardware resources of the computer to provide each process with its virtual resources. This makes the process seem as if it is the sole process running on the machine. The kernel is also responsible for preventing and mitigating conflicts between different processes. Different types of the kernel are:

- Monolithic Kernel
- Hybrid kernels
- Exo kernels
- Micro kernels

System Library

Linux uses system libraries, also known as shared libraries, to implement various functionalities of the operating system. These libraries contain pre-written code that applications can use to perform specific tasks. By using these libraries, developers can save time and effort, as they don’t need to write the same code repeatedly. System libraries act as an interface between applications and the kernel, providing a standardized and efficient way for applications to interact with the underlying system.

Shell

The shell is the user interface of the Linux Operating System. It allows users to interact with the system by entering commands, which the shell interprets and executes. The shell serves as a bridge between the user and the kernel, forwarding the user's requests to the kernel for processing. It provides a convenient way for users to perform various tasks, such as running programs, managing files, and configuring the system.

Hardware Layer:

The hardware layer encompasses all the physical components of the computer, such as RAM (Random Access Memory), HDD (Hard Disk Drive), CPU (Central Processing Unit), and input/output devices. This layer is responsible for interacting with the Linux Operating System and providing the necessary resources for the system and applications to function properly. The Linux kernel and system libraries enable communication and control over these hardware components, ensuring that they work harmoniously together.

System Utility

System utilities are essential tools and programs provided by the Linux Operating System to manage and configure various aspects of the system. These utilities perform tasks such as installing software, configuring network settings, monitoring system performance, managing users and permissions, and much more. System utilities simplify system administration tasks, making it easier for users to maintain their Linux systems efficiently.

Bash in Linux

The Bash is a command language interpreter as well as a programming language. It supports variables, functions, and flow control, like other programming languages. It can also read and execute the commands from a file, which is called a shell script.

Linux Commands

pwd Command

The pwd command is used to display the location of the current working directory.

Syntax:

```
pwd
```

mkdir Command

The mkdir command is used to create a new directory under any directory.

Syntax:

```
mkdir <directory name>
```

Linux Commands

rmdir Command

The rmdir command is used to delete a directory.

Syntax:

```
rmdir <directory name>
```

ls Command

The ls command is used to display a list of content of a directory.

Syntax:

```
ls
```

Linux Commands

cd Command

The cd command is used to change the current directory.

Syntax:

```
cd <directory name>
```

touch Command

The 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> ....
```

Linux Commands

cat Command

The 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>
```

Linux Commands

rm Command

The rm command is used to remove a file.

Syntax:

```
rm <file name>
```

cp Command

The 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>

Linux Commands

mv Command

The mv command is used to move a file or a directory form one location to another location.

Syntax:

1. mv <file name> <directory path>

rename Command

The 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

   rename 's/\.txt$/\.pdf/' *.txt

Linux Commands

head Command

The head command is used to display the content of a file. It displays the first 10 lines of a file.

Syntax:

```
head <file name>
```

tail Command

The tail command is similar to the head command. The difference between both commands is that it displays the last ten lines of the file content. It is useful for reading the error message.

Syntax:

```
tail <file name>
```

Linux Commands

tac Command

The tac command is the reverse of cat command, as its name specified. It displays the file content in reverse order (from the last line).

Syntax:

```
tac <file name>
```

less Command

The less command is similar to the more command. It also includes some extra features such as 'adjustment in width and height of the terminal.' Comparatively, the more command cuts the output in the width of the terminal.

Syntax:

```
less <file name>
```

Linux Commands

more command

The more command is quite similar to the cat command, as it is used to display the file content in the same way that the cat command does. The only difference between both commands is that, in case of larger files, the more command displays screenful output at a time.

In more command, the following keys are used to scroll the page:

ENTER key: To scroll down page by line.

Space bar: To move to the next page.

b key: To move to the previous page.

/ key: To search the string.

Syntax:

```
1.   more <file name>
```

Linux Commands

su Command

The su command provides administrative access to another user. In other words, it allows access of the Linux shell to another user.

Syntax:

1. su <user name>

id Command

The id command is used to display the user ID (UID) and group ID (GID).

Syntax:

1. id

Linux Commands

useradd Command

The useradd command is used to add or remove a user on a Linux server.

Syntax:

```
useradd  username
```

passwd Command

The passwd command is used to create and change the password for a user.

Syntax:

```
1.    passwd <username>
```

Linux Commands

groupadd Command

The groupadd command is used to create a user group.

Syntax:

```
1.    groupadd <group name>
```

Summary

Users and Groups Management:

1. **Adding Users**: sudo adduser username
2. **Adding Users to sudo Group**: sudo usermod -aG sudo username
3. **Switching Users**: su username
4. **Changing Current User's Password**: passwd
5. **Deleting Users**: sudo deluser username

Summary

File Management:

1. **Creating Files**: touch filename
2. **Copying Files**: cp source_file destination
3. **Moving/Renaming Files**: mv source_file destination
4. **Removing Files**: rm filename
5. **Creating Directories**: mkdir directory_name
6. **Copying Directories**: cp -r source_dir destination
7. **Moving/Renaming Directories**: mv source_dir destination
8. **Removing Directories**: rm -r directory_name

Summary

File Permissions:

1. **Changing Permissions**: chmod permissions filename/directory
2. **Viewing Permissions**: ls -l

Searching:

1. **Searching Text in Files**: grep "search_text" filename
2. **Searching with Cut**: cut -d delimiter -f field_number filename

Summary

Services Management:

1. **Starting a Service**: sudo systemctl start service_name
2. **Stopping a Service**: sudo systemctl stop service_name
3. **Restarting a Service**: sudo systemctl restart service_name

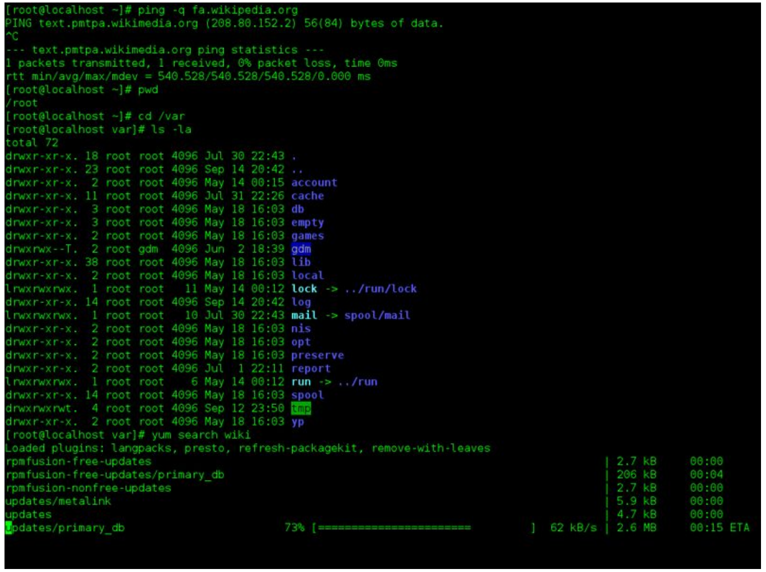
Automating Tasks:

1. **Cron Jobs**: Edit cron jobs with crontab -e, then add entries in the cron file based on the schedule.

Introduction to Bash Scripting

A bash script is a file containing a sequence of commands that are executed by the bash program line by line. It allows you to perform a series of actions, such as navigating to a specific directory, creating a folder, and launching a process using the command line.

By saving these commands in a script, you can repeat the same sequence of steps multiple times and execute them by running the script.



```
[root@localhost ~]# ping -q fa.wikipedia.org
PING text.pmtpa.wikimedia.org (208.80.152.2) 56(84) bytes of data.
^C
--- text.pmtpa.wikimedia.org ping statistics ---
1 packets transmitted, 1 received, 0% packet loss, time 0ms
rtt min/avg/max/mdev = 540.528/540.528/540.528/0.000 ms
[root@localhost ~]# pwd
/root
[root@localhost ~]# cd /var
[root@localhost var]# ls -la
total 72
drwxr-xr-x. 18 root root 4096 Jul 30 22:43 .
drwxr-xr-x. 23 root root 4096 Sep 14 20:42 ..
drwxr-xr-x.  2 root root 4096 May 14 00:15 account
drwxr-xr-x. 11 root root 4096 Jul 31 22:26 cache
drwxr-xr-x.  3 root root 4096 May 18 16:03 db
drwxr-xr-x.  3 root root 4096 May 18 16:03 empty
drwxr-xr-x.  2 root root 4096 May 18 16:03 games
drwxrwx--T.  2 root gdm  4096 Jun  2 18:39 gdm
drwxr-xr-x. 38 root root 4096 May 18 16:03 lib
drwxr-xr-x.  2 root root 4096 May 18 16:03 local
lrwxrwxrwx.  1 root root   11 May 14 00:12 lock -> ../run/lock
drwxr-xr-x. 14 root root 4096 Sep 14 20:42 log
lrwxrwxrwx.  1 root root   10 Jul 30 22:43 mail -> spool/mail
drwxr-xr-x.  2 root root 4096 May 18 16:03 nis
drwxr-xr-x.  2 root root 4096 May 18 16:03 opt
drwxr-xr-x.  2 root root 4096 May 18 16:03 preserve
drwxr-xr-x.  2 root root 4096 Jul  1 22:11 report
lrwxrwxrwx.  1 root root    6 May 14 00:12 run -> ../run
drwxr-xr-x. 14 root root 4096 May 18 16:03 spool
drwxrwxrwt.  4 root root 4096 Sep 12 23:50 tmp
drwxr-xr-x.  2 root root 4096 May 18 16:03 yp
[root@localhost var]# yum search wiki
Loaded plugins: langpacks, presto, refresh-packagekit, remove-with-leaves
rpmfusion-free-updates                                                    | 2.7 kB     00:00
rpmfusion-free-updates/primary_db                                         | 206 kB     00:04
rpmfusion-nonfree-updates                                                 | 2.7 kB     00:00
updates/metalink                                                          | 5.9 kB     00:00
updates                                                                   | 4.7 kB     00:00
updates/primary_db                         73% [=======================          ]  62 kB/s | 2.6 MB     00:15 ETA
```

Advantages of Bash scripting

Bash scripting is a powerful and versatile tool for automating system administration tasks, managing system resources, and performing other routine tasks in Unix/Linux systems. Some advantages of shell scripting are:

- **Automation**: Shell scripts allow you to automate repetitive tasks and processes, saving time and reducing the risk of errors that can occur with manual execution.
- **Portability**: Shell scripts can be run on various platforms and operating systems, including Unix, Linux, macOS, and even Windows through the use of emulators or virtual machines.
- **Flexibility**: Shell scripts are highly customizable and can be easily modified to suit specific requirements. They can also be combined with other programming languages or utilities to create more powerful scripts.

Advantages of Bash scripting

- **Accessibility**: Shell scripts are easy to write and don't require any special tools or software. They can be edited using any text editor, and most operating systems have a built-in shell interpreter.
- **Integration**: Shell scripts can be integrated with other tools and applications, such as databases, web servers, and cloud services, allowing for more complex automation and system management tasks.
- **Debugging**: Shell scripts are easy to debug, and most shells have built-in debugging and error-reporting tools that can help identify and fix issues quickly.

Basics of Bash or Shell Scripting,

How to Create and Execute Bash scripts

Script naming conventions

By naming convention, bash scripts end with .sh. However, bash scripts can run perfectly fine without the sh extension.

Adding the Shebang

Bash scripts start with a shebang. Shebang is a combination of bash # and bang ! followed by the bash shell path. This is the first line of the script. Shebang tells the shell to execute it via bash shell. Shebang is simply an absolute path to the bash interpreter.

```
#!/bin/bash
```

Basics of Bash or Shell Scripting,

You can find your bash shell path (which may vary from the above) using the command:

```
which bash
```

Conditional statements

```
if [ expression ];
then
statements
fi
```

loops

Manipulating files

Introduction to Python

Basics of Python

Conditional statements

Loops

List

Lists are the simplest containers that are an integral part of the Python language. Lists need not be homogeneous always which makes it the most powerful tool in Python. A single list may contain DataTypes like Integers, Strings, as well as Objects. Lists are mutable, and hence, they can be altered even after their creation.

Functions for List

S. No.	Function	Description
1	Append()	Add an element to the end of the list
2	Extend()	Add all elements of a list to another list
3	Insert()	Insert an item at the defined index
4	Remove()	Removes an item from the list
5	Clear()	Removes all items from the list
6	Index()	Returns the index of the first matched item
7	Count()	Returns the count of the number of items passed as an argument
8	Sort()	Sort items in a list in ascending order
9	Reverse()	Reverse the order of items in the list
10	copy()	Returns a copy of the list

Functions for List

11	**pop()**	Removes and returns the item at the specified index. If no index is provided, it removes and returns the last item.
12	**reduce()**	apply a particular function passed in its argument to all of the list elements stores the intermediate result and only returns the final summation value
13	**sum()**	Sums up the numbers in the list
14	**ord()**	Returns an integer representing the Unicode code point of the given Unicode character
15	**cmp()**	This function returns 1 if the first list is "greater" than the second list
16	**max()**	return maximum element of a given list
17	**min()**	return minimum element of a given list
18	**all()**	Returns true if all element is true or if the list is empty
19	**any()**	return true if any element of the list is true. if the list is empty, return false
20	**len()**	Returns length of the list or size of the list

Functions for List

21	**enumerate()**	Returns enumerate object of the list
22	**accumulate()**	apply a particular function passed in its argument to all of the list elements returns a list containing the intermediate results
23	**filter()**	tests if each element of a list is true or not
24	**map()**	returns a list of the results after applying the given function to each item of a given iterable
25	**lambda()**	This function can have any number of arguments but only one expression, which is evaluated and returned.

Tuple

Python Tuple is a collection of objects separated by commas. In some ways, a tuple is similar to a Python list in terms of indexing, nested objects, and repetition but the main difference between both is Python tuple is immutable, unlike the Python list which is mutable.

Creating Python Tuples

There are various ways by which you can create a tuple in Python. They are as follows:

- Using round brackets
- With one item
- Tuple Constructor

Create Tuples using Round Brackets ()

```
var = ("Geeks", "for", "Geeks")
print(var)
```

Create a Tuple With One Item

Python 3.11 provides us with another way to create a Tuple.

```
values : tuple[int | str, ...] = (1,2,4,"Geek")
print(values)
```

Tuple Constructor in Python

```
tuple_constructor = tuple(("dsa", "developement", "deep learning"))

print(tuple_constructor)
```

What is Immutable in Tuples?

Tuples in Python are similar to Python lists but not entirely. Tuples are immutable and ordered and allow duplicate values. Some Characteristics of Tuples in Python.

- We can find items in a tuple since finding any item does not make changes in the tuple.
- One cannot add items to a tuple once it is created.
- Tuples cannot be appended or extended.
- We cannot remove items from a tuple once it is created.

Different Operations Related to Tuples

- Concatenation tuple1 + tuple2
- Nesting tuple3 = (tuple1, tuple2)
- Repetition tuple3 = ('python',)*3
- Slicing tuple1[1:]) (tuple1[::-1]) print(tuple1[2:4])
- Deleting del tuple3
- Finding the length len(tuple2)
- Multiple Data Types with tuples tuple_obj = ("immutable",True,23)
- Conversion of lists to tuples tuple(list1)
- Tuples in a Loop for i in range(n):

```
tup = (tup,)

print(tup)
```

Dictionary

A dictionary in Python is a data structure that stores the value in key:value pairs.

```
Dict = {1: 'Geeks', 2: 'For', 3: 'Geeks'}

print(Dict)
```

Nested Dictionary

```
Dict = {1: 'Geeks', 2: 'For',
        3: {'A': 'Welcome', 'B': 'To', 'C': 'Geeks'}}


print(Dict)
```

Function in Dictionary

S. No.	Method	Description
1	dict.clear()	Remove all the elements from the dictionary
2	dict.copy()	Returns a copy of the dictionary
3	dict.get(key, default = “None”)	Returns the value of specified key
4	dict.items()	Returns a list containing a tuple for each key value pair
5	dict.keys()	Returns a list containing dictionary’s keys
6	dict.update(dict2)	Updates dictionary with specified key-value pairs
7	dict.values()	Returns a list of all the values of dictionary

Function in Dictionary

8	pop()	Remove the element with specified key
9	popItem()	Removes the last inserted key-value pair
10	dict.setdefault(key,default= "None")	set the key to the default value if the key is not specified in the dictionary
11	dict.has_key(key)	returns true if the dictionary contains the specified key.

Functions

References

- https://www.geeksforgeeks.org/introduction-to-linux-operating-system/
- https://www.javatpoint.com/linux-bash
- https://www.javatpoint.com/linux-commands
- https://www.freecodecamp.org/news/bash-scripting-tutorial-linux-shell-script-and-command-line-for-beginners/
- https://www.javatpoint.com/bash
- https://www.geeksforgeeks.org/bash-scripting-introduction-to-bash-and-bash-scripting/
-