PRACTICAL-8

AIM:

a) Perform Port Scanning, File Transfer, Client-server chat and Basic Webserver implementation using netcat.

b) Find the service running on the particular port using netcat.

THEORY:

NetCat:

- netcat (often abbreviated to nc) is a computer networking utility for reading from and writing to network connections using TCP or UDP.
- The command is designed to be a dependable back-end that can be used directly or easily driven by other programs and scripts.
- At the same time, it is a feature-rich network debugging and investigation tool, since it can produce almost any kind of connection its user could need and has a number of built-in capabilities.
- Its list of features includes port scanning, transferring files, and port listening, and it can be used as a backdoor.
- The original netcat's features include:

  i. Outbound or inbound connections, TCP or UDP, to or from any ports
  ii. Full DNS forward/reverse checking, with appropriate warnings
  iii. Ability to use any local source port
  iv. Ability to use any locally configured network source address
  v. Built-in port-scanning capabilities, with randomization
  vi. Built-in loose source-routing capability
  vii. Can read command line arguments from standard input
  viii. Slow-send mode, one line every N seconds
  ix. Hex dump of transmitted and received data
  x. Optional ability to let another program service establish connections
  xi. Optional telnet-options responder

Port Scanning:

- This may useful to know which ports are open and running services on a target machine.
- Try the nc / netcat command as follow.
- The -z flag can be used to tell nc to report open ports, rather than initiate a connection.
- You need to specify hostname / ip along with the port range to limit and speedup operation:

○ Command: nc -z -v hostname port-range
○ Ex: nc -z -v localhost 80

```
┌──(user㉿kali)-[~/Desktop]
└─$ sudo nc -z -v 192.168.43.52 80
DESKTOP-S5UT1SO [192.168.43.52] 80 (http) : Connection refused
```

File Transfer:

- The nc (netcat) command can be used to transfer arbitrary data over the network.
- It represents a quick way for Linux administrators to transfer data without the need for an additional data transfer services such as FTP, HTTP, SCP etc.
- This config will show you an example on how to transfer data between to network hosts.
- We will be transferring data myfile.txt file from a localhost to a destination host with an IP address 10.1.1.2.

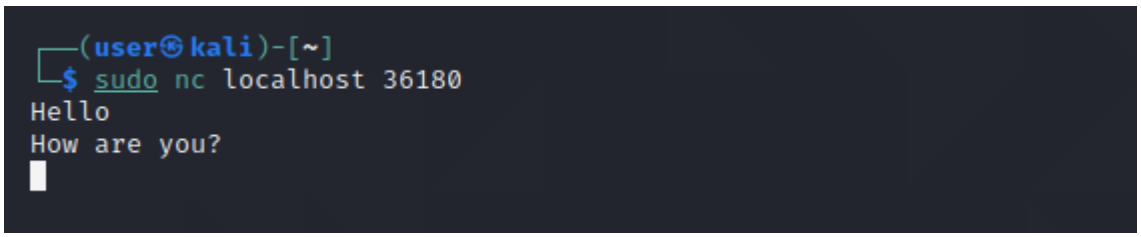
○ Command (Transmitter): nc -v -l -p port < filename
○ Command (Reciever): nc -v hostname port > filename
○ Ex (Transmitter): nc -v -l -p 36180 < hello.txt
○ Ex (Receiver): nc -v localhost 36180 > hello.txt

```
┌──(user㉿kali)-[~/Desktop]
└─$ sudo nc -v -l -p 36180 < hello.txt
listening on [any] 36180 ...
connect to [127.0.0.1] from localhost [127.0.0.1] 36932
```

Client-Server Chat:

- To create a simple chat we need two instances of netcat, one to listen for incoming connections (the server) and another one to start the connection.

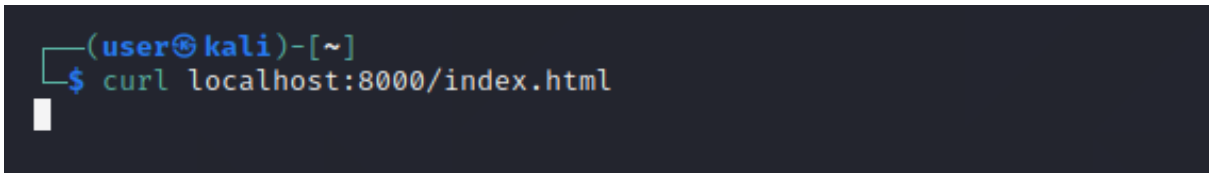
○ Command (Server): nc -l -p port
○ Command (Client): nc hostname port
○ Ex (Server): nc -l -p 36180
○ Ex (Client) : nc localhost 36180



Basic Webserver Implementation:

- The netcat tool nc can operate as a TCP client. Because HTTP works over TCP, nc can be used as an HTTP server!
- Because nc is a UNIX tool, we can use it to make custom web servers: servers which return any HTTP headers you want, servers which return the response very slowly, servers which return invalid HTTP, etc.
- You can also use nc as a quick-and-dirty static file server.

○ Command: nc -l -p 8000
○ Ex: nc -l -p 8000



- After we start listening on port 8000, we can use command "curl localhost:8000/index.html" to send request on port 8000 on localserver.
- We can confirm that by looking at response in server side where GET request will be reflected.

CONCLUSION:

In this practical, we learned about NetCat which has many functionality and it is easy to use tool. We first transferred a file using nc and then implemented a chat server. We also implemented basic web server with single command.