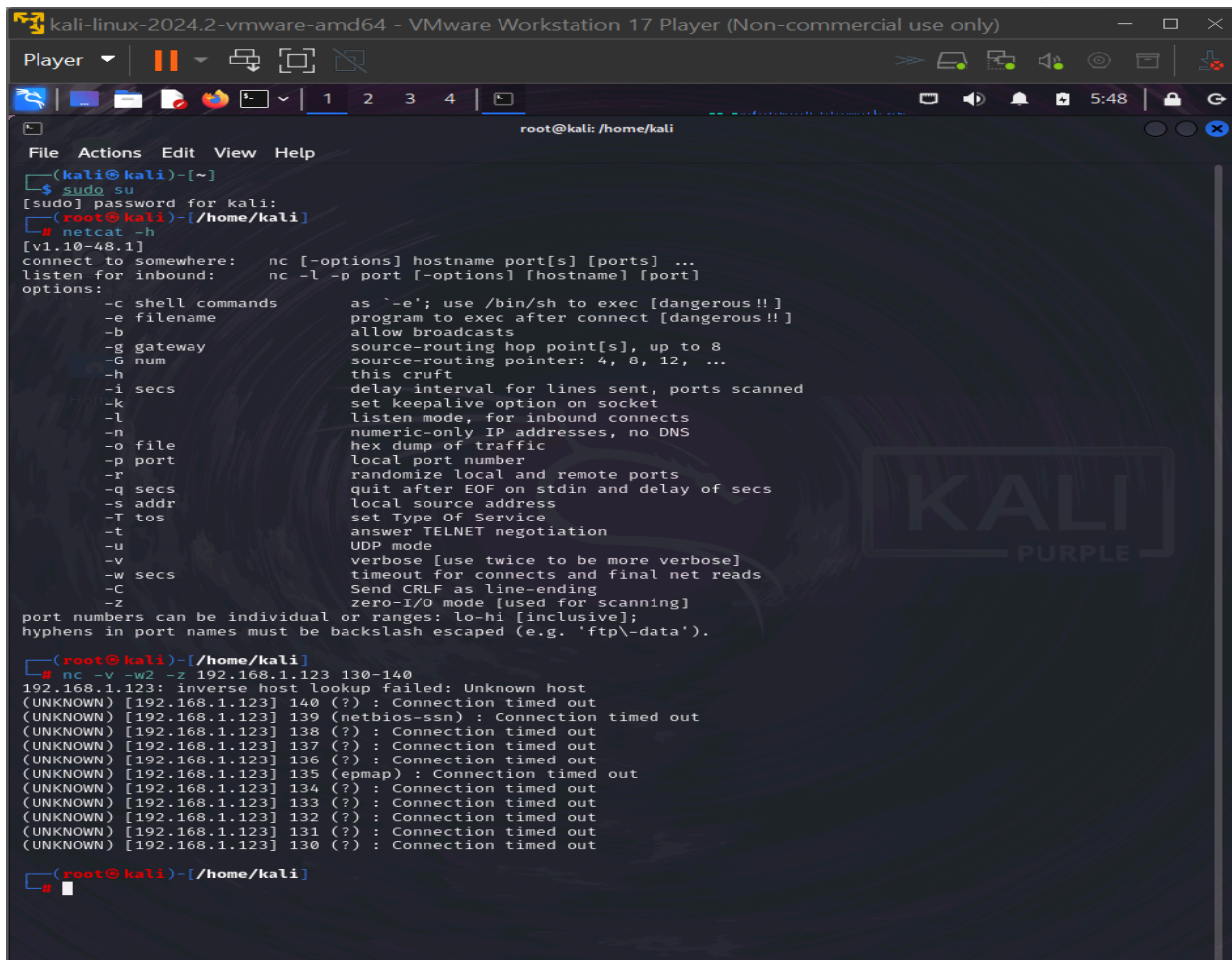


NETCAT AND ITS USES

Tools : NETCAT on KALI

NETCAT (often abbreviated as nc) is a versatile networking utility used for reading from and writing to network connections using the TCP or UDP protocol. It is often called the "Swiss Army knife" of networking tools because of its wide range of functionalities, including port scanning, banner grabbing, and data transfer between machines.

Input from netcat :

A screenshot of a Kali Linux terminal window titled "kali-linux-2024.2-vmware-amd64 - VMware Workstation 17 Player (Non-commercial use only)". The terminal shows a user switching to root with 'sudo su', then running 'netcat -h' to display help information. The help text lists various options like -c, -e, -b, -g, -G, -h, -i, -k, -l, -n, -o, -p, -r, -q, -s, -T, -t, -u, -v, -w, -C, and -Z, along with their functions. Finally, the user runs 'nc -v -w2 -z 192.168.1.123 130-140', which results in a series of connection timeout messages for ports 140 through 130 in descending order.

```
kali@kali:~$ sudo su
[sudo] password for kali:
root@kali:~/home/kali# netcat -h
[v1.10-48.1]
connect to somewhere: nc [-options] hostname port[s] [ports] ...
listen for inbound:   nc -l -p port [-options] [hostname] [port]
options:
  -c shell commands          as '-e'; use /bin/sh to exec [dangerous!!]
  -e filename               program to exec after connect [dangerous!!]
  -b allow broadcasts       allow broadcasts
  -g source-routing hop point[s], up to 8
  -G num                    source-routing pointer: 4, 8, 12, ...
  -h this cruft
  -i secs                  delay interval for lines sent, ports scanned
  -k set keepalive option on socket
  -l listen mode, for inbound connects
  -n numeric-only IP addresses, no DNS
  -o file                  hex dump of traffic
  -p port                  local port number
  -r randomize local and remote ports
  -q secs                  quit after EOF on stdin and delay of secs
  -s addr                  local source address
  -T tos                    set Type Of Service
  -t answer TELNET negotiation
  -u UDP mode
  -v verbose [use twice to be more verbose]
  -w secs                  timeout for connects and final net reads
  -C Send CRLF as line-ending
  -Z zero-I/O mode [used for scanning]
port numbers can be individual or ranges: lo-hi [inclusive];
hyphens in port names must be backslash escaped (e.g. 'ftp\data').

root@kali:~/home/kali# nc -v -w2 -z 192.168.1.123 130-140
192.168.1.123: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.123] 140 (?): Connection timed out
(UNKNOWN) [192.168.1.123] 139 (netbios-ssn): Connection timed out
(UNKNOWN) [192.168.1.123] 138 (?): Connection timed out
(UNKNOWN) [192.168.1.123] 137 (?): Connection timed out
(UNKNOWN) [192.168.1.123] 136 (?): Connection timed out
(UNKNOWN) [192.168.1.123] 135 (epmap): Connection timed out
(UNKNOWN) [192.168.1.123] 134 (?): Connection timed out
(UNKNOWN) [192.168.1.123] 133 (?): Connection timed out
(UNKNOWN) [192.168.1.123] 132 (?): Connection timed out
(UNKNOWN) [192.168.1.123] 131 (?): Connection timed out
(UNKNOWN) [192.168.1.123] 130 (?): Connection timed out

root@kali:~/home/kali#
```

Here we would be exploring NETCAT and its uses.

First of we must be root user using the terminal : `sudo su`

It is also good to know that netcat and nc commands are exactly the same.

We will begin by viewing the help information screen by executing the following command: `netcat -h`

Then we will begin by port scanning using netcat. netcat is quite slow and nmap is a far better option for port scanning, but this is just to show you netcat's functionality. This can be done using the following command: `nc -v -w2 -z 192.168.1.123 130-140` In this example, we started a scan of an IP address for a specific port range. As a result of the scan, we found that ports 135 and 139 are open. This target is probably a Windows machine.

We can then perform banner grabbing to determine which version of a service is running. I will demonstrate this on port 22 for SSH. This can be done using the following command: `nc -v -n 139.162.196.104 22`

When connecting to a web server, we can request information in the form of web requests. We can request the header from this server by using the following command when we are connected:

```
nc 192.168.1.206 80
HEAD / HTTP/1.0
```

This will cause the webserver to respond with useful information like server banner, content size, version, time, etc. To retrieve the top level page on the webserver, we can issue the following command:

```
nc 192.168.1.206 80
GET / HTTP/1.0
```

We can also transfer files between two nodes using netcat. This is very handy when interacting with a server through the command line. In this example, we will assume we want to transfer a file to a target which we have remote command execution of. We will begin by setting up a listener on the target host and then connecting to it from the attack box. `nc -vnlp 8080 > received.file`

This opens a listener on the target on port 8080. We will then connect to it on the attack box and transfer the file. `nc 192.168.1.206 8080 < tobe-send.txt`

We can end the connection as the file has now transferred. With this method, it is possible to transfer large files as well with the help of compression tools.

We can open a UDP server using netcat too, using the following command: `netcat -ul -p 7000`

Connect to listener side with this command; `nc -uv 192.168.1.206 7000`

Netcat can also be used to create a basic shell on a remote system on a port. This can be done by executing this command: `netcat -l -p 7777 -e /bin/bash`

This will start a server on port 7777 and will pass all incoming input to bash command and the results will be sent back. This will basically convert the bash program into a server. Netcat can be used to convert any process into a server. We can connect to this bash shell using the following command: `netcat 139.162.196.104 7777`