# What is Netcat and How to use it

Netcat (also known as ‘nc’ or ‘Swiss Army knife’) is a networking utility used for reading or writing from TCP and UDP sockets using an easy interface. NetCat is designed as a Dependable ‘back-end’ device that can be used directly or easily driven by other programs and scripts. Netcat is a treat to network administrators, programmers, and pen-testers as it’s a feature rich network debugging and investigation tool.

In 2000, Netcat was voted the second most functional network security tool. Also, in 2003 and 2006 it gained fourth place in the same category.

That’s much of theory there; now let’s move on how to use netcat

## Using Netcat

There are many features of Netcat and can be used in many ways, but for this tutorial I'll only focus on some fundamental use.

**Opening Netcat:-**

To open netcat simply go to your Shell and enter 'nc'

Code:

```
  lionaneesh@lionaneesh:~$ nc
```

**Output:-**

Code:

```



  This is nc from the netcat-openbsd package. An alternative nc is available
  in the netcat-traditional package.
  usage: nc [-46DdhklnrStUuvzC] [-i interval] [-P proxy_username] [-p
source_port]
                [-s source_ip_address] [-T ToS] [-w timeout] [-X
proxy_protocol]
                [-x proxy_address[:port]] [hostname] [port[s]]
```

Now let’s use netcat to make a simple Client-Server Chat system.

To make a similar chat client in C we need to write 60-70 lines of code at least. But with netcat we can do it in just 2 simple commands.

**To make a Chat server :-**

Code:

```
nc -l 12345
```

What we just are instructed netcat to listen for connections on port '12345' , Now the machine is listening on the specified port for connections.

**Connecting to this server:-**

Code:

```
nc localhost 12345
```

We instructed netcat to connect to a port '12345' on localhost.

**Testing :-**

Now that we are ready with the Client and the Server set let’s check how it works.

**Client Side :-**

Code:

```
lionaneesh@lionaneesh:~$ nc localhost 12345
Hello i am the client
```

**Server Side :-**

Code:

```
lionaneesh@lionaneesh:~$ nc -l 12345
Hello i am the client
```

Voila ! See what happend our message which we wrote at the client side travelled to the server and was printed o the screen , Isn’t that great!

Using Netcat to transfer files

Netcat can also be used to transfer files , Let’s see how.

**Server Side (The receiver) :-**

Code:

```
  lionaneesh@lionaneesh:~$ nc -l 12345 > file
```

What we did is instructed netcat to listen on port 12345 and redirect all the incoming data to 'file'.

**Client Side (The Sender) :-**

Code:

```
  lionaneesh@lionaneesh:~$ cat article | nc localhost 12345
```

In the above command we used pipes to redirect the output of 'cat article' (which would print the contents of the file named article) to port '12345' of local host.

**Testing :-**

Now let's check whether the transfer of files was successful completed.

**Server Side :-**

Code:

```
  cat file
```

**Output :-**

Code:

```
  It's a test
```

**Client Side :-**

Code:

```
  cat Article
```

**Output:-**

Code:

```
It's a test
```

Voila! We just transferred a file from our client to out server.

**<u>Using Netcat as a port-scanner</u>**

This can easily be done using the '-z' flag which instructs netcat not to initiate a connection but just check if the port is open.

Code:

```
liónaneesh@lionaneesh:~$ nc -z localhost  80-100
```

In the above command we instruct netcat to check which ports are open between 80 and 100 on 'localhost' .

**Output :-**

Code:

```
Connection to 127.0.0.1 80 port [tcp/http] succeeded!
```

The output suggests that port 80 is open on '127.0.0.1'.

**Source: http://www.go4expert.com/articles/netcat-t26082/**