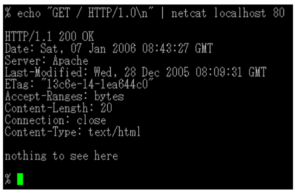
Netcat

netcat



```
% echo "GET / HTTP/1.0\n" | netcat localhost 80

HTTP/1.1 200 OK
Date: Sat, 07 Jan 2006 08:43:27 GMT
Server: Apache
Last-Modified: Wed, 28 Dec 2005 08:09:31 GMT
ETag: "13c6e-14-1ea644c0"
Accept-Ranges: bytes
Content-Length: 20
Connection: close
Content-Type: text/html

nothing to see here

% 
```

Developer(s)	*Hobbit*
Stable release	1.10 / 20 March 1996
Operating system	UNIX
Type	Network utility
License	Permissive free software
Website	nc110.sourceforge.net [1]

Netcat is a computer networking service for reading from and writing to 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.

Netcat is often referred to as a "Swiss-army knife for TCP/IP". Its list of features includes port scanning, transferring files, and port listening, and it can be used as a backdoor.

Features

Some of netcat's major features are:

- Outbound or inbound connections, TCP or UDP, to or from any ports
- Full DNS forward/reverse checking, with appropriate warnings
- Ability to use any local source port
- Ability to use any locally-configured network source address
- Built-in port-scanning capabilities, with randomization
- Built-in loose source-routing capability
- Can read command line arguments from standard input
- Slow-send mode, one line every N seconds
- Hex dump of transmitted and received data
- Optional ability to let another program service established connections
- Optional telnet-options responder
- Featured tunneling mode which allows also special tunneling such as UDP to TCP, with the possibility of specifying all network parameters (source port/interface, listening port/interface, and the remote host allowed to connect to the tunnel.)

Examples

Opening a raw connection to port 25 (like telnet)

```
nc mail.server.net 25
```

Setting up a one-shot webserver on port 8080 to present the content of a file

```
{ echo -ne "HTTP/1.0 200 OK\r\nContent-Length: $(wc -c <some.file)\r\n\r\n"; cat some.file; } | nc -l 8080
```

The file can then be accessed via a webbrowser under http://servername:8080/. Netcat only serves the file once to the first client that connects and then exits, it also provides the content length for browsers that expect it.

Checking if UDP ports (-u) 80-90 are open on 192.168.0.1 using zero mode I/O (-z)

```
nc -vzu 192.168.0.1 80-90
```

Note that UDP tests will always show as “open”. The -uz argument is useless.

Test if UDP port is open: simple UDP server and client

This test is useful, if you have shell access to the server that should be tested, but you do not know whether there is a firewall blocking a specific UDP port on the server.

On the listening host, i.e. on the server whose port needs to be checked, do the following:

```
nc -ul 7000
```

On the sending host, do the following – note that `servname` is the hostname of the listening host:

```
nc -u servname 7000
```

If text typed on the sending host (type something and hit enter) is displayed also on the listening host, then the UDP port 7000 is open. If it is not open, you will get an error such as "Connection refused".

There is a caveat. On some machines, IPv6 may be the default IP version to use by `netcat`. Thus, the host specified by the hostname is contacted using IPv6, and the user might not know about this. Ports may appear closed in the test, even though they would be open when using IPv4. This can be difficult to notice and may cause the false impression that the port is blocked, while it is actually open. You can force the use of IPv4 by using adding `-4` to the options of the `nc` commands.

Pipe via UDP (-u) with a wait time (-w) of 1 second to 'loggerhost' on port 514

```
echo '<0>message' | nc -w 1 -u loggerhost 514
```

Port scanning

An uncommon use of `netcat` is port scanning. Netcat is not considered the best tool for this job, but it can be sufficient (a more advanced tool is nmap)

```
nc -v -n -z -w 1 192.168.1.2 1-1000
```

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.

Proxying

Another useful behaviour is using `netcat` as a proxy. Both ports and hosts can be redirected. Look at this example:

```
nc -l 12345 | nc www.google.com 80
```

Port 12345 represents the request

This starts a `nc` server on port 12345 and all the connections get redirected to `google.com:80`. If a web browser makes a request to `nc`, the request will be sent to google but the response will not be sent to the web browser. That is because pipes are unidirectional. This can be worked around with a named pipe to redirect the input and output.

```
mkfifo backpipe
nc -l 12345 0<backpipe | nc www.google.com 80 1>backpipe
```

The "`-c`" option may also be used with the '`ncat`' implementation:Wikipedia:Verifiability

```
nc -l 12345 -c 'nc www.google.com 80'
```

Another useful feature is to proxy SSL connections. This way, the traffic can not be viewed in wire sniffing applications such as wireshark. This can be accomplished on UNIXes by utilizing `mkfifo`, `netcat`, and `openssl`.

```
mkfifo tmp
mkfifo tmp2
nc -l 8080 -k > tmp < tmp2 &
while [ 1 ]
do
 openssl s_client -connect www.google.com:443 -quiet < tmp > tmp2
done
```

Making any process a server

`netcat` can be used to make any process a network server. It can listen on a port and pipe the input it receives to that process. The `-e` option spawns the executable with its input and output redirected via network socket.

For example, it is possible to expose a bourne shell process to remote computers. To do so, on a computer A with IP 192.168.1.2, run this command:

```
 nc -l -p 1234 -e /bin/sh
```

Then, from any other computer on the same network, one could run this `nc` command:

```
 nc 192.168.1.2 1234
 ls -las
```

And the output one would see might be like this:

```
 total 4288
 4 drwxr-xr-x 15 imsovain users 4096 2009-02-17 07:47 .
 4 drwxr-xr-x  4 imsovain users 4096 2009-01-18 21:22 ..
 8 -rw-------  1 imsovain users 8192 2009-02-16 19:30 .bash_history
 4 -rw-r--r--  1 imsovain users  220 2009-01-18 21:04 .bash_logout
```

```
...
```

In this way, the `-e` option can be used to create a rudimentary backdoor. Some administrators perceive this as a risk, and thus do not allow `netcat` on a computer.

Port Forwarding or Port Mapping

On Linux, NetCat can be used for port forwarding. Below are nine different ways to do port forwarding in NetCat (`-c` switch not supported though - these work with the '`ncat`' incarnation of `netcat`):

```
nc -l -p port1 -c ' nc -l -p port2'
nc -l -p port1 -c ' nc host2 port2'
nc -l -p port1 -c ' nc -u -l -p port2'
nc -l -p port1 -c ' nc -u host2 port2'
nc host1 port1 -c ' nc host2 port2'
nc host1 port1 -c ' nc -u -l -p port2'
nc host1 port1 -c ' nc -u host2 port2'
nc -u -l -p port1 -c ' nc -u -l -p port2'
nc -u -l -p port1 -c ' nc -u host2 port2'
```

Example, see Proxying Netcat#Proxying

Variants

The original version of netcat was a Unix program. The last version (1.10) was released in March 1996.

There are several implementations on POSIX systems, including rewrites from scratch like GNU netcat or OpenBSD netcat, the latter supports IPv6. The OpenBSD version has been ported to the FreeBSD base and Windows/Cygwin as well. Mac OS X users can use MacPorts to install a *netcat* variant. There is also a Microsoft Windows version of *netcat* available. Known ports for embedded systems includes versions for the Windows CE (named "Netcat 4 wince") or for the iPhone.

BusyBox includes by default a lightweight version of netcat.

Solaris 11 includes netcat implementation based on OpenBSD netcat.

Socat is a more complex variant of *netcat*. It is larger and more flexible and has more options that must be configured for a given task.

Cryptcat is a version of *netcat* with integrated transport encryption capabilities.

In the middle of 2005, Nmap announced another netcat incarnation called Ncat. It features new possibilities such as "Connection Brokering", TCP/UDP Redirection, SOCKS4 client and server support, ability to "Chain" Ncat processes, HTTP CONNECT proxying (and proxy chaining), SSL connect/listen support and IP address/connection filtering. Like Nmap, Ncat is cross-platform.

On some systems, modified versions or similar netcat utilities go by the command name(s) `nc`, `ncat`, `pnetcat`, `socat`, `sock`, `socket`, `sbd`.

References

[1] http://nc110.sourceforge.net/

External links

- Official website (http://nc110.sourceforge.net/)
- OpenBSD nc(1) man page (http://www.openbsd.org/cgi-bin/man.cgi?query=nc) via OpenBSD
- GNU netcat (http://netcat.sourceforge.net/)
- Socat (http://www.dest-unreach.org/socat/)
- Adam Palmer (2008-09-16). "NetCat tutorial for Linux & Windows, HOWTO, nc" (http://www.adamsinfo.com/netcat-tutorial-for-linux-windows-howto-nc/). Retrieved 2013-08-11.
- George Notaras (2006-11-06). "Netcat – a couple of useful examples" (http://www.g-loaded.eu/2006/11/06/netcat-a-couple-of-useful-examples/). Retrieved 2013-08-11.
- Jon Crato (2009-04-10). "Netcat for Windows" (http://joncraton.org/blog/netcat-for-windows). Retrieved 2013-08-11.
- Thaoh Myrdania (2011-09-13). "Netcat Mirror" (http://www.thaoh.net/Tools/Netcat/). Retrieved 2013-08-11.
- "Netcat sous Windows - version non détectée et projet CodeBlocks" (http://8pen.net/?p=382) (in French). 2011-06-08. Retrieved 2013-08-11. (Netcat for Windows with `GAPING_SECURITY_HOLE` and `TELNET` enabled)

Article Sources and Contributors

Netcat *Source*: http://en.wikipedia.org/w/index.php?oldid=575585317 *Contributors*: 2A03:3680:0:3:0:0:0:67, Akb4, Andreas Bischoff, Antaeus Feldspar, Apnicsolutions, Axonizer, Bamed, Bauani, Bonadea, Bruceblacklaws, Byrial, CL, Calaka, CanisRufus, Carnesc, Chealer, Claym001, Crh0872, DARTH SIDIOUS 2, Diblidabliduu, Djmckee1, Druiloor, EagleOne, Efa, Ellmist, Episiarch, Family Guy Guy, Feezo, Frap, Frencheigh, Furrybeagle, Gareth McCaughan, Geir3542, Glenn, Gnot, H2g2bob, HopeSeekr of xMule, Incnis Mrsi, Interiot, Isilanes, IttanZ, Jesse V., Jesset77, Jimfbleak, John Vandenberg, Joy, KTC, Kace7, Kalsira, Khatru2, Kl4m-AWB, Kundor, Lorn, Mamaoyot, Mindmatrix, Mortense, Myleslong, Netol, Nnkx00, Nohus, Oracle Techie, Pavlixnet, Pmj005, Polluks, Pradameinhoff, Priyeshgpatel, Prolog, PuerExMachina, Qartis, Qbeep, Rchandra, Redraiment, Remember the dot, Rjwilmsi, Romanc19s, Rosslagerwall, Rousselmanu, Rurik, SPUI, Saifikhan, Samermaz, Scwerllguy, Sietse Snel, SmitherIsGod, Smurfix, Sockseh, Stephan Leeds, Sverdrup, Svick, SymlynX, Techtonik, ThG, The imp, TheParanoidOne, Thorwald, Tobias Bergemann, Tomalak geretkal, Twindruff, Vadmium, Voomoo, Welsh, Widefox, Wiebel, Wikipiero, Wrs1864, Xrblsnggt, Zero Thrust, `Orum, 224 anonymous edits

Image Sources, Licenses and Contributors

Image:Netcat.png *Source*: http://en.wikipedia.org/w/index.php?title=File:Netcat.png *License*: unknown *Contributors*: Original uploader was Interiot at en.wikipedia Later version(s) were uploaded by Mysid at en.wikipedia.

License