Global Information Assurance Certification Paper

Netcat – The TCP/IP Swiss Army Knife

Tom Armstrong
February 15, 2001

Overview

Netcat is a tool that every security professional should be aware of and possibly have in their 'security tool box'. In May/June of 2000, insecure.org conducted a survey of 1200 Nmap users from the Nmap-hackers mailing list to determine their favorite security tools. Netcat was the second most popular tool, not including Nmap[1]. A quick search on securityportal (www.securityportal.com) found 166 matches of netcat. Most of the matches describe or use netcat in some way. Netcat is a utility that is able to write and read data across TCP and UDP network connections. If you are responsible for network or system security it essential that you understand the capabilities of netcat.

Netcat should not be installed unless you have authority to do so. Never install any executable unless you can trust the providor. If possible review the source and compile it yourself. To be safe only use netcat in a test environment.

Hobbit (hobbit@avian.org) created netcat in 1995[2] as a feature-rich network debugging and exploration tool. Its purpose was to be able to create just about any type of network connection. According to Hobbit[2]-

Some of the features of netcat 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 randomizer
- Built-in loose source-routing capability
- Can read command line arguments from standard input
- Slow-send mode, one line every N seconds
- Optional ability to let another program service inbound connections

Some of the potential uses of netcat:

- Script backends
- Scanning ports and inventorying services
- Backup handlers
- File transfers
- Server testing and simulation
- Firewall testing
- Proxy gatewaying
- Network performance testing
- Address spoofing tests

- Protecting X servers
- 1001 other uses you`ll likely come up with

The original version of netcat was released to run on Unix and Linux. Weld Pond (weld@l0pht.com) released the Windows NT version in 1998[3]. The source code is available for both versions.

Remote command prompt anyone?

On a Windows NT server issue the following command in the directory that contains netcat:

```
nc -l -p1234 -d -e cmd.exe –L
```

This –l puts netcat into listen mode, the -p1234 tells netcat to use port 1234, the –d allows netcat to run detached from the console, the –e cmd.exe tells netcat to execute the cmd.exe program when a connection is made, and the –L will restart Netcat with the same command line when the connection is terminated.

On the client system issue the following command:

```
nc destination 1234
```

This command causes netcat to connect to the server named destination on port 1234. Immediately you are given a console connection to the destination server. Be careful! To exit the remote console session type:

```
exit
```

You will be returned to your own console and will be able to reconnect to the destination server because netcat was started on the destination server with the –L option.

FTP & drive mapping blocked?

To receive a file named newfile on the destination system start netcat with the following command:

```
nc –l –p 1234 >newfile
```

On the source system send a file named origfile to the destination system with the following command:

```
nc destination 1234 <origfile
```

Issue a ^C on the source system and your done. Be sure to check the file to be sure it is the same size as the original.

Hiding Netcat on Windows NT

Here are a few ways that a hacker could use to hide netcat on a system or use it behind a firewall:

- Rename the executable or recompile with a different name. Beware that using a copy of netcat that you aren't sure how the source was compiled is very dangerous. If possible review the source code and compile it yourself.
- Detach from the console option (-d)
- Use a port that is well known and allowed through any firewalls between the two systems.

Port Scanning

A scanning example from Hobbit is "nc -v -w 2 -z target 20-30". Netcat will try connecting to every port between 20 and 30 [inclusive] at the target, and will likely inform you about an FTP server, telnet server, and mailer along the way. The -z switch prevents sending any data to a TCP connection and very limited probe data to a UDP connection, and is thus useful as a fast scanning mode just to see what ports the target is listening on. To limit scanning speed if desired, -i will insert a delay between each port probe.[4] Even though netcat can be used for port scanning it isn't its strength. A tool such as nmap is better suited for port scanning.

Netcat + Encryption = Cryptcat[5]

Netcat is a useful tool as it is, but if someone were using it you would be able to at least get a feel for what they were doing. At least you could before Cryptcat! Cryptcat is the standard netcat enhanced with Bruce Schneier's twofish encryption. It can be found at www.farm9.com. Linux, OpenBSD, FreeBSD, and Windows versions are available. So much for sniffing any netcat traffic!

Command Option Overview[6]

Netcat accepts its commands with options first, then the target host, and everything thereafter is interpreted as port names or numbers, or ranges of ports in M-N syntax. Netcat does not currently handle portnames with hyphens.

Option	Description
-d	Allows netcat to detach from the console on Windows NT.
-e	Executes a program if netcat is compiled with the –DGAPING_SECURITY_HOLE.
-i	Sets the interval time. Netcat uses large 8K reads and writes. This basically sends data one line at a time. This is normally used when data is read from files or pipes.

-g	Used to construct a loose-source-routed path for your connection. This is modeled after "traceroute".
-G	Positions the "hop pointer" within the list.
-l	Forces netcat to listen for an inbound connection. An example "nc –l –p 1234 <filename" tells netcat to listen for a connection on port 1234 and once a connection is made to send the file named filename. The file is sent whether the connecting system wants it or not. If you specify a target host netcat will only accept an bound connection only from that host and if you specify one, only from the specified foreign source port.
-L	Restarts Netcat with the same command line that was used when the connection was started.. This way you can connect over and over to the same Netcat process.
-n	Forces netcat to only accept numeric IP addresses and to not do any DNS lookups for anything
-o	Used to obtain a hex dump file of the data sent either way, use "-o logfile". The dump lines begin with "<" or ">" to respectively indicate "from the net" or "to the net", and contain the total count per direction, and hex or ascii representations of the traffic.
-p	Required for outbound connections. The parameter can be numeric or a name as listed in the services file. If –p is not used netcat will bind to whatever unused port the systems gives it, unless the –r option is used.
-r	Causes port scanning to be done randomly. Normally it is done highest to lowest.
-s	Used to specifiy local network source address. Usage "-s ip-addr" or "-s name".
-t	Enables netcat to respond to telnet option negotiation if netcat is compiled with –DTELNET parameter. Telnet daemons will get no useful answers, as they would from a telnet program.
-u	Tells netcat to use UDP instead of TCP.
-v	Controls the level of verbosity. • (without –n) netcat will do a full forward and reverse name and address lookup for the host, and warn you about the all-to-common problem of mismatched names in the DNS. • Usually want to use the –w 3, which limits the time spent trying to make a connection. • If multiple ports are given –v must be specified twice.
-w	Limits the time spent trying to make a connection.
-z	Prevents sending any data to a TCP connection and very limited probe data to a UDP connection. Use –i to insert a delay between each port probe. This is useful as a fast scanning mode just to see what ports the target is listening on.

Conclusion

Netcat is a powerful tool that every security professional should be familiar with. It should be used with caution. I would not recommend installing netcat on your production networks. I would suggest using it to test your firewall, and router configurations in a test environment. It can also be used to test your operating system lockdown procedures. Be certain that you have the authority to install and use netcat on your network before doing so. You might even want to review the source code to learn how Hobbit built netcat and how Weld Pond ported it to the Windows platform.

[1] Insecure.org, "Top 50 Security Tools"
URL: http://www.insecure.org/tools.html (August 21, 2000)
[2] Hobbit, "New tool available: Netcat"
URL: http://lists.insecure.org/bugtraq/1995/Oct/0028.html (October 28, 1995)
[3] Weld Pond, "Netcat 1.10 for NT"
URL: http://www.l0pht.com/~weld/netcat/readment.txt (February 2, 1998)
[4] Hobbit, "Netcat 1.10"
URL: http://www.l0pht.com/~weld/netcat/readme.html (March 20, 1996)
[5] Farm9, "cryptcat = netcat + encryption"
URL: http://farm9.com/content/Free_Tools/Cryptcat (October 2, 2000)
[6] Hobbit, "Netcat 1.10"
URL: http://www.l0pht.com/~weld/netcat/readme.html (March 20, 1996)