War Against Computers

Ncat Cheatsheet

Thu 02 October 2014
By **alex**

Ncat is a modernized implementation of the classic Netcat (nc) networking tool. The current version of nc, which is 1.10, was initially released in 1996 by "The Hobbit". A lot's changed since '96. Thankfully, the wonderful people at The Nmap Project decided that to solve modern problems you need modern tools, and thus Ncat was created. Ncat picks up where nc left off by implementing features such as IPv6 compatibility, multi-protocol support, built in access control, SSL integration, SOCKS and HTTP proxy support, and even connection brokering for sharing connections between concurrent clients. There is a variety of Netcats that have popped up over time. They include netcat-openbsd, netcat6, socat, and cryptcat, to name a few. Of the various Netcats that exist, I find The Nmap Project's Ncat to be the most comprehensive and the best implementation of the classic TCP/IP Swiss Army Knife.

Ncat is typically distributed as part of the Nmap Security Scanner. Nmap packages are readily available for most major Linux distros through their respective package managers. However, the recommended method to install Nmap is to visit the Download page at nmap.org and pickup the current version. Nmap's download page offers precompiled rpms, source rpms, Windows installers, separate binaries for nmap, ncat and nping, and of course source code. Note: If nmap/ncat is installed from source, libssl-dev needs to be installed prior to compiling to enable ssl support.

Netcat's usage has been covered previously, see Netcat tricks, so the following is

going to build on that and detail Ncat's additional features. Care has been taken in developing Ncat to keep the options and basic usage the same as nc. A few minor differences between versions exist, the main being the Ncat doesn't include a port scanning function. This is merely trivial as Nmap is able to fulfill any port scanning requirements.

Send/recieve a file

Server side

```
ncat -vl --send-only 7777 < file
```

Client side

```
ncat -v remote_host 7777 > file
```

Encrypt a session with SSL

Server side

```
ncat -vl --ssl 7777 < file
```

Client side

```
ncat -v --ssl 7777 > file
```

Connect to an HTTPS website

```
ncat -v --ssl google.com 443
```

Forward a local port

```
ncat -vl 7777 -c 'ncat remote_host 8080'
```

Connect through a proxy

```
ncat --proxy proxy_server:port --proxy-type {socks4|http} remot_host port [--pr
oxy-auth username:password]
```

Proxies that return status code 407 require authentication

Create an HTTP proxy

```
ncat -vl --proxy-type http 7777 [--allow 123.45.67.89 --proxy-auth user:pass]
```

Configure access controls for listening sockets

```
ncat -vl 7777 {--allow|--deny}  remotehost
ncat -vl 7777 {--allow|--deny}  10.0.0.10
ncat -vl 7777 {--allow|--deny}  10.0.0.10,10.0.0.20
ncat -vl 7777 {--allow|--deny}  10.0.0.10-20
ncat -vl 7777 {--allow|--deny}  10.0.0.0/24
ncat -vl 7777 {--allow-file|--deny-file} hosts.txt
```