man7.org > Linux > man-pages   **Linux/UNIX system programming training**

ncat(1) — Linux manual page

[Search online pages]

```
NCAT(1)                    Ncat Reference Guide                    NCAT(1)
```

NAME   top

```
    ncat - Concatenate and redirect sockets
```

SYNOPSIS   top

```
    ncat [OPTIONS...] [hostname] [port]
```

DESCRIPTION   top

```
    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 and is the culmination of the
    currently splintered family of Netcat incarnations. It 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.

    Among Ncat's vast number of features there is the ability to
    chain Ncats together; redirection of TCP, UDP, and SCTP ports to
    other sites; SSL support; and proxy connections via SOCKS4,
    SOCKS5 or HTTP proxies (with optional proxy authentication as
    well). Some general principles apply to most applications and
```

thus give you the capability of instantly adding networking
support to software that would normally never support it.

OPTIONS SUMMARY   top

```
      Ncat 7.95SVN ( https://nmap.org/ncat )
      Usage: ncat [options] [hostname] [port]

      Options taking a time assume seconds. Append 'ms' for milliseconds,
      's' for seconds, 'm' for minutes, or 'h' for hours (e.g. 500ms).
        -4                         Use IPv4 only
        -6                         Use IPv6 only
        -U, --unixsock             Use Unix domain sockets only
            --vsock                Use vsock sockets only
        -C, --crlf                 Use CRLF for EOL sequence
        -c, --sh-exec <command>    Executes the given command via /bin/sh
        -e, --exec <command>       Executes the given command
            --lua-exec <filename>  Executes the given Lua script
        -g hop1[,hop2,...]         Loose source routing hop points (8 max)
        -G <n>                     Loose source routing hop pointer (4, 8, 12, ...)
        -m, --max-conns <n>        Maximum <n> simultaneous connections
        -h, --help                 Display this help screen
        -d, --delay <time>         Wait between read/writes
        -o, --output <filename>    Dump session data to a file
        -x, --hex-dump <filename>  Dump session data as hex to a file
        -i, --idle-timeout <time>  Idle read/write timeout
        -p, --source-port port     Specify source port to use
        -s, --source addr          Specify source address to use (doesn't affect -l)
        -l, --listen               Bind and listen for incoming connections
        -k, --keep-open            Accept multiple connections in listen mode
        -n, --nodns                Do not resolve hostnames via DNS
        -t, --telnet               Answer Telnet negotiations
        -u, --udp                  Use UDP instead of default TCP
            --sctp                 Use SCTP instead of default TCP
        -v, --verbose              Set verbosity level (can be used several times)
        -w, --wait <time>          Connect timeout
        -z                         Zero-I/O mode, report connection status only
            --append-output        Append rather than clobber specified output files
            --send-only            Only send data, ignoring received; quit on EOF
            --recv-only            Only receive data, never send anything
            --no-shutdown          Continue half-duplex when receiving EOF on stdin
            --allow                Allow only given hosts to connect to Ncat
            --allowfile            A file of hosts allowed to connect to Ncat
            --deny                 Deny given hosts from connecting to Ncat
            --denyfile             A file of hosts denied from connecting to Ncat
            --broker               Enable Ncat's connection brokering mode
            --chat                 Start a simple Ncat chat server
```

```
           --proxy <addr[:port]>  Specify address of host to proxy through
           --proxy-type <type>    Specify proxy type ("http", "socks4", "socks5")
           --proxy-auth <auth>    Authenticate with HTTP or SOCKS proxy server
           --proxy-dns <type>     Specify where to resolve proxy destination
           --ssl                  Connect or listen with SSL
           --ssl-cert             Specify SSL certificate file (PEM) for listening
           --ssl-key              Specify SSL private key (PEM) for listening
           --ssl-verify           Verify trust and domain name of certificates
           --ssl-trustfile        PEM file containing trusted SSL certificates
           --ssl-ciphers          Cipherlist containing SSL ciphers to use
           --ssl-servername       Request distinct server name (SNI)
           --ssl-alpn             ALPN protocol list to use
           --version              Display Ncat's version information and exit
```

       See the ncat(1) manpage for full options, descriptions and usage examples

CONNECT MODE AND LISTEN MODE   top

    Ncat operates in one of two primary modes: connect mode and
    listen mode. Other modes, such as the HTTP proxy server, act as
    special cases of these two. In connect mode, Ncat works as a
    client. In listen mode it is a server.

    In connect mode, the *hostname* and *port* arguments tell what to
    connect to. *hostname* is required, and may be a hostname or IP
    address. If *port* is supplied, it must be a decimal port number.
    If omitted, it defaults to 31337.

    In listen mode, *hostname* and *port* control the address the server
    will bind to. Both arguments are optional in listen mode. If
    hostname is omitted, it defaults to listening on all available
    addresses over IPv4 and IPv6. If *port* is omitted, it defaults to
    31337.

PROTOCOL OPTIONS   top

    -4 (IPv4 only)
        Force the use of IPv4 only.

    -6 (IPv6 only)
        Force the use of IPv6 only.

    -U, --unixsock (Use Unix domain sockets)
        Use Unix domain sockets rather than network sockets. This
        option may be used on its own for stream sockets, or combined
        with **--udp** for datagram sockets. A description of **-U** mode is

in the section called "UNIX DOMAIN SOCKETS".

-u, --udp (Use UDP)
   Use UDP for the connection (the default is TCP).

--sctp (Use SCTP)
   Use SCTP for the connection (the default is TCP). SCTP
   support is implemented in TCP-compatible mode.

--vsock (Use AF_VSOCK sockets)
   Use AF_VSOCK sockets rather than the default TCP sockets
   (Linux only). This option may be used on its own for stream
   sockets or combined with **--udp** for datagram sockets. A
   description of **--vsock** mode is in the section called
   "AF_VSOCK SOCKETS".

CONNECT MODE OPTIONS   top

-g *hop1*[,*hop2*,...] (Loose source routing)
   Sets hops for IPv4 loose source routing. You can use **-g** once
   with a comma-separated list of hops, use **-g** multiple times
   with single hops to build the list, or combine the two. Hops
   can be given as IP addresses or hostnames.

-G *ptr* (Set source routing pointer)
   Sets the IPv4 source route "pointer" for use with **-g**. The
   argument must be a multiple of 4 and no more than 28. Not all
   operating systems support setting this pointer to anything
   other than four.

-p *port*, **--source-port** *port* (Specify source port)
   Set the port number for Ncat to bind to.

-s *host*, **--source** *host* (Specify source address)
   Set the address for Ncat to bind to.

LISTEN MODE OPTIONS   top

See the section called "ACCESS CONTROL OPTIONS" for information
on limiting the hosts that may connect to the listening Ncat
process.

-l, --listen (Listen for connections)
   Listen for connections rather than connecting to a remote
   machine

-m *numconns*, **--max-conns** *numconns* (Specify maximum number of connections)
> The maximum number of simultaneous connections accepted by an Ncat instance. 100 is the default (60 on Windows).

-k, **--keep-open** (Accept multiple connections)
> Normally a listening server accepts only one connection and then quits when the connection is closed. This option makes it accept multiple simultaneous connections and wait for more connections after they have all been closed. It must be combined with **--listen**. In this mode there is no way for Ncat to know when its network input is finished, so it will keep running until interrupted. This also means that it will never close its output stream, so any program reading from Ncat and looking for end-of-file will also hang.

--broker (Connection brokering)
> Allow multiple parties to connect to a centralised Ncat server and communicate with each other. Ncat can broker communication between systems that are behind a NAT or otherwise unable to directly connect. This option is used in conjunction with **--listen**, which causes the **--listen** port to have broker mode enabled.

--chat (Ad-hoc "chat server")
> The **--chat** option enables chat mode, intended for the exchange of text between several users. In chat mode, connection brokering is turned on. Ncat prefixes each message received with an ID before relaying it to the other connections. The ID is unique for each connected client. This helps distinguish who sent what. Additionally, non-printing characters such as control characters are escaped to keep them from doing damage to a terminal.

SSL OPTIONS   top

--ssl (Use SSL)
> In connect mode, this option transparently negotiates an SSL session with an SSL server to securely encrypt the connection. This is particularly handy for talking to SSL enabled HTTP servers, etc.
>
> In server mode, this option listens for incoming SSL connections, rather than plain untunneled traffic.
>
> In UDP mode, this option enables Datagram TLS (DTLS).

--ssl-verify (Verify server certificates)
 In client mode, **--ssl-verify** is like **--ssl** except that it
 also requires verification of the server certificate. Ncat
 comes with a default set of trusted certificates in the file
 ca-bundle.crt. Some operating systems provide a default list
 of trusted certificates; these will also be used if
 available. Use **--ssl-trustfile** to give a custom list. Use **-v**
 one or more times to get details about verification failures.
 Ncat does not check for revoked certificates.

 This option has no effect in server mode.

--ssl-cert *certfile.pem* (Specify SSL certificate)
 This option gives the location of a PEM-encoded certificate
 files used to authenticate the server (in listen mode) or the
 client (in connect mode). Use it in combination with
 --ssl-key.

--ssl-key *keyfile.pem* (Specify SSL private key)
 This option gives the location of the PEM-encoded private key
 file that goes with the certificate named with **--ssl-cert**.

--ssl-trustfile *cert.pem* (List trusted certificates)
 This option sets a list of certificates that are trusted for
 purposes of certificate verification. It has no effect unless
 combined with **--ssl-verify**. The argument to this option is
 the name of a PEM file containing trusted certificates.
 Typically, the file will contain certificates of
 certification authorities, though it may also contain server
 certificates directly. When this option is used, Ncat does
 not use its default certificates.

--ssl-ciphers *cipherlist* (Specify SSL ciphersuites)
 This option sets the list of ciphersuites that Ncat will use
 when connecting to servers or when accepting SSL connections
 from clients. The syntax is described in the OpenSSL
 ciphers(1) man page, and defaults to
 ALL:!aNULL:!eNULL:!LOW:!EXP:!RC4:!MD5:@STRENGTH

--ssl-servername *name* (Request distinct server name)
 In client mode, this option sets the TLS SNI (Server Name
 Indication) extension, which tells the server the name of the
 logical server Ncat is contacting. This is important when the
 target server hosts multiple virtual servers at a single
 underlying network address. If the option is not provided,
 the TLS SNI extension will be populated with the target
 server hostname.

--ssl-alpn *ALPN list* (Specify ALPN protocol list)
    This option allows you to specify a comma-separated list of
    protocols to send via the Application-Layer Protocol
    Negotiation (ALPN) TLS extension. Not supported by all
    versions of OpenSSL.

PROXY OPTIONS

--proxy *host*[:*port*] (Specify proxy address)
    Requests proxying through *host*:*port*, using the protocol
    specified by **--proxy-type**.

    If no port is specified, the proxy protocol's well-known port
    is used (1080 for SOCKS and 3128 for HTTP). When specifying
    an IPv6 HTTP proxy server using the IP address rather than
    the hostname, the square-bracket notation (for example
    [2001:db8::1]:8080) MUST be used to separate the port from
    the IPv6 address. If the proxy requires authentication, use
    --proxy-auth.

--proxy-type *proto* (Specify proxy protocol)
    In connect mode, this option requests the protocol *proto* to
    connect through the proxy host specified by **--proxy**. In
    listen mode, this option has Ncat act as a proxy server using
    the specified protocol.

    The currently available protocols in connect mode are http
    (CONNECT), socks4 (SOCKSv4), and socks5 (SOCKSv5). The only
    server currently supported is http. If this option is not
    used, the default protocol is http.

--proxy-auth *user*[:*pass*] (Specify proxy credentials)
    In connect mode, gives the credentials that will be used to
    connect to the proxy server. In listen mode, gives the
    credentials that will be required of connecting clients. For
    use with **--proxy-type http** or **--proxy-type socks5**, the form
    should be username:password. For **--proxy-type socks4**, it
    should be a username only.

    These credentials can be alternatively passed onto Ncat by
    setting environment variable **NCAT_PROXY_AUTH**, which reduces
    the risk of the credentials being captured in process logs.
    (Option **--proxy-auth** takes precedence.)

--proxy-dns *type* (Specify where to resolve proxy destination)
    In connect mode, it provides control over whether proxy
    destination hostnames are resolved by the remote proxy server

or locally, by Ncat itself. Possible values for *type* are:

local - Hostnames are resolved locally on the Ncat host. Ncat
exits with error if the hostname cannot be resolved.

remote - Hostnames are passed directly onto the remote proxy
server. This is the default behavior.

both - Hostname resolution is first attempted on the Ncat
host. Unresolvable hostnames are passed onto the remote proxy
server.

none - Hostname resolution is completely disabled. Only a
literal IPv4 or IPv6 address can be used as the proxy
destination.

Local hostname resolution generally respects IP version
specified with options **-4** or **-6**, except for SOCKS4, which is
incompatible with IPv6.

COMMAND EXECUTION OPTIONS   top

     -e *command*, **--exec** *command* (Execute command)
       Execute the specified command after a connection has been
       established. The command must be specified as a full
       pathname. All input from the remote client will be sent to
       the application and responses sent back to the remote client
       over the socket, thus making your command-line application
       interactive over a socket. Combined with **--keep-open**, Ncat
       will handle multiple simultaneous connections to your
       specified port/application like inetd. Ncat will only accept
       a maximum, definable, number of simultaneous connections
       controlled by the **-m** option. By default this is set to 100
       (60 on Windows).

     -c *command*, **--sh-exec** *command* (Execute command via sh)
       Same as **-e**, except it tries to execute the command via
       /bin/sh. This means you don't have to specify the full path
       for the command, and shell facilities like environment
       variables are available.

     --lua-exec *file* (Execute a .lua script)
       Runs the specified file as a Lua script after a connection
       has been established, using a built-in interpreter. Both the
       script's standard input and the standard output are
       redirected to the connection data streams.

All exec options add the following variables to the child's
environment:

NCAT_REMOTE_ADDR, **NCAT_REMOTE_PORT**
>   The IP address and port number of the remote host. In connect
>   mode, it's the target's address; in listen mode, it's the
>   client's address.

NCAT_LOCAL_ADDR, **NCAT_LOCAL_PORT**
>   The IP address and port number of the local end of the
>   connection.

NCAT_PROTO
>   The protocol in use: one of TCP, UDP, and SCTP.

ACCESS CONTROL OPTIONS   top

--allow *host*[,*host*,...] (Allow connections)
>   The list of hosts specified will be the only hosts allowed to
>   connect to the Ncat process. All other connection attempts
>   will be disconnected. In case of a conflict between **--allow**
>   and **--deny**, **--allow** takes precedence. Host specifications
>   follow the same syntax used by Nmap.

--allowfile *file* (Allow connections from file)
>   This has the same functionality as **--allow**, except that the
>   allowed hosts are provided in a new-line delimited allow
>   file, rather than directly on the command line.

--deny *host*[,*host*,...] (Deny connections)
>   Issue Ncat with a list of hosts that will not be allowed to
>   connect to the listening Ncat process. Specified hosts will
>   have their session silently terminated if they try to
>   connect. In case of a conflict between **--allow** and **--deny**,
>   **--allow** takes precedence. Host specifications follow the same
>   syntax used by Nmap.

--denyfile *file* (Deny connections from file)
>   This is the same functionality as **--deny**, except that
>   excluded hosts are provided in a new-line delimited deny
>   file, rather than directly on the command line.

TIMING OPTIONS   top

These options accept a time parameter. This is specified in
seconds by default, though you can append ms, s, m, or h to the

value to specify milliseconds, seconds, minutes, or hours.

-d *time*, **--delay** *time* (Specify line delay)
    Set the delay interval for lines sent. This effectively
    limits the number of lines that Ncat will send in the
    specified period. This may be useful for low-bandwidth sites,
    or have other uses such as coping with annoying **iptables**
    --limit options.

-i *time*, **--idle-timeout** *time* (Specify idle timeout)
    Set a fixed timeout for idle connections. If the idle timeout
    is reached, the connection is terminated.

-w *time*, **--wait** *time* (Specify connect timeout)
    Set a fixed timeout for connection attempts.

OUTPUT OPTIONS   top

-o *file*, **--output** *file* (Save session data)
    Dump session data to a file

-x *file*, **--hex-dump** *file* (Save session data in hex)
    Dump session data in hex to a file.

--append-output (Append output)
    Issue Ncat with **--append-ouput** along with **-o** and/or **-x** and it
    will append the resulted output rather than truncating the
    specified output files.

-v, **--verbose** (Be verbose)
    Issue Ncat with **-v** and it will be verbose and display all
    kinds of useful connection based information. Use more than
    once (**-vv**, **-vvv**...) for greater verbosity.

MISC OPTIONS   top

-C, **--crlf** (Use CRLF as EOL)
    This option tells Ncat to convert LF line endings to CRLF
    when taking input from standard input. This is useful for
    talking to some stringent servers directly from a terminal in
    one of the many common plain-text protocols that use CRLF for
    end-of-line.

-h, **--help** (Help screen)
    Displays a short help screen with common options and
    parameters, and then exits.

--recv-only (Only receive data)
    If this option is passed, Ncat will only receive data and
    will not try to send anything.

--send-only (Only send data)
    If this option is passed, then Ncat will only send data and
    will ignore anything received. This option also causes Ncat
    to close the network connection and terminate after EOF is
    received on standard input.

--no-shutdown (Do not shutdown into half-duplex mode)
    If this option is passed, Ncat will not invoke shutdown on a
    socket after seeing EOF on stdin. This is provided for
    backward-compatibility with OpenBSD netcat, which exhibits
    this behavior when executed with its '-d' option.

-n, **--nodns** (Do not resolve hostnames)
    Completely disable hostname resolution across all Ncat
    options, such as the destination, source address, source
    routing hops, and the proxy. All addresses must be specified
    numerically. (Note that resolution of proxy destinations is
    controlled separately via option **--proxy-dns**.)

-t, **--telnet** (Answer Telnet negotiations)
    Handle DO/DONT WILL/WONT Telnet negotiations. This makes it
    possible to script Telnet sessions with Ncat.

--version (Display version)
    Displays the Ncat version number and exits.

UNIX DOMAIN SOCKETS   top

The **-U** option (same as **--unixsock**) causes Ncat to use Unix domain
sockets rather than network sockets. Unix domain sockets exist as
an entry in the filesystem. You must give the name of a socket to
connect to or to listen on. For example, to make a connection,

ncat -U ~/unixsock

To listen on a socket:

ncat -l -U ~/unixsock

Listen mode will create the socket if it doesn't exist. The
socket will continue to exist after the program ends.

Both stream and datagram domain sockets are supported. Use **-U** on
its own for stream sockets, or combine it with **--udp** for datagram
sockets. Datagram sockets require a source socket to connect
from. By default, a source socket with a random filename will be
created as needed, and deleted when the program ends. Use the
--source with a path to use a source socket with a specific name.

AF_VSOCK SOCKETS   top

The **--vsock** option causes Ncat to use AF_VSOCK sockets rather
than network sockets. A CID must be given instead of a hostname
or IP address. For example, to make a connection to the host,

ncat --vsock 2 1234

To listen on a socket:

ncat -l --vsock 1234

Both stream and datagram domain sockets are supported, but socket
type availability depends on the hypervisor. Use **--vsock** on its
own for stream sockets, or combine it with **--udp** for datagram
sockets.

EXAMPLES   top

Connect to example.org on TCP port 8080.
    ncat example.org 8080

Listen for connections on TCP port 8080.
    ncat -l 8080

Redirect TCP port 8080 on the local machine to host on port 80.
    ncat --sh-exec "ncat example.org 80" -l 8080 --keep-open

Bind to TCP port 8081 and attach /bin/bash for the world to
access freely.
    ncat --exec "/bin/bash" -l 8081 --keep-open

Bind a shell to TCP port 8081, limit access to hosts on a local
network, and limit the maximum number of simultaneous connections
to 3.
    ncat --exec "/bin/bash" --max-conns 3 --allow 192.168.0.0/24
    -l 8081 --keep-open

Connect to smtphost:25 through a SOCKS4 server on port 1080.

```
        ncat --proxy socks4host --proxy-type socks4 --proxy-auth joe
        smtphost 25
```

Connect to smtphost:25 through a SOCKS5 server on port 1080.
```
        ncat --proxy socks5host --proxy-type socks5 --proxy-auth
        joe:secret smtphost 25
```

Create an HTTP proxy server on localhost port 8888.
```
        ncat -l --proxy-type http localhost 8888
```

Send a file over TCP port 9899 from host2 (client) to host1
(server).
```
        HOST1$ ncat -l 9899 > outputfile

        HOST2$ ncat HOST1 9899 < inputfile
```

Transfer in the other direction, turning Ncat into a "one file"
server.
```
        HOST1$ ncat -l 9899 < inputfile

        HOST2$ ncat HOST1 9899 > outputfile
```

EXIT CODE   top

The exit code reflects whether a connection was made and
completed successfully. 0 means there was no error. 1 means there
was a network error of some kind, for example "Connection
refused" or "Connection reset". 2 is reserved for all other
errors, like an invalid option or a nonexistent file.

BUGS   top

Like its authors, Ncat isn't perfect. But you can help make it
better by sending bug reports or even writing patches. If Ncat
doesn't behave the way you expect, first upgrade to the latest
version available from **https://nmap.org** . If the problem persists,
do some research to determine whether it has already been
discovered and addressed. Try Googling the error message or
browsing the nmap-dev archives at **https://seclists.org/** .

Read this full manual page as well. If nothing comes of this,
mail a bug report to <dev@nmap.org>. Please include everything
you have learned about the problem, as well as what version of
Ncat you are running and what operating system version it is
running on. Problem reports and Ncat usage questions sent to
dev@nmap.org are far more likely to be answered than those sent

to Fyodor directly.

Code patches to fix bugs are even better than bug reports. Basic instructions for creating patch files with your changes are available at **https://svn.nmap.org/nmap/HACKING** . Patches may be sent to nmap-dev (recommended) or to Fyodor directly.

AUTHORS   top

- Chris Gibson <chris@linuxops.net>

- Gordon Lyon (Fyodor)<fyodor@nmap.org> (**http://insecure.org**)

- Kris Katterjohn <katterjohn@gmail.com>

- Mixter <mixter@gmail.com>

The original Netcat was written by *Hobbit* <hobbit@avian.org>. While Ncat isn't built on any code from the "traditional" Netcat (or any other implementation), Ncat is most definitely based on Netcat in spirit and functionality.

LEGAL NOTICES   top

Ncat Copyright and Licensing
Ncat is (C) 2005–2022 Nmap Software LLC. It is distributed as free and open source software under the same license terms as our Nmap software. Precise terms and further details are available from **https://nmap.org/man/man-legal.html** .

Creative Commons License for this Ncat Guide
This Ncat Reference Guide is (C) 2005–2022 Nmap Software LLC. It is hereby placed under version 3.0 of the **Creative Commons Attribution License**[1]. This allows you redistribute and modify the work as you desire, as long as you credit the original source. Alternatively, you may choose to treat this document as falling under the same license as Ncat itself (discussed previously).

Source Code Availability and Community Contributions
Source is provided to this software because we believe users have a right to know exactly what a program is going to do before they run it. This also allows you to audit the software for security holes (none have been found so far).

Source code also allows you to port Nmap (which includes Ncat) to

new platforms, fix bugs, and add new features. You are highly
encouraged to send your changes to <dev@nmap.org> for possible
incorporation into the main distribution. By sending these
changes to Fyodor or one of the Insecure.Org development mailing
lists, it is assumed that you are offering the Nmap Project (Nmap
Software LLC) the unlimited, non-exclusive right to reuse,
modify, and relicense the code. Nmap will always be available
open source, but this is important because the inability to
relicense code has caused devastating problems for other Free
Software projects (such as KDE and NASM). We also occasionally
relicense the code to third parties as discussed in the Nmap man
page. If you wish to specify special license conditions of your
contributions, just say so when you send them.

No Warranty

This program is distributed in the hope that it will be useful,
but WITHOUT ANY WARRANTY; without even the implied warranty of
MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the Nmap
Public Source License for more details at **https://nmap.org/npsl/** ,
or in the LICENSE file included with Nmap.

Inappropriate Usage

Ncat should never be installed with special privileges (e.g. suid
root). That would open up a major security vulnerability as
other users on the system (or attackers) could use it for
privilege escalation.

Third-Party Software

This product includes software developed by the **Apache Software
Foundation**[2]. A modified version of the **Libpcap portable packet
capture library**[3] is distributed along with Ncat. The Windows
version of Ncat utilized the Libpcap-derived **Npcap library**[4]
instead. Certain raw networking functions use the **Libdnet**[5]
networking library, which was written by Dug Song. A modified
version is distributed with Ncat. Ncat can optionally link with
the **OpenSSL cryptography toolkit**[6] for SSL version detection
support. All of the third-party software described in this
paragraph is freely redistributable under BSD-style software
licenses.

NOTES   top

1. Creative Commons Attribution License
   http://creativecommons.org/licenses/by/3.0/

2. Apache Software Foundation
   http://www.apache.org

3. Libpcap portable packet capture library
   http://www.tcpdump.org

4. Npcap library
   https://npcap.com

5. Libdnet
   http://libdnet.sourceforge.net

6. OpenSSL cryptography toolkit
   http://www.openssl.org

COLOPHON   top

This page is part of the *nmap* (a network scanner) project.
Information about the project can be found at ⟨http://nmap.org/⟩.
If you have a bug report for this manual page, send it to
dev@nmap.org. This page was obtained from the project's upstream
Git mirror of the Subversion repository
⟨https://github.com/nmap/nmap⟩ on 2024-06-14. (At that time, the
date of the most recent commit that was found in the repository
was 2024-06-13.) If you discover any rendering problems in this
HTML version of the page, or you believe there is a better or
more up-to-date source for the page, or you have corrections or
improvements to the information in this COLOPHON (which is *not*
part of the original manual page), send a mail to
man-pages@man7.org

Ncat   04/23/2024   *NCAT*(1)

THE LINUX
PROGRAMMING
INTERFACE