

# Netcat Cheatsheet

By Dejan Panovski • Updated on May 1, 2026 • [Download PDF](#)

Quick reference for using the Netcat nc command to test ports, listen for connections, transfer files, and send raw network requests

The `nc` command reads and writes data over TCP and UDP network connections. This cheatsheet covers common Netcat commands for port checks, listeners, file transfers, timeouts, and basic troubleshooting.

## Basic Syntax

Core `nc` command forms.

<code>nc host port</code>	Open a TCP connection
<code>nc -u host port</code>	Open a UDP connection
<code>nc -l port</code>	Listen on a local port
<code>nc -h</code>	Show help and options
<code>man nc</code>	Read the local Netcat manual

## Connect and Listen

Create simple client and server connections.

<code>nc <a href="#">example.com</a> 80</code>	Connect to TCP port 80
<code>nc -l 5555</code>	Listen on port 5555
<code>nc server.example.com 5555</code>	Connect to a listening host
<code>nc -v host 22</code>	Connect with verbose output
<code>nc -n 192.168.1.10 22</code>	Skip DNS lookups

## Port Checks

Check whether TCP ports are open.

<code>nc -z -v host 22</code>	Check one TCP port
<code>nc -z -v host 20-80</code>	Check a port range
<code>nc -z -v host 80 443</code>	Check selected ports
<code>nc -z -w 3 host 443</code>	Check with a timeout
<code><a href="#">nmap host</a></code>	Use Nmap for deeper scans

## UDP

Use UDP instead of TCP.

<code>nc -u host 53</code>	Connect to a UDP service
<code>nc -u -l 5555</code>	Listen for UDP datagrams
<code>nc -z -v -u host 53</code>	Check a UDP port
<code>nc -u -w 3 host 123</code>	UDP check with timeout
<code>nc -u 192.168.1.10 5555</code>	Send text to a UDP listener

## File Transfers

Send files between two hosts.

<code>nc -l 5555 &gt; file</code>	Receive a file
<code>nc host 5555 &lt; file</code>	Send a file
<code>nc -l 5555   tar xzvf -</code>	Receive and extract a directory
<code>tar czvf - dir   nc host 5555</code>	Archive and send a directory
<code>nc -w 5 host 5555 &lt; file</code>	Send with a timeout

## HTTP and Raw Requests

Send plain text requests to network services.

<code>printf "GET / HTTP/1.1\r\nHost: example.com\r\n\r\n"   nc example.com 80</code>	Send an HTTP request
<code>echo "QUIT"   nc mail.example.com 25</code>	Send a simple SMTP command
<code>echo "PING"   nc host 5555</code>	Send text to a listener
<code>nc host 80</code>	Type a request manually
<a href="http://example.com">curl http://example.com</a>	Use curl for HTTP work

## Timeouts and Persistent Listeners

Control how long Netcat waits and whether it keeps listening.

<code>nc -w 5 host 80</code>	Timeout after 5 seconds
<code>nc -k -l 5555</code>	Keep listening after disconnect
<code>nc -v -w 3 host 22</code>	Verbose connection with timeout
<code>nc -l -p 8080 &lt; index.html</code>	Serve one file on port 8080
<code>while true; do nc -l -p 8080 &lt; index.html; done</code>	Serve the file repeatedly

## Troubleshooting

Quick checks for common Netcat issues.

<code>nc: command not found</code>	Install <code>netcat-openbsd</code> , <code>netcat-traditional</code> , or <code>nmap-ncat</code>
Connection refused	Confirm the remote service is running and listening on that port
Command hangs	Add <code>-W</code> to set a timeout
UDP result is unclear	UDP has no TCP-style handshake, use Nmap for reliable scans
Option behaves differently	Run <code>man nc</code> and check the Netcat implementation on that system

## Related Guides

Use these guides for broader networking workflows.

<a href="#">Netcat Command in Linux</a>	Full Netcat guide with examples
<a href="#">Nmap Command in Linux</a>	Scan hosts and ports in more detail
<a href="#">curl cheatsheet</a>	Work with HTTP requests and APIs
<a href="#">tcpdump cheatsheet</a>	Capture and inspect network packets
<a href="#">ss Command in Linux</a>	Inspect sockets and listening services