# bandwhich Bandwidth by process

**START HERE**

**`sudo bandwhich`**
Live bandwidth by process

**`bandwhich -i <if>`**
Watch one interface

**Tab**
Switch table layout

**`bandwhich -n`**
Skip DNS lookups

**Space**
Pause / resume

**`bandwhich -p`**
Processes table only

**q**
Quit

**`bandwhich -t`**
Show running totals

## CONCEPTS

bandwhich shows which process, connection and remote host is using bandwidth right now. It needs packet capture and /proc access, so run it with sudo or grant capabilities once.

**Packets** sniffed on interface
↓ `match`
**Socket** port in /proc or lsof
↓ `owner`
**Process** name, PID, remote host
How traffic gets a name

Remote hostnames come from background reverse DNS and appear a moment after the IP. Use -n for raw IPs.

Process mapping uses /proc on Linux, lsof on macOS and the Windows API on Windows. Windows may need npcap installed.

Upstream is in passive maintenance: fixes only, no new features.

## IN THE TUI

**Tab**
Rotate which table comes first
*Three tables: processes, connections, remote addresses. Small terminals show fewer.*

**Space**
Pause or resume the display
*Use it to read a busy screen; totals stop while paused.*

## IN THE TUI (cont.)

**q** Quit

**Ctrl** + **C** Quit (interrupt)

Only q, Space and Tab are handled by the interface; there is no in-app help key, so check bandwhich --help for options.

The tables shrink to fit your window. Make the terminal taller to see more rows.

## RUN

**`sudo bandwhich`**
Start monitoring all interfaces

**`sudo env "PATH=$PATH" bandwhich`**
Keep your PATH under sudo
*Needed when installed under ~/.cargo/bin and sudo says command not found.*

**`bandwhich -i <if>`**
Listen on one interface

**`bandwhich -n`**
Do not resolve IPs to hostnames

**`bandwhich -s`**
Show DNS queries

**`bandwhich -d <ip>`**
Use this DNS server for lookups

**`bandwhich -r`**
Machine-friendlier text output

**`bandwhich -V`**
Print version

**`bandwhich -h`**
Short help (--help for detail)

## VIEWS AND UNITS

**`bandwhich -p`**
Show processes table only

**`bandwhich -c`**
Show connections table only

**`bandwhich -a`**
Show remote addresses table only

**`bandwhich -t`**
Show cumulative totals

**`bandwhich -u bin-bytes`**
Bytes, powers of 1024 (default)

**`bandwhich -u bin-bits`**
Bits, powers of 1024

**`bandwhich -u si-bytes`**
Bytes, powers of 1000

**`bandwhich -u si-bits`**
Bits, powers of 1000 (like ISP speeds)

**`bandwhich -p -t`**
Combine view and totals

## PERMISSIONS

**`sudo bandwhich`**
Simplest: run as root

**`sudo setcap cap_sys_ptrace, cap_dac_read_search, cap_net_raw, cap_net_admin+ep $(command -v bandwhich)`**
Grant capabilities once (Linux)
*Then run bandwhich as a normal user. Any user who can run it can see everyone's traffic, so avoid on shared machines.*

**`getcap $(command -v bandwhich)`**
Check granted capabilities

## PERMISSIONS (cont.)

**`sudo setcap -r $(command -v bandwhich)`**
Remove capabilities

Reinstalling or upgrading replaces the binary and drops its file capabilities. Run setcap again.

## LOGGING AND DEBUG

**`bandwhich -v`**
More log output (repeat for more)

**`bandwhich -q`**
Less log output (repeat for less)

**`bandwhich --log-to <file>`**
Write debug log to a file

**`bandwhich -vv --log-to <file>`**
Verbose debug log to a file

## RECIPES

**Which program is eating my uplink**

```
sudo bandwhich -i eth0 -p -n
```

**Allow without sudo (Linux)**

```
sudo setcap cap_sys_ptrace,
cap_dac_read_search,
cap_net_raw,cap_net_admin+ep $(
command -v bandwhich)
bandwhich
```

**Watch DNS chatter**

```
sudo bandwhich -s -c
```

**Debug a failing start**

```
sudo bandwhich -vv --log-to /
tmp/bw.log
# read /tmp/bw.log afterwards
```