# Find all running hosts, used IPs and ports on your local wireless / ethernet network or how to do a basic network security audit with nmap

**Author :** admin

Find all running hosts / used IPs on your local wireless or ethernet network

**If you're using a Free Software OS such as GNU / Linux or some other proprietary OS such as Mac OS X or Windows and you need a quick way to check all running IPs hosts / nodes locally on your current connected Ethernet or Wireless network, here is how to do it with nmap (Network exploration and security tool port scanner).**

**So why would you do scan that?**

**Well just for fun, out of curiousity or just because you want to inspect your local network whether someone unexpected cracker did not break and is not using your *Wi-Fi* or *Ethernet local network* and badly snoring your network listening for passwords.**

Before you start you should have installed NMAP network scanner on your GNU / Linux, to do so on

Redhat Based Linux (Fedora / CentOS / Redhat Enterprise RHEL):

```
 yum -y install nmap
...
```

On Deb based GNU / Linux-es such as Ubuntu / Mint / Debian etc.

```
 apt-get install --yes nmap
...
```

To install **nmap on FreeBSD / NetBSD / OpenBSD OS issue from console or terminal:**

```
 cd /usr/ports/security/nmap
make install clean
...
```

or if you prefer to install it from latest binary instead of compiling

```
pkg_add -vr nmap
...
```

On a proprietary Mac OS X (I don't recommend you to use this obnoxious OS which is designed as a proprpietary software to steal your freedom and control you, but anyways for Mac OS victims), you can do it to with Macs equivalent tool of **apt-get / yum called *homebrew***:

Open *Mac OS X terminal* and to ***install homebrew*** run:

```
ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
brew install nmap
brew search nmap
brew info nmap
```

If you want to do it system wide become root (super user) from Mac terminal with

```
su root
```

and run above commands as administrator user.

Windows users might take a look at Nmap for Windows or use the M$ Windows native **portqry**

**command line port scanner**

Test whether nmap is properly installed and ready to use with command:

```
  nmap --help
Nmap 6.00 ( http://nmap.org )
Usage: nmap [Scan Type(s)] [Options] {target specification}
TARGET SPECIFICATION:
  Can pass hostnames, IP addresses, networks, etc.
  Ex: scanme.nmap.org, microsoft.com/24, 192.168.0.1; 10.0.0-255.1-254
  -iL : Input from list of hosts/networks
  -iR : Choose random targets
  --exclude : Exclude hosts/networks
  --excludefile : Exclude list from file
HOST DISCOVERY:
  -sL: List Scan - simply list targets to scan
  -sn: Ping Scan - disable port scan
  -Pn: Treat all hosts as online -- skip host discovery
  -PS/PA/PU/PY[portlist]: TCP SYN/ACK, UDP or SCTP discovery to given ports
  -PE/PP/PM: ICMP echo, timestamp, and netmask request discovery probes
  -PO[protocol list]: IP Protocol Ping
  -n/-R: Never do DNS resolution/Always resolve [default: sometimes]
  --dns-servers : Specify custom DNS servers
  --system-dns: Use OS's DNS resolver
  --traceroute: Trace hop path to each host
SCAN TECHNIQUES:
  -sS/sT/sA/sW/sM: TCP SYN/Connect()/ACK/Window/Maimon scans
  -sU: UDP Scan
  -sN/sF/sX: TCP Null, FIN, and Xmas scans
  --scanflags : Customize TCP scan flags
  -sI : Idle scan
  -sY/sZ: SCTP INIT/COOKIE-ECHO scans
  -sO: IP protocol scan
  -b : FTP bounce scan
PORT SPECIFICATION AND SCAN ORDER:
  -p : Only scan specified ports
    Ex: -p22; -p1-65535; -p U:53,111,137,T:21-25,80,139,8080,S:9
  -F: Fast mode - Scan fewer ports than the default scan
  -r: Scan ports consecutively - don't randomize
  --top-ports : Scan most common ports
  --port-ratio : Scan ports more common than
SERVICE/VERSION DETECTION:
  -sV: Probe open ports to determine service/version info
  --version-intensity : Set from 0 (light) to 9 (try all probes)
  --version-light: Limit to most likely probes (intensity 2)
  --version-all: Try every single probe (intensity 9)
  --version-trace: Show detailed version scan activity (for debugging)
SCRIPT SCAN:
  -sC: equivalent to --script=default
  --script=: is a comma separated list of
           directories, script-files or script-categories
  --script-args=: provide arguments to scripts
  --script-args-file=filename: provide NSE script args in a file
  --script-trace: Show all data sent and received
  --script-updatedb: Update the script database.
  --script-help=: Show help about scripts.
            is a comma separted list of script-files or
           script-categories.
OS DETECTION:
  -O: Enable OS detection
  --osscan-limit: Limit OS detection to promising targets
  --osscan-guess: Guess OS more aggressively
TIMING AND PERFORMANCE:
  Options which take are in seconds, or append 'ms' (milliseconds),
  's' (seconds), 'm' (minutes), or 'h' (hours) to the value (e.g. 30m).
  -T: Set timing template (higher is faster)
  --min-hostgroup/max-hostgroup : Parallel host scan group sizes
  --min-parallelism/max-parallelism : Probe parallelization
  --min-rtt-timeout/max-rtt-timeout/initial-rtt-timeout : Specifies
      probe round trip time.
  --max-retries : Caps number of port scan probe retransmissions.
  --host-timeout : Give up on target after this long
  --scan-delay/--max-scan-delay : Adjust delay between probes
  --min-rate : Send packets no slower than per second
  --max-rate : Send packets no faster than per second
FIREWALL/IDS EVASION AND SPOOFING:
  -f; --mtu : fragment packets (optionally w/given MTU)
  -D : Cloak a scan with decoys
  -S : Spoof source address
  -e : Use specified interface
  -g/--source-port : Use given port number
  --data-length : Append random data to sent packets
  --ip-options : Send packets with specified ip options
  --ttl : Set IP time-to-live field
  --spoof-mac : Spoof your MAC address
  --badsum: Send packets with a bogus TCP/UDP/SCTP checksum
OUTPUT:
  -oN/-oX/-oS/-oG : Output scan in normal, XML, s|
```