Network Scan Commands

This file collects common commands and examples for working with `nmap`, `ipconfig` (Windows), and Wireshark/tshark. All examples below use the IP address **10.204.197.132** (your machine) and the network `10.204.197.0/24` (subnet mask 255.255.255.0).

```
```

1) Check local IP (Windows)

Open Command Prompt (cmd) and run:

```
ipconfig
```

Look for your Wireless or Ethernet adapter. Example fields you should see:

- IPv4 Address: `10.204.197.132`
- Subnet Mask: `255.255.255.0`
- Default Gateway: `10.204.197.185`

---

2) Nmap version (verify nmap installation)

```
nmap --version
```

3) Basic network discovery (ping sweep / host discovery)

Scan the full local `/24` network to find live hosts (no port scan):

```
# ICMP/ARP ping-scan (fast host discovery)
# Replace interface/privileges as needed; run with admin/root on Linux/
Windows
nmap -sn 10.204.197.0/24
```

Scan a single host to check if it's up (treating 10.204.197.132):

```
nmap -Pn 10.204.197.132
```

`-Pn` disables host discovery (useful when pings are blocked) and runs port scanning directly.

---

4) Port/service scan and version detection

A standard service detection with default scripts and OS detection:

```
# Service/version detection, default scripts, and OS detection
nmap -sV -sC -A 10.204.197.132
```

If you want to scan all TCP ports (0-65535):

```
# Scan all TCP ports with service detection
nmap -p- -sV 10.204.197.132
```

Combine UDP scan (slower, requires elevated privileges):

```
# UDP scan (requires root/administrator, can be slow)
nmap -sU -p- 10.204.197.132
```

Target a set of common ports only:

```
nmap -p 22,80,443,139,445,3306 -sV 10.204.197.132
```

---

5) Save scan results (nmap output formats)

Nmap supports multiple output formats. Example filenames use a timestamp or a dedicated `scans/` folder.

```
# Normal human-readable
nmap -sV 10.204.197.132 -oN scans/10.204.197.132-scan.txt

# Grepable (for scripts)
nmap -sV 10.204.197.132 -oG scans/10.204.197.132-scan.gnmap

# XML (good for parsers and tools)
nmap -sV 10.204.197.132 -oX scans/10.204.197.132-scan.xml

# All three at once (nmap will append extensions)
nmap -sV 10.204.197.132 -oA scans/10.204.197.132-scan
```

Append `-p-` if you want all ports and keep the same `-o*` options to save results.

---

6) Example quick scan scripts

scripts/scan_local_network.sh (Bash)

```
#!/usr/bin/env bash
# Quick network discovery and save results
NET=10.204.197.0/24
OUTDIR=scans/$(date +%F_%H%M)
mkdir -p "$OUTDIR"

# Host discovery
nmap -sn "$NET" -oN "$OUTDIR/network-hosts.txt"

# Full TCP port scan + service detection for discovered hosts (example:
single host)
nmap -p- -sV 10.204.197.132 -oA "$OUTDIR/10.204.197.132-full"

echo "Results saved to $OUTDIR"
```

scripts/scan_host.sh (Bash)

```
#!/usr/bin/env bash
TARGET=10.204.197.132
OUTDIR=scans/$(date +%F_%H%M)
mkdir -p "$OUTDIR"

# Quick service detection with default scripts
nmap -sC -sV "$TARGET" -oN "$OUTDIR/$TARGET-default-scan.txt"
```

Make scripts executable:

```
chmod +x scripts/*.sh
```

---

7) Wireshark (GUI) and tshark (CLI) commands

List available capture interfaces (tshark)

```
tshark -D
```

This prints numbered interfaces; use the number or the interface name in the next commands.

Capture to file (CLI) for only traffic to/from your IP

```
# Capture all traffic where host is 10.204.197.132
# Replace 1 with the interface number from `tshark -D`
sudo tshark -i 1 -f "host 10.204.197.132" -w captures/10.204.197.132-$(date +
%F_%H%M).pcapng
```

If you prefer to filter by IP in the GUI, add a capture filter: `host 10.204.197.132`.

Filter traffic in Wireshark (display filters)

- Show packets where the IP is source or destination:

```
ip.addr == 10.204.197.132
```

- Show only HTTP traffic to/from your host:

```
ip.addr == 10.204.197.132 and http
```

- Show only traffic from your host:

```
ip.src == 10.204.197.132
```

Read a capture and export human-readable details (tshark)

```
# Verbose packet dump to text
tshark -r captures/yourfile.pcapng -V > captures/yourfile.txt

# Summary of endpoints
tshark -r captures/yourfile.pcapng -z endpoints,ip
```

---

8) Useful combined workflows

1. Discover live hosts on the /24 network and save output:

```
nmap -sn 10.204.197.0/24 -oN scans/network-discovery.txt
```

1. Run targeted service detection on an interesting host:

```
nmap -p- -sV -sC -oA scans/10.204.197.132-targeted 10.204.197.132
```

1. Capture traffic while you run a service scan (useful for protocol debugging):

```
# Start capture (background or separate terminal)
sudo tshark -i 1 -f "host 10.204.197.132" -w captures/10.204.197.132-
scan.pcapng

# Run nmap scan
nmap -sV 10.204.197.132 -oN scans/10.204.197.132-during-capture.txt

# Stop capture after scan completes (Ctrl+C in the capture terminal)
```

End of file.