

NetEye User Manual (v1.0.0): Windows Edition

High-Performance Network Diagnostic Tool

NetEye is a high-performance, single-file network diagnostic tool built with Go, designed to give you real-time visibility into the health and route of your network connections through a clean web interface.

1 Prerequisites and Setup

To run NetEye successfully, the following conditions must be met on the target Windows machine:

- **Executable:** The single, statically compiled backend file, `NetEye.exe`.
- **Dependency (CRITICAL):** NetEye relies on the underlying Windows packet capture features. You must install the latest **Npcap driver** (available on the official Nmap website) for network traffic analysis to function.
- **Permissions (CRITICAL):** The application must be run from an **Administrator Command Prompt** or PowerShell to utilize system diagnostic commands (`ping`, `tracert`) and access low-level network features.
- **Browser:** Any modern browser (Chrome, Edge, Firefox) is required to view the visualization.

2 Running and Accessing NetEye

Follow these steps to launch the application and connect to the dashboard:

1. **Open Terminal as Administrator:** Search for "Command Prompt" or "PowerShell," right-click, and select "Run as administrator."
2. **Navigate to Executable:** Use the `cd` command to go to the directory where you saved `NetEye.exe`.
Example: `cd C:\Users\Admin\Desktop\NetEye\`
3. **Execute the Application:** Run the file directly.
Example: `NetEye.exe`
4. **Terminal Output:** The server will start on a random, available port. Check the output for the access URL:
Example: Starting NetworkEye server on `http://localhost:49152`
5. **Access Dashboard:** The application will attempt to auto-open your default browser. If this fails, manually copy the `http://localhost:...` URL into your browser.

3 Frontend Features: Dashboard Deep Dive

The NetEye web interface is split into two primary diagnostic tools:

3.1 Smart Dashboard (Continuous Pinging)

This tool executes ping commands every second and analyzes the raw output to provide real-time performance metrics.

- **Input Field:** Enter the IP address or Hostname (e.g., `google.com` or `8.8.8.8`).
- **Live Metrics:**
 - **Latency (ms):** The Round-Trip Time (RTT) of the last successful ping packet.
 - **Loss (%):** The percentage of packets that failed to return a response in the total history of the session.
 - **Jitter (ms):** The variation in latency between consecutive packets, indicating network stability. High jitter is the primary cause of poor voice/video calls.
 - **Uptime (%):** The percentage of successful pings (Loss was 0%) out of the total pings attempted.
- **Export CSV:** Generates a raw data file (`.csv`) containing up to 1000 historical ping entries with timestamp, RTT, loss, and jitter data.

3.2 Traceroute Analyzer (Route Mapping)

This tool performs a single, detailed trace to map the exact path your data takes across the internet.

- **Target Input:** Enter the IP address or Hostname you want to trace.
- **Protocol Selector:** Allows selection of the protocol type (ICMP, UDP, or TCP). *(Note: On Windows, this defaults to ICMP because the native tracert utility strictly uses ICMP.)*
- **Visualization Details:**
 - **Hop IP:** The IP address of each router or device along the path.
 - **Location:** The network name (Reverse DNS) or geographic location (GeoIP) of the hop.
 - **Latency:** The average RTT to that specific hop, calculated from the three measurements reported by `tracert`.
- **Path Control:** The Go backend monitors the trace output and automatically kills the process after 3 consecutive timeouts to prevent long hangs.
- **Alerts:** A hop is flagged if its latency exceeds 100ms or if it results in a timeout.

4 Troubleshooting and Debugging

Problem: "Failed to start ping/tracert command"

Cause: You did not run `NetEye.exe` from an Administrator Command Prompt.

Solution: Close the application, and re-open your terminal explicitly as "Run as administrator."

Problem: No Latency/Only Errors on Ping

Cause: The local Windows firewall is blocking the outgoing ICMP (Ping) packets.

Solution: Temporarily disable the Windows Firewall (for testing only), or ensure an exception is created for `NetEye.exe`.

Problem: Traceroute shows only * * *

Cause: The remote firewall is configured to drop TTL-expired packets (the core function of traceroute).

Solution: This is a normal and expected result when tracing to highly secure hosts (e.g., banks, major cloud platforms) and indicates the path is firewalled.

5 API Endpoints (For Advanced Users)

The NetEye backend provides the following live API endpoints:

- `/ws/ping`: (WebSocket) Manages continuous ping sessions and streams real-time RTT, loss, and jitter data to the client.
- `/api/traceroute?target=IP/Host`: (HTTP GET) Executes a single traceroute command, returning the full path, IP addresses, and GeoIP data in a JSON object.