

AmiLeaked Documentation

AmiLeaked is a browser extension that detects network leaks in an active VPN connection. It captures your real network fingerprint as a baseline, then continuously monitors for IPv4, IPv6, DNS, and WebRTC leaks that could expose your true identity. No data is sent to a 3rd party service or verifier, all processing is done locally.

The problem

Other VPN leak detectors on the internet are based on one-off tests that the user must manually conduct, and they usually do not use a baseline as a reference, meaning leaks can't be reliably determined. This leaves them without a way to easily check their leak status without resorting to advanced methods.

Our solution

AmiLeaked aims to be what most other leak detectors aren't. The baseline system ensures absolute certainty that your detected IP's are different or the same as your initial ones, and the background detection feature allows for periodic scanning, giving the user peace of mind while they use their VPN when browsing. There's also a dedicated scan button for one-time tests, offering absolute control for leak detection and IP monitoring.

How It Works

1. **Baseline Capture** — On first setup, you disconnect your VPN and AmiLeaked records your real IP fingerprint (IPv4, IPv6, DNS resolver, WebRTC IPs).
2. **On-Demand Scan** — With your VPN active, press the power button to run a full scan. If any scanned values match your baseline, a leak is flagged.
3. **Background Detection** — Optionally runs scans every 10-60 seconds in the background and alerts you via Badge notifications if a leak is detected.

Features

- **Multi-vector leak detection** - Checks IPv4, IPv6, DNS, and WebRTC simultaneously
- **Background monitoring** - Periodic scans with extension Badge notifications
- **WebRTC leak detection** - Uses STUN servers to discover public and local IPs exposed via `RTCPeerConnection`
- **DNS leak detection** - Queries Google DNS to identify your resolver IP
- **Dual IPv4 sources** - Primary fetch via [ipify](#), automatic fallback to [AWS checkIP](#)

- **First-time setup page** - Guided baseline capture with confirmation steps
- **Re-capture baseline** - Update your fingerprint anytime from the Settings page
- **Dark / Light theme** - Toggle between themes
- **Hide sensitive info** - Mask IP address information in the popup UI

Tech Stack

- **React 18 + React Router** - Popup UI and page navigation
- **Vite + vite-plugin-web-extension** - Build tooling with hot reload
- **Chrome Extension APIs** - chrome.storage, chrome.notifications, chrome.action
- **WebRTC** - RTCPeerConnection for local/public IP discovery

Project Structure

src/

```

|— background.js           # Service worker and background script worker
|— manifest.json          # Extension manifest (Contains Chrome template)
|— popup.html / popup.jsx  # Initial popup html and jsx file
|— offscreen.html         # Loads an offscreen JavaScript module for the WebRTC
background service
|— offscreen.js           # Minimal JavaScript file for fetching WebRTC IPs from the
background service
|—--- pages/
|   |— Popup.jsx          # Main script file for leak detection behavior and other services
|   |— Settings.jsx       # Settings page (theme, notifications, re-capture baseline)
|   |— Setup.jsx          # First-time baseline capture set-up
|—--- services/
|   |— ipService.js        # Public IPv4/IPv6 and DNS resolver fetching
|   |— webrtcService.js    # WebRTC IP fetching via STUN
|   |— backgroundIPService.js # Periodic background scan service
|   |— webrtcBackgroundService.js # Background service for WebRTC STUN requests
|   |— notificationService.js # Badge notification alerts for detected leaks

```

```
|--- storage/
|   |--- storageService.js      # Persistent storage for baseline and settings
|--- content_scripts/
|   |--- content.js            # Content script (injected into pages)
|--- utils/
|   |--- timeUtils.js          # Timestamp utilities
```

Getting Started

Prerequisites

- [Node.js](#) (v18+)
- npm or yarn

Install & Build

npm install

npm run build

Development

npm run dev

Load in Chrome

1. Navigate to `chrome://extensions`
2. Enable **Developer mode**
3. Click **Load unpacked** and select the dist folder

Usage

1. **Setup** - Disconnect your VPN, open the extension, and complete the baseline capture.
2. **Scan** - Reconnect your VPN, then press the power button to check for leaks.
3. **Background mode** - Enable background detection for continuous monitoring with notifications.

License

This project is licensed under the GNU General Public License v3.0.