PrivacyRadar

SRS 1.0

Software Requirements Specification

Authors:
David Osborne
Luke Therieau
Madhur Deep Jain
Yashna Praveen
Sujal Charak

Table of Contents

Introduction

1.1 Overview

PrivacyRadar is a privacy-focused network monitoring platform designed to give users clear visibility into how their applications interact with the internet and access system resources. The product captures network traffic, monitors hardware access (camera, microphone, location services, screen capture, and clipboard), associates all activity with specific processes, and presents the results in a clear, application-level view. This allows users to understand which applications are communicating, what destinations they contact, and how much data they transmit, and what privacy-sensitive hardware they access.

Unlike traditional packet capture tools that overwhelm users with raw traffic, PrivacyRadar simplifies analysis by normalizing data and presenting it through intuitive dashboards. The platform monitors both network activity and hardware resource access, providing comprehensive privacy intelligence. The platform's goal is to transform raw network activity and hardware access events into actionable privacy intelligence, allowing users to detect unauthorized data collection, hardware exploitation, or suspicious behavior without requiring expert networking knowledge.

1.2 Benefits

An application-level network monitoring tool provides many clear benefits to its users. By providing structured, process-aware visibility into network and privacy activities, the product helps users quickly identify relevant traffic, troubleshoot connectivity issues, and make informed decisions about application behavior:

- **Application Transparency:** By linking captured traffic with specific processes, the user can see which applications are transmitting data, what resources they are connecting to, and the nature of those connections.
- **Efficient Analysis:** Instead of producing a raw packet stream that requires technical knowledge to interpret, the system normalizes traffic into structured records, making analysis faster and more accessible.

- **Privacy Protection:** The system highlights access to sensitive resources such as cameras, microphones, and location services, allowing users to detect potential risks.
- **Troubleshooting Support:** By linking bandwidth usage, connection states, and destinations to individual processes, users can quickly identify applications that are consuming excessive resources or communicating with suspicious endpoints.
- **Historical Reporting:** Logged activity, visualizations, and export options allow both casual users and professionals to review trends over time and conduct further analysis using external tools.
- **Cross-Platform Usability:** Support for Windows, Linux, and MacOS ensures that users across different operating systems will benefit from consistent monitoring and reporting.
- **AI-Powered Intelligence:** The integrated AI assistant provides natural language querying, allowing users to ask questions in plain English and receive actionable insights without requiring technical expertise, making advanced network analysis accessible to everyone.

1.3 Scope

PrivacyRadar is designed as a monitoring utility that provides network visibility and privacy-related activities, but does not actively block or restrict connections. Its purpose is to observe, record, and present traffic and privacy events in a structured manner that highlights process-level behavior. The product will operate as a cross-platform tool, supporting Windows, Linux, and MacOS, to ensure broad usability across different system environments. PrivacyRadar's dashboard includes a 'Basic' mode and 'Advanced' mode to satisfy the scope of its users; a general user with little to no networking knowledge, to developers, researchers, or network administrators with years of expertise.

The product provides a comprehensive alert and notification system that allows users to configure custom thresholds for data usage, monitor connections to suspicious destinations, and receive real-time notifications about privacy-sensitive hardware access. These alerts can be configured through both traditional UI controls and natural language commands via the AI assistant, enabling users to set up sophisticated monitoring rules without technical complexity.

High-Level Requirements

2.1 Functional Requirements

The following functional requirements specify all services and observable behaviors that PrivacyRadar must provide:

FR1: Network Monitoring Core Functions

- FR1.1: The product shall continuously monitor all network traffic generated by installed applications
- FR1.2: The product shall record upload and download data volumes for each application since monitoring began
- FR1.3: The product shall identify and log connection destinations including IP addresses, domain names, and ports
- FR1.4: The product shall perform geographical lookup of IP addresses to determine data destination countries
- FR1.5: The product shall maintain historical records of all network activities for trend analysis

FR2: Privacy Monitoring Functions

- FR2.1: The product shall detect and log all camera access attempts by applications
- FR2.2: The product shall detect and log all microphone access attempts by applications
- FR2.3: The product shall detect and log all location service requests by applications
- FR2.4: The product shall detect when applications capture screenshots or record the screen
- FR2.5: The product shall monitor and log clipboard access by applications

FR3: Alert and Notification System

- FR3.1: The product shall allow users to configure application-level data usage thresholds

- FR3.2: The product shall allow users to configure global network usage thresholds
- FR3.3: The product shall generate real-time alerts when configured thresholds are exceeded
- FR3.4: The product shall notify users of connections to suspicious or blacklisted destinations
- FR3.5: The product shall alert users when privacy-sensitive hardware is accessed

FR4: Data Visualization and Reporting

- FR4.1: The product shall provide graphical representations of network activity over time
- FR4.2: The product shall generate application-specific network usage reports
- FR4.3: The product shall create visual maps showing geographical data destinations
- FR4.4: The product shall display real-time network activity dashboards
- FR4.5: The product shall provide comparative analysis between applications

FR5: AI Assistant Integration

- FR5.1: The product shall provide natural language query processing for data analysis
- FR5.2: The product shall generate plain-language summaries of network activities
- FR5.3: The product shall classify network traffic types (telemetry, updates, user data, malicious)
- FR5.4: The product shall provide AI-powered privacy risk assessments
- FR5.5: The product shall enable voice or text-based alert configuration through AI

FR6: Data Management and Export

- FR6.1: The product shall export monitoring data in CSV format
- FR6.2: The product shall export monitoring data in JSON format
- FR6.3: The product shall export monitoring data in XML format
- FR6.4: The product shall provide data filtering before export
- FR6.5: The product shall maintain data integrity during export operations

2.2 Non-Functional Requirements

PrivacyRadar is designed to provide high-quality network monitoring while maintaining strong performance, reliability, and security under typical and heavy workloads. Traffic capture and display must be fast and consistent to ensure users can make timely decisions based on current network activity. Scalability is a key consideration, allowing the system to handle thousands of packets per second without loss or significant delays. Reliability and stability are critical, ensuring the product remains responsive, avoids crashes, and handles errors appropriately. The RAIL performance model is used as a guideline to define and measure these goals, resulting in a smooth, user-focused experience.

2.2.1 RAIL Performance Model (Response, Animation, Idle, Load)

Response:

- The product shall detect and display available network interfaces within 100ms of startup.
- When the user selects an interface, capture shall begin within 200ms.
- Filtering network traffic by an product shall update the GUI within 200ms.

Animation:

- Visualizations shall refresh smoothly at 60 FPS or degrade gracefully during heavy load traffic.
- Transition from Basic to Advanced mode shall complete within 100ms to feel responsive.
- Graphical elements such as charts and tables shall update incrementally to maintain smooth motion and avoid flickering, even when large volumes of traffic data are processed.

Idle:

- Background processes such as device polling and logging shall not impact overall system responsiveness.
- Background analytics and data aggregation tasks shall run without noticeable impact on the user interface or network capture performance.
- Periodic database maintenance, such as indexing or cleanup, shall occur in the background without interrupting real-time traffic monitoring or causing UI lag.

Load:

- The product shall handle at least 10,000 packets/sec without packet loss on supported hardware.
- For high-throughput interfaces (1 Gbps+), traffic capture shall remain stable for sessions lasting 24 hours or more.
- Application startup time shall not exceed 2 seconds on supported systems.

2.3 User Stories

General User Stories - Network Monitoring:

GU-NM-001: As a general user, I want to see a summary dashboard showing total network usage since PrivacyRadar started, broken down by application, so that I can quickly identify which apps consume the most bandwidth.

GU-NM-002: As a general user, I want to click on any application to see detailed connection information including websites visited and data transferred, presented in non-technical language.

GU-NM-003: As a general user, I want to see a world map highlighting countries where my data is being sent, with privacy risk indicators for each location.

General User Stories - Privacy Protection:

GU-PP-001: As a general user, I want to receive immediate pop-up notifications when any application accesses my camera or microphone, with options to investigate further or take action.

GU-PP-002: As a general user, I want a privacy score for each application based on its behavior, helping me understand which apps respect my privacy.

GU-PP-003: As a general user, I want to see a timeline view of when applications accessed privacy-sensitive features throughout the day.

GU-PP-004: As a general user, I want to receive an alert when an application connects to a suspicious or blacklisted domain, with a short explanation and options to Allow/Block/Learn More, so that I can protect my data from unsafe connections and stay in control of my privacy.

General User Stories - AI Integration:

GU-AI-001: As a general user, I want to ask the AI assistant questions like "Which apps sent data to China today?" and receive clear, actionable answers.

GU-AI-002: As a general user, I want the AI to proactively suggest privacy improvements based on detected patterns in my application usage.

GU-AI-003: As a general user, I want weekly AI-generated privacy reports summarizing important findings and recommendations.

Advanced User Stories - Technical Analysis:

AU-TA-001: As an advanced user, I want to filter network connections by specific TCP/UDP ports to identify service-specific traffic patterns.

AU-TA-002: As an advanced user, I want to see raw packet count and byte-level statistics for each connection.

AU-TA-003: As an advanced user, I want to correlate process IDs with network connections for forensic analysis.

AU-TA-004: As an advanced user, I want to create custom alert rules using regex patterns for domain names and IP ranges.

Advanced User Stories - Data Management:

AU-DM-001: As an advanced user, I want to export monitoring data.

AU-DM-002: As an advanced user, I want to create custom SQL queries against the monitoring database for specialized analysis.

AU-DM-003: As an advanced user, I want to configure data retention policies to automatically purge old records based on my requirements.

Detailed Requirements

3.1 Application Details

3.1.1 Capture Engine and Monitoring Layer

The capture engine in PrivacyRadar is built on the Cap library for Node.js, utilizing Npcap on compatible systems (Windows, Linux, MacOS) to provide raw packet capture. To reduce processing overhead, the engine supports Berkeley Packet Filter (BPF) syntax so that only relevant traffic is captured. Relevant traffic includes all application-level communication associated with a process ID (PID) as well as system-level traffic, ensuring that the monitoring data is meaningful and actionable.

Network Monitoring: Captured packets are processed and associated with active processes using system tools like netstat and ps-list. The system maintains a mapping of (IP, Port, Protocol) tuples to their corresponding PIDs and executable names. Connection state tracking and protocol decoding for common protocols are handled at this layer.

Hardware Monitoring: In addition to network traffic, this layer monitors privacy-sensitive hardware access including:

- Camera access detection through system API hooks
- Microphone usage monitoring via audio device APIs
- Location services access tracking through geolocation APIs
- Screen capture and recording detection
- Clipboard access monitoring

3.1.2 Processing and Data Layer

All captured traffic and hardware events are normalized into a structured format with consistent fields to simplify further analysis. The processing layer handles data transformation, enrichment, and storage.

Data Processing:

- Normalizes raw packet data into standardized records

- Handles protocol-specific parsing (HTTP headers, DNS queries, TLS SNI)
- Resolves domain names through reverse DNS lookups
- Performs IP geolocation using MaxMind GeoIP2 or similar databases
- Calculates per-process statistics including bandwidth, packet counts, and latency distributions
- Handles unmatched traffic or system service packets with no PID association

AI-Powered Analysis: The AI Processing Module provides following analysis:

- Classifiers for traffic categorization (telemetry, updates, user data, tracking, malicious)
- Natural language query processing for user questions
- Automated recommendation generation for privacy improvements

Data Storage: All monitoring data is stored in an encrypted SQLite database.

Alert Management: The processing layer generates real-time alerts based on:

- User-configured thresholds (bandwidth limits, connection counts)
- Connections to blacklisted or suspicious destinations
- Hardware access by applications (camera, microphone, location)
- Anomalous traffic patterns and behavioral changes
- Alerts are persisted in an Alerts Panel with filtering by application, domain, and user action
- Each alert provides Allow, Block, and Learn More options, with user decisions stored to avoid repeated notifications

3.1.3 User Interface Layer

The desktop interface is built using Electron (cross-platform application container), React.js (component-based UI framework), and Node.js (runtime environment), providing a responsive and intuitive experience for both general and advanced users.

Technology Stack:

- **Electron**: Desktop application framework providing native system integration
- **React.js**: Frontend UI library for building interactive components
- **Node.js**: Backend runtime environment for application logic
- **Tailwind CSS**: Utility-first CSS framework for responsive styling

- **Chart.js & D3.js**: Data visualization libraries for charts, graphs, and geographic maps

Interface Modes: The UI displays all detected network interfaces and can suggest the most active one to simplify setup. PrivacyRadar offers two interface modes:

- **Basic Mode**: Summarizes overall traffic and highlights top applications by usage, privacy scores, and recent hardware access events. Includes geographic maps showing global traffic destinations.
- **Advanced Mode**: Provides detailed packet breakdown with header information for Ethernet, IP, and Transport layers. Includes well-known port identification, real-time traffic viewer, custom filtering, interactive timelines, and raw packet inspection.

Graphical Visualizations:

- Real-time bandwidth usage charts
- Per-process network usage charts
- Protocol distribution pie charts
- Interactive world map displaying global traffic paths to/from the user's device
- Timeline view of hardware access events
- Alert dashboard with severity indicators
- Light and dark mode themes

3.1.4 Dependencies and Privileges

PrivacyRadar relies on several key external libraries, system utilities, and runtime environments to provide its comprehensive monitoring capabilities. Understanding these dependencies is crucial for deployment, troubleshooting, and security considerations.

Core Runtime Dependencies:

1. *Node.js (v16.x or higher)*
    - JavaScript runtime environment for all backend processing
    - Event-driven architecture for efficient I/O operations
    - Native addon support for system-level integration
    - Package management via npm

2. *Electron (v22.x or higher)*
    - Cross-platform desktop application framework
    - Chromium-based rendering engine for UI
    - Native menu, notification, and dialog APIs
    - Inter-process communication (IPC) between main and renderer processes
3. *React.js (v18.x or higher)*
    - Frontend UI library for building the user interface
    - Component-based architecture for modular design
    - Virtual DOM for efficient rendering
    - Hooks API for state management and side effects

Network Capture Dependencies:

1. *Cap Library (latest stable)*
    - Node.js binding for packet capture functionality
    - Cross-platform abstraction over OS-specific capture mechanisms
    - Real-time packet delivery to JavaScript layer
2. *Npcap (Windows) / libpcap (Linux/MacOS)*
    - System-level packet capture drivers
    - Kernel-mode packet filtering
    - Network interface enumeration
    - Installation requirements:
        - Windows: Npcap installer (WinPcap API-compatible mode)
        - Linux: libpcap-dev package
        - MacOS: Built-in libpcap (no installation needed)

System Utility Dependencies:

1. *netstat*
    - Connection-to-process mapping on all platforms
    - Active connection enumeration
    - Socket state information
    - Platform-specific implementations:
        - Windows: netstat -ano
        - Linux: netstat -tupn or ss -tupn
        - MacOS: netstat -anvp tcp and netstat -anvp udp
2. *ps-list (Node.js package)*

- Cross-platform process enumeration
- Process metadata extraction (PID, name, path, user)
- CPU and memory usage statistics
- Parent-child process relationship mapping

UI and Visualization Dependencies:

1. *React-related packages:*
   - react-dom: DOM rendering for React components
   - react-router-dom: Client-side routing
   - react-redux: State management (optional)
2. *Chart.js (v4.x)*
   - Charting library for data visualizations
   - Responsive canvas-based rendering
   - Animation and interaction support
3. *D3.js (v7.x)*
   - Advanced data visualization toolkit
   - Geographic map rendering
   - Custom interactive visualizations
4. *Tailwind CSS*
   - Utility-first CSS framework
   - Responsive design system
   - Dark mode support
5. *Electron Notifications API*
   - Native Cross-platform notifications
   - Used for real-time alert delivery (e.g., suspicious domain connections)

Data Storage Dependencies:

1. *SQLite3 (Node.js binding)*
   - Embedded relational database
   - No separate server process required
   - ACID compliance and transaction support

AI and Machine Learning Dependencies:

1. *LangChain.js*
   - Cloud and local model support
2. *Template-based fallback*

    - Provides plain-language explanations when AI services are unavailable
    - Ensures suspicious domain alerts remain understandable without external dependencies

Privilege Requirements:

PrivacyRadar requires different privilege levels depending on the operation being performed:

Administrator/Root Privileges Required For:

- Raw packet capture initialization
- Npcap/libpcap driver access
- Network interface manipulation
- System-wide process enumeration on some platforms
- Hardware access monitoring hooks (camera, microphone)
- Installation and first-time setup

Standard User Privileges Sufficient For:

- UI rendering and interaction
- Database queries and visualization
- Export operations
- Configuration changes
- AI assistant interactions
- Viewing historical data

Accessing raw packet capture requires administrator or root privileges. Standard privileges are sufficient for monitoring, visualization, and UI operations once capture is initiated. The application will warn users and disable capture if the necessary privileges are missing, ensuring security and stability.

3.2 Functional Specifications

The PrivacyRadar GUI provides comprehensive access to all monitoring, analysis, and configuration features through an intuitive interface designed for both general and advanced users.

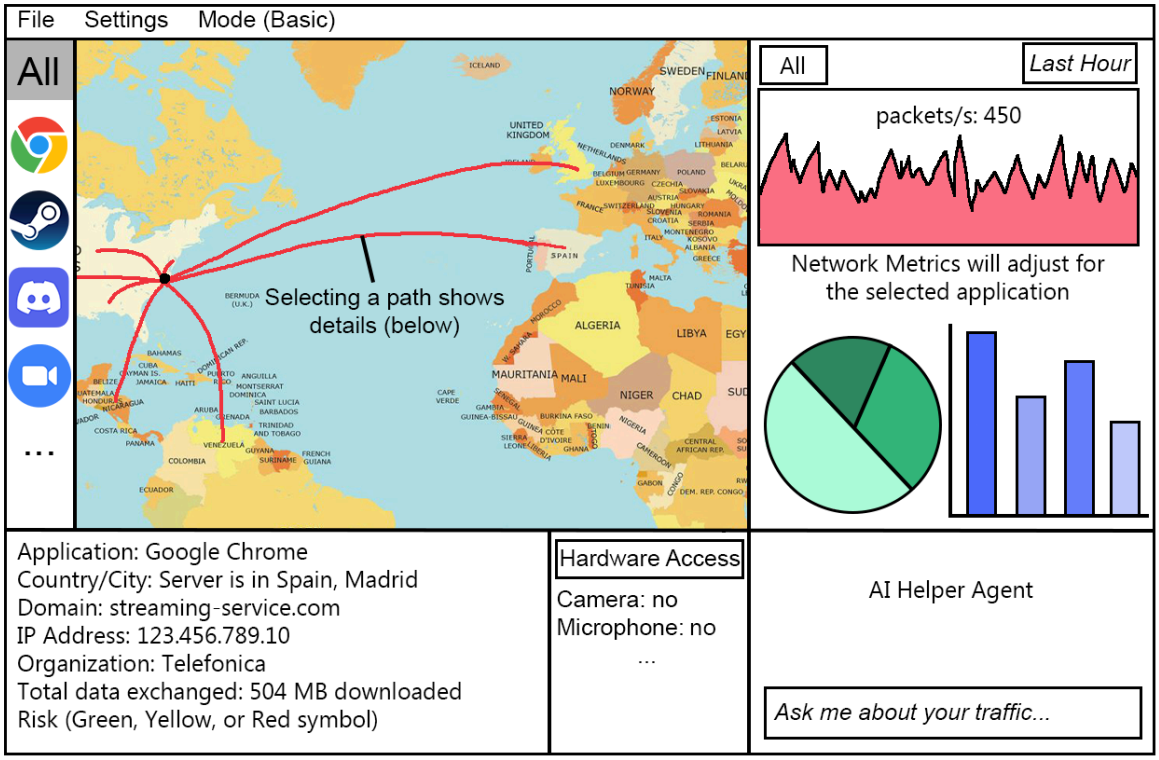
Main Dashboard

The Main Dashboard serves as the primary landing page and provides an at-a-glance overview of network activity and privacy events.

Basic Mode Components:

- **Summary Statistics Panel**: Total data uploaded/downloaded since monitoring began, active connections count, number of monitored applications
- **Top Applications Widget**: Bar chart showing the 5-10 applications with highest bandwidth usage, with clickable entries for detailed views
- **Privacy Score Overview**: Visual cards displaying privacy scores for all monitored applications, color-coded by risk level (green/yellow/red)
- **Recent Hardware Access Timeline**: Scrollable timeline showing recent camera, microphone, location, and clipboard access events
- **Geographic Overview Map**: World map with highlighted countries where data is being sent, with size/color indicators for data volume
- **Quick Status Indicators**: Active alerts count, capture status, system health indicators

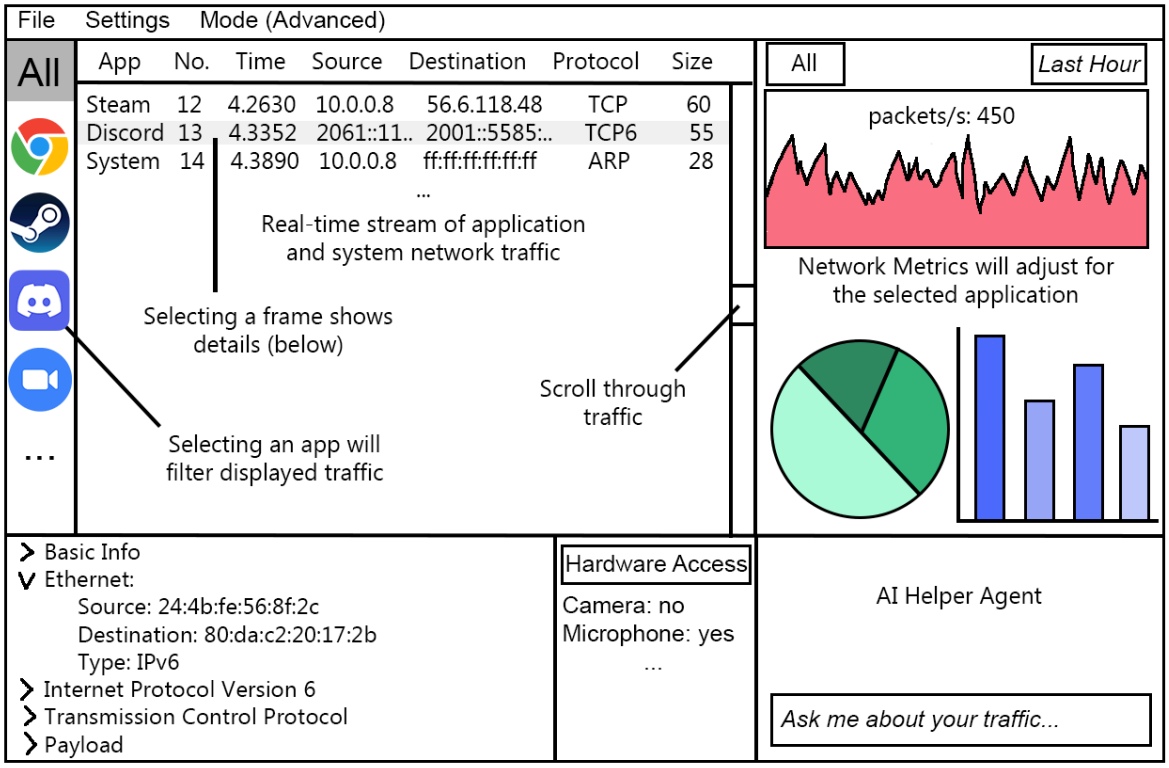
Basic Mode UI Concept:



Advanced Mode Components:

- All Basic Mode components plus:
- **Real-Time Traffic Viewer:** Auto-scrolling table with newest connections at top with columns consisting of Timestamp, Application Name, Source Address:Port, Destination Address:Port, Protocol, Bytes Sent/Received, Duration
- **Connection State Table**: Detailed table of all active connections with columns for PID, application, protocol, local/remote addresses, state, and bytes transferred
- **Bandwidth Graph**: Real-time line chart showing network throughput over the last 5 minutes/1 hour/24 hours (user-selectable)
- **Protocol Distribution**: Pie chart breaking down traffic by protocol type
- **Alert Configuration Panel**: Quick access to create and manage alert rules
- **System Resource Monitor**: CPU and memory usage of PrivacyRadar itself

Advanced Mode UI Concept:



System Requirements and Assumptions

4.1 Requirements

Hardware Requirements:

- Minimum: Dual-core CPU, 4GB RAM, 500MB disk space
- Recommended: Quad-core CPU, 8GB RAM, 2GB disk space for extended monitoring

Operating System Requirements:

- Windows 10/11 (64-bit)
- Linux (kernel 4.x or higher) with libpcap support
- MacOS 11.0 (Big Sur) or higher

Network Requirements:

- Administrative privileges for packet capture
- Compatible network adapter drivers
- Npcap (Windows) or libpcap (Linux/MacOS) installed

4.2 Assumptions

User Knowledge:

- Basic users understand fundamental networking concepts (e.g., "internet connection")
- Advanced users have technical networking knowledge
- Users can follow privilege escalation prompts when needed

System Environment:

- System has stable power and disk space for continuous monitoring
- Network traffic volume is typical for consumer or small business use
- Applications generate identifiable network traffic