Requirements Document

The Superb Security Specialists

**Feature Requirements:**

**GUI Features**

* Tabbed interface to separate functionalities:
  + HIDS: Monitor traffic, SYN packets, and thresholds.
  + Process Monitor: Real-time CPU/memory monitoring.
  + Custom/NIDS tab for advanced features and future extensions.
* Provide visual feedback for alerts, with dynamic, real-time charts.

**HIDS Features**

* Monitor incoming traffic for SYN packets.
* Allow users to define ports to monitor.
* Set thresholds for SYN packet counts and alert users upon breaches.
* Plot real-time SYN packet data dynamically.

**Process Monitoring Features**

* Allow users to set thresholds for CPU and memory usage.
* Continuously monitor system processes.
* Alert users when thresholds are breached, including:
  + Pop-up alerts showing process details (Name, PID, CPU, Memory usage).
* Provide an option to terminate flagged processes from the interface.

**Future NIDS Features (WIP)**

* Packet filtering with user-defined rules.
* Anomaly detection system.
* Network traffic analysis and intrusion simulation scripts.

**Platform Features**

* Cross-platform functionality.
* Threaded monitoring to maintain GUI responsiveness.

**Deployment Features**

* Include VM/machine deployment scripts for intrusion simulation.

**Logging and Customization**

* Logs for auditing (future development).
* User-defined behavior customization for alerts.

**Performance Requirements:**

**Real-Time Monitoring**

* GUI must remain responsive with a refresh rate of less than 100ms during monitoring.
* Traffic monitoring and process monitoring should operate in separate threads to avoid lag.

**SYN Packet Monitoring**

* Process up to 10,000 SYN packets per second without dropping packets or slowing down the GUI.
* Visual charts must update within 500ms of data changes.

**Process Monitoring**

* Poll system processes every 1 second for CPU and memory usage.
* Alerts must trigger within 2 seconds of threshold breaches.

**Resource Usage**

* CPU usage by the HIDS system should not exceed 10% on a standard machine.
* Memory usage should not exceed 500 MB during operation.

**Scalability**

* Must support up to 10,000 monitored processes and 100 monitored ports concurrently without degradation in performance.

**Privacy Requirements:**

**Data Handling**

* All monitored data (SYN packets, process information) should be kept local to the user's machine.
* No external servers or cloud storage for monitoring data unless explicitly authorized by the user.

**Data Minimization**

* Collect only essential data needed for monitoring thresholds and detecting anomalies.
* Avoid logging or displaying sensitive user information unnecessarily.

**Access Control**

* Restrict access to monitoring and alerts to the user of the system.
* If logs are implemented, provide encryption and password protection for log files.

**Compliance**

* Adhere to privacy regulations such as GDPR or CCPA if user data logging/alerting involves personal information.

**Security Requirements:**

**System Integrity**

* Ensure the HIDS itself is secure against external tampering or misuse.
* Employ code signing or checksum validation to verify program authenticity.

**Secure Communication**

* If remote logging or communication is added, ensure encrypted connections using TLS/SSL.

**Process and Resource Management**

* Ensure that flagged processes are terminated safely and only with user consent.
* Validate user-defined thresholds to prevent abuse or incorrect inputs that may harm the system.

**Access Control**

* Limit administrative actions (e.g., defining thresholds, terminating processes) to authorized users.

**Attack Mitigation**

* Harden the system against Denial of Service (DoS) attacks on monitored ports.
* Protect the GUI and system threads from injection attacks or privilege escalation.

**Alerts and Logging**

* Prevent alert spoofing or false logging by validating all monitored data before triggering alerts.
* Logs must be tamper-proof, potentially using encryption or integrity checks.

**Fail-Safe Mechanisms**

* Ensure the system can recover gracefully from unexpected crashes or resource overloads.
* Notify users of failure states and allow restart options without compromising functionality.