SWAVLAMBAN 2025 HACKATHON CHALLENGE – 2

DEVELOPMENT OF A UNIFIED CROSS-PLATFORM IPSEC SOLUTION FOR SECURITY OF DATA IN TRANSIT IN A NETWORK

1. **Challenge Overview**. Enterprises and government networks consist of numerous devices running diverse operating systems such as Windows, Linux (Debian/ Ubuntu, RHEL, BOSS, Fedora) and MacOS etc. Ensuring secure and consistent encryption of all network traffic originating from these devices remains a critical challenge. While each operating system supports inherent IPsec, the configuration methods, management tools and APIs vary widely. This inconsistency leads to deployment delays, user overheads, operational complexity and security misconfigurations.

2. The challenge is to design and develop a unified IPsec solution that can be deployed on every node in a network, irrespective of its operating system. Once enabled by a network administrator, the solution should automatically protect network traffic based on centrally defined security policies. The solution must support all IPsec modes, allow the administrator to select the desired cryptographic and operational parameters and ensure persistent, automated and interoperable secure communication between devices.

3. **Core Technical Objectives**

3.1 **Unified Cross-Platform IPsec Solution**. Develop a cross-platform IPsec solution capable of automatically configuring and managing encrypted network traffic on Windows, Linux (Debian/ Ubuntu, RHEL, BOSS, Fedora) and MacOS etc**.** The solution should operate uniformly across all devices in the network. The solution should be deployable once by the administrator on each device and automatically enable itself at system start-up without manual intervention.

3.2 **Supports for all IPsec Modes**. The solution must support all IPsec operational modes (ESP Tunnel mode, ESP Transport mode, AH tunnel mode, AH transport mode, combined ESP + AH) enabling network administrator to select an option most suited for the organisation’s security model.

3.3 **Complete or selective Traffic Encryption**. The solution must support policies allowing the network administrator to choose between encryption of all outgoing traffic or encryption of only specific subnets, destinations or exception of ICMP or any requirement for network configuration. The solution should support both site-to-site and host-to-host encrypted tunnels. Allow seamless routing of encrypted traffic through the appropriate tunnel.

3.4 **Latency**. Implementation of packet encryption should not affect latency of the network traffic to the extent feasible.

3.5 **Flexible Cryptographic and Key Exchange controls.** Network administrator must be able to configure and enforce various parameters such as Encryption algorithms (AES-128, AES-256 or other supported options), Integrity algorithms (SHA1, SHA-256, SHA-384 etc), Key Exchange parameters (IKEv1/ IKEv2/ DH group selection etc.) and Authentication methods (PSK or certificate based) uniformly across all devices in the network.

3.6 **Persistent and Automated Operation.** The solution should be enabled once by the administrator on each device. After initial setup, the client should auto-start on system boot and automatically re-establish tunnels if disconnected. The solution should persist configurations securely without manual user re-entry.

3.7 **Automation and Interoperability**. Automate Security Association (SA) negotiation, renewal and tear-down. It should automatically manage firewall/ routing requirements (e.g. opening UDP 500/ 4500, ESP protocol 50, AH protocol 51). It should ensure interoperability across various OS platforms. Provide status checks, error handling and logs for administrator visibility.

4. **Evaluation and Scoring**. Evaluation of the solution would consider the following key aspects:-

4.1 **Unified Configuration**. A centralised or per-device configuration approach that supports all OS using the same policies and parameter structure.

4.2 **Automatic Encryption**. All outgoing network traffic from the device is automatically encrypted using the selected IPsec mode and policies.

4.3 **Cross-Platform Support**. Functionality across Windows, Linux, MacOS, and BOSS OS with uniform security parameters.

4.4 **Multi-Tunnel Capacity**. Support for simultaneous multiple encrypted tunnels to different peers, subnets or workloads.

4.5 **Auto-Start and Recovery**. Automatic activation at boot and reconnection of dropped tunnels.

4.6 **Monitoring and Logs**. Maintain detailed logs of tunnel status, uptime, rekey events and traffic summaries.

4.7 **Error Handling**. Detect and report configuration errors, key mismatches or peer unavailability.

4.8 **Operability with all network devices**. For seamless user experience, availability of network devices such as printers, scanners etc should be feasible even post implementation of the solution.

5. **Bonus Features**

5.1 **Tunnel Visualization**. Graphical representation of active tunnels and endpoints.

5.2 **Real-Time Traffic Monitoring**. Bandwidth, latency, packet loss and encryption statistics.

5.3 **Event Logs and Alerts**. Notifications for tunnel downtime, security warnings, or configuration issues.

5.4 **Configuration Dashboard**. Interface for adding/ removing tunnels, viewing SA details and managing keys.

6. Final Deliverables

6.1 **Fully Functional IPSEC Solution implementation**. Complete IPsec solution, packages and ready to deploy on Windows, Linux, MacOS and Boss OS. To be submitted as part of the private repository on GitHub classroom.

6.2 **Source Code.** Complete source code with clear directory structure, comments and coding standards. Build scripts for each supported OS and a README explaining how to build and run the solution. To be submitted as part of the private repository on GitHub classroom.

6.3 **Technical Documentation (2-3 pages)**. Covering technical architecture, code description, installation procedure (along with dependencies, Admin/ user manual. To be submitted as part of the Product Description Document.

6.4 **Test set-up and validation document**. Test environment description, test scenarios, Test results (Logs, screenshots, evidences of encrypted traffic, performance analysis and latency reports (comparison with clear traffic). To be submitted as part of the private repository on GitHub classroom.

6.5 **Demonstration Video (5 minutes)**. Demo video showing installation, configuration, starting client, verifying tunnels and SAs, encrypted traffic across multiple OS, Multi-tunnel scenario, auto-start demo and logs etc. The first round of the evaluation would be completely based on the video submitted.

6.6 **Presentation (8-10 slides)**. Short presentation containing problem statement, solution approach, architecture, Key features, demonstration summary, challenges faced and future enhancements.

Note – Participants may submit their solutions that may incorporate all/ some of the abovementioned features. Reasons for noncompliance may be indicated.

The Code may be submitted in the mentioned link-

https://classroom.github.com/a/YvbnXGej

You need to submit your GitHub user name and link to the private repository provided by the organizer via the GitHub Classroom.