SageVPN

Tucker Cook, Ayush Verma

University of CINCINNATI CEAS

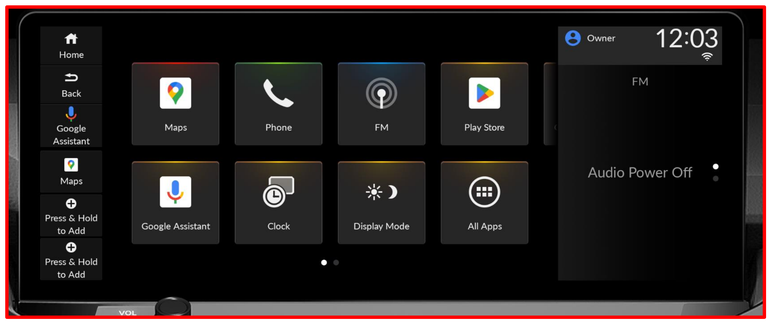
CEAS EXPO

Abstract

Most newly manufactured vehicles utilize wireless network connections for various vehicle functionalities. From using services like *Google Maps*, mirroring your phone to your infotainment display, or even your driver habits contain data that is sent within the vehicle or remotely over a network. Use of this technology in automotive raises serious concerns for user privacy and security if these network connections are not secure. By 2030 about 95 percent of new vehicles sold globally will have internet connectivity. To circumvent these concerns, SageVPN provides a VPN service that all vehicle network traffic can be routed through and provide drivers with options to view, intercept, or even spoof network data.

Purpose

SageVPN aims to address increasing security and data privacy concerns in the automotive sector. Using a virtual private network deployed in our application, we can regulate the data packets sent from vehicles and prevent cases where user data is unfoundedly used against consumers.

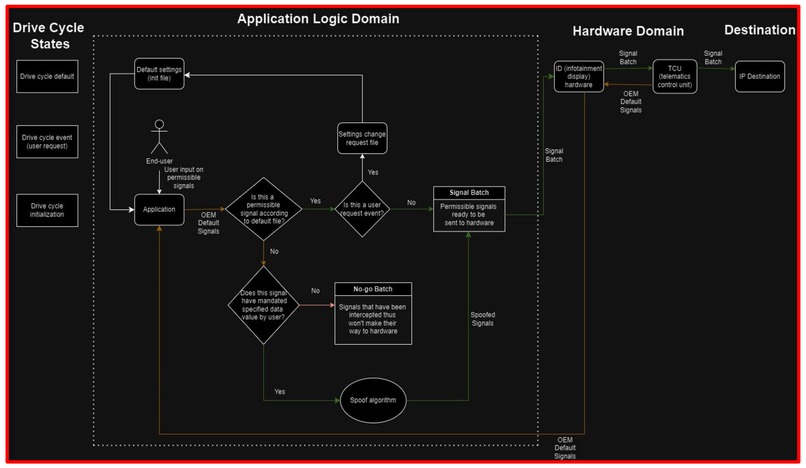


Method

SageVPN is designed to intercept network traffic and perform various operations that enhance security beyond OEM specifications. The following methods outline its key functionalities and implementation:

- **Intercept**
  - While VPN service is up, all network traffic from vehicle is routed to and displayed in the application UI
  - All data about the network connection is viewable, source/dest, protocol, size, etc.
- **Block**
  - Network connections may be completely blocked from reaching their destinations, this is at driver discretion.
  - Safety features that rely on network connectivity such as e911 are not available to be changed by driver.
  - In the future we hope to implement a *smart block* that automatically flags and blocks potentially malicious signals from third parties.
- **Spoof**
  - SageVPN implements spoofing through the following methods:
    - **IP Address Spoofing** – Sets up virtual IP for data to be sent to
    - **DNS Spoofing** – Modify DNS responses to redirect traffic
    - **Headers and Payload Modification** – modifies packet data and reconstructs packet with modified request
    - **Location Spoofing by Manipulating Network Identifiers** – replace actual cell tower IDs, MCC, MNC with spoofed values

Architecture



The diagram describes **SageVPN's** main intended execution flow. Sage relies on a permission system for network signals—connections will be flagged as blocked/spoofed based on driver choice. These settings are saved in default settings & persist across vehicle drive cycles, i.e. turning on or off vehicle has no effect on default settings.

When user input is received on permissible signals:

- Authenticates the user against stored credentials
- Validates the permission level for requested modifications protocol, size, etc.
- Creates an audit trail for permission changes

Protocol handling:

- **TCP/IP Traffic**: Validated at packet level before forwarding
- **UDP Datagrams**: Screened for permissible destination addresses
- **HTTP/HTTPS**: Deep packet inspection for header and content validation

The persistent nature across drive cycles is achieved by storing the VPN configuration in the vehicle's secure storage area that survives power cycles, ensuring immediate protection upon vehicle startup.

Tech Stack

Challenges

- **Compatibility** – developing application to be compatible with older versions of Android which is common in the automotive sector
- **Android restrictions** – In order for our application to get full access to vehicle network traffic, Android restrictions must be disabled, leading to more protection our app needs to offer
- **Manufacturer restrictions** – Different manufacturers will put different restrictions/policies on network traffic in their infotainment systems based on their policy
- **Safety concerns** – Under no circumstance can our application interfer with a safety critical signal or any of that signal's dependencies
- **Performance overhead** – Ensuring head unit is still able to perform all same functionalities with our application running with minimal or no performance loss

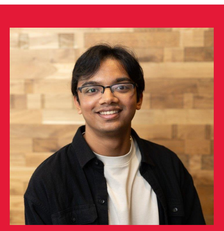
Results

- **8%** more CPU overhead & **15 MB** increased memory usage
- Packet processing time only increased by **.3 ms**
- **98%** packet acceptance rate by VPN as compared to bassline
- Minimal effect on TCP, UDP, & HTTP protocol performance (**+.3 ms each)**

References

- Gözübüyük, B., Tang, B., Shin, K. G., & Pesé, M. D. (2024). Analyzing Privacy Implications of Data Collection in Android Automotive OS. arXiv:2409.15561v1 [cs.CR]. https://doi.org/10.48550/arXiv.2409.15561
- Pese, M., Shin, K., Bruner, J., & Chu, A. (2020). Security Analysis of Android Automotive. SAE Technical Paper 2020-01-1295. https://doi.org/10.4271/2020-01-1295

Tucker Cook
Computer Science



Ayush Verma
Computer Science

Advisor: Giovani Abuaitah
Department: Computer Science