Intrusion Detection System on Automotive CAN Bus

Senior Design Team sdmay24-39

Team: Cole Burkle, Alec Cose, Tiffanie Fix, Trace Haage **Advisor**: Dr. Manimaran Govidarasu

Introduction

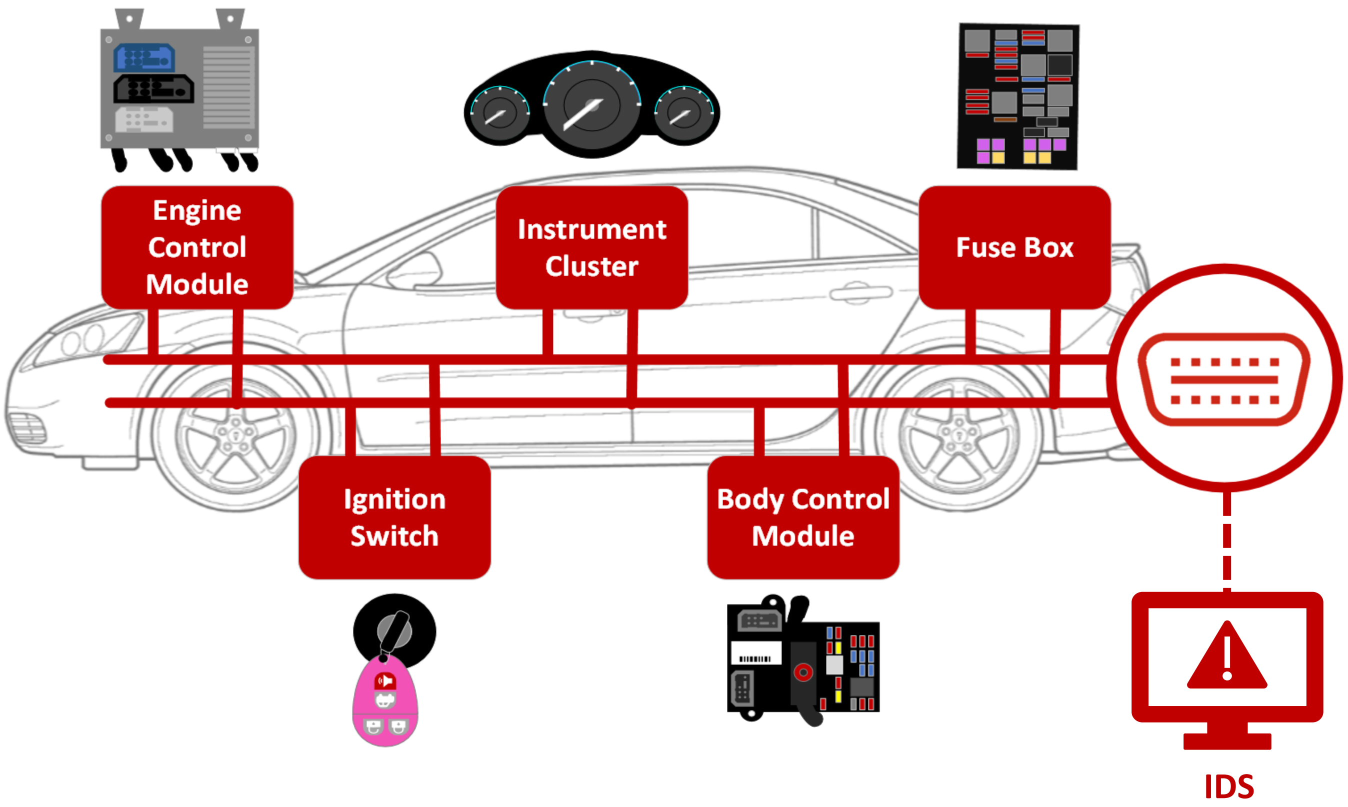
Most modern vehicles are interconnected through CAN Bus, a protocol on vehicle networks that facilitate communication among internal modules essential for vehicle operation such as the engine, dashboard, transmission, and brakes.

Problem

CAN Bus networks often do not consider cyber security and are vulnerable to attacks that involve injecting, altering or intercepting CAN messages to manipulate vehicle operation.

Solution

Implement an IDS on automotive CAN Bus to monitor network traffic for any malicious activity defined in the rules set to alert the user to promptly take action.



Use Case

Users:
- Security researchers
- GRC within automotive Industry

Uses:
- Detect malicious activity on CAN Bus network
- Simulate cyber attacks on the vehicle

Design Requirements

IDS
- Offline and Real-time detection

Pi Test Bed
- Simulate vehicle ECU
- Generate CAN messages using potentiometer

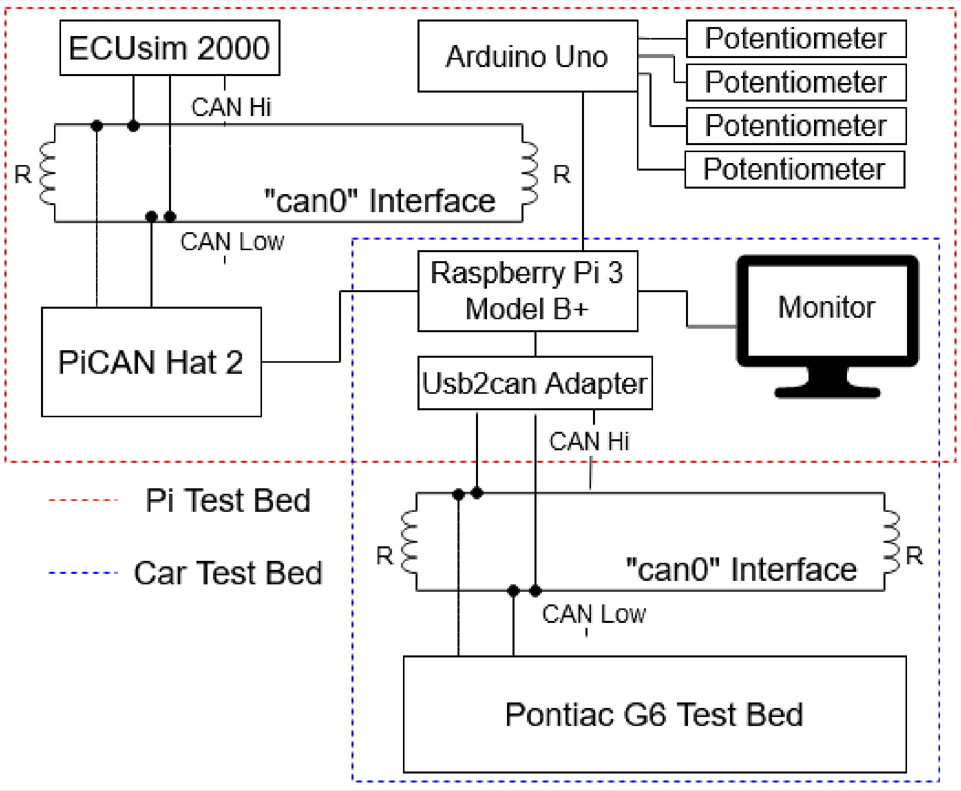
Car Test Bed
- Send/Receive CAN messages
- Fuse box, TCU, BCM, ECU, multi-function switch, dashboard, ignition switch, steering column, main window switches

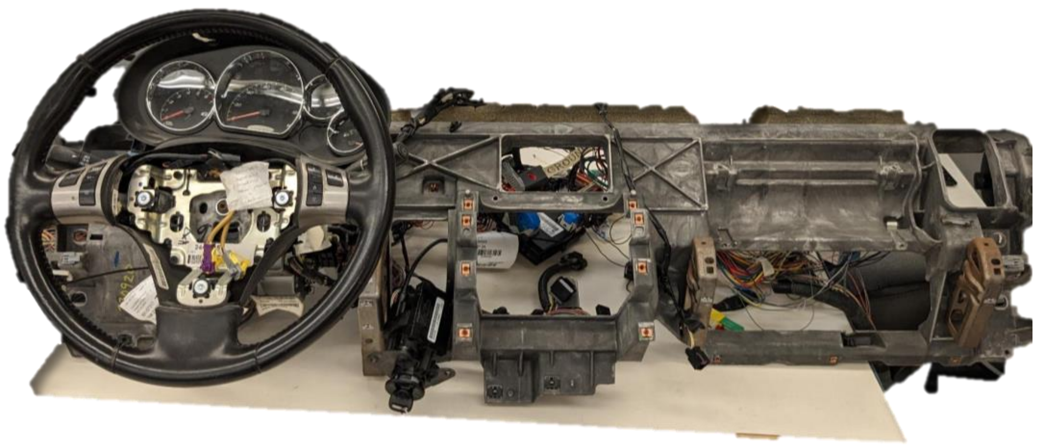
Attack Code
- Simulate cyber attacks

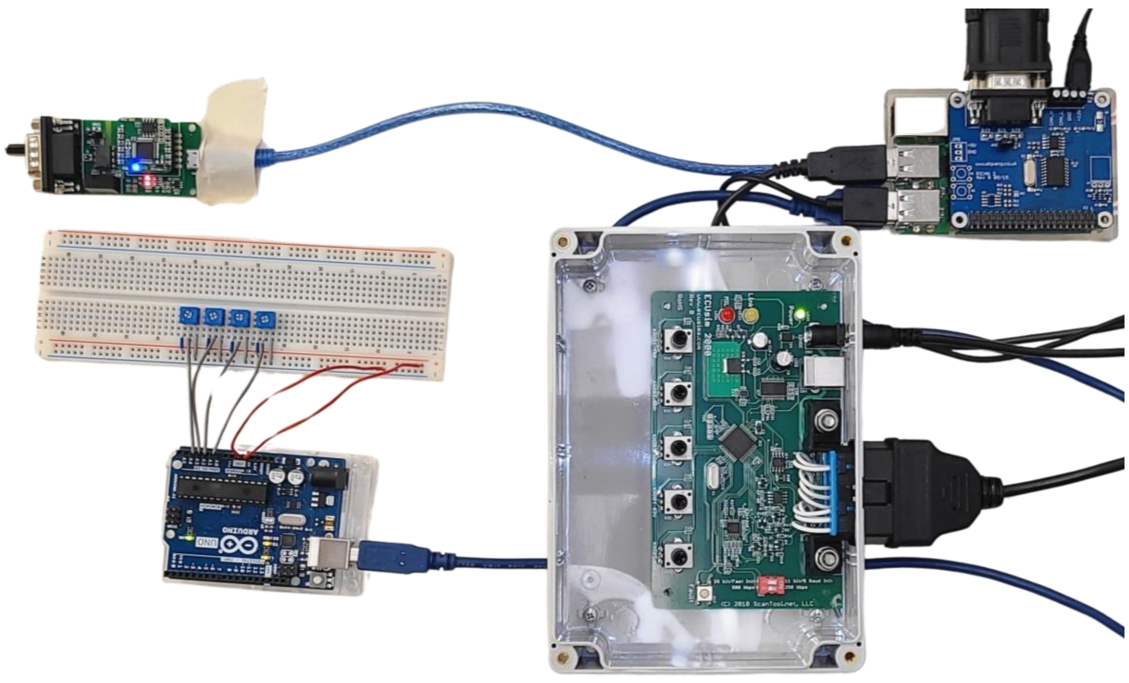
Technical Details

IDS:	Platform: Snort v3 OS: Linux (Raspian)
Pi Test Bed:	Hardware: Raspberry Pi 3 Model B+, PiCAN Hat 2, ECUsim 2000, Arduino uno, potentiometer
Car Test Bed:	Hardware: 2007 Pontiac G6, Innomaker usb2can, 13V power supply
Attack Code:	Language: Python Library: CAN-utils

Design Approach







Testing

Denial of Service (DOS) Attack
- Send large amount of traffic such as low ID messages or remote requests

Injection Attack
- Injecting messages at random (fuzzing) or targeting IDs

Timing Attack
- Executed by sending more messages than expected within a given timeframe

Results

Attack	Pi Bed Detection	Car Bed Detection
DOS	X	X
Injection	X	X
Timing	X	X