IOWA STATE UNIVERSITY

Senior Design Team sdmay24-39

Intrusion Detection System on Automotive CAN Bus

Team: Cole Burkle, Alec Cose, Tiffanie Fix, Trace Haage

Advisor: Dr. Manimaran Govindarasu

Our Team

Trace Haage
Client Liaison and Pi Testbed Lead

Alec Cose
Testbed Design and IDS Rule Development

Tiffanie Fix
Vulnerability Research and Development Lead

Cole Burkle
Lead Vulnerability Tester and G6 Testbed Design

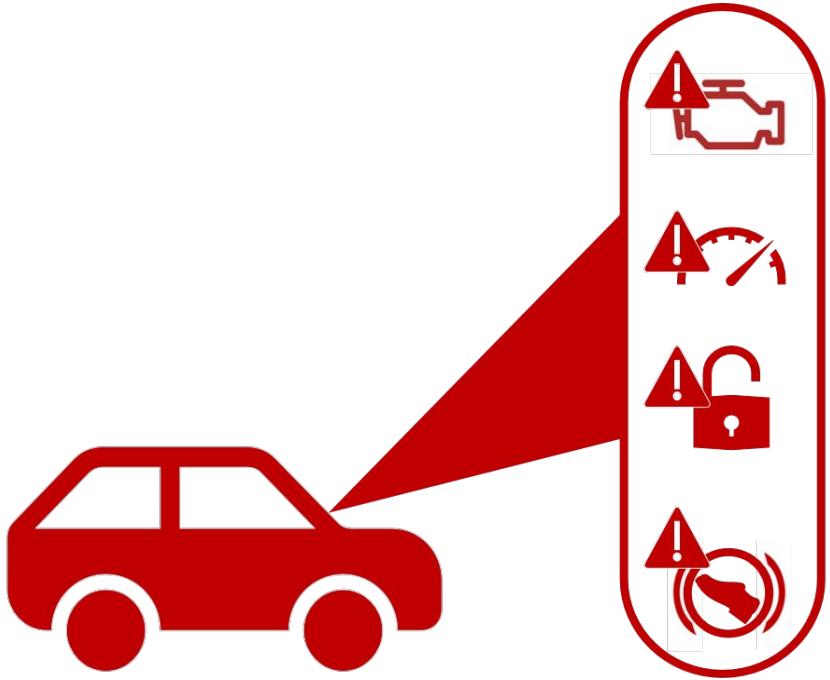
Outline

Introduction

CAN Bus and its Significance

Problem

- Most modern vehicles are interconnected through CAN Bus
- CAN Bus networks often do not consider cyber security
- Vulnerable to attacks that manipulate vehicle operation and may result in unauthorized access.



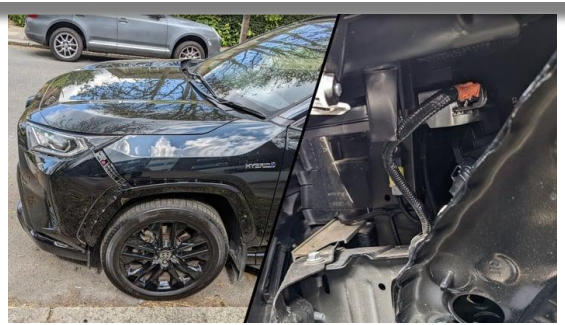
Real Life Cases

- 2016: Jeep Cherokee controlled wirelessly through entertainment system
- 2016: Tesla firmware vulnerability led to remote control
- 2024: Rav4 exploited through headlight connector

Hackers Remotely Kill a Jeep on the Highway—With Me in It

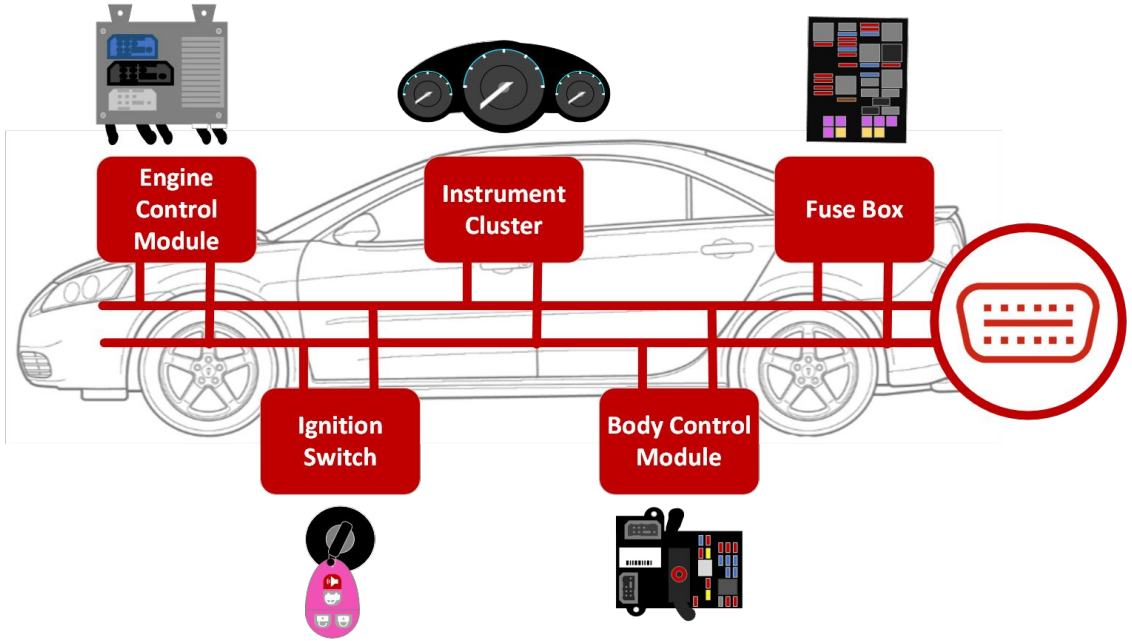
Hackers crack Tesla CAN Bus, DoT issues policy for securing connected car

Thieves Steal Toyota RAV4 by Hacking Into Its Headlights



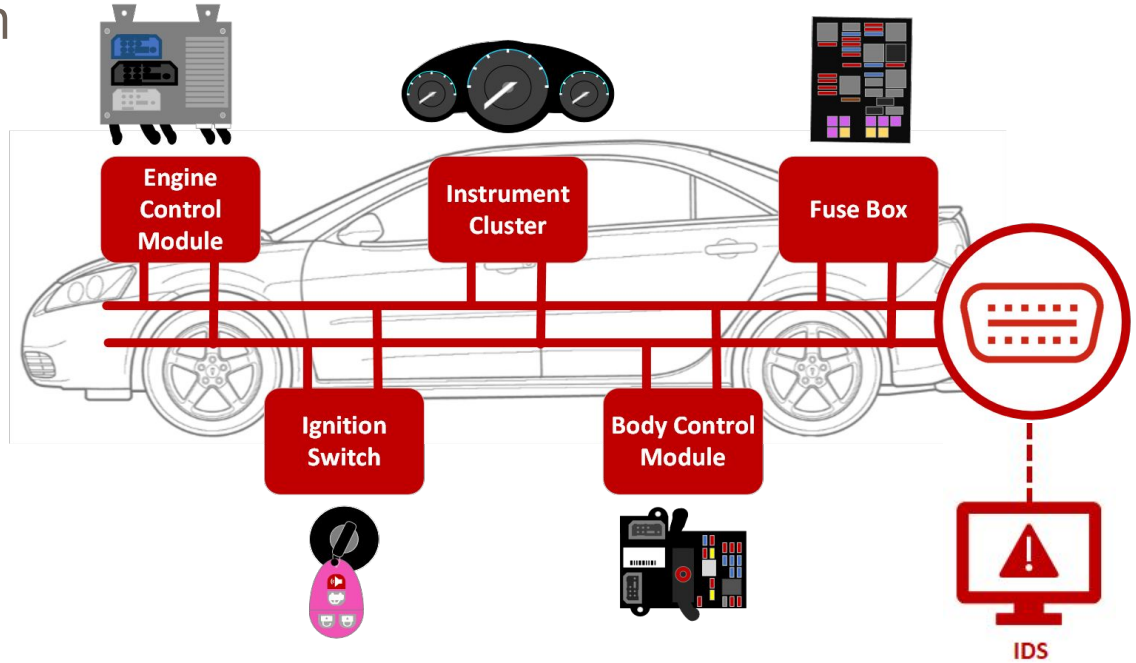
CAN Bus Background

- CAN Bus is a protocol on the vehicle network that enables internal modules within the vehicle to communicate
  - Such as the engine, transmission, and brakes
- Essential for vehicle operation



Intrusion Detection System Solution

- An Intrusion Detection System (IDS) is a software that monitors the network and reports any anomalies.
- Rules are set and when triggered, promptly alerts the user.

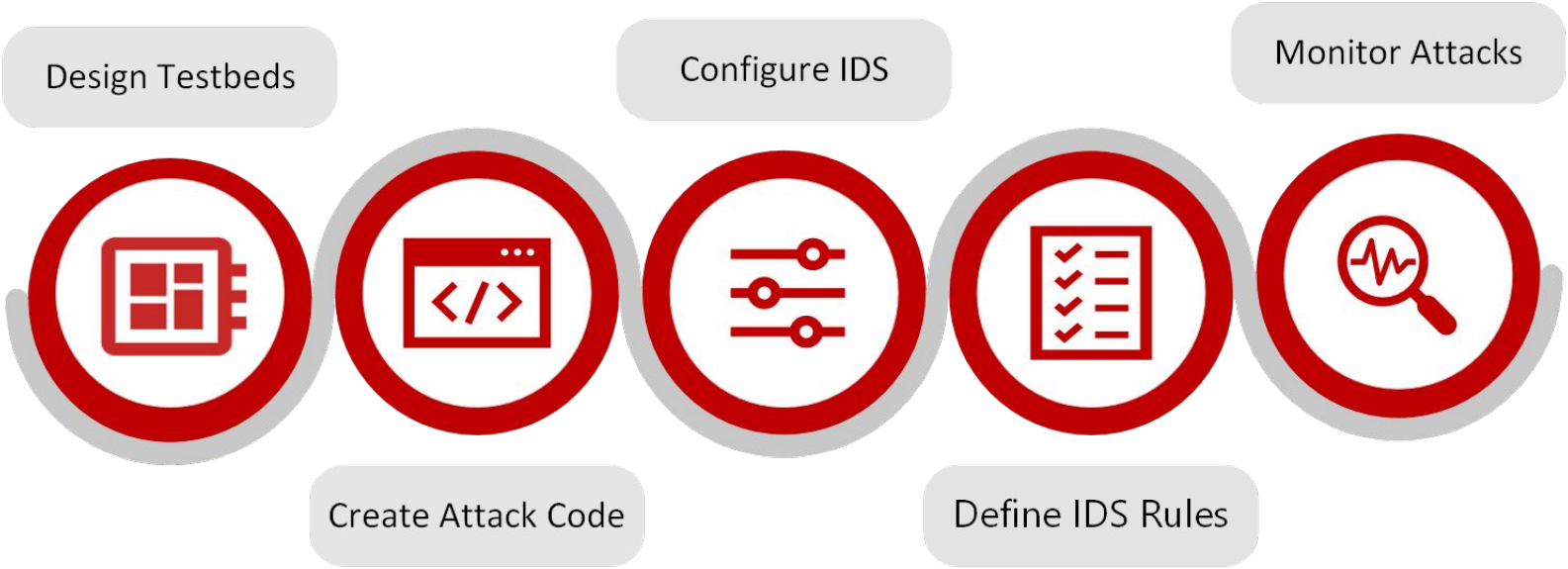


Design Approach

Creating and Proposing an IDS and Testing Platform

Design Overview

Implementing an Intrusion Detection System (IDS) on an automotive CAN Bus network:



Design Requirements

Testbed Design

Pi Testbed

- Create multiple nodes
- Emulate vehicle ECU
- Adjust data values using potentiometers

Car Testbed:

- Utilize vehicle CAN Network
- Send/Receive CAN Messages

Attack Code

- Compromise integrity, availability, and confidentiality of data
- Manipulate or deceive ECUs or modules into unauthorized actions
- Congest or disrupt network traffic
- Timing control

IDS

- Enable uploading of offline logs
- Configured to analyze traffic in real time
- Define rules to effectively detect attack code executed on testbeds

Technical Details

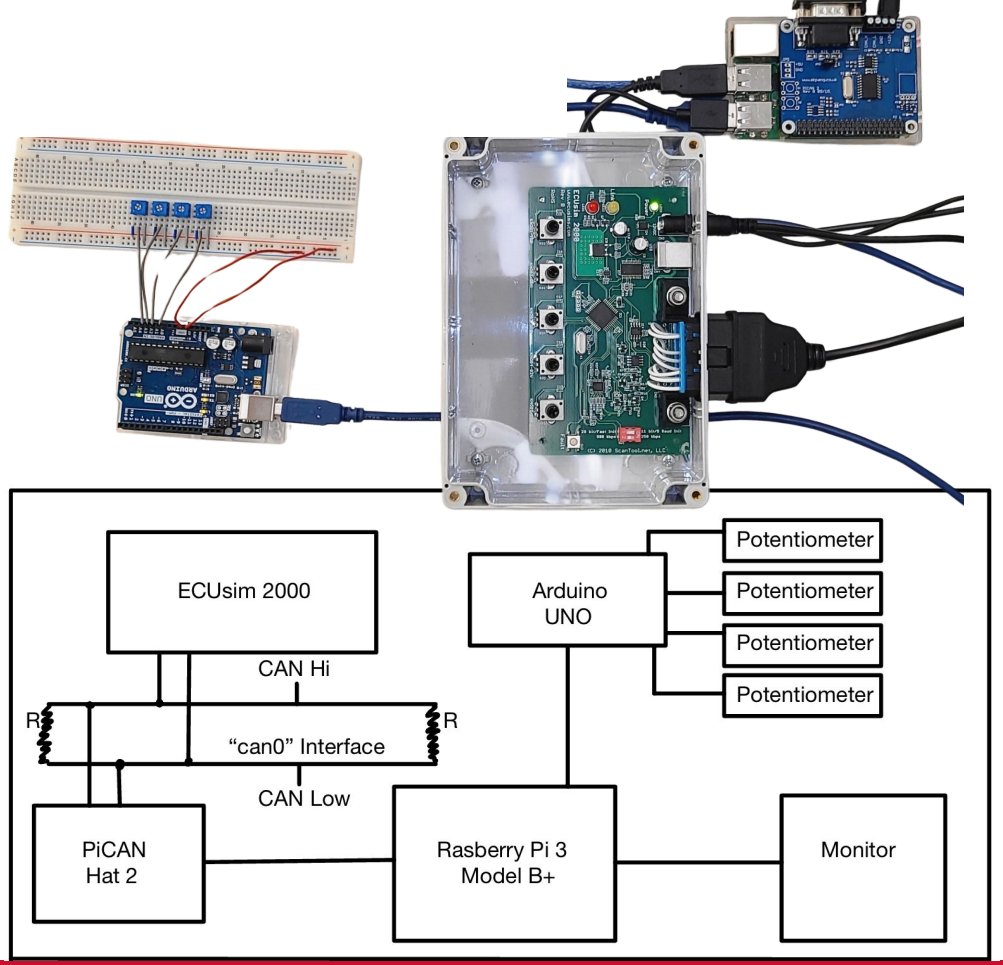
	Car Testbed	Pi Testbed	IDS	Attack Code
Hardware	• 2007 Pontiac G6 • Innomaker usb2can Adapter • 13V power supply	• Raspberry Pi 3 Model B+ • PiCAN Hat 2 • ECUsim 2000 • Arduino UNO • potentiometers	• Raspberry Pi 3 Model B+ • Monitor	• Raspberry Pi 3 Model B+
Software	• CAN-util • usb2can program	• Raspbian OS • CAN-util • Python code	• Snort v3	• Python language • CAN-util

Testbed Development

Constructing an effective testing platform

Pi Testbed

- PiCAN Hat 2 and ECUsim 2000 create CAN channel
- Arduino sends to Pi, Pi puts on the CAN channel
- 4 sensors, ID 0-3



Pi Testbed Evaluation

Benefits:

- Emulates car network structure
- Values can be changed as the system is running
- Completely mutable

Challenges:

- ECUsim 2000
- Size of the network
- Transmission speeds
- Not real CAN frames

Car Testbed

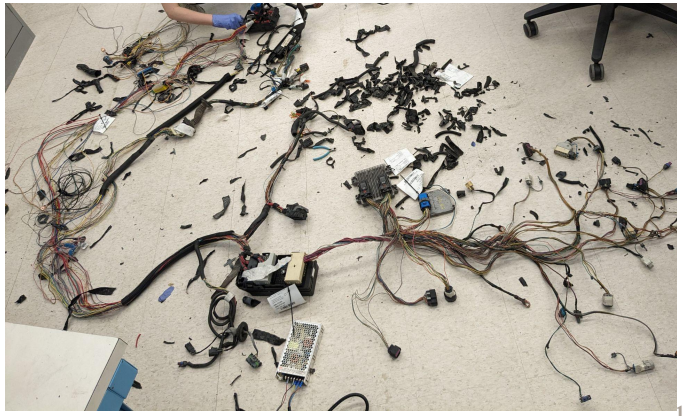
- 2007 Pontiac G6
- Bought from a local junkyard
- Two CAN networks
- USB2CAN to read/capture data

Car Testbed Benefits

- Realistic testing environment
- Immediate physical feedback
- Access to genuine data
- Message Variety
- Two separate networks

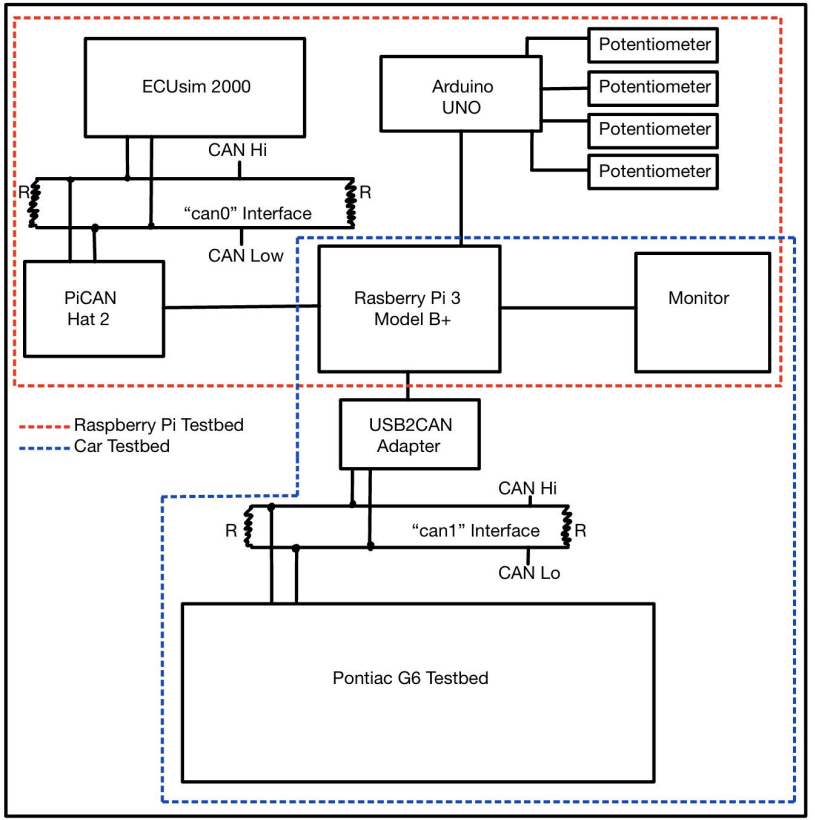
Car Testbed Challenges

- Nothing connected
- Schematics are proprietary
- High speed network bus off
- No straightforward solution to power



Final Testbed Design

- Car testbed
- Pi testbed
- IDS has access to both



IDS Implementation

Curating rules to detect malicious anomalies

Attacks and Detection

- Open Source IDS
- Allows for rule development
- Only functions on TCP/IP

Ruleset Strategy

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

Pi Testbed Rules

alert tcp any any -> any 12345 (msg: “Injection Attack: ID out of range >4”;

byte_test:8,>,4,8,string,dec; sid:10000006;)

- Denial of Service Attack - Low ID and Remote Requests
- Injection Attack - Mismatch ID and message
- Timing Attack - More messages than typical behavior

Car Testbed Rules

alert any any -> any 12345 (msg: “Mismatching ID and message - 670:47”;

content:”670”; content:!”47”, distance 7, within 4; sid 30000006;)

- Rules for both high and low transmission
- Denial of Service Attack - Low ID and Remote Request
- Injection Attack - ID Ranges and Matching ID and Messages
- Timing Attack - Limited by speed of sending packets over TCP

IDS Possibilities

100500	VEHICLE_SPEED_INFORMATION	0x28

Second and third bytes
Formula x/10 = Vehicle speed (likely in KPH but unconfirmed)

alert tcp any any -> any any (msg: “Impossible Speed Detected”; byte_test:3,=,28,6,string,dec; byte_test:2,>,1450,9,string,dec; sid: 10000001;)

Recap

- Testbed Design
- Testbed Testing
- Attack Simulation
- IDS Rule Development
- IDS Rule Testing

Questions?