HoliSec

Holistic Approach to Improve Data Security

Intrusion Detection for In-vehicle Networks: Reflection on Practical Challenges and the Road Ahead

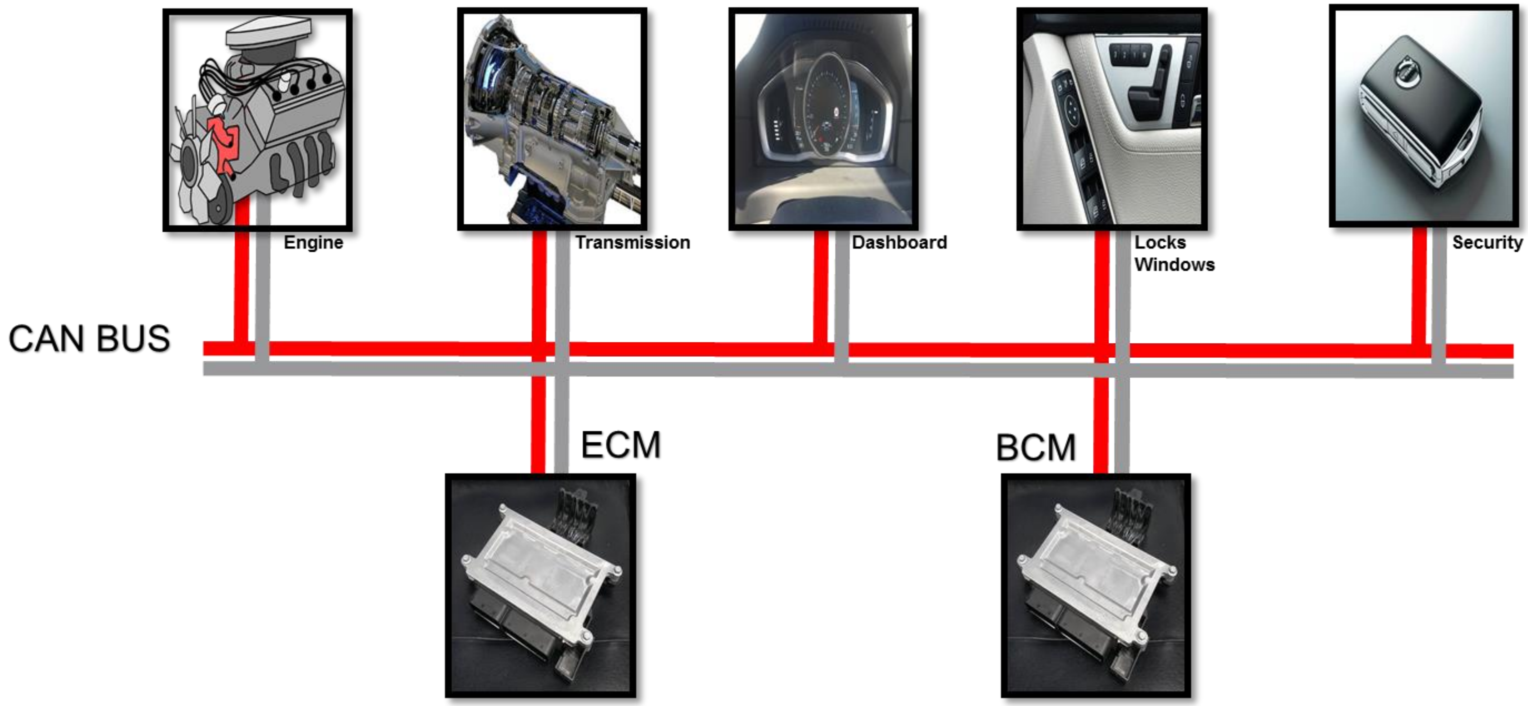
David Thiringer, Nasser Nowdehi, Sebastian Kvarnström

March 26, 2019. Time 09:30 – 10:15

Recent Automotive Hacks

- 2010
  - Vehicles Disabled Remotely Via Web Application (Austin, Texas)
  - Center for Automotive Embedded Systems Security (CAESS)
    - disable the brakes, stop the engine, falsify information on the vehicle's speedometer, and more
- 2011
  - CAESS: radio's MP3 parser, the vehicle's Bluetooth system, and the cellular connection used for the vehicle's telematics
- 2013
  - Miller & Valasek: Tightened seat belt, speedometer attacks, horn blast, acceleration, headlights, disable power steering, slam on brakes at any speed
- 2015
  - Miller & Valasek: Remote hack of Jeep Cherokee
- 2017-2018
  - Tencent's Keen Security Lab: Tesla and BMW i3 hack

CAN bus



Engine
Transmission
Dashboard
Locks
Windows
Security
CAN BUS
ECM
BCM

IDS types

- Signature Based
  - Recognizing bad patterns, such as malware
  - Precise detection for known attacks
  - Unable to detect unknown attacks: Need for constant signature updates
- Anomaly based
  - Detecting deviations from a model of "good" traffic, which often relies on machine learning
  - Able to detect unknown attacks, doesn't need signatures
  - Frequent false positives

State-of-the-art approaches

- Specification based
  - Message timing
- Low-level properties of ECUs
  - Clock behavior
  - Voltage behavior

Our Work

- Step 1:
  - What are the challenges?
  - Answers:
    - Static checks + Anomaly detection
    - Low memory and processor footprint
    - Start of deployment from most critical ECUs
- Step 2:
  - The gaps in the state-of-the-art IVN IDS? Can we address them?
  - Answers:
    - Stealthy attacks
    - Specification agnostic

A Lightweight Intrusion Detection System for In-Vehicle Communication on CAN

David Thiringer

Sebastian Kvarnsträm

Introduction

- Smarter cars with more functionalities
  - Self-driving cars
  - Bluetooth, mobile networks
  - More possible attack vectors
- Mechanical parts are replaced by electronic signals
  - Hackers can cause greater harm than before

2015: Remote Access to Jeep

Chris Valasek and Charlie Miller were able to

- Disable the brakes
- Force the car to brake
- Steer the vehicle
- Max the volume of the radio

Remotely, from anywhere in the United States!

2018: Who Killed My Parked Car?

Work by K.-T. Cho, Y. Kim & K.G. Shin, where they:

- Drain the car's battery
- Lock the driver out of the car

All while the ignition was turned off!

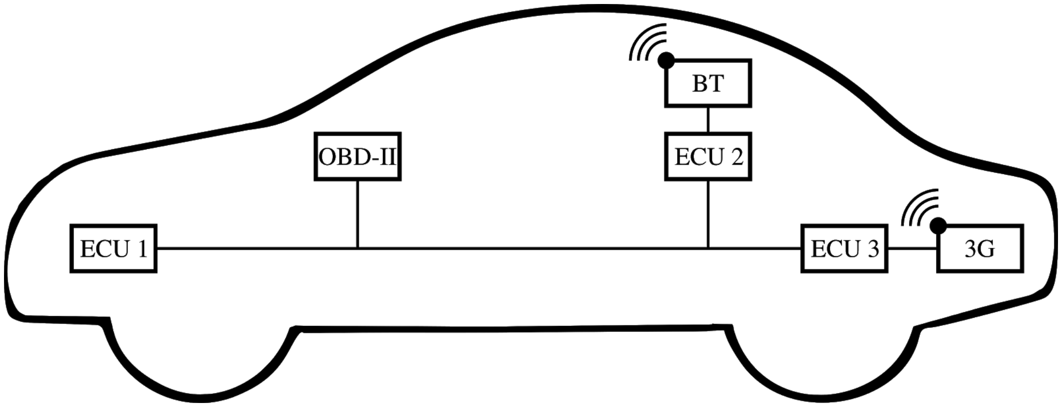
Question

“Is it possible to detect attacks against in-vehicle networks, within a realistic environment in real time?”

- **Background**
- Evaluation and Test Benches
- Results: Arduino
- Results: Box Car (Offline)
- Results: Box Car (Online)
- Conclusion

In-vehicle Networks

- ECU (Electronic Control Unit)
- CAN (Controller Area Network)



Security concerns with CAN

- **C**onfidentiality
- **I**ntegrity
- **A**vailability

Question

"Is it possible to detect attacks against in-vehicle networks, within a realistic environment in real time?"

- The in-vehicle network investigated is CAN

Attacks on CAN

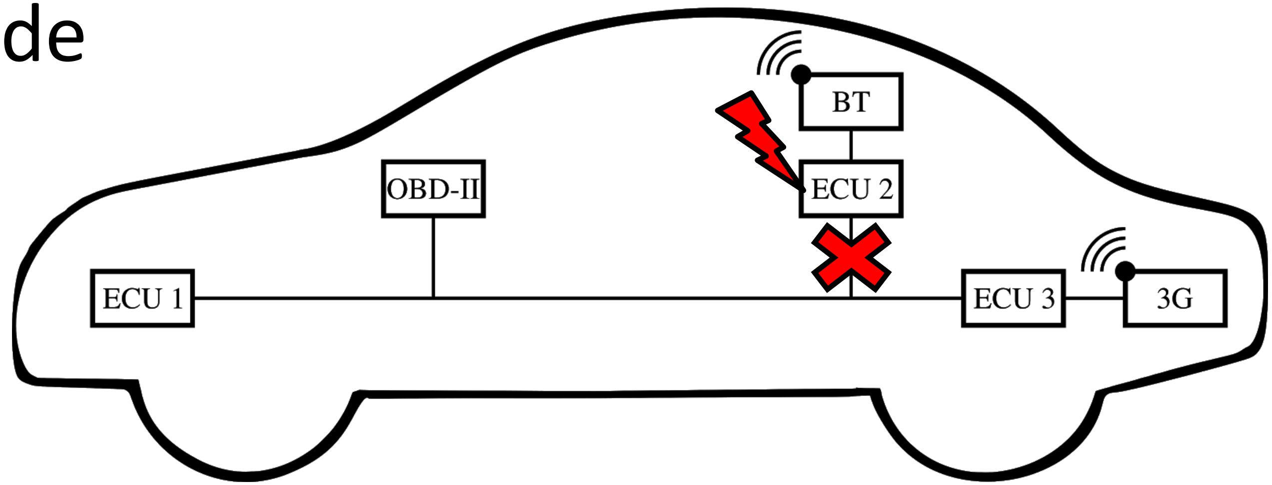
Defined in related literature:

- Suspension
- Fabrication
- Masquerade
- Conquest

Attacks on CAN

Defined in related literature:

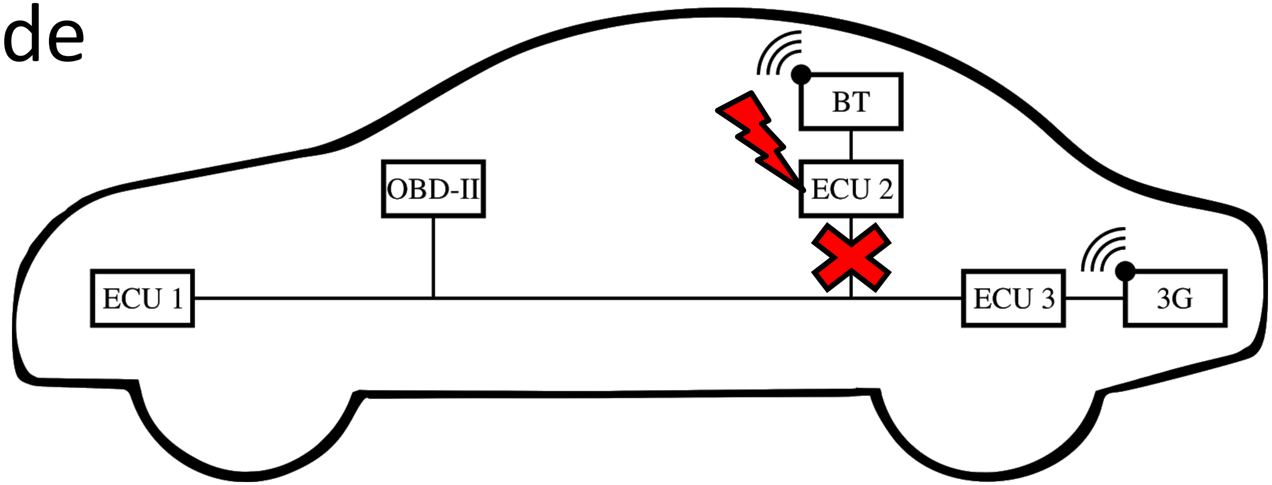
- **Suspension**
- Fabrication
- Masquerade
- Conquest



Attacks on CAN

Defined in related literature:

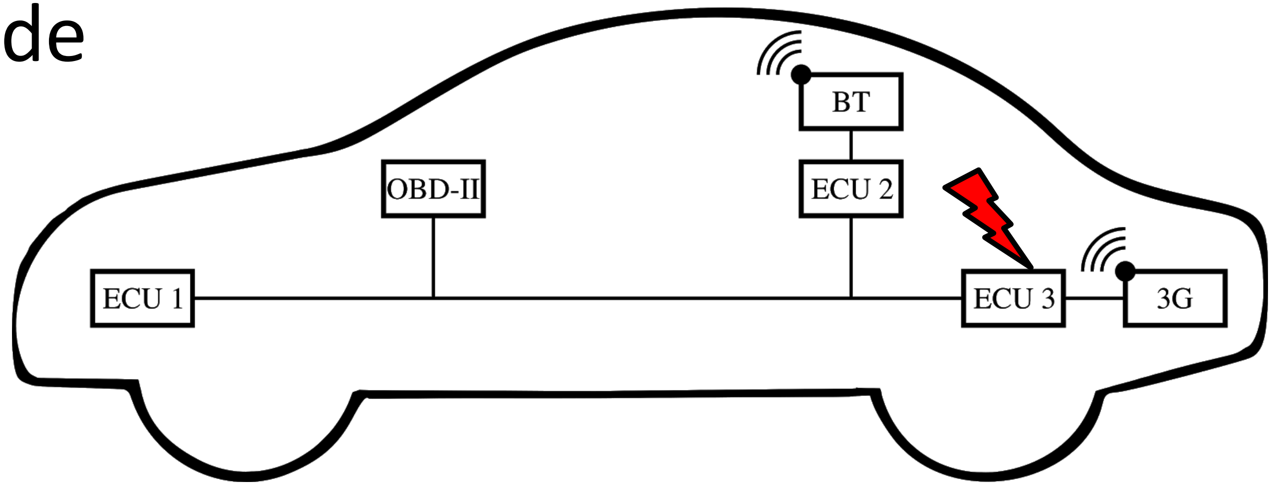
- **Suspension**
- Fabrication
- Masquerade
- Conquest



Attacks on CAN

Defined in related literature:

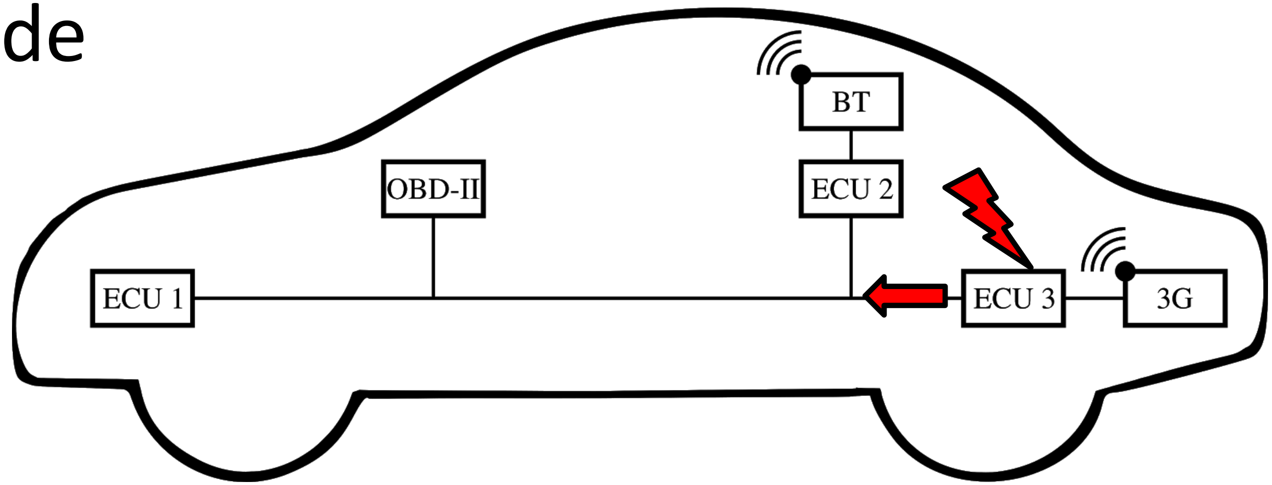
- Suspension
- **Fabrication**
- Masquerade
- Conquest



Attacks on CAN

Defined in related literature:

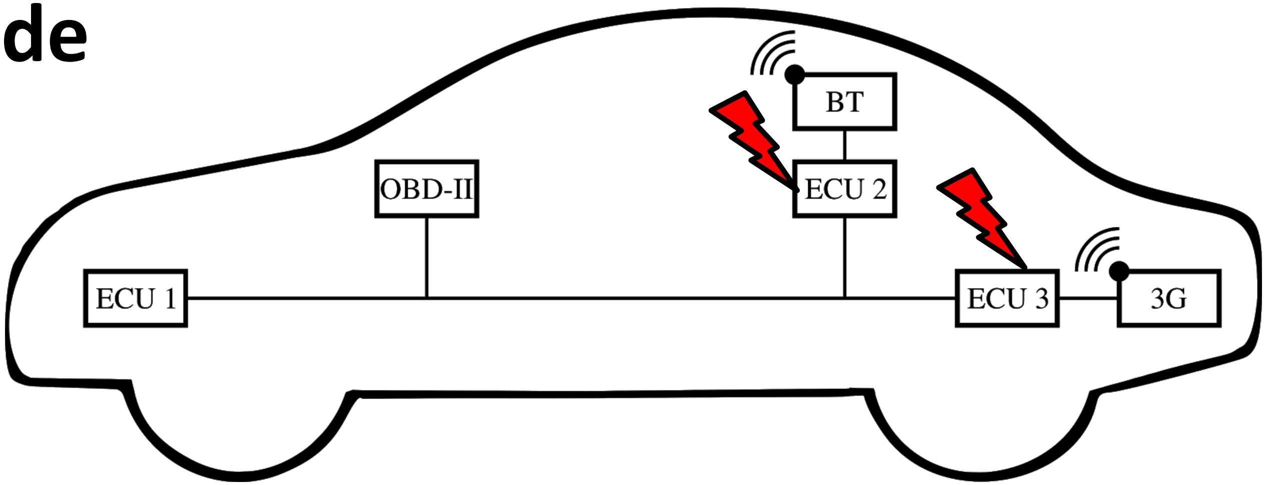
- Suspension
- **Fabrication**
- Masquerade
- Conquest



Attacks on CAN

Defined in related literature:

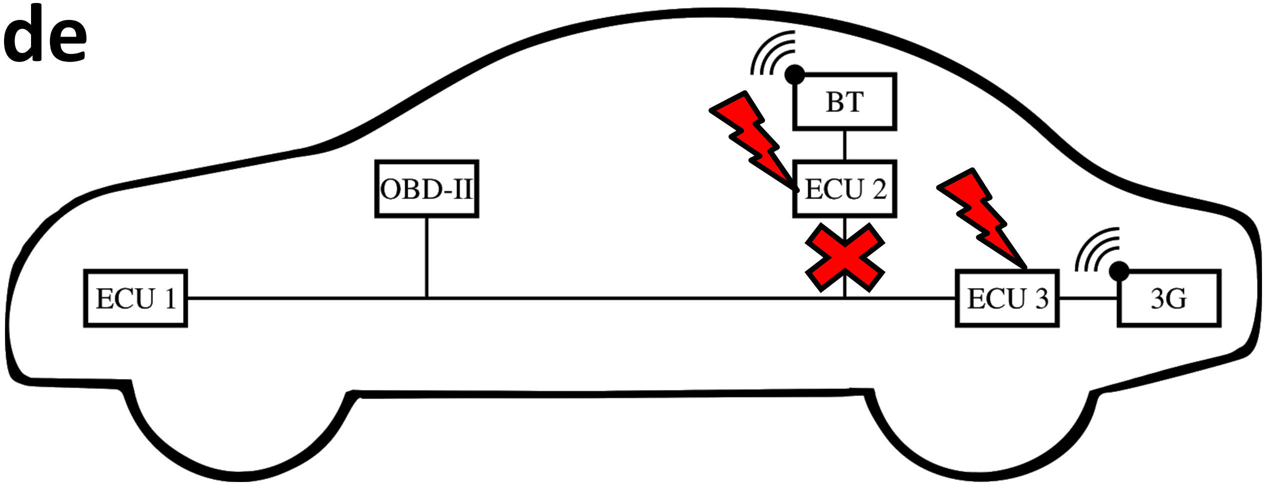
- Suspension
- Fabrication
- **Masquerade**
- Conquest



Attacks on CAN

Defined in related literature:

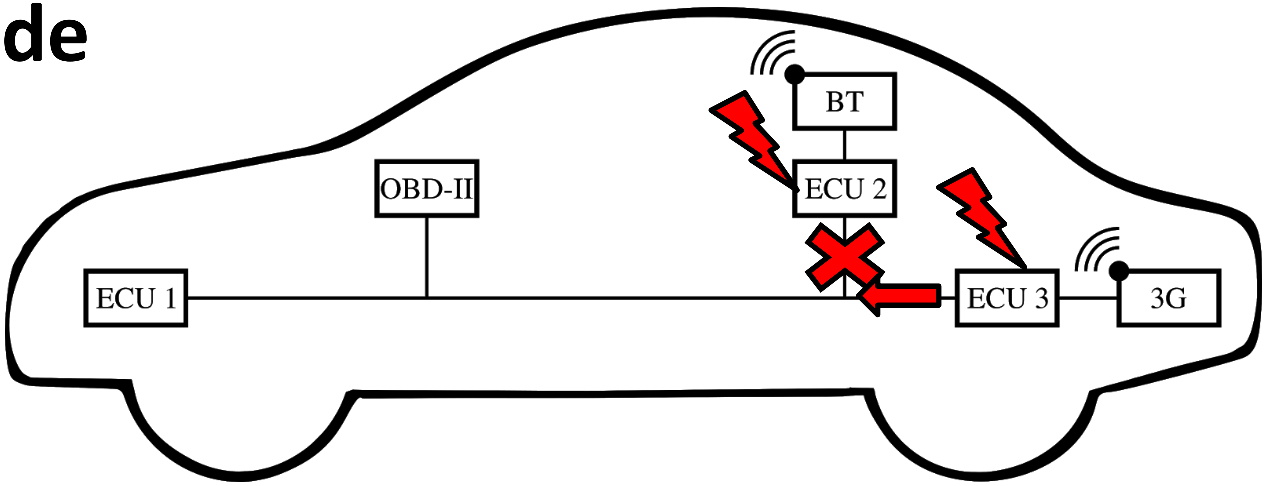
- Suspension
- Fabrication
- **Masquerade**
- Conquest



Attacks on CAN

Defined in related literature:

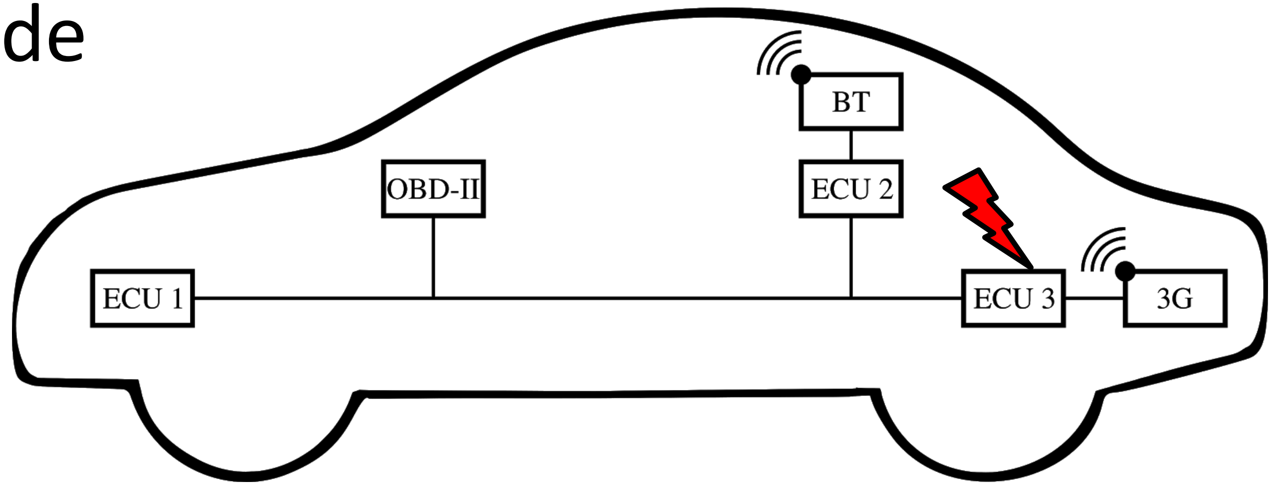
- Suspension
- Fabrication
- **Masquerade**
- Conquest



Attacks on CAN

Defined in related literature:

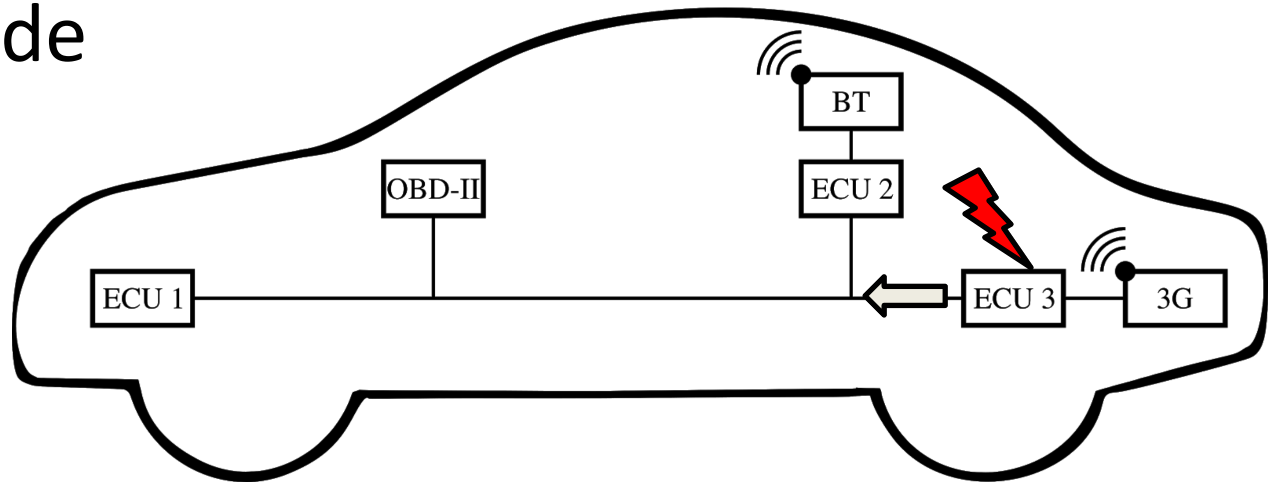
- Suspension
- Fabrication
- Masquerade
- **Conquest**



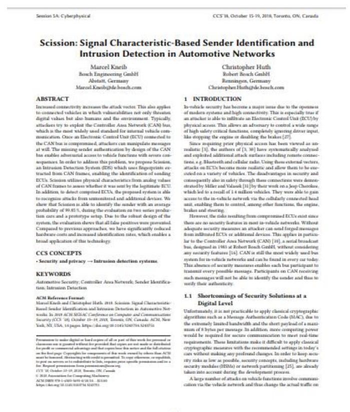
Attacks on CAN

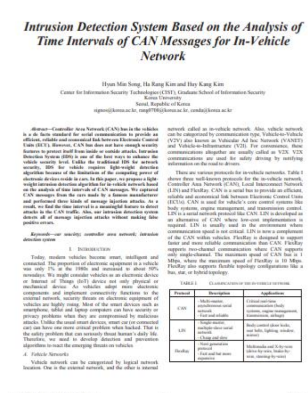
Defined in related literature:

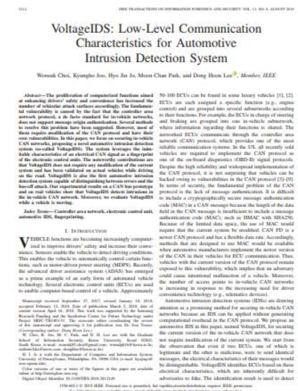
- Suspension
- Fabrication
- Masquerade
- **Conquest**



Intrusion Detection for CAN







Question

“Is it possible to detect attacks against in-vehicle networks, within a realistic environment in real time?”

- The in-vehicle network investigated is CAN
- Realistic environment corresponds to low resource ECU

Choice of Algorithm

- Low-end hardware → Algorithm needs to be **lightweight**
- Low number of false positives → Algorithm needs to be **accurate**
- Preferably detects the **four attack types**

One algorithm fits these criteria: **CASAD**

CASAD: CAN-Aware Stealthy Attack Detection

- Anomaly-based
- Data-driven
- Claims to be lightweight
- Has been shown to have a high detection accuracy

What does CASAD do?

Training phase

- **Given** a value L and a time series of payload data (byte-by-byte works best):
- **Outputs:**
  - A value **r**, its *statistical dimension*.
  - A projection matrix: U[L][r], describing the *signal subspace.*
  - A centroid, the mean of all training vectors projected onto the signal subspace
- Based on this, a threshold for attacks can also be determined.

What does CASAD do?

Detection phase

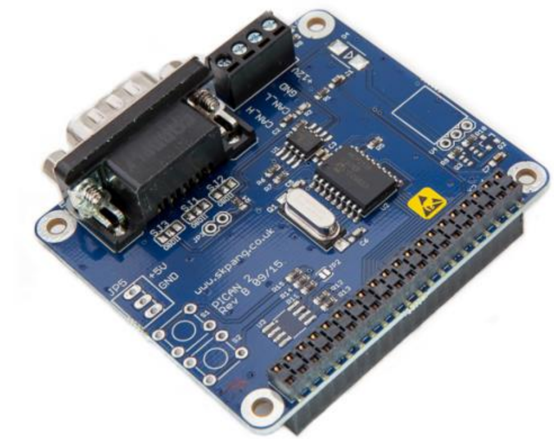
- Fill a “lag vector” with the **L** most recent bytes
- Project the lag vector onto the signal subspace
- Calculate the Euclidean distance to the centroid from the projected lag vector

If the distance is greater than the threshold, an attack is detected!

- Background
- **Evaluation and Test Benches**
- Results: Arduino
- Results: Box Car (Offline)
- Results: Box Car (Online)
- Conclusion

What we have done

- Implemented CASAD on a Raspberry Pi 3 Model B
- CASAD was written in C
- Interface to CAN bus via PiCAN v2



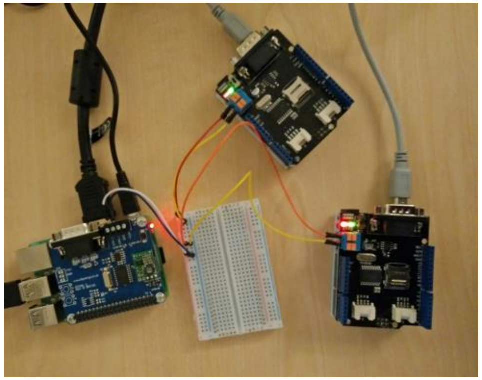
Question

“Is it possible to detect attacks against in-vehicle networks, within a realistic environment in real time?”

- The in-vehicle network investigated is CAN
- Realistic environment corresponds to low resource ECU
- Real time means detecting attacks while they are happening

Test benches and experiment set-ups

- Arduino Network
- Box Car Logged Data (*Offline*)
- Box Car Live Data (*Online*)

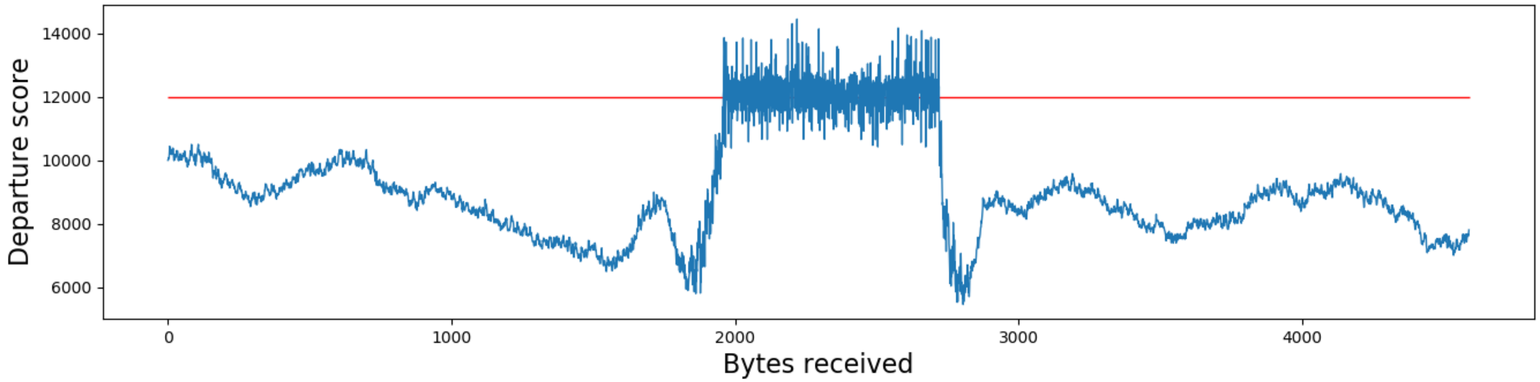


Design of attacks

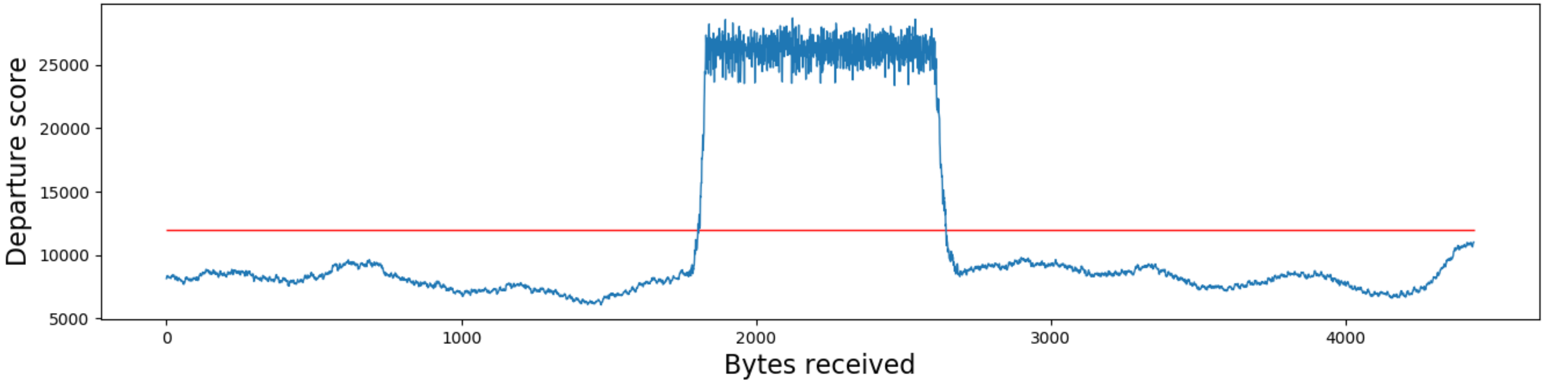
- All attack types were implemented on both test benches
- Experiments: 20s normal → 20s attack → 20s normal
- Not realistic attacks on Arduino
- Realistic attacks on the Box Car

- Background
- Evaluation and Test Benches
- **Results: Arduino**
- Results: Box Car (Offline)
- Results: Box Car (Online)
- Conclusion

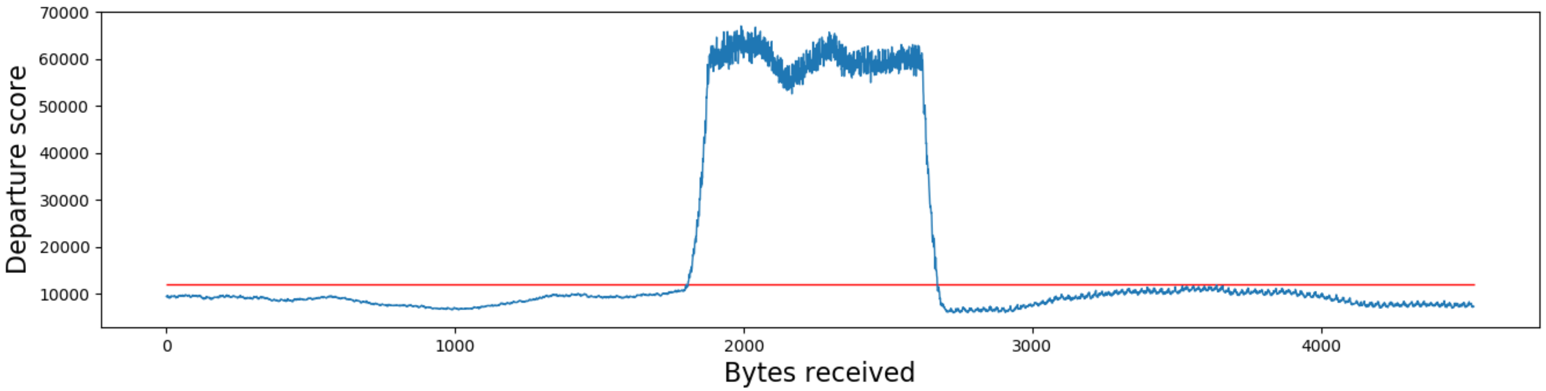
Arduino - Suspension attack



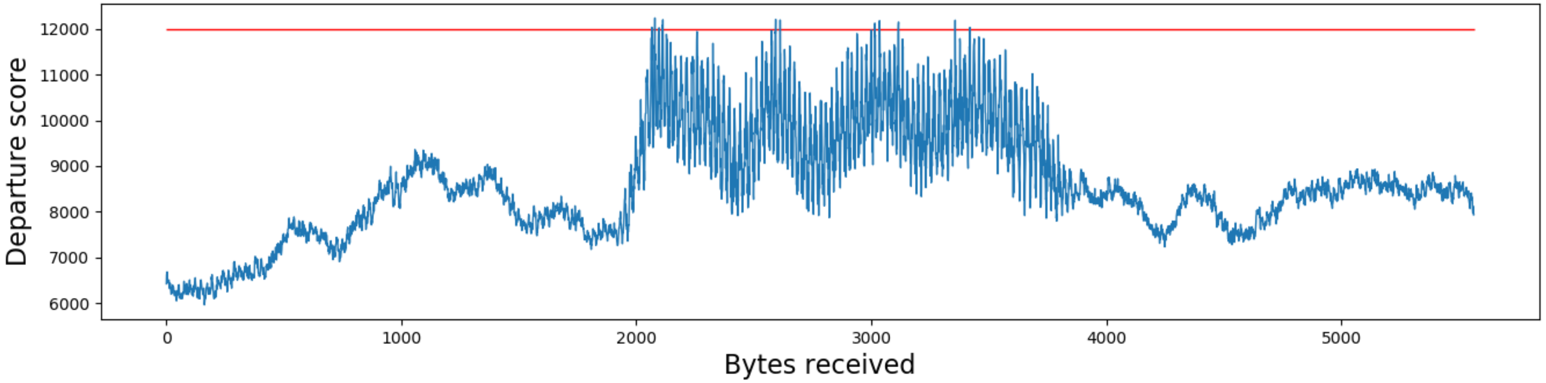
Arduino - Fabrication attack



Arduino – Masquerade attack



Arduino – Conquest attack



- Background
- Evaluation and Test Benches
- Results: Arduino
- **Results: Box Car (Offline)**
- Results: Box Car (Online)
- Conclusion

Preparation to meet real-time requirements

CAN bus we investigated:

- Busload ~67%
- 24 000 bytes per second

  Using parameters: L=10 000 & r=24
- $24 * 10^3 * 10 * 10^3 * 24 =$
  $5.76*10^9$ multiplications per second

Optimization of CASAD (1)

Overall improvements

- Threading
- Circular buffer instead of array

Improved accuracy

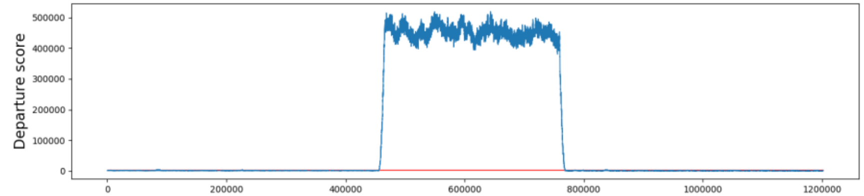
- Implemented rolling average of departure scores

Optimization of CASAD (2)

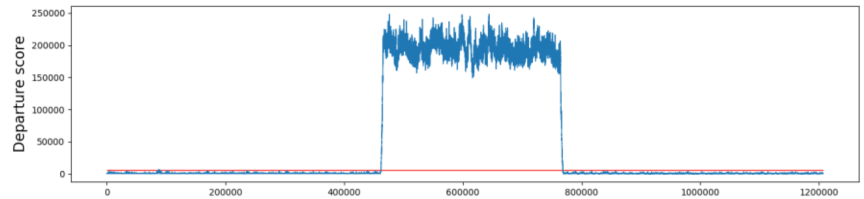
Multiplication reduction

- Lagvector, L=10 000 and L=5 000
- Input downsampling, by 15 and 63
- Output downsampling, by 1 000

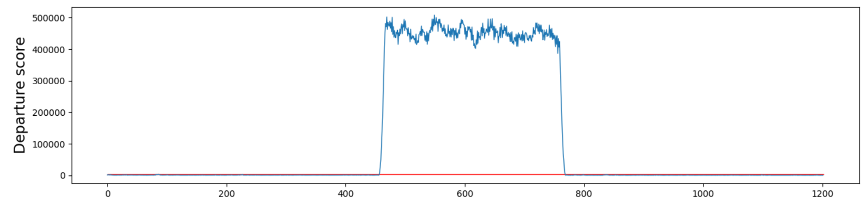
L = 10 000



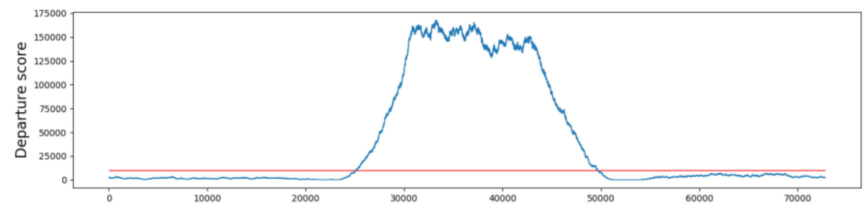
L = 5 000



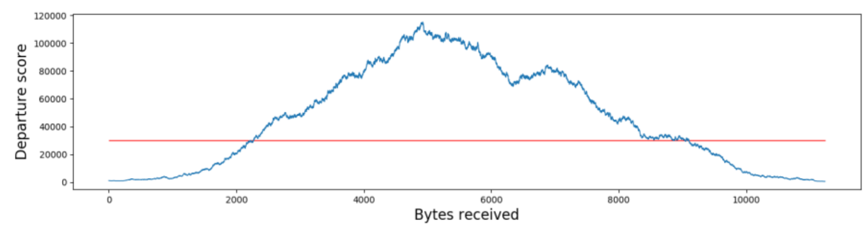
L = 10 000, output ds 1 000



L = 8 000, input ds 15



L = 8 000, input ds 63



Boxcar (offline) - Suspension attack

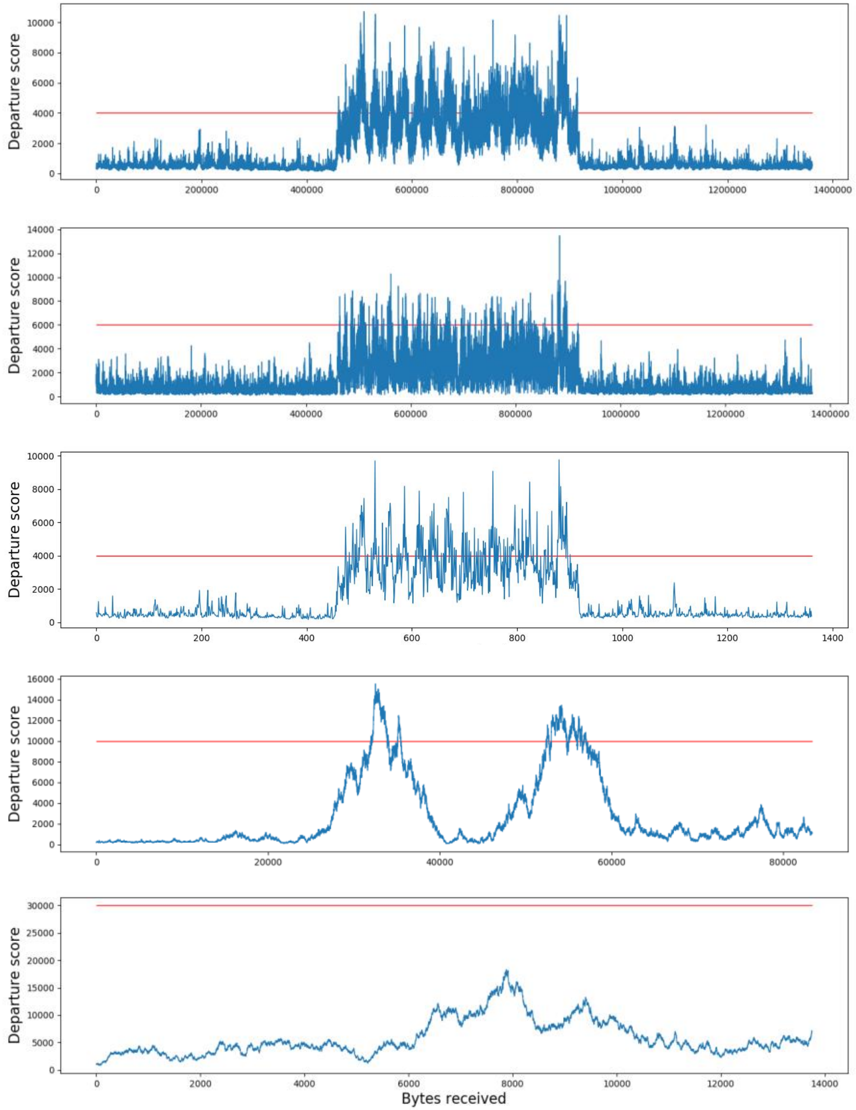
L = 10 000

L = 5 000

L = 10 000, output ds 1 000

L = 8 000, input ds 15

L = 8 000, input ds 63



Boxcar (offline) - Fabrication attack

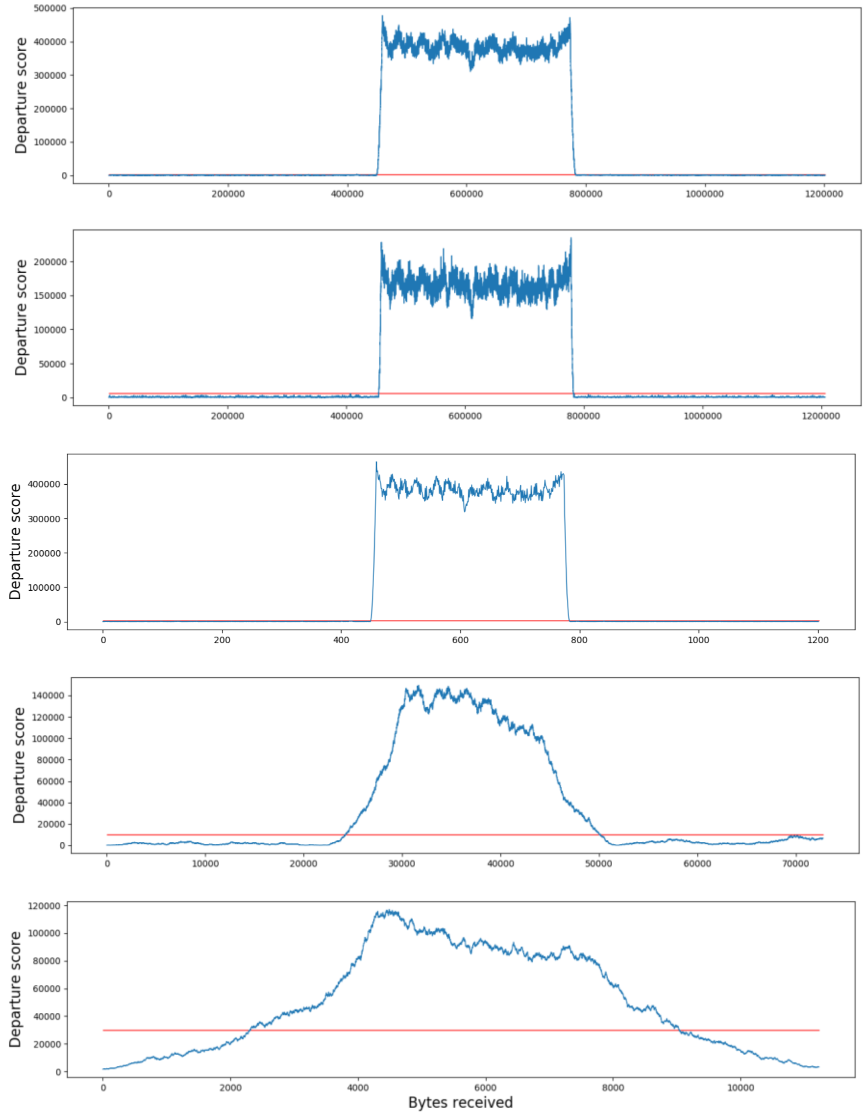
L = 10 000

L = 5 000

L = 10 000, output ds 1 000

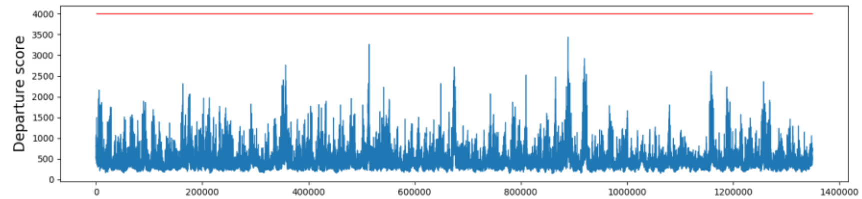
L = 8 000, input ds 15

L = 8 000, input ds 63

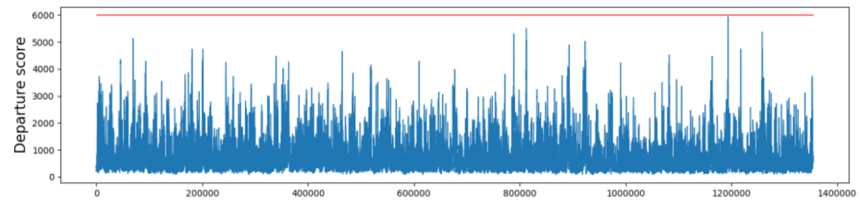


Boxcar (offline) - Masquerade attack

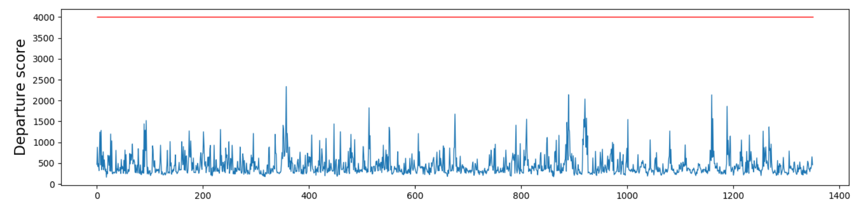
L = 10 000



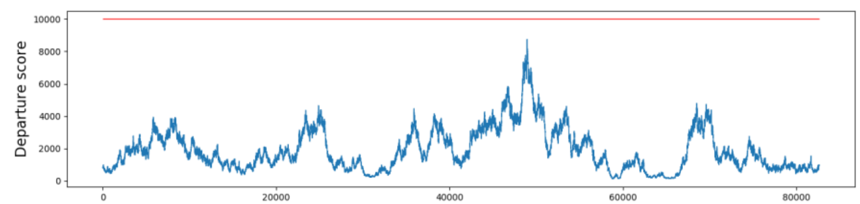
L = 5 000



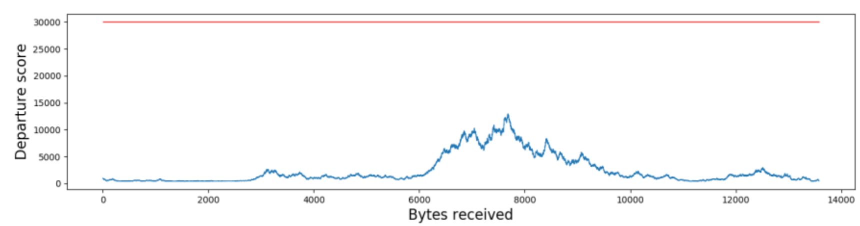
L = 10 000, output ds 1 000



L = 8 000, input ds 15



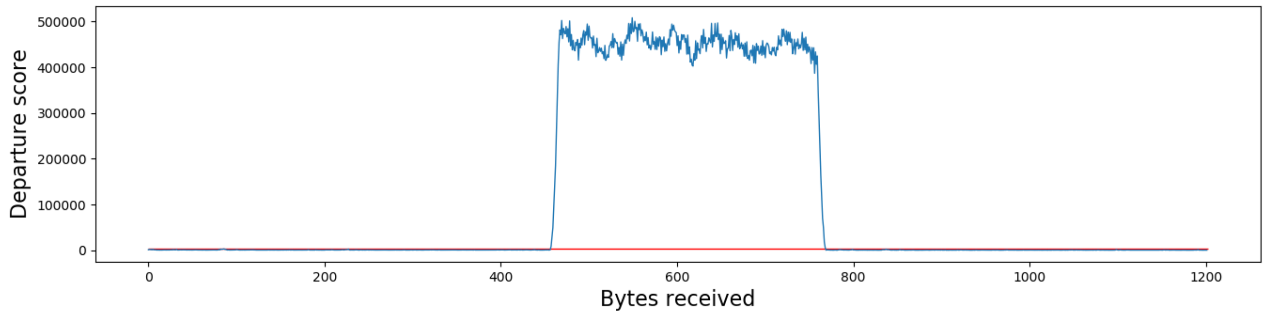
L = 8 000, input ds 63

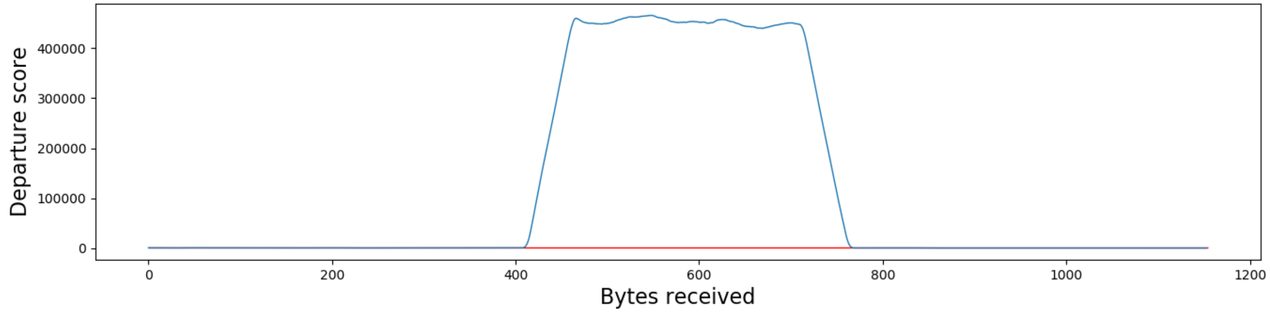


Boxcar (offline) - Conquest attack

Rolling average, Boxcar (Offline)

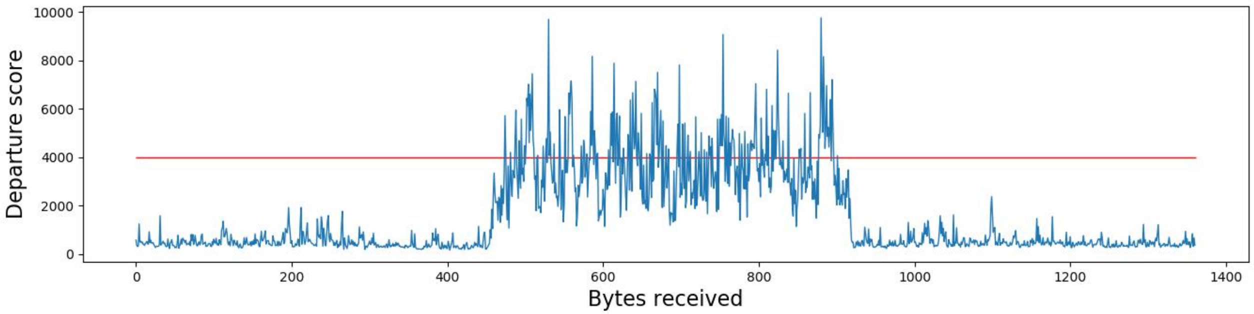
- Suspension attack

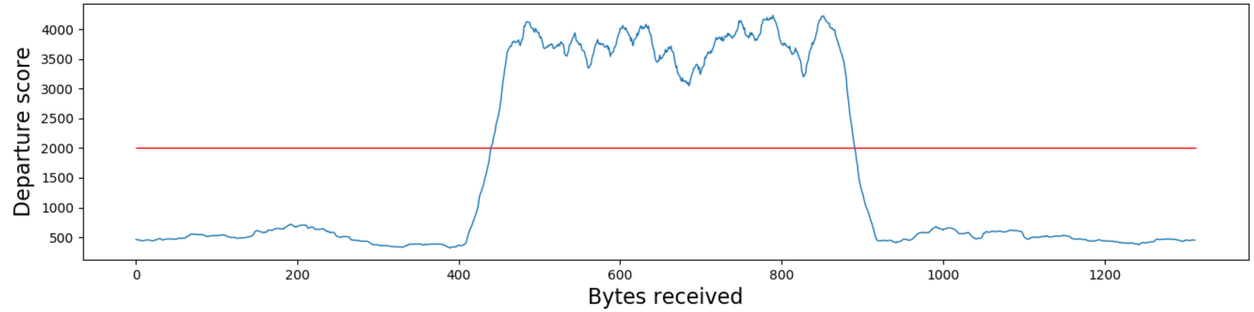




Rolling average, Boxcar (Offline)

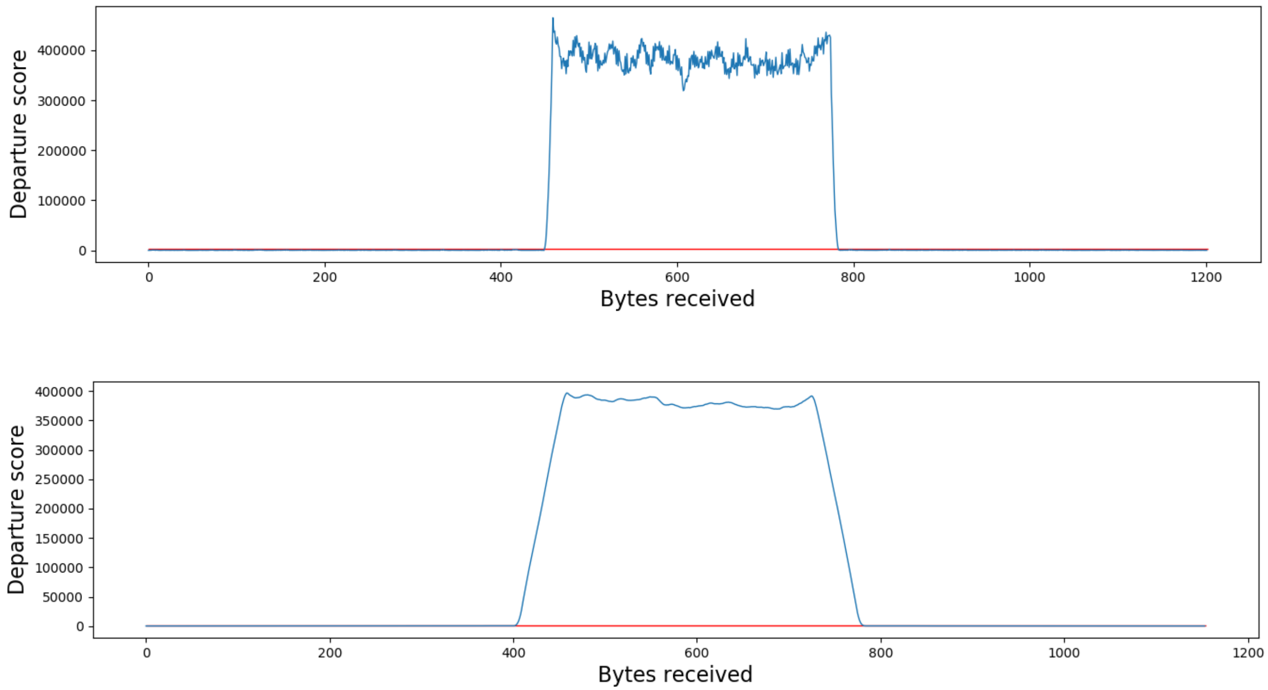
- Fabrication attack





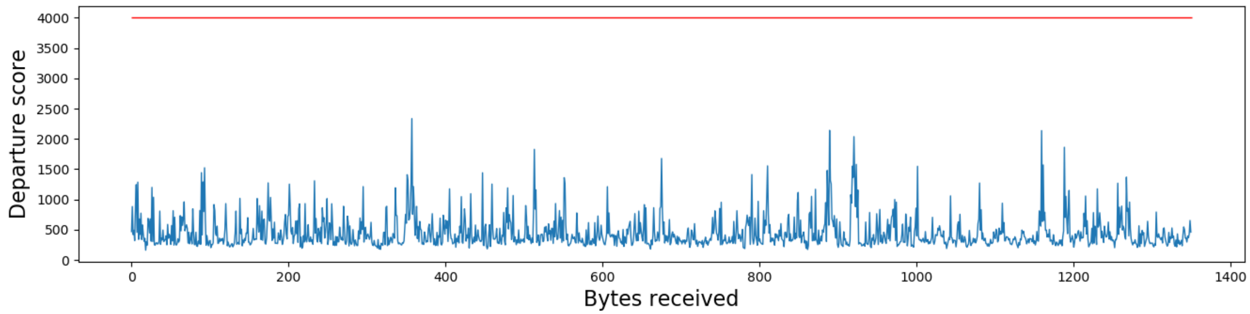
Rolling average, Boxcar (Offline)

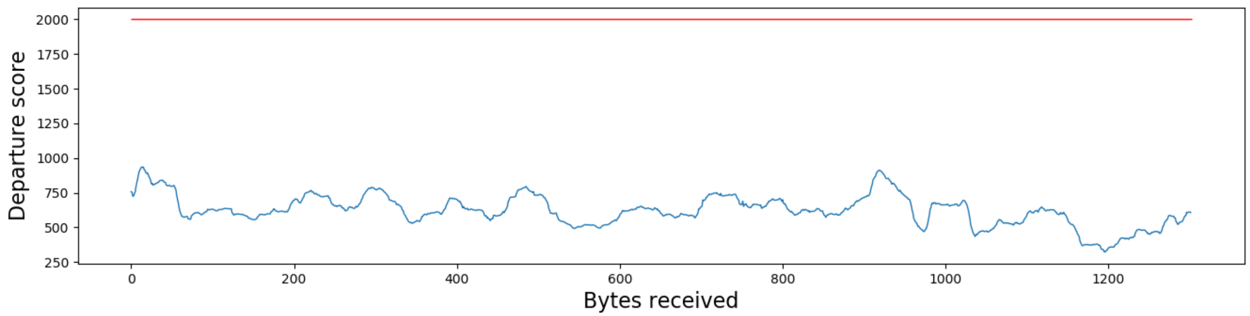
- Masquerade attack



Rolling average, Boxcar (Offline)

- Conquest attack



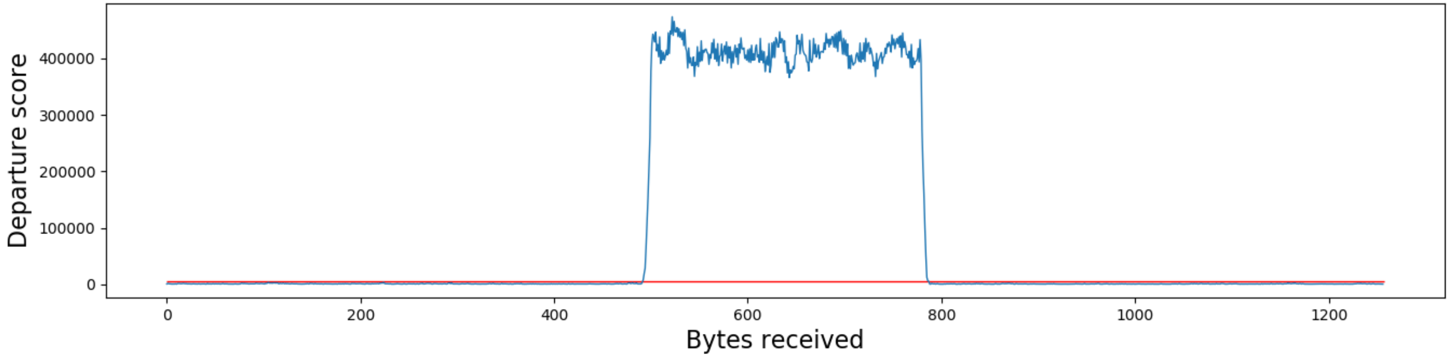


- Background
- Evaluation and Test Benches
- Results: Arduino
- Results: Box Car (Offline)
- **Results: Box Car (Online)**
- Conclusion

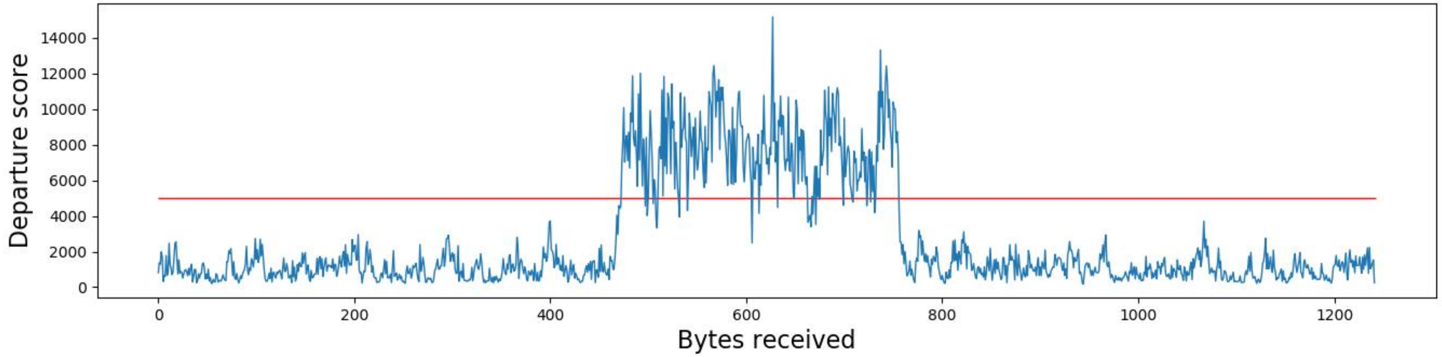
Meeting the real-time requirements

- Producer-Consumer problem for reading CAN messages.
- Output downsample of 2 000 required!

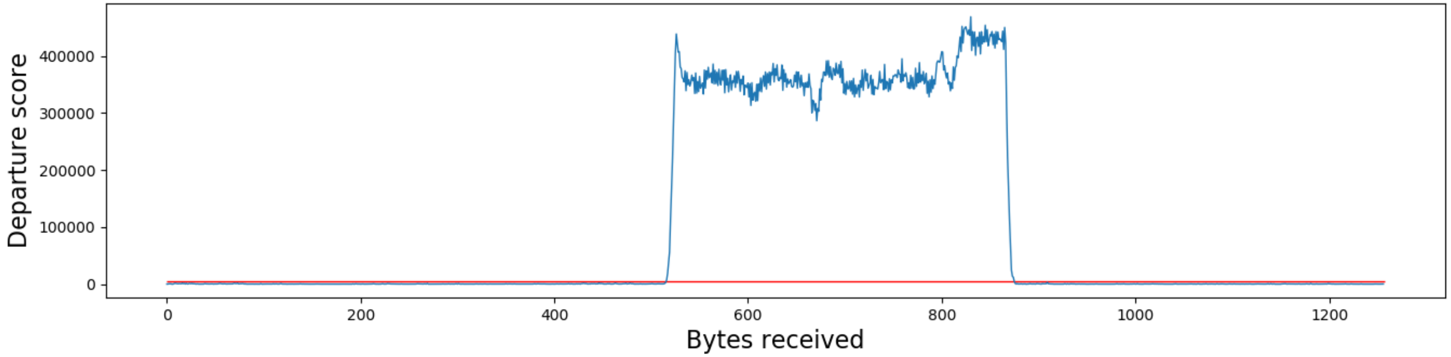
Box Car (Online) - Suspension attack



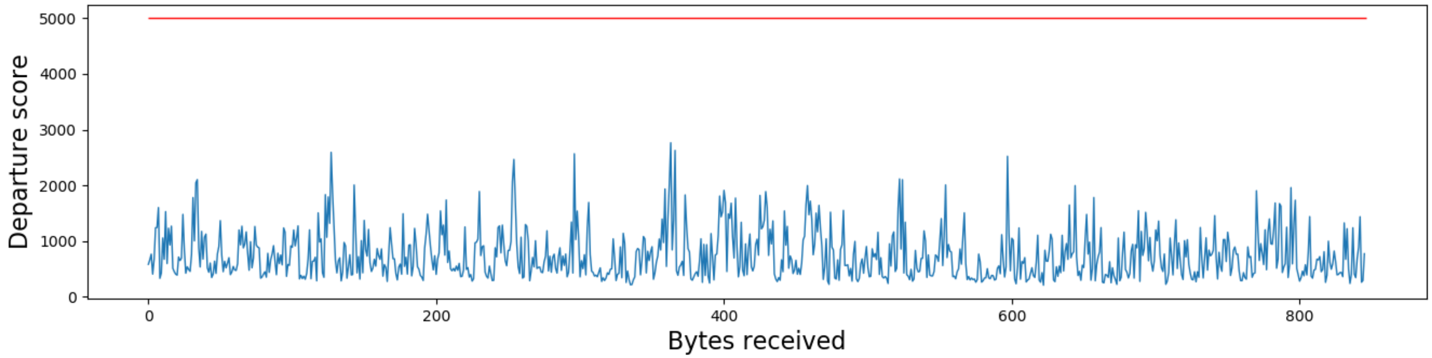
Box Car (Online) - Fabrication attack



Box Car (Online) - Masqurade attack



Box Car (Online) - Conquest attack



- Background
- Evaluation and Test Benches
- Results: Arduino
- Results: Box Car (Offline)
- Results: Box Car (Online)
- **Conclusion**

Conclusion

- We investigated the feasibility of IDS in vehicles
- 4 / 4 attacks were detected on the Arduino
- 3 / 4 attacks were detected on the Box car
- Downsampling makes CASAD more lightweight
- Implementing rolling average reduces false negatives/positives

Future work

- GPUs - Optimized for matrix operations
- Source Detection - Detect which ECU is attacking!
- Intrusion Prevention System - Stop the attacker

References

[1] Valasek, C., & Miller, C. (2015). Remote Exploitation of an Unaltered Passenger Vehicle. Technical White Paper, 2015, 1–91. https://doi.org/10.1088/2041-8205/762/2/L23

[2] Cho, K.-T., Kim, Y., & Shin, K. G. (2018). Who Killed My Parked Car?, 1–27. Retrieved from http://arxiv.org/abs/1801.07741

[3] Cho, K.-T, & Shin, K. G. (2016). Fingerprinting Electronic Control Units for Vehicle Intrusion Detection. Usec, 911–927.

[4] N. Nowdehi, W. Aoudi, M. Almgren and T. Olovsson, "CASAD: CAN-aware stealthy-attack detection for in-vehicle networks", Not yet published, N.D.

HoliSec

Holistic Approach to Improve Data Security

Thank you for your attention!