Detect Denial of Service on an Open Embedded Automotive Platform

Miltos D. Grammatikakis, HMU

(Joint work with Lefteris Kypraiois, Nikos Mouzakitis, Nikos Papatheodorou, and Voula Piperaki)
In-Vehicle Networks & CAN Bus (ISO 11898)

- Bus-based networks
  - LIN, CAN, CAN-FD, MOST, Flexray

- CAN enables reliable, priority-based real-time communication
- CAN data frame
  - 11 or 29-bit ID
  - 8 byte data frame
  - DLC (data length code), CRC, Rx ack
- CAN Arbitration based on CSMA/CA policy
  - Lower ID ⇒ Higher Priority
  - Receivers may filter which IDs they receive
- CAN nodes (ECUs, sensors, actuators) asynchronously bcast using different IDs
  - Baud rate 125, 250, 500 to 1000Kb/s
- No security support
Security Standards

- Cybersecurity standards are protection methods and techniques to secure several environments
  - Devices
  - Networks
  - Systems
  - Applications
  - Services
- IEC 62443 are multi-industry cybersecurity standards that reduce risks and prevent cyber attacks
Threat Models on CAN Bus & DoS

- **Masquerade/Spoof**: Attacker sends a message claiming to be a node other than itself
- **Replay**: Attacker sends a previously received message
  ⇒ Lightweight Crypto-based Solutions
- **Other, e.g. Denial of Service (DoS)?**
  ⇒ Anomaly-based Solutions
  - specification-based,
  - frequency/delay-based: FFT, entropy, neural networks, etc
Use Case: Interactions among CAN Buses

- A central Gateway (GW) can isolate a critical in-vehicle network
- Dashboard subsystem displaying metrics from Drive subsystem

![Diagram of CAN Buses and Gateway]

- Compromised node
Our Detection Approach

- Intrusion detection can be based on changes in random processes
  - power consumption profiles
  - temperature zones
  - frequency-based detection, and
  - round-trip time
- Sliding window statistics approach
  - simple thread computations to detect disruption
Experimental Platform (MCP2515, MCP2551)

- Odroid XU3 as GW (Ubuntu 18.04 LTS) with 2 OBD Dev Kits
  - Extended serial code (2 threads each controlling 1 UART-to-CAN IF, STN2120 2.5k LoC)
  - Sensors: INA231 (power per CPU (big-little)/memory/GPU), thermal zones (per big CPU)
  - Extra: low power, CAN sleep/wakeup routines (OBD Dev Kit Power module)
- Raspberry Pi3 (2019-04-08-Raspbian) with 2 Canberry SPI-to-CAN IF (max 4 IFs)
  - Linux can-utils (cansend, candump, canplay, cangw, …)
Experimental Platform: Energy & Thermal Sensors

- Our serial code on Odroid XU3 reads data from ina231/thermal sensors via IOCTL calls
Legitimate Transactions: Dashboard to Engine

Replay normal engine dataset from:
http://ocslab.hksecurity.net/Dataset/CAN-intrusion-dataset
Legitimate Transactions: Dashboard to Engine
Legitimate Transactions: Dashboard to Engine
Legitimate Transactions: Dashboard to Engine
Legitimate Transactions: Engine to Dashboard
Legitimate Transactions: Engine to Dashboard
Malicious DoS Transactions & Detection
Malicious DoS Transactions & Detection
Malicious DoS Transactions & Detection

- Accurate prediction of DoS results in fast, results in
  - shutting down
  - throttling down, or
  - sleeping the outgoing GW interface, thus safeguarding the engine ECUs
Experimental Approach & Metrics

- Using 500K CAN Baud Rate
  - 2Mb/s UART rate
- Examine four metrics for DoS detection
  - Power consumption
  - Thermal zones
  - ID frequency count
  - Round trip time
- Sliding time-window (EWMA-like)
  - Time window of 10 messages
  - Not fixed time, e.g. every 0.2s to counter random kernel activity
Conclusions

Our ARM-based platform offers the ability to detect denial-of-service using different metrics

**On Gateway (Odroid XU3)**
- Power Consumption
- Thermal zones
- **Frequency Count of IDs**

**On Legitimate CAN side (Dashboard Raspberry)**
- Round trip time

Frequency-based is better (large region of high sensitivity)
Future work

- Include more CAN nodes
- Lightweight crypto, e.g. VATICAN to safeguard from
  - Masquerade/Spoofing
  - Replay/Playback
- Extend Gateway to multiple producer/consumer solutions
  - heterogeneous GW: LIN with CAN, MOST with CAN?
- Test limits of all UART-to-CAN solutions using concurrent lock-free queues
- Linux RT policy support to improve sensitivity (delays, power)

- **Code & troubleshooting guide on sourceforge/github (Q3/20)**
- **Contact: mdgramma@cs.hmu.gr**

Low-cost boards that integrate power sensors are interesting!
Troubleshooting & Development

- Initial experimentation, debugging
References

● OBD-development-kit, Scantool. Available:  
  https://www.scantool.net/obd-development-kit/

● CanberryDual_isolated, Industrialberry. Available:  
  http://www.industrialberry.com/canberry-v-2-1-isolated/

● Ginkgo-usb-can, Viewtool. Available:  

● Serial Console, Sourceforce. [Online]. Available:  
  https://sourceforge.net/projects/serialconsole/files/serialconsole

● Car-Hacking Dataset, HCRL. [Online]. Available:  
  https://sites.google.com/a/hksecurity.net/ocslab/Datasets/CAN-intrusion-dataset

● Several scientific papers related to anomaly-detection
Demo & Questions ?
Thank you
Accelerating deployment in the Arm Ecosystem