Embedded Intrusion Detection based on AI
Dr.-Ing. Andreas Weichslgartner, Bonn, 17.10.2019
evil-hacker-control-server.ninja

execv("pwnd.elf")
Why Should we Use an IDS?

The Ethernet network should be analyzed for anomalies. While, in general, the problem of generic intrusion detection is difficult and often leads to false positives, in this case it works well. That is because this is a network devoid of human users like we are used to in an enterprise environment. All the traffic is periodically generated from machine to machine. [...] Like Ethernet, the CAN network traffic should be observed in real time to identify anomalies. All the attacks outlined in the historical section could have been detected (and prevented) with even the most trivial CAN network intrusion detection software.

Miller, Valasek: Securing Self-Driving Cars (one company at a time), 2018
UNECE WP 29 is coming

Draft Recommendation on Cyber Security of the Task Force on Cyber Security and Over-the-air issues of UNECE WP.29 GRVA:

- The use of combinations of gateways, firewalls, intrusion prevention or detection mechanisms, and monitoring are employed to defend systems
- System monitoring (mentioned in various places)
- Limit and monitor message content and protocol
- Measures to protect systems against embedded viruses/malware should be considered
- System monitoring for unexpected messages/behaviour
- ...

- See also ISO/SAE CD 21434 Road Vehicles — Cybersecurity engineering
Agenda

Motivation

Intrusion Detection

Anomaly Detection & Machine Learning

Results

Summary
Intrusion Detection Systems

A Taxonomy
Intrusion Detection Systems (IDS) Classification

**IDS**

*Host-based* (e.g. system calls)

*Network-based* (e.g. connections)

**Misuse** (e.g. signature)
- Can only detect known attacks

**Anomalies** (e.g. suspicious behavior)
+ Can detect unknown attacks

Can be combined to detect reliably known attacks and also flag suspicious traffic of unknown attacks
## Differences of IDS

<table>
<thead>
<tr>
<th><strong>Enterprise Domain</strong></th>
<th><strong>Embedded Automotive Domain</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>› Vast availability of computing and memory resources</td>
<td>› Limited computing and memory resources</td>
</tr>
<tr>
<td>› Special hardware (GPUs, FPGAs, Many-Core-Chips) available</td>
<td>› No special hardware such as GPUs available</td>
</tr>
<tr>
<td>› Databases with vulnerabilities, malware, IDS rules</td>
<td>› No database with known attacks</td>
</tr>
<tr>
<td>› Unstructured/unpredictable communication &amp; computation</td>
<td>› Structured/predictable communication &amp; computation</td>
</tr>
<tr>
<td>› Human to machine communication</td>
<td>› Machine to machine communication</td>
</tr>
</tbody>
</table>
Anomaly Detection

What is Normal and What Abnormal?
What is AI?

Artificial Intelligence (AI)

Machine Learning (ML)

Deep Learning (DL)
What is an Anomaly?

Example of an Anomaly and an Outlier
What to do with Anomalies?
Anomaly Detection

› Point Anomaly

Length

Interval

Encoder
Latent
Representation
Decoder

Length

Reconstruction
Error

Interval
Anomaly Detection

- Context-based Anomaly

\[ x_t \rightarrow x_{t-1} \rightarrow x_{t-2} \rightarrow x_{t-3} \rightarrow x_{t+1} \]

RNN/Regressor

Reconstruction Error
Implementation

Bringing AI to the ECU
Challenges of an Embedded Implementation

› No GPUs, FPGAs, or accelerators for linear algebra are available for security

› Memory limitations on ECU prevent large models:
  › Algorithms like k-NN are not suitable
  › Pruning, quantization, precision reduction

› Real-Time requirements:
  › Each packet should be classified within a fixed time window
Logging and Training

ECU

- OS/Runtime
  - Feature Extractor
  - Sensor (Host)
  - Logfiles

- Ethernet Packets
  - Feature Extractor
  - Sensor (ETH)
  - Logfiles

- CAN
  - Feature Extractor
  - Sensor (Net)
  - Logfiles

ECU

- Trained Model
- Backend/Host
Inference Pipeline (Embedded)

1. **OS/Runtime**
   - Sensor (Host)
   - Feature Extractor

2. **Ethernet Packets**
   - Sensor (ETH)
   - Feature Extractor

3. **CAN**
   - Sensor (Net)
   - Feature Extractor

4. **Preprocessing**
5. **Classifier**
6. **Alert/Forensic/ReTrain**
   - Backend
   - Trained Model
   - Specification

Time vs. Anomaly Score

- Time: X-axis
- Anomaly Score: Y-axis

- Detected anomalies (points)

Results

CAN Anomaly
Results

Payload Fuzzing

CAN Attack

![Graph showing the results of Payload Fuzzing with Ground Truth and Anomaly Score.](image)

- **Time / s**
- **Reconstruction Error**
- **Ground Truth**
- **Anomaly Score**
Summary

Wrapping Up
Intrusion Detection Systems are necessary in future automotive systems:
- To detect unknown malicious attacks
- Norms and regulations (UNECE WP 29)
- For the automotive domain no databases with malware (binaries and communication) exists
- A data-driven approach based on Machine Learning (ML) can detect unknown attacks
- Anomaly detection is a ML technique which requires only data/traffic from the normal case (no labeling needed)
- Embedded implementation requires thoughtful algorithm selection

Contact:
Dr.-Ing. Andreas Weichslgartner
Audi Electronics Venture GmbH
andreas.weichslgartner@audi.de