A discussion about

Critical Challenges with Autonomous Cars

March 13, 2018

Prepared for:

George Mason University
Overview: Three critical challenges exist in the introduction of autonomous cars

Introduction

- **Security-minded Engineering**: Reducing vulnerabilities through design
- **Socially-minded Decision Logic**: Integrating morality into machine logic
- **Validation of Implementation**: Limiting malicious activity through proper implementation
Critical Challenges with Autonomous Cars

Reducing vulnerabilities starts with security-minded engineering; a design problem

Security-minded Engineering

Challenge: Reducing vulnerabilities prior to implementation

Design a resilient automotive network ecosystem for a hostile world. Issues are:

➔ Providing for security at the component boundaries (e.g., communication)
➔ Providing proper coverage for the most common vulnerabilities

A Solution: Implementing defensible design

Build resilience into design. Introduce security-minded engineering to:

➔ Form a trust network (e.g., Segregate-able ECU's with protective filtering at the borders to form trust zones)
➔ Introduce ECU's with basic sanity-checks to weed out common networking hacks
➔ Introduce strong cryptography where it makes sense (e.g., UDS/Firmware Updates)
Getting the algorithms “right” considers morality and malicious activity; a social impact problem

Socially-minded Decision Logic

**Challenge: Getting the algorithms “right”**

*Who programs the machine logic? Issues include:*  
➔ The moral principle: Consequentialist moral reasoning vs. Categorical moral reasoning  
➔ Sanity check: Positive identification of “trickery” attempts

**A Solution: Integrating morality and reality checks into machine logic**

*Determine which choice to make.*  
➔ Programming algorithms “correctly” (e.g., which choice: A/B/C?)  
➔ Programming to identify false messages, outside of the expected (e.g., sensors)
Limit successful malicious activity through implementation validation; a post-implementation security problem

Validation of Implementation

Challenge: Limiting implementation bugs

**Designs are not always implemented properly. Issues include:**

- Implementation flaws may undermine secure design in unexpected ways (e.g., proper memory buffer management)
- “Correct” implementations can still have vulnerabilities

A Solution: Validating the implementation

**Validating the implementation can reduce the attackers “playground”. For example:**

- A post-design security check identifies implementation flaws
  - Robustness Testing (Fuzzing)
  - Static code analysis
  - Red-teaming
  - Periodic analysis (e.g., Internal, 3rd Party)

- Security checks also provide feedback loop for re-design / implementation
  - Authentication / validation of OTA firmware updates
  - Validation of control messages
A discussion about

Critical Challenges with Autonomous Cars

March 13, 2018