Experimental Security Analysis of a Modern Automobile

Matthias Lange

TU Berlin

June 29th, 2010
Karl Koscher, Alexei Czeskis, Franziska Roesner, Shwetak Patel, and Tadayoshi Kohno

University of Washington

Stephen Checkoway, Damon McCoy, Brian Kantor, Danny Anderson, Hovav Shachman, and Stefan Savage

University of California, San Diego

published in 2010 IEEE Symposium on Security and Privacy
Outline

1. Introduction
2. Automotive Embedded Systems
3. Threat Model
4. Security Analysis
5. Results
6. Discussion
Introduction

- automobile remained static for 80 years
  - gasoline engine
  - four wheels
  - familiar user interface
- today many computers coordinate and monitor sensors
  - 100MB of binary code spread over 50 - 70 ECUs
Goals

- **safety**
  - Anti-lock Brake System
  - standard access through OBD-port

- value added features
  - automatic crash response
  - remote diagnostics
  - stolen vehicle recovery

- future: App Store
Consequences

New Threats
Computerized environments bring new array of potential new threats.

New Attack Vectors
Trend will open a wide range of attack vectors for attackers.
Automotive Embedded Systems

- ECUs found in cars since late 70s
- partly due to legislation
- complex interactions between ECUs
  - Electronic Stability Control
  - steer-by-wire
- Interconnection
  - past: bilateral physical wire
  - today: digital buses like CAN and FlexRay
Connectivity

- high speed bus for real-time telemetry
- low speed bus for binary actuators
- buses are bridged
- cellular based uplinks
  - remote unlock
  - track car location
  - remote stop
Threat Model

Purpose

What can an attacker do if she is able to maliciously communicate on the car’s internal network?

- analysis of attack surface intentionally left blank
  - through wireless interfaces
  - OBD-port
  - malicious component
## Experimental Setup

### Bench

Physically extracted hardware hooked up to a power supply, CAN-to-USB converter and a oscilloscope.
Experimental Setup

**Bench**
Physically extracted hardware hooked up to a power supply, CAN-to-USB converter and a oscilloscope.

**Stationary car**
Car elevated on jack stands, laptop connected to OBD-port.
Experimental Setup

**Bench**
Physically extracted hardware hooked up to a power supply, CAN-to-USB converter and a oscilloscope.

**Stationary car**
Car elevated on jack stands, laptop connected to OBD-port.

**On the road**
Experimented with car at speed on a de-commissioned airport with wireless control.
Experimental Setup

**Bench**
Physically extracted hardware hooked up to a power supply, CAN-to-USB converter and a oscilloscope.

**Stationary car**
Car elevated on jack stands, laptop connected to OBD-port.

**On the road**
Experimented with car at speed on a de-commissioned airport with wireless control.

**CarShark**
A custom CAN bus analyzer and packet injection tool.
## CAN Security Challenges

### Broadcast

<table>
<thead>
<tr>
<th>Packets broadcasted to all nodes</th>
</tr>
</thead>
</table>

DoS
Packet flooding attack, priority based arbitration allows node to assert dominant state indefinitely

No Authentication
Packets do not contain any source identifier

Weak access control
Challenge response sequence to protect ECU against certain actions without authorization.

Firmware Updates
Malicious firmware updates
## CAN Security Challenges

### Broadcast
- Packets broadcasted to all nodes

### DoS
- Packet flooding attack, priority based arbitration allows node to assert dominant state indefinitely
CAN Security Challenges

Broadcast
Packets broadcasted to all nodes

DoS
Packet flooding attack, priority based arbitration allows node to assert dominant state indefinitely

No Authentication
Packets do not contain any source identifier
CAN Security Challenges

**Broadcast**
Packets broadcasted to all nodes

**DoS**
Packet flooding attack, priority based arbitration allows node to assert dominant state indefinitely

**No Authentication**
Packets do not contain any source identifier

**Weak access control**
Challenge response sequence to protect ECU against certain actions without authorization.
# CAN Security Challenges

<table>
<thead>
<tr>
<th><strong>Broadcast</strong></th>
<th>Packets broadcasted to all nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DoS</strong></td>
<td>Packet flooding attack, priority based arbitration allows node to assert dominant state indefinitely</td>
</tr>
<tr>
<td><strong>No Authentication</strong></td>
<td>Packets do not contain any source identifier</td>
</tr>
<tr>
<td><strong>Weak access control</strong></td>
<td>Challenge response sequence to protect ECU against certain actions without authorization.</td>
</tr>
<tr>
<td><strong>Firmware Updates</strong></td>
<td>Malicious firmware updates</td>
</tr>
</tbody>
</table>
Deviations from Standards

- standards prescribe risk-mitigation which components *should* comply

disable communication

Reject 'disable CAN communication' when it is unsafe

reflashing ECU while driving

Failed, reflash firmware while car wheels moving, engine stopped

noncompliant access control

Failed, hardcoded key pair for ALL units, result not used at all, release breakes while car in motion

network segregation

Failed, some bridge devices only reprogrammable from low-speed bus, malicious code may access high-speed bus
Deviations from Standards

- standards prescribe risk-mitigation which components *should* comply

Disable Communication

Failed, disable communication while car wheels moving
Deviations from Standards

- standards prescribe risk-mitigation which components *should* comply

**Disable Communication**
Failed, disable communication while car wheels moving

**Reflashing ECU While Driving**
Reject reflashing when it is unsafe
Deviations from Standards

- standards prescribe risk-mitigation which components *should* comply

Disable Communication
Failed, disable communication while car wheels moving

Reflash ECU While Driving
Failed, reflash firmware while car wheels moving, engine stopped
Deviation from Standards

- standards prescribe risk-mitigation which components \textit{should} comply

Disable Communication
- Failed, disable communication while car wheels moving

Reflashing ECU While Driving
- Failed, reflash firmware while car wheels moving, engine stoped

Noncompliant Access Control
- Safety functionality \textit{must} be protected by challenge response and unsafe
- DeviceControl must be denied
Deviations from Standards

- standards prescribe risk-mitigation which components should comply
  
  **Disable Communication**
  Failed, disable communication while car wheels moving

  **Reflashing ECU While Driving**
  Failed, reflash firmware while car wheels moving, engine stopped

  **Noncompliant Access Control**
  Failed, hardcoded key pair for ALL units, result not used at all, release breaks while car in motion
Deviations from Standards

- standards prescribe risk-mitigation which components *should* comply

**Disable Communication**
Failed, disable communication while car wheels moving

**Refloshing ECU While Driving**
Failed, reflash firmware while car wheels moving, engine stopped

**Noncompliant Access Control**
Failed, hardcoded key pair for ALL units, result not used at all, release breakes while car in motion

**Network Segregation**
Gateway between low- and high-speed bus must only be reprogrammable from the high-speed bus
Deviations from Standards

- standards prescribe risk-mitigation which components should comply

Disable Communication
Failed, disable communication while car wheels moving

Reflashing ECU While Driving
Failed, reflash firmware while car wheels moving, engine stopped

Noncompliant Access Control
Failed, hardcoded key pair for ALL units, result not used at all, release breaks while car in motion

Network Segregation
Failed, some bridge devices only reprogrammable from low-speed bus, malicious code may access high-speed bus
Deviations from Standards

- standards prescribe risk-mitigation which components \textit{should} comply

Disable Communication
Failed, disable communication while car wheels moving

Reflashing ECU While Driving
Failed, reflash firmware while car wheels moving, engine stoped

Noncompliant Access Control
Failed, hardcoded key pair for ALL units, result not used at all, release breakes while car in motion

Network Segregation
Failed, some bridge devices only reprogrammable from low-speed bus, malicious code may access high-speed bus
Packet Sniffing and targeted probing with CarShark
Fuzzing, aid in reverse engineering
reverse engineering with IDA Pro
Results

- control of radio, disable user control, increase volume, clicks and chimes etc.
- display arbitrary messages on the instrument panel cluster
- honk the horn, lock doors, shoot windshield fluids etc.
- boost engine RPM, disturb engine timing, disable all cylinders, forge 'airbag deployed'
- lock individual brakes (even resistant), release brakes, prevent enabling of brakes
- turn on/off fans and AC
- disabling communication led to reported speed be 0 mph, arbitrary offset to reported speed
- lights out, 'self destruct', self wiping code
Pwned by CarShark
CARSHARKED X_X
PAND321
Discussion

- Manufacturers unaware of security issues?
- How to handle complexity?
- Can a micro kernel system consolidating different ECUs help solving some issues with CAN?