Exposing New Vulnerabilities of Error Handling Mechanism in CAN

Khaled Serag†, Rohit Bhatia†, Vireshwar Kumar*, Z. Berkay Celik†, Dongyan Xu†

† Purdue University
*Indian Institute of Technology Delhi
Background and Motivation

- CAN: Communication protocol for automobiles and industrial automation
  - Wiring
  - Decentralization
  - Noise Resistance
  - Effective error handling and fault confinement mechanism
- We investigate CAN’s error handling and fault confinement mechanism
Background and Motivation
Background and Motivation

- CAN Operation
  - Format

Standard Data Frame Format

<table>
<thead>
<tr>
<th>S</th>
<th>O</th>
<th>F</th>
<th>ID (11 b)</th>
<th>R</th>
<th>I</th>
<th>D</th>
<th>R</th>
<th>E</th>
<th>DLC (4 b)</th>
<th>Data (0-8 B)</th>
<th>CRC (2 B)</th>
<th>ACK (2 b)</th>
<th>EOF (7 b)</th>
</tr>
</thead>
</table>

Arbitration
Background and Motivation

- CAN Operation
  - Format
- CAN error handling and fault confinement mechanism
  - Error Counters: TEC, REC
  - Error States

![Diagram showing error states and standard data frame format](image-url)
Background and Motivation

• Attacker can remotely compromise certain ECUs (i.e., telematics)
  • Weak security of ECUs has been demonstrated
Background and Motivation

• Attacker can remotely compromise certain ECUs (i.e., telematics)
  • Weak security of ECUs has been demonstrated

• New: Attacks against error handling
  • Simultaneous transmission and collisions
  • Attacker can dictate a victim’s error state
  • Security impact of error handling is understudied

Causing Deliberate Collisions
Background and Motivation

- Attacker can remotely compromise certain ECUs (i.e., telematics)
  - Weak security of ECUs has been demonstrated
- **New**: Attacks against error handling
  - Simultaneous transmission and collisions
  - Attacker can dictate a victim’s error state
  - Security impact of error handling is understudied
Background and Motivation

- Attacker can remotely compromise certain ECUs (i.e., telematics)
  - Weak security of ECUs has been demonstrated
- New: Attacks against error handling
  - Simultaneous transmission and collisions
  - Attacker can dictate a victim’s error state
  - Security impact of error handling is understudied

[Diagram of error states and deliberate collisions]
CANOX: A Protocol Testing Tool For CAN

- CAN Operation eXplorer (CANOX)
  - Explores the impact of operating outside of the error active state
  - Reveals possible vulnerabilities
CANOX: A Protocol Testing Tool For CAN

- **CAN Operation eXplorer (CANOX)**
  - Explores the impact of operating outside of the error active state
  - Reveals possible vulnerabilities
- **Node under Test (NUT)**
  - Logs its metrics throughout the experiment
CANOX: A Protocol Testing Tool For CAN

- Scenarios
  - Single Collision Scenario
  - Successive Transmission Scenario
  - Single Transmission Scenario
CANOX: A Protocol Testing Tool For CAN

• Scenarios
  • Single Collision Scenario
  • Successive Transmission Scenario
  • Single Transmission Scenario

• Behavioral Metrics
  • Standby Delay (SD)
  • TEC Change (TECC)
CANOX: A Protocol Testing Tool For CAN

- Scenarios
  - Single Collision Scenario
  - Successive Transmission Scenario
  - Single Transmission Scenario

- Behavioral Metrics
  - Standby Delay (SD)
  - TEC Change (TECC)

- Vary error state and bus traffic
CANOX: A Protocol Testing Tool For CAN

• Scenarios
  • Single Collision Scenario
  • Successive Transmission Scenario
  • Single Transmission Scenario

• Behavioral Metrics
  • Standby Delay (SD)
  • TEC Change (TECC)

• Vary error state and bus traffic
• Log analyzer detects violations
Vulnerability 1: Passive Error Regeneration

- Failure to send a passive error frame generates a new error

**Passive Error Frame**

<table>
<thead>
<tr>
<th>Error Flag</th>
<th>Error Delimiter</th>
</tr>
</thead>
<tbody>
<tr>
<td>6 b (recessive)</td>
<td>8 b (recessive)</td>
</tr>
</tbody>
</table>
Vulnerability 1: Passive Error Regeneration

- Failure to send a passive error frame generates a new error
Vulnerability 1: Passive Error Regeneration

• Failure to send a passive error frame generates a new error
Vulnerability 1: Passive Error Regeneration

- Failure to send a passive error frame generates a new error
Vulnerability 2: Deterministic Recovery Behavior

- At recovery, an ECU will send the same message that failed to transmit.
Vulnerability 2: Deterministic Recovery Behavior

- At recovery, an ECU will send the same message that failed to transmit
Vulnerability 2: Deterministic Recovery Behavior

- At recovery, an ECU will send the same message that failed to transmit
Vulnerability 3: Error State Outspokenness

- The error state of a message sender is detectable by any node on the bus
- This could be exploited to map the network
Vulnerability 3: Error State Outspokenness

• The error state of a message sender is detectable by any node on the bus
• This could be exploited to map the network
Vulnerability 3: Error State Outspokenness

- The error state of a message sender is detectable by any node on the bus.
- This could be exploited to map the network.
"Scan-Then-Strike" (STS) Attack

- **Threat Model:**
  - Remotely compromised ECU able to execute arbitrary code
  - No physical access or previous knowledge of the vehicle
“Scan-Then-Strike” (STS) Attack

- Threat Model:
  - Remotely compromised ECU able to execute arbitrary code
  - No physical access or previous knowledge of the vehicle

[Diagram showing vehicle network mapping with ECU-1, ECU-2, ECU-3, ECU-4, ECM, BCM, TCM, EBCM]
“Scan-Then-Strike” (STS) Attack

• **Threat Model:**
  • Remotely compromised ECU able to execute arbitrary code
  • No physical access or previous knowledge of the vehicle

Attacker observing a vehicular CAN bus

1-Network Mapping

2-Victim Identification
“Scan-Then-Strike” (STS) Attack

- **Threat Model:**
  - Remotely compromised ECU able to execute arbitrary code
  - No physical access or previous knowledge of the vehicle

Attacker observing a vehicular CAN bus

1-**Network Mapping**

2-**Victim Identification**

3-**Learning Recovery Behavior**
“Scan-Then-Strike” (STS) Attack

• Threat Model:
  • Remotely compromised ECU able to execute arbitrary code
  • No physical access or previous knowledge of the vehicle

1-Network Mapping

2-Victim Identification

3-Learning Recovery Behavior

4-Suppression and Recovery Prevention
## Results

### Testbed Results

<table>
<thead>
<tr>
<th>ECU #</th>
<th>Suppression Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECU-1</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-2</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-3</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-4</td>
<td>99.9%</td>
</tr>
</tbody>
</table>
## Testbed Results

<table>
<thead>
<tr>
<th>ECU #</th>
<th>Suppression Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECU-1</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-2</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-3</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-4</td>
<td>99.9%</td>
</tr>
</tbody>
</table>

$$S_{rate} = \frac{Bus\ Off\ Time}{Total\ Time}$$
## Results

### Testbed Results

<table>
<thead>
<tr>
<th>ECU #</th>
<th>Suppression Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECU-1</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-2</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-3</td>
<td>99.9%</td>
</tr>
<tr>
<td>ECU-4</td>
<td>99.9%</td>
</tr>
</tbody>
</table>

**Suppression Rate**

\[
S_{rate} = \frac{\text{Bus Off Time}}{\text{Total Time}}
\]

### Vehicle Results

<table>
<thead>
<tr>
<th>ECU #</th>
<th>Function</th>
<th>Suppression Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ECU-1</td>
<td>EBCM (Brake)</td>
<td>97.5%</td>
</tr>
<tr>
<td>ECU-2</td>
<td>BCM (Body)</td>
<td>91.4%</td>
</tr>
<tr>
<td>ECU-3</td>
<td>TCM (Transmission)</td>
<td>85%</td>
</tr>
<tr>
<td>ECU-4</td>
<td>ECM (Engine)</td>
<td>83%</td>
</tr>
</tbody>
</table>
Demo

ID 1: 07A

ID 2: 07F

Attacker

Monitors

Victim
Responsible Disclosure

- Reported vulnerabilities to:
  - Bosch Product Security Incident Response Team (PSIRT).
  - Cybersecurity and Infrastructure Security Agency (CISA)
    - Case opened
  - Society of Automotive Engineers (SAE)
    - Committee review for next standard revision
- Proposed mitigations to each of the discovered vulnerabilities
Conclusion

• CAN’s error handling mechanism – a security weakness

• We introduced CANOX
  • A protocol testing tool to identify possible vulnerabilities

• Three new error-handling vulnerabilities revealed by CANOX
  • Each could be exploited separately
  • STS: an end-to-end attack via exploiting all three vulnerabilities

• Attack Implementation on a testbed and a real vehicle
  • Mapping Accuracy: 100%
  • Single Frame Bus Off Effectiveness: 100%
  • Persistent Bus Off Suppression Rate: 83-100%
Thank You!

Questions?

Khaled Serag: kserag@purdue.edu

This work was supported in part by the Office of Naval Research (ONR) under Grant N00014-18-1-2674.