Heavy/Medium Duty Vehicle Hacking
From Myth to Reality

Subhojeet Mukherjee
Colorado State University
So What Type of Vehicles Are We Talking?

- Transport goods worth about $53 billion were moved each day in 2015 [BTS].
  - Commercially motivated attacks can have severe financial consequences.
- Emergency vehicle response time is critical.
  - Personally motivated attacks can have life threatening consequences.
- Capital equipment bear high asset value.
  - Commercially motivated attacks can have severe financial and legal consequences.
- Military vehicles are mission critical.
  - Politically motivated attacks can have widespread negative consequences.
Scope of Today’s Talk

- Not all heavy/medium vehicles are alike
  - Internal operations might be similar
  - Externally design can vary
    - Depends on the purpose of use
- Heavy duty vehicles
  - Trucks (with or without trailers) and buses are alike.
  - Let us focus on them for now.
The Rise of the Electronic Vehicle [1]
Trucks Nowadays
The Era of Electronic Transportation

- Why the electrification?
  - Early initiative was emission control
  - From then onwards
    - Reduced wiring
    - Enhanced functionality
    - Diagnostics
    - Fleet Management

- Just some examples of electronic control....
  - Entertainment
    - Headunit
  - Diagnostic and fleet management
    - Telematics
  - Safety
    - Electronic stability program (ESP)
    - Advanced driver-assistance system (ADAS)
  - Security
    - Immobilizer

WHAT'S THE PROBLEM?
The Good is the Bad and the Ugly

- There is a need
  - To know
    - Operators/Owners -- the truck's runtime stats
    - Drivers -- the directions, emergency situations.
    - Garage people -- the fault codes, vital forensics.
  - To act
    - Owner/Operator manipulating internal operations to control driving and fuel economy
    - Automatic control from ESP, ADAS etc.
    - Theft prevention by immobilizer.
  - For comfort
    - Latest songs, news, on internet radio.

- The needs are often served/met by remote/local connections to the vehicle bus.

- The needs are often served by the same control unit.

- Security was not a major concern in making design decisions, until this happened....
  - At CAN and J1939 protocol level
  - At the ECU design level
### Bad Guys and their Cruel Intentions!!

Wolf et al. [1]

<table>
<thead>
<tr>
<th>Physical Theft</th>
<th>Data Theft</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exemplary attacks</strong></td>
<td><strong>Exemplary attacks</strong></td>
</tr>
<tr>
<td>Theft of airbags, navigation systems, whole car</td>
<td>IP theft, privacy invasions, counterfeit parts</td>
</tr>
<tr>
<td>Typical attacker</td>
<td>Typical attacker</td>
</tr>
<tr>
<td>Organized crime</td>
<td>Plagiarist, competitor, third parties (e.g., insurance companies), OEM, government, organized crime</td>
</tr>
<tr>
<td>Attack probability</td>
<td>Possible</td>
</tr>
<tr>
<td>Possible</td>
<td>Possible</td>
</tr>
<tr>
<td>Damaged party</td>
<td>Owner</td>
</tr>
<tr>
<td>Owner</td>
<td>Driver, owner, OEM</td>
</tr>
<tr>
<td>Damage potential</td>
<td>Significant</td>
</tr>
<tr>
<td>Critical</td>
<td>Critical</td>
</tr>
<tr>
<td>Resulting cybersecurity risk</td>
<td>Medium</td>
</tr>
</tbody>
</table>

### Manipulation

<table>
<thead>
<tr>
<th>Physical Theft</th>
<th>Data Theft</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exemplary attacks</strong></td>
<td><strong>Exemplary attacks</strong></td>
</tr>
<tr>
<td>Chip tuning, hijack critical functions, hijack driving functionality</td>
<td>Delete critical data, lock critical functions, hijack driving functions</td>
</tr>
<tr>
<td>Typical attacker</td>
<td>Typical attacker</td>
</tr>
<tr>
<td>Extortionist, terrorist, nation-state</td>
<td>Extortionist, terrorist, nation-state</td>
</tr>
<tr>
<td>Attack probability</td>
<td>Very rare</td>
</tr>
<tr>
<td>Unlikely</td>
<td>Unlikely</td>
</tr>
<tr>
<td>Damaged party</td>
<td>Driver, society</td>
</tr>
<tr>
<td>OEM, third party, society</td>
<td>Driver, operator, customer, society</td>
</tr>
<tr>
<td>Damage potential</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>Catastrophic</td>
<td>Catastrophic</td>
</tr>
<tr>
<td>Resulting cybersecurity risk</td>
<td>Medium</td>
</tr>
</tbody>
</table>
Today, I will try to chalk out bottom-up plan for the bad guys to achieve their respective goals.

- Show common attack surfaces between cars and trucks.
- Use an example automotive network that they will break into.
- Perform any intermediary steps.
- Inject necessary messages to achieve the eventual goal.

Although, I am using references, a large bulk of this is part of my research.
Common Attack Surfaces
Example Network
(Created from descriptions in [1,2,3])

- **OBD-II passthrough device**
  - Connected to the OBD-II port

- **Service computer**
  - Can communicate remotely and physically with the passthrough device

- **Headunit**
  - Includes a GPS, CD player and Bluetooth for handsfree entertainment

- **Telematics Control Unit**
  - For communication with operator/owner
  - Uses a cellular network

- **Engine Control Unit**
  - Controls driving critical functions

- **Transmission control unit**
  - Controls transmission related functions.

- **CAB Control Unit**
  - Controls cabin functions like temperature, door control etc.

- **Trailer Control Unit**
  - Controls trailer functions like temperature, door control etc.

- **Bridge**
  - Sits between networks.
  - Filters traffic across networks to control bandwidth utilization.

- **Gateway**
  - Same as bridge, except with protocol conversion functionalities.

**Image Sources**
Steps to HACK, from a High Level

(Created from descriptions in [1,5])

- Access to the bus through a compromised Surface
- Target ECU Reached
- Target ECU Attacked
- Data theft
- Physical Theft Achieved
- Attack on Safety/Reliability Achieved
- Data theft
Let's dive a bit deeper
Compromise an Attack Surface [2]
Steps to HACK, from a High Level
(Created from descriptions in [5])
Before we move forward,
Let's understand some fundamentals of in-vehicle networking.
MECHATRONICS In-Vehicle Communication Technology

- Electronic control units (ECU) regulate critical vehicle functions.
- Need to communicate with other units.
  - Request information.
  - Command operations.
  - Report parameter status and faults if any.
- Sensor readings and possible actions are embedded in messages according to the SAE J1939 standards.
  - RPM reading from the crankshaft position sensor.
  - Initiate fuel injection.
- Transmitted as sequence of bits on the CAN bus
  - Bundled into CAN frames.
- ECUs receive, interpret and use messages
  - RPM displayed on the dash.
  - Fuel injection initiated.
- Communication model is requirement dependent
  - Might vary across manufacturers.

Here are the current fuel levels

Here is the current engine speed and other stuff

Thanks guys, Let me process the messages and display that stuff on the dash.

Here are the current fuel levels

010101010….

010101010….
The CAN protocol

• Facilitates broadcast communication over a 2-wire multi-master serial bus.
  • Shielded wires
  • Reduced EMI.
  • Twisted wires
  • Cancel EMI.
  • Terminating resistors
    • Maintain recessive level.
    • Reduce reflection.

• Differential voltage
  • Cancel out any residual noise.
  • Dominant bit: 0
  • Recessive bit: 1

• CSMA/CD with message priority-based arbitration
  • Nodes transmit when bus is free.
  • Arbitration is performed on the Identifier field.

• CAN Frame format
  • 11/29 bit identifier
  • 0…64 bits of data.

https://www.linkedin.com/pulse/automotive-can-bus-system-explained-kiril-mucevski
SAE J1939: A Layered Networking Paradigm

- Broadcast networks.
  - High speed CAN, 250 kbps.
  - Readily connectable using bridge/gateway.
  - Possible message filtering.
  - Nodes/ECUs connected to main or implement networks/sub-networks.

- Messages composed and interpreted at the application layer.
  - Identifier and data.

- Message data can be longer than 64 bits, eg. VIN number.
  - Split into ~128 bit J1939 PDU
  - CAN frames can only be maximum 128 bit.
  - Reliable message delivery.

- No message encapsulation
  - Modular functionalities invoked when required.
    - Eg. Reliable delivery functions at layer 2.

- Transmitted as stream of bits at the Physical/CAN layer.
J1939 PDU Interpretation

- CAN frame is obtained
  - start-of-frame and end-of-frame bits are used
- CAN specific bits are removed to obtain a J1939 PDU
- PDU has a 29 bit identifier and 64 bits data
- Parameter Group Number (PGN) is obtained
  - E.g. PGN 64993 “Cab/AC Climate Information”.
- Associated Suspect Parameter Numbers (SPNs) are obtained from J1939 Digital Annex
  - SPN 2609 “Cab/AC Refrigerant Compressor Outlet Pressure”
  - SPN 7853 “Air Conditioner Compressor Status”
- SPNs bits extracted using Length and Starting bit position
- SPNs interpreted using Resolution and Offset
  - SPN 2609 [Cab/AC Refrigerant Compressor Outlet Pressure] \(\rightarrow 01100100_{10} \times 16 + 0_{10} \rightarrow 100_{16} \rightarrow 1600 \text{ kPa} \).
Lets get Back to business
Steps to HACK, from a High Level
(Created from descriptions in [5])
Reach the Target ECU [3]

Target ECU Reached

OR

Jump Bridge/Gateway

AND

Access to the OBD-II port

A lot of effort
For the next couple of months

Access to desired network

Access to a multi-homed ECU

ECU On network

Access to bus through a single-homed ECU

Makeshift Bridge/Gateway

AND

Re-flashing enabled on multi-homed ECU

Access to the Headunit

Access to the desired network

On network

OR

OR

Access to the multi-homed ECU

On network

Access to the desired network

On network

Access to the multi-homed ECU

On network
Steps to HACK, from a High Level
(Created from descriptions in [5])

1. Access to the bus through a compromised Surface
2. Target ECU Reached
3. Target ECU Attacked
4. Data theft
5. Data theft
6. Access to the bus through a compromised Surface
Before we move forward,
Let's enjoy some real world ATTACKS!!!

on embedded Truck networks
Practical dos attacks on target ecus

Experiment set-up: A Remote testbed
Truck in a box

- A remotely accessible testbed for conducting heavy vehicle research.
  - Sandboxexperiments.
- Nodes connected to the network
  - Engine control module
    - Engine and retarder controller.
  - Brake controller
  - Telematics unit
  - Beaglebone node controllers.
    - Remote access.
    - Allows access to CAN bus.
**Some background** Multi-packet request transfer protocol

<table>
<thead>
<tr>
<th>Label</th>
<th>PGN</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Request</td>
<td>EA00</td>
<td>Requested PGN in reverse byte order</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RTS</td>
<td>EC00</td>
<td>10</td>
<td>Total num bytes</td>
<td>Total num packets</td>
<td>Max num packets to be sent in response to one CTS</td>
<td>Requested PGN in reverse byte order</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTS</td>
<td>EC00</td>
<td>11</td>
<td>Num packets</td>
<td>Next seq number</td>
<td>FF</td>
<td>FF</td>
<td>Requested PGN in reverse byte order</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td>EB00</td>
<td>seq</td>
<td></td>
<td></td>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Experiment testbed

- Remotely accessible testbed
- J1939 speaking Engine Control Module and Brake Controller.
- High speed (250 kbps) CAN bus with terminating 120 ohm resistors.
- 2 Beagle Bone Black devices running 32 bit Ubuntu@Linux operating systems running on an ARM rev 2 v7l@1000MHz processor with 500 MB of RAM.
- 14-15% bus load.
Request Overload – Attack Scenario $@CSUSource$

- **Issue**
  - When ECUs follow the above mentioned algorithm, they de-queue items and process requests.
  - Multi-packet requests have a higher processing requirement.

- **Attack**
  - Destination specific multi-packet request messages are sent at a rapid interval.

- **Impact**
  - Periodic message volume decreases significantly
  - Failure to respond to legitimate requests
Request overload Results

- Experiment independent Factors
  - number of concurrent thread [1, 4, 8]
  - injection time interval in ms [0.4, 0.8, 1.2]
  - source address [00, 0B, F9]

- High Priority messages
  - Average drop: 46%

- Low Priority Messages
  - Average drop: 65%

- Pearson correlation coeff.
  - Source: -0.01
  - Thread: 0.137
  - Interval: 0.66

- Two-tailed Mann-Whitney U test
  - p-value of 0.01468 (<= .5)
  - 5% confidence interval
False Rts – The Warning

• **Theory**
  - Message size is piggybacked on the RTS message.
  - RTS retransmission shall be acted upon.
  - Sender is not notified.

• **Attack**
  - Listen for an RTS
  - Send a spoofed RTS with reduced number of bytes or packets.

• **Impact**
  - Crash [Buffer Overflow]

Allocate

Reallocate

Request

CTS

Data

RTS Tot-bytes = 21

RTS Tot-bytes = 5

CRASH !!
Connection exhaustion

- Theory
  - Only 255 possible nodes.
  - Each pair of ECUs can have at most one connection at any given time.
  - Keep connections open by sending CTS messages within a specified time period.

- Attack
  - Scan the network for all available source addresses.
  - A max of 255.
  - Create connections by spoofing all source addresses.
  - Keep the connections alive.

- Impact
  - Legitimate connection attempts are rejected.
## Connection exhaustion Trace

<table>
<thead>
<tr>
<th>Connection</th>
<th>Source Address</th>
<th>Destination Address</th>
<th>Type</th>
<th>Protocol</th>
<th>Options</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>BB1-&gt;Engine-#1 request</td>
<td>00EA0011</td>
<td>EB FE 00 00 00 00 00 00 00</td>
<td>EB FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 RTS</td>
<td>18EC1100</td>
<td>10 2C 00 07 FF EB FE 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB1-&gt;Engine-#1 CTS</td>
<td>00EC0011</td>
<td>11 07 01 FF FF EB FE 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB1-&gt;Engine-#1 request</td>
<td>00EA000B</td>
<td>EB FE 00 00 00 00 00 00 00</td>
<td>EB FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 RTS</td>
<td>18EC0000</td>
<td>10 2C 00 07 FF EB FE 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB1-&gt;Engine-#1 CTS</td>
<td>00EC000B</td>
<td>11 07 01 FF FF EB FE 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB1100</td>
<td>01 43 4D 4D 4E 53 2A 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB1100</td>
<td>02 43 20 75 30 37 44 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB1100</td>
<td>03 38 33 30 30 30 30 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB1100</td>
<td>04 30 30 30 30 30 30 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB1100</td>
<td>05 30 30 30 30 30 30 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB1100</td>
<td>06 36 42 42 42 42 42 42</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB1100</td>
<td>07 42 2A FF FF FF FF FF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB0000</td>
<td>01 43 4D 4D 4E 53 2A 36</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB0000</td>
<td>02 43 20 75 30 37 44 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB0000</td>
<td>03 38 33 30 30 30 30 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB0000</td>
<td>04 30 30 30 30 30 30 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB0000</td>
<td>05 30 30 30 30 30 30 30</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB0000</td>
<td>06 36 42 42 42 42 42 42</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Engine-#1-&gt;BB1 Data Transfer</td>
<td>18EB0000</td>
<td>07 42 2A FF FF FF FF FF</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA0011</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA000B</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA0011</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA000B</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA0011</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA000B</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA0011</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA000B</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA0011</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA000B</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA0011</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA000B</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA0011</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BB2-&gt;Engine-#1 request</td>
<td>00EA000B</td>
<td>EC FE 00 00 00 00 00 00 00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Data Transfer

- **Connection Initiate: SRC 11**
- **Connection Initiate: SRC 0B**
- **Data Transfer**
- **Legitimate conn requests**
- **No response**
- **Keep Alive CTS**
Making the ecus do bad things

Experiment Set up
## Attacks in a Nutshell

<table>
<thead>
<tr>
<th>PGN</th>
<th>Pre-Conditions</th>
<th>Payload</th>
<th>Impact</th>
<th>Severity</th>
</tr>
</thead>
<tbody>
<tr>
<td>0xFEEE</td>
<td>Ignition ON</td>
<td>Arbitrary value</td>
<td>Set Oil &amp; Coolant Temperature Gauges</td>
<td>Low</td>
</tr>
<tr>
<td>0xFEED</td>
<td>Ignition ON</td>
<td>Arbitrary value</td>
<td>Set Oil Pressure Gauge</td>
<td>Low</td>
</tr>
<tr>
<td>0xFEAE</td>
<td>Ignition ON</td>
<td>Arbitrary value</td>
<td>Set Service Brake Pressure Gauge</td>
<td>Moderate</td>
</tr>
<tr>
<td>0xFEFE</td>
<td>Ignition ON</td>
<td>Arbitrary value</td>
<td>Set Speedometer Gauge</td>
<td>Low</td>
</tr>
<tr>
<td>0x00F4</td>
<td>Ignition ON</td>
<td>Arbitrary value</td>
<td>Set RPM Gauge</td>
<td>--</td>
</tr>
<tr>
<td>0xFEFC</td>
<td>Ignition ON</td>
<td>Arbitrary value</td>
<td>Set Battery Gauge</td>
<td>--</td>
</tr>
<tr>
<td>0xFEED</td>
<td>Ignition ON</td>
<td>Arbitrary value</td>
<td>Set Fuel Level Gauge</td>
<td>--</td>
</tr>
<tr>
<td>0x0</td>
<td>Fuzzing, speed control-mode, pause before engine controller timeout, forward motion</td>
<td>High RPM</td>
<td>Increase Engine RPM, sudden acceleration</td>
<td>High</td>
</tr>
<tr>
<td>0x0</td>
<td>Active acceleration</td>
<td>Low torque</td>
<td>Decrease Engine RPM, idle engine, no further acceleration</td>
<td>High</td>
</tr>
<tr>
<td>0x0</td>
<td>Speed below 30 mph, actively decelerating</td>
<td>0% torque</td>
<td>Disable Engine Brake</td>
<td>High</td>
</tr>
</tbody>
</table>
Let's get Back to business
Steps to HACK, from a High Level
(Created from descriptions in [5])

1. Access to the bus through a compromised Surface
2. Target ECU Reached
3. Target ECU Attacked
4. Data theft

Data theft
Access to the bus through a compromised Surface
The Big Bang!! [4,6,7]

- False RTS and Request Overload are attacks that I introduced in 2016 [6], and turned out to be successful at the US TARDEC event in Michigan last year.
  - In False RTS
    - We kill the engine ECU by making it respond to a flurry of request messages.
  - In Request Overload
    - We attempt to crash the engine ECU firmware with a buffer-overflow attack, only theoretical though.
- Ramping up RPM and disabling engine braking were demonstrated by the group at Umich [7].
Steps to HACK, from a High Level
(Created from descriptions in [5])

- Access to the bus through a compromised Surface
- Target ECU Reached
- Data theft
- Target ECU Controlled
- Attack on Safety/Reliability Achieved
- Physical Theft Achieved
- Data theft
- Attack on Safety/Reliability Achieved
MISSION

ACCOMPLISHED!
References....


