Technical Report

Vehicle Malicious Insider: The Remote Rogue Mechanic

UL Mobility Security
2019

Empowering Trust™
Abstract ........................................................................................................... 4
I. Introduction .................................................................................................... 5
II. Threat landscape ............................................................................................ 6
III. Attack methodology ....................................................................................... 7
   OBD-II port analysis ......................................................................................... 8
   Sniffing on the CAN bus ............................................................................... 8
   Fuzzing ........................................................................................................... 9
   Characterization process .............................................................................. 9
   Hardware/CAN error leading to peculiar behaviors .................................. 10
   Alternative attack .......................................................................................... 10
IV. Summary ....................................................................................................... 11
V. Recommendations .......................................................................................... 11
References .......................................................................................................... 11
Acknowledgments ............................................................................................... 11
Introduction

We were asked to examine the feasibility of a malicious insider, such as a rogue mechanic, being able to install devices in vehicles that would impede responsiveness of first responders, with a focus on law enforcement. The scenario presented was to leverage existing OBD-II devices or install a device of our own. We set several fixed parameters — the rogue mechanic device, the targeted use cases and the testing of the vehicles from multiple OEMs — in order to ensure that our test scenarios demonstrate how a rogue mechanic could hack an LE vehicle remotely by accessing its CAN bus through a hidden device and take control of the vehicle.

The rogue mechanic device

Instead of fashioning a remote rogue mechanic device, we found that there were several suitable, off-the-shelf consumer devices that could be used as an attack platform. We selected the Raspberry Pi-based device, AutoPi, as the rogue mechanic device for our test. This device is readily available and can be bought online at http://autopio.io. The easy availability of the AutoPi and similar devices increases the likelihood and risk of these devices being purchased by rogue mechanics who service and maintain fleet vehicles for LE or other government agencies.

The device needs wireless communications, such as 3G/4G, Bluetooth or Wi-Fi connectivity, to enable remote control by the rogue mechanic. The device must be connected to the vehicle’s network via the CAN or FlexRay bus to execute the attacks and be small enough to be hidden from the driver (see Figure 1). Once the attacker has hidden the device in the vehicle, they can attack the vehicle’s electronics system remotely through Bluetooth, remote keyless entry system (RKES), tire pressure monitoring system (TPMS) and other remote vehicle surfaces. These are all standard characteristics of an AutoPi device.

Using basic automotive electronics knowledge, a rogue mechanic could configure the AutoPi device to execute the attack. It has a short set-up time and requires no previous vehicle-hacking experience, which allows nonautomotive security individuals to succeed with this device and is easy to replicate with minimal instructions. In our test scenario, we assume that the mechanic has complete access to the LE vehicle fleet during repair or maintenance and is able to enter a vehicle without arousing suspicion to hide the rogue mechanic device inside it.

We hid the device on an LE vehicle’s CAN bus via the OBD-II port and subjected the vehicle to penetration testing attacks.

Targeted use cases

We focused our attention on testing specific targeted use cases that all have a potentially detrimental and, in some instances, dangerous impact on the personal safety of LEOs and their ability to perform their duties effectively. The use cases include:

- Preventing the engine from starting or turning off
- Locking and unlocking doors or trunk
- Lowering and raising windows
- Disengaging brakes
- Altering the revolutions per minute (RPM)
- Disengaging the antilock braking system (ABS)

Challenges

In order to conduct the study, we sought access to actual LE vehicles. Finding local LE departments willing to cooperate and potentially provide access to vehicles that were going to be decommissioned and sold was a challenge. We sought help from an OEM, under a nondisclosure agreement, that was willing to cooperate with the study.

We also interviewed LEOs and surveyed their vehicle systems, communications, policies and procedures to learn how plausible a malicious insider scenario may be and if there were additional means besides attacking the vehicles via OBD-II.

Multiple OEM vehicle makes and models

The high cost of vehicles is an impediment for most researchers, since many tests result in the vehicle being rendered undriveable. Under a nondisclosure agreement, we were able to find an OEM willing to lend a recent 2018 LE vehicle with cybersecurity mitigation technology, such as gateways, for two weeks to perform our testing. We reproduced the use cases on a wrecked vehicle — that still had intact electronics — from a second OEM.
Threat landscape

Modern vehicles are substantially computerized and, therefore, potentially vulnerable to attacks. However, while previous research has shown that the internal networks within some modern vehicles are insecure, the associated threat model requiring prior physical access has justifiably been viewed as unrealistic.\(^1\) As aftermarket CAN bus devices become more common, researchers will be challenged to redefine a vehicle’s threat model.

**Entry points of attack**

An attacker only needs to successfully install the device directly on the OBD-II port of the vehicle. As with most LE vehicles in operation, the vehicle is potentially susceptible to multiple entry points of attack:

- Vehicle build
- Vehicle delivery
- Installation of specialized equipment
- Maintenance during operations
- Major or minor repairs experienced during operation
- Vehicle service decommissioning

To the right are examples of situations where LE vehicles might be vulnerable to hacking alongside their associated level of risk. The level of risk is measured by its likelihood of occurrence, e.g., servicing occurs on a regular basis and is predictable, therefore it is rated high.

### How to hide a rogue mechanic device

Once the malicious insider attacker has successfully gained physical access to the vehicle, they can hide the AutoPi (shown in Exhibit A) or rogue mechanic device within the vehicle (Exhibits B and C).

#### In-vehicle cyberattack mitigation systems

We conducted penetration tests on two LE vehicles; the 2018 vehicle was equipped with cyberattack mitigation systems. However, we found that these systems are still vulnerable to a rogue mechanic device.

<table>
<thead>
<tr>
<th>Entry point event</th>
<th>Attacker opportunity</th>
<th>Risk</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vehicle service</td>
<td>The vehicle is turned in for servicing</td>
<td>High</td>
</tr>
<tr>
<td>Vehicle crashed</td>
<td>Contracted tow truck driver has vehicle in their possession</td>
<td>Medium</td>
</tr>
<tr>
<td>Vehicle left unattended</td>
<td>The vehicle is taken home by an LE officer</td>
<td>Low</td>
</tr>
</tbody>
</table>

**Figure 1**

This illustrates how easy it is for attackers to connect the device with an extension cable and hide it within the vehicle.

**Exhibit A:** The AutoPi  
**Exhibit B:** AutoPi connected to OBD-II  
**Exhibit C:** Device and cables hidden

**Attack methodology**

We used the following process to perform cyberattacks on an LE vehicle via remote access through our rogue mechanic device.

1. **Step 1 – Purchase and configuration of a rogue mechanic device**
   - We bought a rogue mechanic device from http://autopi.io and configured it using information that is available online, including instructions and videos. Configuring the device for our needs required only a few hours.

2. **Step 2 – Hide the device in an LE vehicle**
   - After gaining physical access to the vehicle in the simulated repair shop, we hid the rogue mechanic device and cables on the LE vehicle’s CAN bus where it would not be easily discovered, attaching it to the OBD-II port (Figure 1).

3. **Step 3 – Control the vehicle via remote access using our rogue mechanic device**
   - We went to a location without physical access to the vehicle to test a rogue mechanic’s potential ability to remotely take control of the LE vehicle via the wireless access (4G network).

4. **Step 4 – Conduct penetration testing using remote attacks**
   - We conducted several remote attacks to test our ability to control functions of the LE vehicle from a remote location.

5. **Step 5 – Analyze results**
   - Using the rogue mechanic device to conduct attacks from a remote location, we found that we could perform our use cases. For each vehicle, we were able to:
     - Prevent the engine from starting or turning off
     - Lock and unlock doors or trunk
     - Lower and raise the windows
     - Disengage the brakes
     - Alter the RPM
     - Disengage the ABS

   - **Preliminary discoveries**
     - In the process of performing these attacks, we discovered that, via remote access, we were also able to:
       - Interfere with the 911 call functions — such as making an emergency call
       - Activate the wiper blades
       - Falsify the information on the dashboard
       - Switch on the left and right turn signals

We were surprised by how easy it is to bypass the gateway installed in vehicles manufactured as recently as 2018. When we found the gateway, we simply connected directly to the CAN behind the gateway.

First, we analyzed the available connected CAN buses via the OBD-II port. Once the OBD-II port was utilized and the CAN buses identified, we used our rogue mechanic device to:

1. Sniff and capture the series of CAN messages caused by driver actions, e.g., accelerate, lock/unlock the doors. These actions are performed while CAN bus data is being logged to ensure that the recorded CAN messages pertain to the specific driver actions performed.
2. Fuzz the CAN bus by sending random arbitration IDs and payloads in order to trigger vehicle functionality that cannot be initiated directly from a driver’s action.
3. Characterize vehicle function from a sequence of CAN messages.
4. Probe the hardware or a CAN bus error leading to peculiar vehicle behavior.

\(^1\) Quoted and adapted from previous work that the authors conducted while employed by the Federal Bureau of Investigation (FBI) and the Department of Justice (DOJ).
OBD-II port analysis

The standardized OBD-II connector gives direct access to a high speed (HS1) bus via pin 6 for CAN High and pin 14 for CAN Low. The vehicle manufacturer included a second high speed CAN bus (HS2) using pin 3 for CAN High and pin 11 for CAN Low.

We could have used other potential buses — HS3 and medium speed 1 — available behind the gateway module to penetrate into the vehicle system but did not exploit them during this evaluation.

Sniffing on the CAN bus

The goal of this phase was to build up a profile of CAN frames that appear on the CAN bus over time, under various operating conditions. By plotting the values of CAN data associated with a particular arbitration ID over time and correlating the resulting time graphs with other time-stamped “side” data, e.g., instrument cluster observations, brake pedal pressed events, etc., we were able to accurately monitor and interpret most messages. To achieve this, the tester plugged a sniffing device into the OBD-II port and listened to the CAN traffic (Figure 2).

The “cansniffer” tool in the package “can-utils” helps the evaluator to observe the fluctuation of the CAN frames by highlighting the changes (Figure 3). If there is a high amount of traffic on the CAN bus, the evaluator can quickly identify the CAN message resulting from a driver action.

Fuzzing

For purposes of triggering undiscovered functionalities, we applied fuzzing. Fuzzing is the sending arbitrary CAN messages in order to activate random functionalities. Once the expected functionality, e.g., doors unlock, is observed, the fuzzing sequence of CAN messages is sent onto the characterization phase (Figure 4). We used the “Caring Caribou” tool during this phase for fuzzing and capturing the CAN sequences.

Characterization process

The goal of this process is to identify the CAN frame associated with a specific action and functionality of the vehicle. This process is implemented by replaying the exact sequence of frames captured during the sniffing phase or created during the fuzzing phase. If an expected action does not occur, the process stops and is considered a failure and the CAN message sequence is discarded. If functionality was observed, the sequence is split in half, and the process is reapplied to each subsequence until one CAN message remains (Figure 5). The final message is the identifier of the observed action and concludes the process.
Hardware/CAN error leading to peculiar behaviors
In this phase, the focus is on the hardware and CAN bus related errors. A hardware error is caused by a wrong connection of the CAN bus pinout and may result in a denial of service (DoS) of some electronic control units (ECUs). By connecting CAN High to ground on the test vehicle, we observed an odd behavior affecting most of the ECUs onboard. During the test, we noticed that the dashboard and the in-vehicle infotainment system stopped working, some error indicators started blinking, and we were not able to start the engine. While testing on a running vehicle, the engine also stopped working after a few seconds.

This attack can be executed remotely by using an AutoPi or Raspberry Pi with a relay. The relay simply connects the ground to CAN High of the OBD-II port when requested by the Raspberry Pi through general-purpose input/output (GPIO) pins. The Raspberry Pi with an embedded 4G modem provides the attacker with the CAN High of the OBD-II port when requested by the Raspberry Pi — or similar — device with a garage door remote controller — and a LEO uses sniffers to detect unexpected Wi-Fi, Bluetooth or other 802 protocols coming off the vehicle, we investigated if the relevant procedures are updated for the repair or maintenance of vehicles — detecting malicious hidden remote devices — and a LEO uses sniffers to detect unexpected Wi-Fi, Bluetooth or other OBD2 protocol coming off the vehicle, we investigated an alternative remote attack with a garage door opener that would still remain undocumented. This approach simply replaces the Raspberry Pi — or similar — device with a garage door remote controller. While this significantly reduces the attack distance, the garage door opener remote receiver will be connected to the relay and can trigger the connection of ground to CAN High. The attacker can now disable the car engine with a simple push of a button (Figure 6).

Summary
We conducted this research project on a typical LE vehicle using commonly available hardware and a skill set compatible with a mechanic’s level of vehicular cyberknowledge. Once we selected our rogue mechanic device, we installed various demonstrations of its ability to interfere with the mission of the vehicle onto the device.

It took a week to find the reported vulnerabilities, exploit and reproduce them to enable us to execute these attacks remotely.

- Executed remotely, critical functions such as stopping the engine and locking a person in or out of the car can have serious safety impacts.
- Noncritical functions such as lights, sounds, alarms and dashboard dials should not be discounted. When used in specific conditions, even these noncritical functions can impact safety and prevent the user from operating the vehicle properly, e.g., turning lights off at night, distracting the driver and directing unwanted attention to the vehicle by activating sounds/alarms.

It is important to note the scale of vulnerability to rogue mechanistic devices since our methodology enabled us to reproduce these same attacks on two vehicles, one of which was a 2018 model that remained vulnerable despite the sophisticated gateway protection built in.

Also, the risk is greater than we expected, especially if the relevant procedures are not updated for the repair or maintenance of vehicles, such as detecting malicious hidden remote devices. Detection, mitigation and prevention of such attacks should be given a greater priority. Even with sniffers that can detect unexpected Wi-Fi, Bluetooth or other OBD2 protocols coming off the vehicle, substituting the garage door opener remote would still remain undetected.

This was done in coordination with the U.S. Department of Transportation Volpe Center and the Massachusetts Institute of Technology (MIT) World Wide Web Consortium (W3C), and all results were responsibly disclosed to the OEM.

Recommendations
Beyond the reported vulnerabilities, we would like to emphasize the importance of updating the training, policies and procedures for LE and LEOS to not only learn about the risks but also reduce vulnerability to a rogue device with remote hacking capabilities or to a compromised telematics unit. This could be a quick first step in mitigating the risk and is fully under LE control.

Telematics devices are now common in vehicles. Once hacked, they can become a backdoor to access vehicles’ potentially producing the same risks on a wider scale since physical access is not required. This highlights the importance of telematics device security conformance, to ensure those units come with baseline cybersecurity mitigation, preventing rapid escalation to an entire fleet without physical intervention — no rogue device and no insider necessary.

References
2. UL Paper ON THE SECURITY OF IN-VEHICLE SOFTWARE (a vision on security for road safety) 2016
5. Reflection and projection: year of automotive exploitation, Matthew Carpenter, GRIMM (escar 2018 presentation)

Acknowledgments
UL would like to thank the Kevin Harnett, Graham Watson and Daniel Chin from the U.S. Department of Transportation Volpe Center (Volpe Center) for the extensive and detailed technical support as well as providing law enforcement vehicle hack use cases; Ted Guild from MIT and W3C for the proactive technical support and information shared, resulting in a sound technical white paper for the automotive cybersecurity sector.

UL and Volpe Center would like to also thank Michigan State Police and California Highway Patrol for contributing with information on law enforcement officers and their procedures for maintaining law enforcement vehicles; as well as GRIMM for sharing information and tools.