# Truck Hacking: An Experimental Analysis of the SAE J1939 Standard

Yelizaveta Burakova, Bill Hass, Leif Millar, and André Weimerskirch

*The University of Michigan*
{ybura, billhass, ltmillar, andrewmk}@umich.edu

## Abstract

Consumer vehicles have been proven to be insecure; the addition of electronics to monitor and control vehicle functions have added complexity resulting in safety critical vulnerabilities. Heavy commercial vehicles have also begun adding electronic control systems similar to consumer vehicles. We show how the openness of the SAE J1939 standard used across all US heavy vehicle industries gives easy access for safety-critical attacks and that these attacks aren't limited to one specific make, model, or industry.

We test our attacks on a 2006 Class-8 semi tractor and 2001 school bus. With these two vehicles, we demonstrate how simple it is to replicate the kinds of attacks used on consumer vehicles and that it is possible to use the same attack on other vehicles that use the SAE J1939 standard. We show safety critical attacks that include the ability to accelerate a truck in motion, disable the driver's ability to accelerate, and disable the vehicle's engine brake. We conclude with a discussion for possibilities of additional attacks and potential remote attack vectors.

## 1 Introduction

Although academic research has shown vulnerabilities in consumer automobiles as early as 2010 [12], the general public has only recently been made aware of such vulnerabilities through media reports in 2015. Now, both industry and consumers are paying more attention to the security of their own cars. However, not much has been said or done in public about the heavy vehicle industry.

All modern heavy duty trucks and buses in the United States use the SAE J1939 Standard (J1939) for their internal networks. Motivation for J1939 stems primarily from a desire to electronically control drivetrain components of a vehicle, which is typically the core component of a concerted effort to maximize fuel efficiency. Because so many different organizations are involved in the building of heavy vehicles, a standard was needed to minimize engineering effort and the complications of integrating systems. J1939 is not the first standard for heavy vehicles, but rather is the successor of the SAE J1587 and SAE J1708 standards. While standardizing these communications has proven crucial in allowing various suppliers and manufacturers to work together and cut costs, it also means that all heavy vehicles currently on the road in the US, from semi tractor-trailers to garbage trucks and cement mixers to buses, utilize the same communication protocol on their internal networks.

Heavy vehicles play an important role in our nation's economy. In 2002, the value of freight shipments was $11 trillion, of which trucks hauled 64% [3], and there were over 6.5 million heavy trucks in fleets in the United States in 2013 [20]. While physically different from consumer automobiles in many ways, heavy vehicles are similar internally in that they are composed of a distributed system of electronic control units (ECUs) that communicate over a CAN-based network.

Additionally, as with cars, the trend for heavy vehicles is to move away from purely mechanical systems towards more electronically controlled ones thanks to the promise of fuel efficiency, driver comfort, and safety. For example, heavy trucks are mandated in the US to employ electronically controlled anti-lock brake, anti-slip regulation, and active rollover protection systems. Furthermore, active lane keep assist, collision avoidance, and adaptive cruise control systems are available, and a couple companies are even touting their autonomous trucking capabilities [2]. These systems bring electronic control to safety critical components which necessitates a focus on robustness, reliability, and security.

Heavy trucks are typically part of a larger fleet of vehicles which are monitored over long distances using fleet management systems (FMS). The FMS standard is a worldwide standard developed in 2002 which combines satellite and cellular communication to provide information about vehicle location and status. Some status messages defined by FMS include vehicle and driver identification as well as the state of the electronic engine controller, cruise control module, and fuel levels [6]. The FMS standard enables third party systems to integrate with the API across manufacturers which is a nice benefit for fleet owners, but as we've seen in the consumer segment, third party devices don't always prioritize security [4]. In fact, a blogpost in March, 2016 [16] revealed over 1,500 third party fleet management systems with connection to the vehicle's internal network whose Telnet port was wide open. This indicates a viable long-range attack surface on heavy vehicles.

In this paper, we focus on what an adversary can accomplish physically connected to the internal network, and analyze the impact of insecure ECUs in heavy vehicles that use the J1939 standard. This is a topic of concern to many agencies and parties, including various government agencies, heavy vehicle manufacturers, the freight industry, and of course the general public. Our goal was to experimentally analyze the security of the J1939 protocol and determine whether heavy vehicles are more or less secure than consumer automobiles.

### 1.1 Contributions

We are the first to experimentally demonstrate that heavy vehicle networks are vulnerable to attacks similar to those implemented on consumer car networks. A strength of our work is that by focusing on the J1939 standard, our results can be applied to all vehicles using that standard. We summarize our contributions as follows:

1. Using publicly available information of a common, standardized vehicle network, we show that it is possible to mount safety critical attacks.

2. We demonstrate that safety critical systems are vulnerable to an adversary with access to the vehicle's internal network through the diagnostics port.

3. We verify that attacks developed on a semi-tractor also work on a bus, providing evidence that all heavy vehicles with the J1939 standard are affected.

4. We provide an outlook on further attacks that are highly likely to be successful and give recommendations for future areas of research.

### 1.2 Overview

We first cover related work, provide terms from the heavy vehicle industry, and present a technical overview of CAN and J1939 in Section 2. Then, in Section 3 we describe our threat model, and Section 4 covers our methodology. We present our results in Section 5, followed by a discussion of the individual attacks. We end with future work in Section 6 and give our conclusion in Section 7.

## 2 Background

### 2.1 Related Work

Recent attention has been paid to consumer automobile security thanks to several prominent demonstrations of vehicle vulnerabilities on an unnamed car in 2010 [12], a Toyota Prius and Ford Escape in 2014 [14], and a Jeep Grand Cherokee in 2015 [15]. Notably, exploits were first developed and reported using a physical connection to the vehicle's internal CAN network through the car's on-board diagnostics (OBD-II) port, as is the case in 2010 and 2014. Follow up research to the 2010 report by some of the same authors in 2011 [13] demonstrated a wide array of remote exploits thanks in part to buffer overflows at the remote interfaces and a general lack of security on behalf of the vehicle system engineers. Similarly, the 2015 vulnerabilities were extensions of the authors' prior research in 2014, which showed they were able to remotely control the vehicle across the country over a cellular network. In a similar manner, we wish to first explore the capabilities of an adversary with a physical connection to the heavy vehicle's internal network via the OBD port.

### 2.1.1 Terminology

Before we begin, we will define a few terms from the heavy vehicle industry.

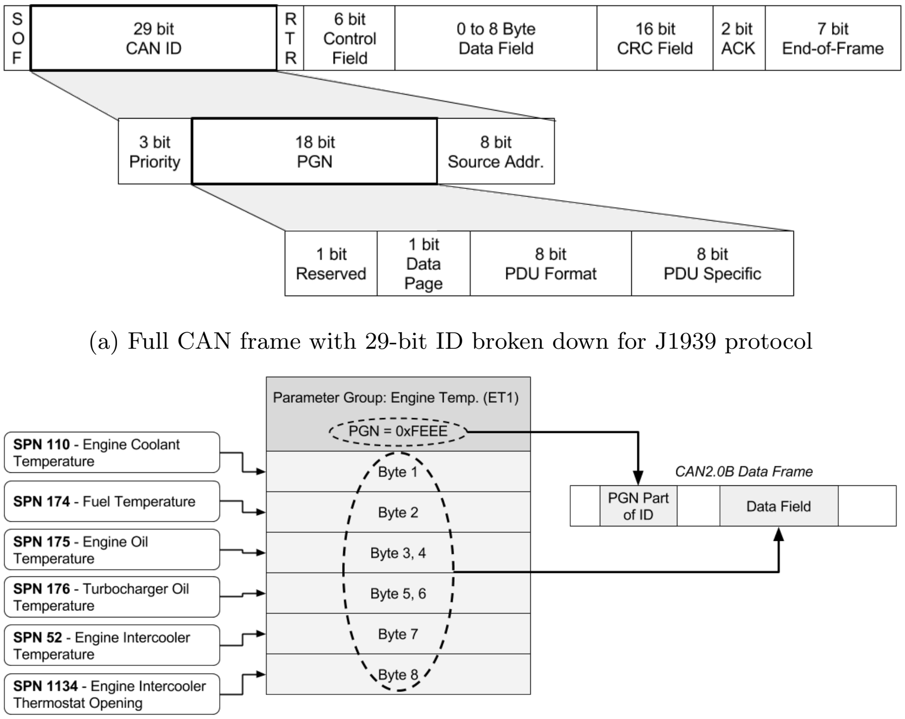
- Control Area Network (CAN) Bus - The standard method of communication for electronic modules in automobiles. The most common CAN bus configuration is a twisted pair of wires which are connected to each module in the vehicle that needs to send or receive data to other modules.
- CAN Message / Frame - A complete CAN2.0B packet that contains various headers, an 8 byte data payload, which itself is composed of several signals, and various footers as specified by ISO 11898.
- CAN Signal - An individual piece of information which is contained in the data payload of a CAN message. It can be one or many bits in length.
- Electronic Control Unit (ECU) - One of the numerous electronic modules that collectively constitute the distributed control system of the vehicle.
- Foundation / Service Brake - The physical braking mechanism at the wheel ends or axle which uses friction to cause the truck to decelerate.
- Engine Brake - On heavy vehicles, an important aspect of braking is the use of the engine to slow down the vehicle. This is especially important when going downhill, as using foundation brakes for this task would quickly cause the brakes to overheat. When the brakes overheat, not only do they wear substantially faster, but more importantly a partial or total loss of braking results (Also known as brake fading).
- Original Equipment Manufacturer (OEM) - The manufacturer of the end product. The relationship between OEM and suppliers is complex. Examples from automotive include Ford, Kenworth, Mercedes-Benz, and General Motors.
- Supplier - Companies that supply OEMs with various parts or services. Suppliers may sell raw materials, individual parts, or an entire system they have designed and developed.
- Gross Vehicle Weight Rating (GVWR) - The GVWR is a commercial vehicle classification used in the United States based on the maximum loaded weight, ranging from Class-1 to Class-8. A Class-8 truck is classified as a truck whose GVWR exceeds 33,000 lbs and requires a Class-A commercial driver's license to operate.
- Commercial Driver's License (CDL) - The driver's license that is required to operate heavy vehicles.

## 2.2 CAN Protocol

The CAN standard was first developed at Robert Bosch GmbH in 1983 for the purpose of networking various electronics modules without the need for a dedicated computer in automobiles.

The original standard was specified in ISO 11898, which defines the physical and data link layers of the CAN protocol. The most common physical layer describes a two wire differential bus, one high and one low voltage wire, terminated at both ends by a 120-ohm resistor and connected at each node to a transceiver. The wires are typically twisted together, which gives the bus a high tolerance for interference as both wires will experience the same level of distortion, leaving the voltage difference between the two wires largely unaffected. The data link layer usually consists of a peripheral micro-controller that implements arbitration and packet framing in hardware which is controlled by the host processor.

Data is sent using frames which are also described in the specification. These packets include a priority identifier, data length, data payload, error detection bits, and an ACK bit. The priority ID and data bytes are the primary components of the frame which can be controlled by the host controller. The original specification only allows for up to 8 bytes of data per frame and an 11 bit ID. Since the release of CAN2.0B in 1991 [5], an extended frame format was defined that allows for a 29 bit ID. The extended frame format can be seen at the top of Figure 1a. Recently, CAN-FD was defined which inter-operates with CAN2.0B and allows up to 64 bytes of data per frame. Larger frames will one day enable the ability to include message authentication codes with the data, but it will be at least several years until CAN-FD is widely adopted.



(a) Full CAN frame with 29-bit ID broken down for J1939 protocol

(b) Example SPN layout for the "Engine Temperature" PGN

Figure 1: Diagrams that describe the J1939 identifier and data fields [21]

The messages sent on the bus depend greatly on the make and model of the vehicle. For example, the messages sent on consumer vehicle networks are proprietary to the OEM that designed that particular vehicle and kept secret. Often times the message formats will change, not only from model to model within an OEM, but also from year to year. For this reason, deciphering consumer vehicle network traffic involves the tedious process of reverse engineering any messages observed on the bus to determine their function.

### 2.3 J1939 Protocol

The SAE J1939 standard describes the vehicle bus used in heavy vehicles such as buses and semi-trucks. J1939 defines five of the seven layers of the OSI model, with CAN2.0B being used for the physical and data-link layers [11]. The 29 bit ID encodes a 3 bit priority, 18 bit Parameter Group Number (PGN), and 8 bit source address, as seen in Figure 1a. The PGN is a message identifier that specifies which signals are contained in the data payload. In most cases, the data sent for a given PGN consists of 8 bytes, but if a PGN requires more data, a set of transport protocol messages can be used to send up to 1,785 bytes for a given PGN. The data bytes are grouped using Suspect Parameter Numbers (SPNs) which define what the data means. An example of how PGNs and SPNs relate is shown in Figure 1b .

The J1939 standard is open and used across many industries that employ diesel engine vehicles, such as bus and train transportation, construction, agriculture, forestry, mining, and the military [18]. This is a very different model from OEM's proprietary application level CAN protocols which change across make, model, and model year and are heavily guarded secrets within the OEMs.

The openness of J1939 allows anyone who can make a payment receive technical details about standard PGNs, allowing a potential adversary to easily gain the knowledge necessary to attack safety critical components. PGNs 0x00FF00 through 0x00FFFF are reserved for proprietary use, but every attack described in this paper uses standard PGNs that are the same for different vehicles. Interesting messages that are part of the standard include brake, engine, transmission, and cruise control. These messages could lead to similar vulnerabilities as seen in the consumer automotive sector. A recent NSF Grant for a J1939 security testbed in Jan. 1, 2016 [17] suggests government and heavy trucking industry are starting to analyze the security implications of the standard.

We provide supporting evidence in Section 4 to say that attacks developed for the J1939 protocol can potentially be used across a wide variety of vehicles. While there may still be slight implementation differences from supplier to supplier and some vehicles won't have certain features implemented, we hypothesize that most basic attacks will work for any vehicle that uses this standard.

## 3 Threat Model

In contrast to the consumer car segment, there is more incentive for an adversary to attack the heavy vehicle industry due to the size of the vehicles and the variety of goods they carry. The biggest motivation for an attacker is usually financial, and the freight transportation industry contributes nearly 10 percent of the United States GDP [19]. When you add other potentially affected industries such as construction and agriculture, that number only goes up. So our adversary can be anyone who could stand to make a profit off manipulating the vehicles, be it from hijacking their goods, adversely manipulating a competition's fleet, extorting fleet owners and drivers, or selling their tools and services on the black market. Another type of adversary we consider is one who wishes to cause the most harm and damage as possible, such as a terrorist or nation state.

We assume that our adversary has the ability to transmit arbitrary messages on the vehicle's J1939 bus. This is most readily accomplished with physical access to the vehicle through the OBD port. Given that CAN hardware is common and easy to obtain, we can assume that someone with physical access also has access to the correct hardware. With a small enough dongle, an adversary with brief physical access could gain persistent access to the vehicle's bus since the OBD port is commonly located out of sight under the dash and is not part of the CDL pre-trip inspection. Alternatively, a more sophisticated attacker could inject malware into other ECUs on the network leaving no external trace anything is amiss. A common argument to the physical assumption is that an adversary can already cut the brakes or loosen some nuts with physical access, but we think that argument takes a nearsighted view of the threat model. With a foothold in the car's internal network, vehicle status or external events can be monitored to trigger various behaviors, and the software can avoid forensic analysis by erasing itself.

While our research focused on direct physical access through the OBD port, there are a couple other attack vectors common to heavy vehicles that are worth mentioning. First, is the trailer in a semi tractor-trailer configuration. Typically, there is a bridge between the J1939 network and the trailer network as can be seen in Figure 2, which if exploited could be a viable attack vector. Second, are fleet management systems (FMS) which can be found in many commercial vehicles. They are used by fleet owners to wirelessly track and log various statistics of a given fleet, from GPS, speed, and brake usage to notifications in the case of an accident or airbag deployment. FMS should be read-only from the J1939 network, but they are complex systems which could provide a wireless attack vector.

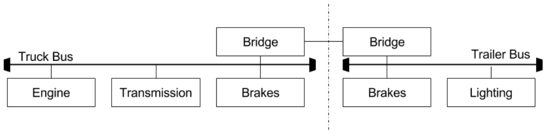


Figure 2: Tractor-trailer network configuration

It's reasonable to assume that given a physical exploit, a remote exploit will soon follow. Many cases of remote attacks in the consumer vehicle space derive from physical exploits [13, 15, 4]. There has even been at least one documented vulnerability of hardware that is specifically used in trucks today [16]. This particular security flaw was due to an open Telnet port accessible without authentication on the public IP space. It's not news that embedded devices have open ports [1], so we can reasonably expect that this is not the only instance of a telematics unit with an exploitable default configuration. Like FMS, telematics units are used widely in the trucking industry to track vehicles and many are connected to CAN to report diagnostics back to the base.

It takes more sophisticated knowledge of the protocol to understand what messages to modify or send in order to get some kind of desired behavior, but the fact that J1939 is an open standard means that anyone with enough technical skill can craft attacks before even connecting to a vehicle. This differs from consumer vehicles, where the proprietary CAN standard requires significant reverse engineering and narrows the possible attackers to people with access to the specific vehicle's protocol version. On top of that, an adversary attacking consumer vehicles would need to modify their attack for different models of cars, whereas an attack on heavy vehicles can remain unmodified and work on a variety of vehicles, from trucks to buses.

## 4 Methodology

All of our experiments specifically exploit messages present in J1939 without the use of backdoors or software vulnerabilities. We focused primarily on a Class-8 2006 model year semi tractor, and we had limited access to a 2001 model year school bus. Since both use the J1939 standard, our hypothesis was that the exact same attacks should work on each vehicle, and on any other vehicle that uses the standard. All of our attacks were first developed on the semi truck while idling in neutral with the parking brake applied, then tested on a closed track with a certified CDL driver while the truck was driven under normal conditions. Then, for each of our successful attacks, we also tested them unmodified against the school bus while parked and idling to check our hypothesis.

Our test setup is pictured in Figure 3. We connected a laptop to the Vector and PEAK tools described in Section 3.1 which were connected to the J1939 OBD port via a Y-cable in order to collect and transmit data. The CANoe application proved to be the most useful for packet snooping thanks to a publicly available Communication Database (DBC) file which enabled messages to be parsed and inspected in real-time based on the J1939 standard. We were quickly able to identify PGNs that controlled various functions of the truck as we manipulated them from within the cabin. The PEAK tool on the other hand was much easier to use for packet injection thanks to the simple, intuitive user interface. By injecting packets with the PEAK tool and snooping with the Vector tool, we were able to easily verify our injected messages.

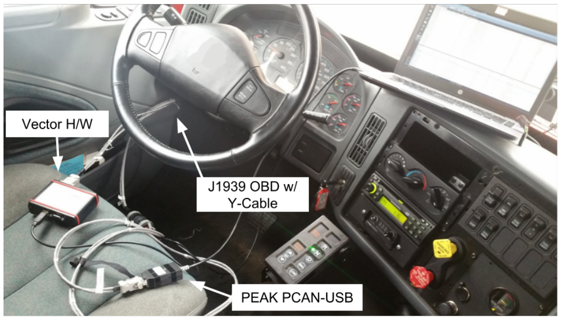


Figure 3: Example setup for experimentation.

In order to gather data while the truck was under normal driving conditions, we used public roads due to the limited availability of testing facilities. For these tests, we put our tools into listen-only mode so no data packets could be injected onto the CAN bus. These sessions were aimed at gathering data only to be stored for later analysis. Once we had collected that data, we went back to the parked and idle setup to replay the sequence of messages we recorded. Using this method, we discovered a reaction by the powertrain and other electronic systems.

Finally, we needed to test our attacks while the truck was in motion in order to realize the true impact of our findings. For this, we used MCity - The University of Michigan Mobility Transformation Center's 32-acre closed-course test track. All of the experiments were carried out at low speed on a slight uphill gradient with large roundabouts at either end so that the vehicle could be put in neutral and coast to a stop in case of an emergency.

### 4.1 Tools

We used a variety of diagnostic tools to analyze the bus traffic and inject our own packets. All of these tools are available to purchase and are designed for the typical mechanic or engineer to use.

### Vector CANoe

The Vector CANoe is a high-cost, industry-standard CAN analysis and simulation software tool. It uses a hardware interface device which allows the user to interface via USB with various CAN protocols such as the J1939 standard or other proprietary protocols that use CAN. The software application allows the user to snoop and record CAN traffic, inject and replay CAN messages, and write scripts for efficiently describing complex interactions with modules on the bus. It uses a proprietary C-like event-driven scripting language called CAPL. The application layer CAN protocols can be described in the DBC which can then be imported to CANoe for real-time identification of the CAN messages and signals on the bus, as well as for easier creation of scripts.

We used CANoe for data gathering, and our more sophisticated attacks were implemented in CAPL and executed using the CANoe system.

### PEAK USB-PCAN

The PEAK tool is a low-cost alternative to the Vector tool. It is similar in that there is a software application and hardware device that interfaces between CAN and USB. Its free application, PCAN-view, has fewer features, but it is also more intuitive and works very well for simple tasks. PEAK provides a set of libraries for interfacing with PCAN APIs which we used with Python to create a fuzzing script [7] which enumerates all possible data and ID fields. Prior research has had success with such fuzzing methods, but we have not. Additionally, the lack of database files which describe the identification and purpose of various messages makes for a much less straightforward experience with the PEAK tool.

The PEAK tool was used for data gathering, especially while using CANoe, and our earlier, simple attacks were implemented with PCAN-view and the PEAK tool.

### Diagnostics Tool

The generic diagnostics tool we used had capability to retrieve diagnostic codes and general status information about the semi truck it is connected to. It uses a standard J1939 OBD connector and is used primarily by mechanics for ECU diagnostics. We found that the software also has the option to update ECUs and cut off each of the six engine cylinders, one cylinder at a time, and we found a freely available software module from the ABS manufacturer's website for the generic diagnostics tool which enabled us to actuate the ABS valves on the wheel ends.

We used the diagnostic tool to setup diagnostics sessions and perform various diagnostics tasks while logging the messages with CANoe via the Y-Cable.

## 5 Results

We find that an adversary with network access can control safety critical systems of heavy vehicles using the SAE J1939 protocol. The specific message PGNs for each behavior are listed in Table 1. The instrument cluster attack required different PGNs to control each gauge, whereas we found we could get a lot of control from the truck by changing the SPNs within the torque/speed control 1 (TSC1) message. Not only is the TSC1 message a public PGN, but the documentation provides a detailed overview of the purpose of the different SPNs associated with it, making this powerful attack relatively easy to replicate.

### 5.1 Instrument Cluster

By spoofing the status messages that originate in various ECUs of the truck, we were able to control all gauges on the instrument cluster, which include oil temperature, oil pressure, coolant temperature, RPM, speed, fuel level, battery voltage, and air pressure of the foundation brake system. [8] The temperature, oil pressure, air pressure, fuel level, and battery voltage gauges all cause an alarm to sound accompanied by a bright red light at each gauge when the reading goes above or below a certain point, and in all cases we were able to make the gauges go beyond said thresholds, causing the alarm to sound. Our control was precise, we could make the gauges point to the value of our choosing - even while the truck was in motion. This attack did not work on the 2001 model year bus.

For the safety critical rating, we deemed fuel level, battery, and RPM as non-critical because the driver has other indicators which can be relied on such as odometer and engine noise. For speedometer, oil pressure, and temperature, we gave a low rating because they are harder to verify by the driver and could put the driver or other vehicles on the road in a dangerous situation if the underlying system fails. The service brake pressure has moderate severity because the driver has no other indicator and needs to know the air pressure to avoid activating the emergency brake system which would lock up all of the wheels.

### 5.2 Powertrain

The powertrain attack was somewhat harder to discover, but just as straightforward to implement. By replaying a sequence of captured messages that we recorded during normal

| **Attack Messages** | | | |
|---|---|---|---|
| **Behavior** | **PGN** | **Acronym** | **Safety Critical** |
| Set Oil & Coolant Temperature Gauges | 0xFEEE | ET1 | Low |
| Set Oil Pressure Gauge | 0xFEEF | EFl/P1 | Low |
| Set Service Brake Pressure Gauge | 0xFEAE | AIR1 | Moderate |
| Set Speedometer Gauge | 0xFEF1 | CCVS | Low |
| Set RPM Gauge | 0xF004 | EEC1 | — |
| Set Battery Gauge | 0xFEF7 | VEP1 | — |
| Set Fuel Level Gauge | 0xFEFC | DD1 | — |
| Increase Engine RPM | 0x0 | TSC1 | High |
| Decrease Engine RPM | 0x0 | TSC1 | High |
| Disable Engine Brake | 0x0 | TSC1 | High |

Table 1: List of PGNs and attack severity

driving conditions, we observed the engine RPM increase in a specific pattern. There were eleven unique PGNs that repeated at various intervals related to the engine within that sequence, leading us to believe it was a complex interaction of messages causing the RPM to spike. However, by shrinking the sequence and probing with specific messages we were surprised to find that just one PGN, TSC1, enabled powertrain control. [10]

According to the specification, the TSC1 message is used for engine control and retarding by various ECUs, such as accelerator pedal, cruise control, or power take-off governor. It is received by the engine or retarder and commands a given RPM value if speed control mode is specified or percentage of torque output if torque control mode is specified. We ran further experiments while idle and found that by injecting the TSC1 message with a specified RPM in speed control mode we could physically command the engine's RPM to that specific value. Torque control mode behaved similarly, but a specific RPM value was harder to hit.

After seven seconds of continuous control, the engine wouldn't obey our messages and returned to idle, but we quickly overcame this limitation by pausing for 40ms before the seven second timeout expired, then repeating. With just a 40ms pause, we could indefinitely hold the RPM to a given value. We did not try to blow the engine. This attack also worked while idle without modification on the 2001 model year school bus. [9]

We developed a set of messages that exercised edge cases of various signals within the TSC1 message for testing on MCity while driving. By commanding the RPM to any value in speed control mode, we were able to override the driver's input. If a high RPM value was issued (e.g. 3000 RPM) we caused the truck to very quickly accelerate to max out the speed provided by the currently selected gear. This attack didn't work while the truck was completely stopped or rolling backwards down an incline, but it did work if the truck was rolling forward. By commanding the torque percentage to a low value (e.g. 0%) in torque control mode, we were able to override the driver's input to the accelerator as he was actively accelerating causing the truck's engine to idle; effectively cutting off the engine. This even prevented the driver from accelerating from a standstill. The TSC1 message also had other side effects explained in the engine brake section below.

To summarize, we were able to override the driver's input to the accelerator pedal and simultaneously cause either direct acceleration or remove the ability to provide torque to the wheels while the truck was in motion. We gave both a high safety critical rating because the CAN message directly influences the vehicle's engine without operator control; the best the driver can do is brake and pull over which isn't possible in all situations.

### 5.3 Engine Brake

In addition to overriding accelerator input, the TSC1 message can be configured to disable the truck's ability to use engine braking at speeds below 30 miles per hour. When we commanded the torque percentage to 0% while the truck's engine brakes were actively decelerating the vehicle, the truck's engine brakes let off completely and deceleration ceased. This is dangerous because engine braking is heavily relied on by heavy vehicles to save the foundation brakes from overheating, which can lead to brake fade or total loss of braking power. Even with a 30 mph threshold, we still give this a high safety critical rating because on long and steep winding roads a fully-loaded truck has the potential for catastrophic brake loss.

## 6 Future Work

We were not able to turn every discovery into an attack. Here we list a few possibilities for further research.

### Other Messages of Interest

We found additional J1939 messages in documentation that seem safety critical on their own, similar to the TSC1 message, which suggest network control of transmission and brakes.

### Diagnostic Tool Spoofing

We found that the diagnostic tool can disable the engine cylinders and actuate the ABS valves through the OBD port, but these messages are sent over the J1708/J1587 network, so future research will involve determining how the request is made, and what kind (if any) authentication the diagnostics tool uses.

### Engine Braking

Additional research is required to investigate how engine braking might be disabled at speeds greater than 30 mph.

### Remote Extension

Further research is needed to determine the number of telematics unit models available, the security measures used by different models, and the utilization rates of attacks on models deemed insecure. Additionally, for those models that are secure, the level of sophistication that would be required to bypass their security enough to write malicious packets on the CAN bus or gather sensitive information.

### Truck Trailers

We focused on the truck tractor in our research, but future work could also involve looking into the bus of the trailer as well. There are theoretical possibilities à la Stuxnet of how a malicious trailer could affect multiple tractors. Future work is required to investigate how feasible this would be.

### Other Industries

The scope of our experiments was limited to two vehicles from similar industries, but we would be interested in applying the same experiments to trucks and buses from different manufacturers as well as vehicles from other industries. We have shown that an unmodified version of the powertrain attack from a 2006 model year truck worked on a 2001 model year bus, and we believe this is only the tip of the iceberg. It would not be surprising to us to see that diesel engine vehicles from agriculture, forestry, construction, locomotive, marine, and military industries are affected by the same or similar attacks.

## 7 Conclusion

There has been a lot of prior work done to analyze the security of consumer automobiles. However, we are the first to show that some of the same vulnerabilities are very much present in the heavy vehicle industry. Similar to cars, semi trucks are designed to protect against safety failures, but the idea of an active attacker does not go into the design of the safety mechanisms.

By using publicly available information of a popular vehicle network standard, we have developed concrete examples of attacks that affect safety-critical systems of a semi-truck and a bus which use the same standard. Since our attacks focus on the J1939 protocol, not the software on the truck itself, the attacks aren't limited to just semi trucks, and they can be implemented on a wide scale. The impact of protocol vulnerabilities that we showed stretches across many industries in the US and abroad. We only needed one message to implement a series of safety-critical attacks, and while we required physical access to the internal network, it is reasonable to assume that a remote extension to our attacks is feasible given how similar the vulnerabilities are to consumer vehicles and the complexity of fleet management systems already widely employed.

It is imperative that the trucking industry begins to take software security more seriously. Our attacks took us less than two months to implement and did not require any proprietary PGNs. It is reasonable to assume that with more time an adversary could create an even more sophisticated attack, one that could be implemented remotely. With Bluetooth, cellular, and WiFi, modern trucks are becoming much more connected to the outside world, which present new attack vectors. Our hope is the heavy vehicle industry begins to include the possibility of an active adversary in the design of their safety features.

## 8 Acknowledgements

We would like to thank the University of Michigan Transportation Research Institute (UMTRI) for providing access to the truck and bus, as well as the Michigan Mobility Transformation Center (MTC) for allowing us to test on MCity between construction crews. Specifically we thank Steve Stachowski and Russ Bielawski for discussions over coffee and help getting set up, Ken Winzeler for being our CDL certified driver, and Regina Ferguson and Lindsey Scheer for helping with logistics.

## References

[1] Botnet, C. Port scanning /0 using insecure embedded devices. http://internetcensus2012.bitbucket.org/paper.html, aug 2012. Accessed: 2016-03-17.

[2] Davies, A. The world's first self-driving semi-truck hits the road. http://www.wired.com/2015/05/worlds-first-self-driving-semi-truck-hits-road/, may 2015. Accessed: 2016-02-01.

[3] Felix Ammah-Tagoe, U.S. Department of Transportation, O. o. T. A. Freight shipments in america. http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/freight_shipments_in_america/pdf/entire.pdf, 2004.

[4] Foster, I., Prudhomme, A., Koscher, K., and Savage, S. Fast and vulnerable: A story of telematic failures. In *9th USENIX Workshop on Offensive Technologies (WOOT 15)* (Washington, D.C., Aug. 2015), USENIX Association.

[5] GmbH, R. B. Can specification version 2.0. http://www.kvaser.com/software/7330130980914/V1/can2spec.pdf, 1991.

[6] Group, H. . B. W. Fms-standard description, version 03. http://www.fms-standard.com/Truck/down_load/fms_document_ver03_vers_14_09_2012.pdf, sept 2012. Accessed: 2016-04-18.

[7] Hass, B., and Burakova, Y. Pyfuzz can. https://github.com/bhass1/pyfuzz_can.

[8] Hass, B., Burakova, Y., and Millar, L. Video of attack on instrument cluster. https://youtu.be/HZmNKkdlYmM.

[9] Hass, B., Burakova, Y., and Millar, L. Video of attack on schoolbus. https://youtu.be/nZDiCRBdSF0.

[10] Hass, B., Burakova, Y., and Millar, L. Video of powertrain attack. https://youtu.be/ZfXkDPU3WMQ.

[11] Junger, M. Introduction to j1939. http://vector.com/portal/medien/cmc/application_notes/AN-ION-1-3100_Introduction_to_J1939.pdf, 2010.

[12] Karl Koscher, Alexei Czeskis, e. a. Experimental security analysis of a modern automobile. *IEEE Symposium on Security and Privacy* (2010).

[13] Karl Koscher, Stephen Checkoway, e. a. Comprehensive experimental analyses of automotive attack surfaces. *Usenix Security* (2011).

[14] Miller, C., and Valasek, C. Adventures in automotive networks and control units. http://www.ioactive.com/pdfs/IOActive_Adventures_in_Automotive_Networks_and_Control_Units.pdf, 2014.

[15] Miller, C., and Valasek, C. Remote exploitation of an unaltered passenger vehicle. *DEF CON 23* (2015).

[16] Norte, J. C. Hacking industrial vehicles from the internet. http://jcarlosnorte.com/security/2016/03/06/hacking-tachographs-from-the-internets.html, mar 2016. Accessed: 2016-03-17.

[17] NSF. EAGER: Collaborative: Toward a Test Bed for Heavy Vehicle Cyber Security Experimentation. https://www.nsf.gov/awardsearch/showAward?AWD_ID=1619690. Accessed: 2016-03-02.

[18] SAE. The SAE J1939 Communications Network. http://www.sae.org/misc/pdfs/J1939.pdf. Accessed: 2016-03-17.

[19] U.S. Department of Transportation, O. o. T. A. Economic characteristics of the freight transportation industry. http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/data_and_statistics/by_subject/freight/freight_facts_2015/chapter5, 2015. Accessed: 2016-03-17.

[20] U.S. Department of Transportation, O. o. T. A. National transportation statistics. http://www.rita.dot.gov/bts/sites/rita.dot.gov.bts/files/publications/national_transportation_statistics/index.html, 2016. Accessed: 2016-04-17.

[21] W., V. Sae j1939 serial control and communications vehicle network. http://www.esd-electronics-usa.com/online-seminars.html.