Automotive Car Hacking Technique Using ICS Simulator

Panaganti Priyanka
Computer Science and Engineering,
CyberSecurity
(Amrita School of Engineering)
Chennai, India

T. Mahima Singh
Computer Science of Engineering,
CyberSecurity
(Amrita School of Engineering)
Chennai, India

Velaga Sohith Raghava
Computer Science of Engineering,
CyberSecurity
(Amrita School of Engineering)
Chennai, India

***Abstract*— As cars become highly automated and linked, concerns about Cyber security in the automotive industry have grown significantly. Recent successful automotive hacks show the major security vulnerabilities with automobiles. Even said, forecasting what kinds of security vulnerabilities will emerge is exceedingly difficult, if not impossible. This study will detail the design and execution of an ICS Simulator as well as hacking tactics on the modern car network. The Controller Area Network (CAN) bus, which is part of the ISO 11898 protocol family, serves as the foundation for the primary network in automobiles that are now being created. The car contains a huge number of electronic control units (ECUs) that are linked together through the CAN bus to run various systems. The CAN bus allows ECUs to interact with one another via a single bus, which has improved fuel and emission efficiency while also increasing vulnerabilities by allowing access to CPS (Cyber Physical Systems) on the same network. This paper will summarize current research on the car network, provide ways for collecting network data for replay, and demonstrate how to construct and install an IC Simulator for continuing CAN bus research.**

Keywords—CAN Bus, ECUs, ICS Simulator,OBD, Replay attack,Cyber-attacks,

1. Introduction :

An automated vehicle has an automatic transmission and does not require the driver to shift gears manually. Transmissions, also known as gearboxes, help to direct the rotational force and speed of a vehicle. As a result, automatic gearboxes change gear ratios as the vehicle drives.

On-board diagnostics refers to the automotive electronic system that allows for self-diagnosis of the vehicle as well as reporting capabilities for repair technicians (OBD).

An integrated security system is one of the most important components of modern automotive manufacturing and design. As automobiles become more networked and software-controlled, security issues in the automotive industry are becoming increasingly severe. Furthermore, only a few components, such as immobilizers, are designed with security in mind in today's automobiles, which is insufficient. Cars now have a variety of electronic controls that can help them meet emissions standards while also providing their users with a comfortable and safe driving experience.

Given the state of modern vehicles, it is easy to envision a scenario in which a car is controlled by a smartphone. Furthermore, this leads to an increase in autonomous vehicles and self-driving cars, which represents the next logical step and is a reality in the current scenario. Because of the increasing complexity of vehicle electronic circuits, it is necessary to understand these electronic control units (ECUs) as well as their importance in monitoring the various subsystems of a car. Furthermore, modern vehicles can communicate with other devices via wireless interfaces, potentially exposing the car's internal network to vulnerabilities.

ECUs are now widely used in automobiles to control and perform the majority of functions. A vehicle's ECUs can range from dozens to hundreds. In this case, CAN serves as a link between ECUs. The CAN bus is the name given to the CAN hardware.

Automotive hacking is the act of identifying security flaws in software, hardware, or communication technologies such as Bluetooth or Wi-Fi, which allows a vehicle to be accessed. Cars have more lines of code than planes at times. This implies that other flaws could be discovered and exploited. Many modern vehicle technologies make driving easier, more enjoyable, and, in some cases, safer. The more linked your vehicle is and the more functions that can be controlled, the more complex the system. According to the figure, the global automobile sector will grow by $9 trillion by 2030. As a result, it appears that the automobile industry has a bright future.

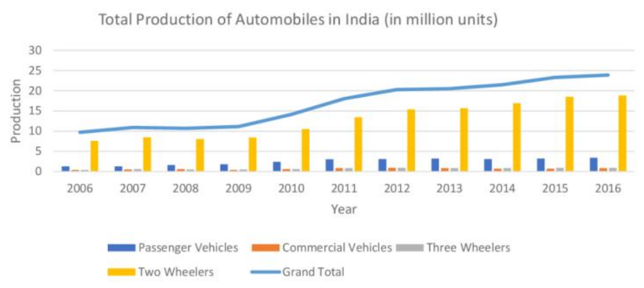

Fig 1: Automobile production in India

Furthermore, Cyber-attacks are a significant barrier that this cutting-edge technology must overcome. Because they connect via Bluetooth and Wi-Fi, automobiles are becoming increasingly vulnerable to Cyber-attacks. The Cyber-attacks ranged from close-range physical assaults to long-distance digital assaults. Solitary in 2020 Automotive hacking incidents increased by approximately 138% just this year. According to Upstream, growth will reach 99% in the coming years. Toyota suffered a data breach in February that exposed the personal information of 3.1 million of its customers.

Literature Review:

Many studies have been carried out in order to identify vulnerabilities and provide solutions for improving the security of embedded automobile systems. The authors present an overview of all the various methodologies used to analyse these security vulnerabilities in [1,] as well as a list of recommended solutions to improve their safety.

Despite the complexity of automotive electrical architecture, studies like [2] and [3] provide a taxonomy of various threats like systematic manipulation and infiltration. According to the authors of [4,] the internal automotive computer network was not protected against a local attacker with physical access to the vehicle.

They show an attack that takes advantage of flaws in various subsystems, such as inserting malicious code into an Electronic Control Unit (ECU) and erasing all traces of its existence after a crash. We extend this idea by looking into the feasibility of repeating such attacks from a distance by remotely controlling a vehicle's internal control systems.

Papers like [5] and [6] provide an overview of the many vulnerabilities, potential attacks, and perspectives on the security challenges of networked cars. [7] is an outstanding book that analyses various assaults on automobile subsystems with a highly practical approach, in which authors detail how to manipulate and control a variety of Ford Escape and Toyota Prius safety critical subsystems. A year later, in [8,] the same authors present an intriguing assessment of remote assaults, examining possible infiltration scenarios for 20 different vehicles. The authors of [9] present a review of autonomous car technology achievements from recent US and European research initiatives, as well as highlighting the risks of current embedded systems.

2. CAN Bus:

The Controller Area Network (CAN) bus is a message-based protocol that allows the Electronic Control Units (ECUs) found in today's automobiles, as well as other devices, to communicate with one another in a dependable, priority-driven manner. Messages or "frames" are received by all network devices without the need for a host computer. CAN is supported by a comprehensive set of international standards defined by ISO 11898.

2.1. Overview of CAN Bus:

By the early 1980s, cars contained an increasing number of ECUs (electronic control units), and companies such as Germany's Robert Bosch were looking for a type of bus communication system that could be used as a communication system between multiple ECUs and vehicle systems. They searched the market but couldn't find exactly what they needed, so they began developing the "Controller Area Network" in collaboration with Mercedes-Benz, Intel®, and several German universities.

Bosch introduced the CAN standard at the SAE Congress in Detroit in 1986. After one year, Intel Corporation began shipping the first CAN controller chips, and the automotive world was forever changed.

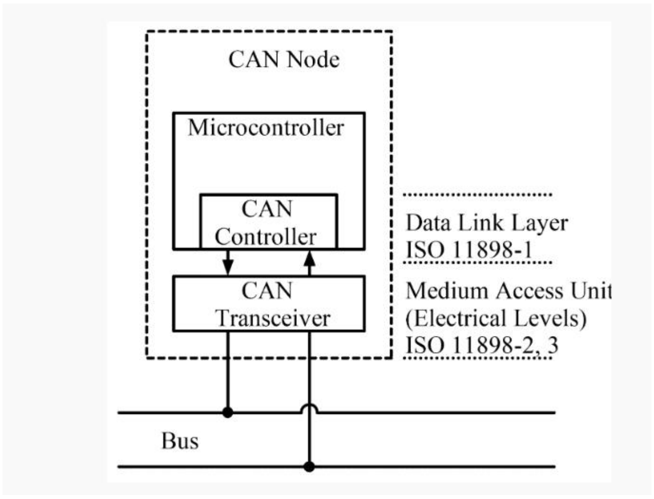

Fig 2: CAN bus network schematic

2.2. CAN Bus messaging:

The devices on a CAN bus are referred to as "nodes." Each node is made up of a CPU, a CAN controller, and a transceiver that adjusts the signal levels of data sent and received by the node. All nodes can send and receive data, but not simultaneously.

Nodes cannot communicate with one another directly. Instead, they broadcast their data across the network, where it is accessible to any node to which it has been addressed. The CAN protocol is lossless, using a bit wise arbitration

All data is sent in frames with CAN, and there are four types:

- **Data frames** are used to send data to one or more receiver nodes.
- **Remote frames** request information from other nodes.
- **Error frames** display errors.
- **Overload frames** are used to report overload

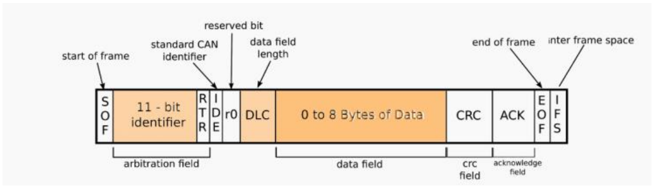

Fig 3: Standard CAN Data Message

2.3. CAN Bus Architecture:

The CAN bus is referred to as the "heart" of any modern vehicle's interconnected systems. The CAN bus is a single, centralized network bus that broadcasts all of a vehicle's data traffic.

Every command from the operator, such as "apply the brakes" or "roll down the windows," is carried by the CAN bus system, as are readouts from sensors reporting engine temperature or Tyre pressure. The introduction of the CAN bus improved efficiency and reduced complications, lowering wiring costs as well.

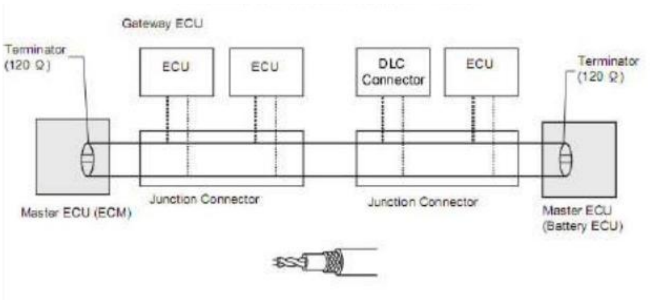

Fig 4: CAN Bus Architecture

2.4. CAN Bus Hacking:

An attacker can send wireless commands remotely from a computer by injecting malicious code into the CAN ports. It may only take 5 minutes or less to enter the action and then walk away. A hacker could wait for a minute or a year and then trigger it to do whatever they had programmed it to do.

Once hackers gain access to this network, they can control locks, lights, steering, and even brakes.

Hackers have already tested the car hacking toolkit (CHT) on various vehicles and successfully performed tricks such as setting off alarms, affecting steering, applying brakes, and turning off headlights. This was done with Bluetooth, but it could also be done with a Raspberry Pi or a WiFi router, allowing the CHT to control the car from a distance.

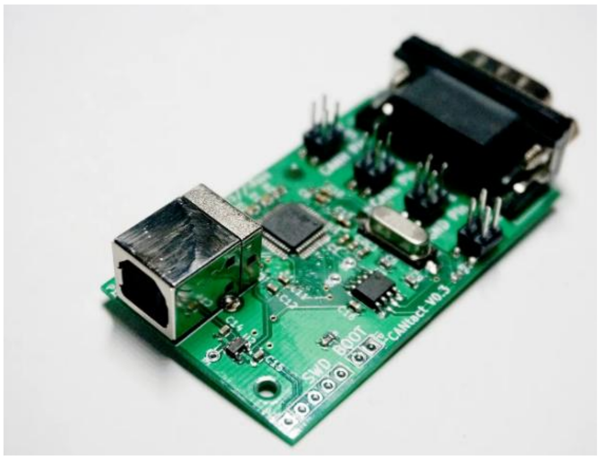

Fig 5: Car Hacking tool

3. Electronic Control Unit(ECU):

The term ECU can refer to an Engine Control Unit, but it can also refer to an Electronic Control Unit, which is a component of any automotive mechatronic system and not just for engine control.

The term ECU is frequently used in the automotive industry to refer to an Engine Control Unit (ECU) or an Engine Control Module (ECM). When this unit controls both the engine and the transmission, it is referred to as a Power train Control Module (PCM).

ECUs were first used in the 1980s to precisely control the amount of gasoline delivered into the motor at each cycle. Modern ECUs can, among other things, control the air intake, fuel injection, torque, and gather data from a variety of sensors. They can serve as door control units, human computer interfaces, and cruise control systems, among other things. It is critical that these systems operate continuously in accordance with manufacturer specifications. When necessary, an electronic control unit uses data received from one or more automotive components to perform the required actions.

3.1. Types Of ECU:

Vehicles with multiple ECUs are divided in terms of what tasks they perform. The following are some examples of these types.

Engine Control Module:

The ECM controls the amount of fuel and ignition timing with its sensors to get the most power and economy out of the engine.

Brakes Control Module:

The BCM, which is used in vehicles with ABS, ensures that the wheels are not skidding and determines when to initiate braking and when to let go of the brake to prevent the wheels from locking up.

Transmission Control Module:

When used on an automatic vehicle, the TCM ensures the smoothest shifts possible by analyzing the engine RPM and acceleration.

Telematic Control Module:

Another TCU with the same abbreviation, this one ensures that the car's onboard services are operational. It is in charge of the vehicle's satellite navigation, Internet, and phone connectivity.

Suspension Control Module:

The SCM, which is found in vehicles with active suspension systems, ensures proper ride height and optimal suspension changes based on driving conditions.

3.2. How does ECU Work:

ECU is an electronic device with memory filled with base numbers and parameters. With multiple sensors around a vehicle feeding data to the ECU, it can efficiently manage and control the electronic systems by issuing orders to improve their output.

For example, an airbag ECU collects data from seat sensors and collision sensors. The ECU determines which airbags to deploy based on where the passengers are seated during a collision. The actuators are then told to deploy themselves. The electrical signal is then converted into the physical value required by the actuators, which are comprised of valves, injectors, or relays. A car may have over 100 ECUs that manage comfort and safety features such as parking assistance, memory seats, and airbag activation, in addition to controlling vital elements such as power steering and engine performance.

The ECU controls the injection of fuel as well as the timing of the spark to ignite it. It uses a Crankshaft Position Sensor to determine the position of the engine's internals, allowing the injectors and ignition system to be activated at precisely the right time. While this appears to be something that can be done mechanically (and it once was), there is now a bit more to it.

An internal combustion engine is essentially a large air pump that runs on fuel. As the air is drawn in, enough fuel must be provided to generate enough power to keep the engine running while still having enough left over to propel the car when needed.

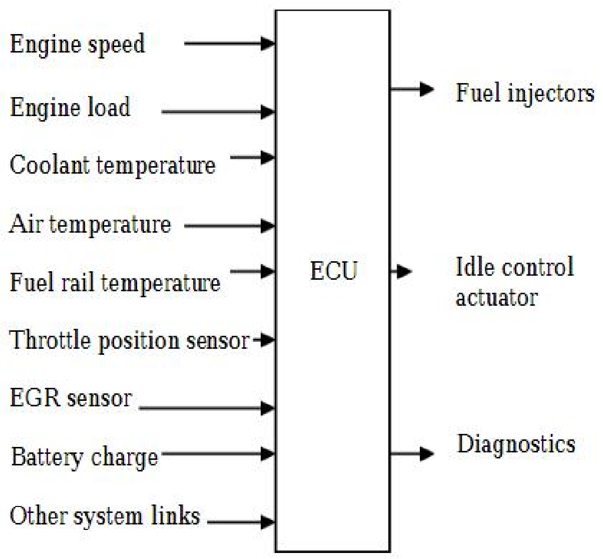

Fig 6: Architecture Diagram Of ECU

4. Various ways to hack an Automotive Cars:

The growing number of embedded subsystems in automobiles, as well as their internal communication, have created opportunities for attackers to launch various attacks against them. In this section, we'll start talking about potential attacks in the automotive scenario.

An attacker who wants to take control of a vehicle learns how to compromise an electronic control unit (ECU) and discovers the feature of an attack.

Physical attacks : This requires the attacker to gain access to physical ports such as USB and OBD-II and sniff the data.

Wireless attacks : These include Wi-Fi, Bluetooth Low Energy (BLE), NB-IoT, and GSM attacks on short and long-range wireless communication.

After compiling a list of all significant automobile hacking incidents worldwide over the previous ten years, Upstream discovered numerous patterns. According to its analysis, automakers should focus their Cyber security efforts here.

A. ***Replay Attack***:

A replay attack is a type of man-in-the-middle attack in which an attacker sniffs messages being sent over a channel in order to intercept and resend them as authentic messages. The replay attack is particularly dangerous because the attacker does not even need to decrypt the message they resend in order to fool the receiver into thinking the received message is legitimate.

A replay attack occurs when a Cyber criminal intercepts a secure network transmission, retrieves it, and then falsely freezes or re-transmits it to deceive the receiver. The additional risk of replay attacks is that a hacker does not even need technical skills to decode a communication once it has been captured from the network. Simply re-sending the entire thing could be enough to complete the attack.

B. Overriding Cars key-less entry:

Key-less car entry is a cutting-edge car lock system that communicates with the vehicle within a certain range, eliminating the need for a physical key to unlock the doors. When you bring the fob close to the car, whether in your hand, pocket, or bag, the car detects your presence and unlocks the doors.

While key-less entry is a convenient feature, it may raise some security concerns. The main one is that, theoretically, thieves with the right technology can override a key-less entry system and start a car without the ignition key.

5. ICS Simulator:

Operating a simulation of instrument clusters is the simplest and least expensive way to practise automobile hacking. Craig Smith's ICS Simulator open-source repository is greatly appreciated. Learning CAN-Bus exploitation with ICS is simple to set up and reasonably priced.

In this Paper, We will set up a simulation environment in which you will be able to apply some of your knowledge to analyse and hack a simulated vehicle.

Step 1: Setup Dependencies

The first step is to add the required dependencies to your Kali system.

```
Kali > apt-get install libsdl2-dev libsdl2-image-dev -y
```

Step 2: Download and install Can Utils.

The CAN utils must then be installed. Bosch of Germany created this set of Linux-native utilities. You've probably already installed these utilities if you followed my second tutorial in the series. If you haven't already, you can do so right now by downloading and installing them from the Kali repository.

```
apt - get install can-utils -y
```

Step 3: Get ICSim.

Craig Smith, the author of The Car Hackers Handbook and the creator of opengarages.org, has created a small CAN simulator that we will download and install next. You can get it from GitHub.com.

```
kali > git clone https://github.com/zombieCraig/ICSim
```

Step 4:

After cloning the ICS Simulator we need to execute the setup_vcan .sh shell script.

```
kali > ./setup_vcan.sh
```

Step 5:

To start the instrument panel of our simulated vehicle, we simply need to execute ICS simulator followed by the name of the virtual CAN interface, in this case, vcan0.

```
kali > ./icsim vcan0
```

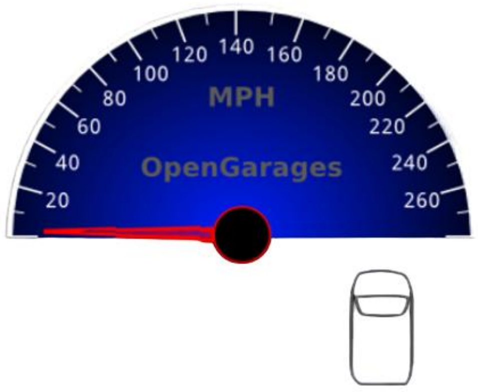

Fig 7: ICS Simulator

The instrument panel should look like this on your desktop. It has a speedometer, turn signal, and a virtual vehicle silhouette that, like modern vehicles, indicates open and closed doors for the driver.

Step 6:

Enter This Command to activate the vehicle's controller.

```
kali > ./controls vcan0
```

So, With this we have completed setting up ICS Simulator and Controller. Now, we will be able to perform replay attack on the virtual car.

6. Replay Attack:

Basic CAN traffic display, recording, generation, and replay tools are

cansniffer : shows differences in CAN data content (just 11bit CAN IDs)

candump : candump is a programme that displays, filters, and logs CAN data to files.

canplayer : play back CAN log files

cansend : send a single frame with cansend.

Step 1: Turn on the cansniffer.

To begin, use cansniffer to sniff the CAN traffic. You must specify the interface (vcan0 in our case) and the -c option if you want to see the colorized output.

```
kali > cansniffer -c vcan0
```

We can see the CAN Network Below

Step 2: Filter for specific traffic with cansniffer.

Rather than watch all the traffic go past our terminal, we can filter traffic similarly to the more widely used sniffer, Wire shark.

We can use the command -f to filter the specific network.

```
Cansniffer  -f
```

We could then enter if we only wanted to see traffic from ID=161.

```
kali > cansniffer -c vcan0
```

Once the sniffer has started, we can then enter;

```
-000000
+161
```

It is important to note that the above commands will not appear on the screen when you enter them. After you enter the ID number, the sniffer will start filtering out all traffic those with the ID= 161.

Step 3: Capture CAN traffic with candump.

While the cansniffer utility, like Wire shark, can sniff CAN network traffic, the candump utility in can-utils can capture CAN traffic and store it in a file for later analysis or replay.

To accomplish this, we can simply use the -l option to log and the -c option to colourize the output.

```
kali > candump -c -l vcan0
```

We can use the -s 0 option to log AND view the output (silent mode 0). In addition, we can use the -a (ASCII) option to convert the output from hex to ASCII (human readable). This starts candump in colourize mode with ASCII output, storing the data in a log file and sending it to the terminal at the same time (stout).

```
kali > candump -c -l -s 0 -a vcan0
```

Step 4: Use canplayer

Canplayer is another important CAN network tool. This tool allows us to "play" the candump output. As a result, we could capture data from the CAN network and replay it on the network. We only need to use the -I option, followed by the

```
kali >canplayer -I candump-xxxxxxxxxxx.log
```

Step 5: Sending Custom Frames with cansend

Finally, there's the cansend tool. This tool allows us to replay a specific frame or send a custom CAN frame. If we want to resend a single frame with ID=161, we can do so.

we do so by entering;

```
kali > cansend vcan0 161#000005500108000d
```

When we press enter, the custom CAN frame will be transmitted over the network. I hope it is clear that when we reverse engineer the network, this is the command we will use to initiate the CAN network actions we desire, such as accelerating, opening the door, initiating brakes, and so on.

Now that we have installed the ICS Simulator and understand the basics of the key can-utils tools, we can now able to use these tools to reverse engineer the CAN bus on our ICS Simulator and take control of the vehicle!

7. Prevention Methods:

You should also be aware of any software security patches issued by your vehicle's manufacturer. On their websites, vendors typically provide links to download software onto personal USB flash drives. Simply insert the flash drive into your car's USB port to transfer the fix or update to your vehicle.

Avoid installing software that has not been approved by the manufacturer on your vehicle. This includes diagnostic software to monitor your vehicle's performance as well as various types of entertainment software that require Internet access. Third-party software may contain flaws that hackers can use to steal or control your vehicle.

Limit wireless or remote systems: Systems that can remotely disable or monitor your vehicle put you at risk. Unlike many other systems that are hard-wired into your vehicle's computer, wireless or remote systems are frequently controlled online, making them more vulnerable and appealing to hackers.

Don't leave your password in your vehicle: Hacking can occur both inside and outside of your vehicle. A car thief, for example, who discovers your On Star password, can take over your account. That means the feature that allows you to remotely turn off your engine when you report a stolen vehicle will be rendered ineffective.

Don't install rogue apps or use your car's Web browser: Your car's infotainment system is vulnerable and open to attack. Untrusted apps in your infotainment system have the potential to introduce malware. You should never use your vehicle's Web browser, either. Simply use your phone while safely parked instead.

Keep track of vehicle recalls: One Cyber Security-related vehicle recall has already occurred for the Jeep Grand Cherokee UConnect entertainment system. The vulnerability left access open to the car’s acceleration, radio, brakes, windshield wipers, and more. Customers who were affected received a USB device to upgrade their vehicle's software with new security features. Vehicle owners should be on the lookout for similar recalls.

8. Conclusion :

Cyber security is now a necessity. Smart cars are the most vulnerable and susceptible to all types of exploits. Consider the possibility of being hacked while driving. Even the airbags, brakes, and accelerators may be beyond one's control while driving. As a result, manufacturers must prioritize the CAN bus system by strengthening its hardware security and employing secret codes. We can patch that vulnerability and potentially save lives if we find all possible ways for a hacker to attack the car.

We are currently watching the shift from theory to terrsor. In this research, we demonstrated an experimental platform capable of remotely accessing and interacting with a vehicle's internal systems. This platform is made up of an instrument cluster simulator also known as ICSim that serves as the communication interface. This proposal's purpose was to remotely control the airplane's vital safety components. We addressed vehicle security challenges from three perspectives: communication, system, and software architecture. We begin by discussing several security problems with communication mediums and protocols such as CAN bus and Ethernet .

9. Future work:

Our future work will consist in adding additional equipment to the vehicle, including sensors and actuators for the braking and steering.

10. References:

[1]Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaˆaniche, and Y. Laarouchi, “Survey on security threats and protection mechanisms in embedded automotive networks,” in Dependable Systems and Networks Workshop (DSN-W), 2013 43rd Annual IEEE/IFIP Conference on. IEEE, 2013, pp. 1–12.
[2] M. Wolf, A. Weimerskirch, and T. Wollinger, “State of the art: Embedding security in vehicles,” EURASIP Journal on Embedded Systems, vol. 2007, no. 1, p. 074706, 2007.
[3] P. Kleberger, T. Olovsson, and E. Jonsson, “Security aspects of the in-vehicle network in the connected car,” in Intelligent Vehicles Symposium (IV), 2011 IEEE. IEEE, 2011, pp. 528– 533.
[4] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham et al., “Experimental security analysis of a modern automobile,” in Security and Privacy (SP), 2010 IEEE Symposium on. IEEE, 2010, pp. 447–462.
[5] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno et al., “Comprehensive experimental analyses of automotive attack surfaces.” in USENIX Security Symposium, 2011.
[6] R. Moalla, H. Labiod, B. Lonc, and N. Simoni, “Risk analysis study of its communication architecture,” in Network of the Future (NOF), 2012 Third International Conference on the. IEEE, 2012, pp. 1–5.
[7] C. Miller and C. Valasek, “Adventures in automotive networks and control units,” in DEF CON 21 Hacking Conference. Las Vegas, NV: DEF CON, 2013.
[8] ——, “A survey of remote automotive attack surfaces,” Black Hat USA, 2014.
[9] J. Ibaˇnez-Guzman, C. Laugier, J. D. Yoder, and S. Thrun, “Autonomous driving: Context and state-of-the-art,” in Handbook of Intelligent Vehicles. Springer, 2012, pp. 1271–1310.