A Survey on Connected Vehicles Vulnerabilities and Countermeasures

Caleb Riggs, Carl-Edwin Rigaud, Robert Beard, Tanner Douglas, and Karim Elish
Florida Polytechnic University
4700 Research Way
Lakeland, FL, 33805
Email: {criggs2892, carledwinrigaud0841, rbeard3043, Tannerdouglas0256, kelish}@floridapoly.edu

***Abstract*—With the growing ease of connecting a vehicle and other technologies to the Internet, the need for security is growing. In a connected vehicle, there are many different connections and therefore many different systems can be exploited. In this paper, we shed the light on the security of several features of connected vehicles to determine whether or not they are vulnerable to attacks and identify possible mitigations. We focus on four features, namely, Bluetooth, OBD (On Board Diagnostics) System, Infotainment System, and OTA (Over the air).**

***Index Terms*—Connected vehicle, Vulnerability, OTA (Over the Air), OBD (On Board Diagnostics), Infotainment, Telematics**

I. Introduction

There are many issues to consider when thinking about putting autonomous connected vehicles on the road. Are they safe? Are they efficient? The answers to these questions are important. The smallest mistake can result in a car accident with severe injuries or even death. To introduce connected vehicles into chaos, we have to be certain that they are not susceptible to attacks from attackers that aim to purposefully cause car accidents. The objective of this paper is to shed the light on the security of several features of connected vehicles to determine whether or not they are vulnerable to attacks and identify possible mitigations. These features include Bluetooth, OBD (On Board Diagnostics) System, Infotainment System, and OTA (Over the air) mechanism which is used by connected vehicles to upgrade and maintain their software.

II. Feature 1: Bluetooth

Bluetooth has been around since the 90's and it has been implemented into many of our everyday devices for convenience, including our cars. Though there have been many advancements in the security and implementation of Bluetooth, there are still many vulnerabilities in this technology. With these vulnerabilities we risk an attacker gain control of Bluetooth but also gaining access to other aspects of the car itself. In this section, we discuss some of the flaws and weaknesses of Bluetooth as well as the steps that can be used to mitigate these problems. There are four versions of Bluetooth security modes [3]:

- Bluetooth Security Mode 1: This mode is non-secure. The authentication and encryption functionality are bypassed, and the device is susceptible to hacking. Security Mode 1 is only supported up to Bluetooth 2.0 + EDR and not beyond.
- Bluetooth Security Mode 2: For this Bluetooth security mode, a centralized security manager controls access to specific services and devices. The Bluetooth security manager maintains policies for access control and interfaces with other protocols and device users.
- Bluetooth Security Mode 3: In this mode, the Bluetooth device initiates security procedures before any physical link is established. The authentication and encryption are used for all connections to and from the device.
- Bluetooth Security Mode 4: The security procedures are initiated after link setup. Secure Simple Pairing uses what are termed Elliptic Curve Diffie Hellman (ECDH) techniques for key exchange and link key generation. The security modes for services protected by Security Mode 4 are as follows: (1) authenticated link key required, (2) unauthenticated link key required, (3) no security required.

Most cars that were made after 2010 should run on the Bluetooth security mode 4. With the earlier version of Bluetooth security, it ran little to no security and was very susceptible to attack from an outside source and even now can be a liability if used regularly to connect to your vehicle. Even more, the most up to date version of Bluetooth is at risk to attack.

A. Tools and Programs

Through our research into the vulnerabilities in Bluetooth we came across many tools and programs used to hack into Bluetooth. Many have been used to not only get information from your device but to also take control of it as well. There are many tools that are available [4], but we look at the most common tools used.

1) BlueScanner

This tool stores more technical information: device type, IP address, Mac address, manufacturer, etc. It also provides the name the user gave to each device. For example, if three cell phones and two computers are in

Manuscript received February 1, 2018; revised June 1, 2018.

©2018 Journal of Traffic and Logistics Engineering
doi: 10.18178/jtle.6.1.11-16

range, BlueScanner will identify if the user named his/her computer as PC, My PC,…etc. It also gathers this information without being discovered since it does not try to log in.

2) Bloover II

Blooover II is a J2ME-based auditing tool. It is intended to serve as an auditing tool to check whether a mobile phone is vulnerable.

3) Carwhisperer

Can be used to listen or transmitted audio through the hands-free system integrated into your car.

4) Bluediving

Bluediving is a Bluetooth penetration testing suite. It implements attacks like Bluebug, BlueSnarf, BlueSnarf++, BlueSmack, and has features such as Bluetooth address spoofing, an AT and a RFCOMM socket shell. It also implements tools like car whisperer, L2CAP packet generator, L2CAP connection resetter, RFCOMM scanner and green plaque scanning mode.

B. Potential Vulnerabillities

BlueBorne is an attack vector by which hackers can leverage Bluetooth connections to penetrate and take complete control over targeted devices. BlueBorne affects ordinary computers, mobile phones, and the expanding realm of IoT devices. The attack does not require the targeted device to be paired to the attacker's device, or even to be set on discoverable mode. These are the eight vulnerabilities discovered [5]:

1) Linux kernel RCE vulnerability - CVE-2017-1000251

The native Bluetooth stack in the Linux Kernel (BlueZ), starting at the Linux kernel version 3.3-rc1 and up to and including 4.13.1, are vulnerable to a stack overflow vulnerability in the processing of L2CAP configuration responses resulting in Remote code execution in kernel space.

2) Linux Bluetooth stack (BlueZ) information Leak vulnerability - CVE-2017-1000250

All versions of the SDP server in BlueZ 5.46 and earlier are vulnerable to an information disclosure vulnerability which allows remote attackers to obtain sensitive information from the Bluetooth process memory. This vulnerability lies in the processing of SDP search attribute requests

3) Android information leak vulnerability - CVE-2017-0785

An information disclosure vulnerability in the Android system (Bluetooth). Product: Android. Versions: 4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0. Android ID: A-63146698.

4) Android RCE vulnerability #1 - CVE-2017-0781

A remote code execution vulnerability in the Android system (Bluetooth). Product: Android. Versions: 4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0. Android ID: A-63146105

5) Android RCE vulnerability #2 - CVE-2017-0782

A remote code execution vulnerability in the Android system (Bluetooth). Product: Android. Versions: 4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0. Android ID: A-63146237.

6) The Bluetooth Pineapple in Android - Logical Flaw CVE-2017-0783

An information disclosure vulnerability in the Android system (Bluetooth). Product: Android. Versions: 4.4.4, 5.0.2, 5.1.1, 6.0, 6.0.1, 7.0, 7.1.1, 7.1.2, 8.0. Android ID: A-63145701.

7) The Bluetooth Pineapple in Windows - Logical Flaw CVE-2017-8628

Microsoft Bluetooth Driver in Windows Server 2008 SP2, Windows 7 SP1, Windows 8.1, Windows RT 8.1, Windows 10 Gold, 1511, 1607, 1703 allows a spoofing vulnerability due to Microsoft's implementation of the Bluetooth stack, aka "Microsoft Bluetooth Driver Spoofing Vulnerability".

8) Apple Low Energy Audio Protocol RCE vulnerability - CVE-2017-14315

In Apple iOS 7 through 9, due to a BlueBorne flaw in the implementation of LEAP (Low Energy Audio Protocol), a large audio command can be sent to a targeted device and lead to a heap overflow with attacker-controlled data. Since the audio commands sent via LEAP are not properly validated, an attacker can use this overflow to gain full control of the device through the relatively high privileges of the Bluetooth stack in iOS. The attack bypasses Bluetooth access control; however, the default "Bluetooth On" value must be present in Settings.

Blueborn is the most current vulnerability that an attacker can use to not only get information but gain full control of that device. With a connected car this could be troublesome being that many of its vital controls are connected with the devices that use Bluetooth as well.

The dongle as well can be an area of interest for an attacker. With many third party add-ons that you can use to diagnose your vehicle, many attackers can use that point of entry to attack the system directly. Drive log for example was known for several vulnerabilities that could give an attacker access to the CAN bus and send messages directly to it from a compromised phone using the Bluetooth connection that they share. In 2017, there was a study to see how exactly would an attacker would use drive log to hack into a car. They found that it could not only send patches directly to the firmware using the compromised phone but after several firmware updates it could send messages directly to the CAN itself affecting many functions of that vehicle.

This brings to light another aspect needing attention even if the Bluetooth in your vehicle is secure, its security is only as strong as to what you connected to. If you are connecting to compromised devices you can still be a target of an attack. Especially with the blueborn vulnerabilities. Many attackers can use the weakest point of entry into your system, e.g., it can be a device that you have connected to your system.

C. Mitigation

One of the most obvious ways to mitigate the vulnerabilities is to make sure your software is always up to date [6]. Many of the exploits and vulnerabilities have been addressed and patches have been sent out, but it is up to the user to implement them. Another way is to be mindful of what you are connecting to the system itself. It seems that the biggest weakness for many of these connected cars are their network since they have many critical systems (brakes, transmission, steering) on the same network with non-critical systems (radio, Bluetooth, WIFI). This makes an attacker to get access to a lot of things that they should not have access to.

For example, 2014 jeep Grand Cherokee is listed as one of the most hackable cars according to researchers Charlie Miller and Chris Valasek [7] while present their findings at the Black Hat USA global security event in Las Vegas. "The 2014 Jeep Cherokee operates both "cyber physical" features and remote access functions on the same network. Intuitively, the cars with the least amount of integrated systems on the same network are the least hackable. Miller and Valasek point out that cars like the 2014 Dodge Viper, and the 2014 Honda Accord have steering and engine management systems that are isolated from other systems [7]. So, another way to mitigate these issues is to use separate networks for these systems and remove them off from other systems not even giving the access.

In summary, though this is just one aspect of the connected cars system, it is something that can be a very attractive area to an attacker. At a user level, the only protection you can have is the common sense and being mindful of what you are doing. But most people will say that the risk is worth the reward. The same can be said about cars many people would prefer that convenience that cars offer now than the non-connected counterparts.

III. Feature 2: Infotainment Systems

A major attack vector for autonomous vehicles is their infotainment systems. The term infotainment system is basically a fancy way of saying the stereo and user accessible computer in the car. In some cars, these systems are running Android auto or Apple car play. However, some manufacturers still use their own custom infotainment systems. All of these systems are generally connected to the CAN bus which makes them crucial in a car's ecosystem. In this section, we investigate Android auto and what potential vulnerabilities it has.

A. Potential Vulnerabilities

Android is known for being an open platform and often times being vulnerable. A large part of Android auto is simply relaying information from your phone. It is possible a phone app could be created that while appearing harmless could unknowingly interact with Android auto. Since the car already has a trust relationship with the phone exploiting that relationship, it could be the first step into cracking Android auto. However, the Android auto API is currently not documented well and does not seem to have many features that could be exploited. Two of the main features that are documented are displaying messages and playing media [8].

1) Displaying messages

Since Android auto can show messages sent from applications, it opens the potential for a well-known vulnerability to be exploited in what is known as stage fright [9]. The stage fright vulnerability takes advantage of the fact that Android would automatically download data from an MMS. The files that are downloaded can be malicious and have the potential to do many things including crash the system. If Android auto behaves similarly when receiving messages this would be a serious vulnerability, although crashing the infotainment system does not seem like a major vulnerability the fact that the infotainment system is connected with the CAN bus could lead to further problems.

If you can control a phone by sending an MMS perhaps the same concepts apply to Android auto where you could potentially crash the system with a message sent to the display. Aside from the above two methods Android auto is not that fully featured right now, but as more features and functionalities become available your phone and car will become increasingly important.

2) Playing media

In order for this functionality to be exploited, Android auto would have to be vulnerable to some form of malicious media file like an MP3. This, however, is not likely because Android has dealt with media files since its beginnings. More than likely there is not a vulnerability in the media player on Android auto. Although there is not likely a direct vulnerability involving media files an app could still potentially send audio to the infotainment system to inconvenience a user, in the form of adware.

3) Other

Given recent events and the exposure of the KRACK attack, this raises concern to other vulnerabilities in a car. Some connected cars have built-in wifi hotspots. Assuming these hotspots use WPA for authentication you could exploit the newfound weakness to gain access to the system.

B. Case Study

While researching infotainment system exploitation, it is important to note that this in conjunction with the telematics system is what was used to remotely shutdown a Jeep in 2015. By exploiting a trust relationship with sprint phones and the Jeep they gained access to the infotainment system that was connected to the CAN Bus controller, they flashed a custom firmware to the controller that allowed them to send commands from it to other systems and control the car [10].

C. Mitigation

Although the infotainment system can sometimes be exploited, the main vulnerability here is the CAN bus technology. The CAN bus was designed before the Internet was commonplace and therefore lacks security features that it should have in a modern society. A potential solution is securing the CAN through encryption, one such protocol that does this is SecureCAN. More detailed solutions to securing the CAN bus can be found in [10].

Although no many common exploits exist in infotainment systems like Android auto, the ability to compromise infotainment systems and take control of the vehicle is drastic. Like every technology, infotainment systems are growing in both their capability and frequency in vehicles. As their capabilities grow there will be more attack vectors and more vulnerabilities.

IV. Feature 3: OBD System

The standard data collection method for modern automobiles is called On Board Diagnostics (OBD) port [11]. Originally when an automobile encountered an error a "Check Engine" light would illuminate. While this was good for reporting issues to the driver, it did not tell the specific issue or even narrow down the possible issues causing the problem. Afterwards in the 1980's the beginning of the OBD System was being created. This "new" OBD port allowed for the automobile to report a variety of information to the user. The issue with this system is that each make and model of car could have their own type of connector as well as what data is being transmitted. This prompted for a standardization of the OBD system. The new standard has changed to the OBD-II recently which allows for more standardized data transport methods. Before OBD-II was the standard, the port still has a small variety of possible connector types as well as pinout designs depending on country of origin and the manufacturer.

A. Potential Vulnerabilities

Currently there is an exploit where European BMW's can be stolen by using the OBD port to give the electronic key so that a random key fob can be reprogrammed to start the car [12]. While this allows you to steal the car, it does not allow for remote access to the vehicle.

Currently, it does not seem to be a way to send data to the vehicle through the OBD port. Further investigation of the port's technical specifications is needed to check if it is possible that any data can be sent through the port.

V. Feature 4: OTA Updates

With all the recent advancements in car manufacturing and car technologies, in-vehicle software has become increasingly important over recent years. This is due to such factors as development of ADAS, represented by automatic braking, sensors to detect nearby vehicles and objects, rearview cameras to assist in parking, self-parking systems and the introduction of autonomous driving technologies. In order to maintain the efficiency of the car, it has become a big issue as to how you would go about upgrading and improving the quality of such software [13].

However, car owners would feel it inconvenient and time consuming if they have to bring their vehicles to dealerships every time they needed the software updated to maintain and improve its quality. The problem now is that if they do not want to bother themselves to bring their vehicles to the dealerships and the software is not updated, that may put the vehicle, the car owner, and other car owners in danger. How can this situation be properly resolved? The solution to this problem is Over the Air (OTA) updates.

A. Benefits

When car owners need to get their cars serviced and checked, they bring it to a dealership. They also bring it to the dealership if the software needs to be updated for certain systems on the car. These take time and also money. When a car manufacturer has to recall vehicles because of software defects that need be addressed, they have to pay for these recalls themselves. So updating cars at dealerships and recalling them takes time money. The time it takes for this to be done is time that a vulnerability could be exploited and an accident could happen.

If it becomes frequent that the in-car software needs to be updated, it would be unreasonable to ask the service engineers to constantly be trained and service cars going into the dealer for software updates. That will create bottlenecks in car dealerships across the country and be very costly, because the dealerships wouldn't make money when updating software. There is money when they have to fix the transmission, the exhaust system, and the brakes because these are hardware parts that are purchasable. So, car manufacturers do not want cars to come back to the dealer for software updates because of economic reasons, and because of convenience.

That is why OTA updates are so beneficial, since they reduce the time it would take to perform these corrections. OTA updates would eliminate the need to bring vehicles into dealerships for software updates and allow data driven improvements to minimize maintenance. Ford, General Motors, Cadillac, and Fiat all experienced recalls related to embedded software bugs in 2014, while Tesla avoided a potential recall related to defective adapter plugs by issuing a remote software update using OTA, and also used the center console as a gateway to upgrade the transmission systems of Model S sedans. Minimizing recalls, which overall totaled more than $900M for General Motors (GM) alone in 2016, would be a significant cost saving opportunity. According to IHS Markit, OTA updates could save the global automotive industry more than $35B by 2022 [14].

B. Potential Vulnerabilities

There are many different potential vulnerabilities. When you are dealing with around 100 million lines of code it becomes difficult to secure all vectors [15]. One of the biggest issues is that vehicles are growing in the ways they connect and interact with the world around them.

When vehicles are connected to networks, a specific vehicle can be identified, targeted, and accessed even from a remote location. This means a higher risk of cyberattacks. OTA updates allow a car manufacturer to transmit and execute code on a vehicle to update its software. If the OTA update mechanism were to be used by an attacker, they would be able to gain access to a vehicle, transfer a malicious file to it, execute that file, and now the vehicle is compromised. Securing an OTA update mechanism is tough work though, there are a lot of different attack vectors to consider. The update file should be encrypted and delivered via a secure protocol, the package content must be cryptographically verified, and the endpoint device should be authenticated before any operation takes place.

From a security perspective, there is several steps that need to be taken when updating software Over the Air. If any of those steps were vulnerable to attack or could be misused by a hacker, the integrity of the vehicle and the safety of the passengers would be at risk. The first step in an OTA update is the security of the connection itself, whether that's through Secure Socket Layer protocol or the newer Transport Layer Security protocol or some of the other security mechanisms that the manufacturers would have for the connectivity. After establishing the connection, there's the authentication. Is the car talking to the server it should be talking to and is that server talking to the device it believes it is talking to? There is also the payload itself. Is the vehicle receiving the payload it should be receiving? has it been tampered with? And then there is the installation of that payload.

The main issue here is that if the OTA mechanism is not properly secured and the update package comes from a non-authorized back end, or a hacker, and it convinces the car that the IP address of the hacker's server is the IP address that the car should use in order to perform the software update, an unwanted firmware will go to the car and be installed which will grant the hacker complete control of certain parts or the entire vehicle.

TABLE I. VULNERABILITIES AND ASSOCIATED REMEDIATION

Problem	Type	Countermeasures
Bluescanner	Bluetooth	Ensure that the Bluetooth device is deactivated or is not in discoverable mode.
Bloover II	Bluetooth	Ensure that the Bluetooth device is deactivated or is not in discoverable mode.
Car whisperer	Bluetooth	Ensure that the Bluetooth device is deactivated or is not in discoverable mode.
Bluediving	Bluetooth	Because this is a suite of different Bluetooth cracking tools, Bluetooth should be deactivated and not in discoverable mode. Keep up to date on all updates.
BlueBorne	Bluetooth	Because this is a newly discovered vulnerability there is a patch released but not all of the issues have been solved.
Attackers gaining access through a compromised device.	Bluetooth	All devices should be up to date on all security patches and updates. Never use a jailbroken or rooted device and limit what devices are allowed to connect to you vehicle.
Stage Fright	Infotainment	Disable automatically downloading MMS message content.
Attackers gaining access through the use of a third party dongle	OBD	Only use approved devices to connect to the dongle itself, unplug when not in use, make sure that it is up to date on all updates and patches.
Attackers exploiting the connection between OTA center and vehicle to upload malicious software to take control of the vehicle	OTA	Ensure that the connection between the vehicle and the OTA center is secure and has no vulnerabilities.
Attackers gaining access to critical system components (brakes, transmission, ect…) through non-critical components(radio).	CAN	Separate all non-critical system components from critical ones, putting them on their own protected isolated network.

C. Associated CAN Vulnerabilities

After initial development by Bosch in 1983, the CAN protocol was officially released in 1986 and was first featured in production vehicles in 1989. In 1993, the International Organization for Standardization (ISO) accepted CAN as a standard and published ISO 11898 for road vehicles. Since then, CAN has been used as a standard for practically every light-duty vehicle currently in circulation today, and was being pushed to be the only acceptable one in the US federal courts [16].

CAN is the network protocol that connects all in-vehicle equipment such as parking sensors, airbags, active safety system and systems such as navigation and infotainment, and allows them to communicate. It allows the different components to send messages to each other to communicate and work together or let each other know when they are malfunctioning.

The CAN messages, including errors, are called frames. Errors arise when a device reads values that do not correspond to the original expected value on a frame. When a device detects such an event, it writes an error message onto the CAN bus in order to "recall" the errant frame and notify the other devices to entirely ignore the recalled frame [17]. This mishap is very common and is usually due to natural causes, a transient malfunction, or simply by too many systems and modules trying to send frames through the CAN at the same time.

If a device sends out too many errors, then, according to the CAN standards, it goes into a so-called Bus Off state, where it is cut off from the CAN and prevented from reading and/or writing any data onto the CAN. This feature is helpful in isolating clearly malfunctioning devices and stops them from triggering the other modules/systems on the CAN. But if an attacker were to abuse this system by using the OTA mechanism to introduce malware to a subsystem of the vehicle, causing it to malfunction and send out too many error messages, and thus rendering the system inert/inoperable by forcing it into the Bus Off state. This, in turn, can drastically affect the car's performance to the point that it becomes dangerous and even fatal, especially when essential systems like the airbag system, the self-driving system, or the anti-lock braking system are deactivated.

D. OTA Summary

The OTA update mechanism has several good points and benefits in regards to saving money and time when updating a vehicle's software. The benefits are even more apparent with autonomous vehicles considering how important it is that these autonomous vehicles' software is always functioning properly and is secured against vulnerabilities. A major issue with the OTA update mechanism is that it is still relatively new since the technology is considered as emerging. It is not yet apparent what exact vulnerabilities and exploits it is exposed to.

Another factor to consider in assessing OTA updates for connected vehicles is that these updates, if done improperly or maliciously by a hacker, can cause the updated system to be turned off by the CAN, which could result in accidents and death. Because OTA updates can reach software throughout the car, it is a giant attack vector in taking control or compromising a vehicle if the transmission method is not secure. To make sure that does not happen, much research and rigorous testing are necessary.

VI. Concolusions

This paper sheds the light on the security of several features of connected vehicles to determine whether or not they are vulnerable to attacks and identify possible mitigations. The features we investigate include Bluetooth, OBD (On Board Diagnostics) System, Infotainment System, and OTA (Over the air) mechanism. Table I summarizes the vulnerabilities and associated countermeasures for each one of the studied features. For future work, we plan to perform security assessment for more features of connected vehicles. Also, we plan to identify possible mitigation and countermeasures to the exploitable vulnerabilities present in the connected vehicles.

References

[1] NATIONAL VULNERABILITY DATABASE. (December 9, 2017). [Online]. Available: https://nvd.nist.gov/

[2] L. Constantin, Critical Bluetooth Flaws Put Over 5 Billion Devices at Risk of Hacking, Forbes, September 12, 2017.

[3] R. Lekowski, Bluetooth: Just How Secure Is It? May 20, 2016

[4] D. Browning and G. C. Kessler, Bluetooth Hacking: A Case Study, 2009.

[5] T. Bécsi, P. Gáspár, and S. Aradi, Security Issues and Vulnerabilities in Connected Car Systems, June 15, 2015.

[6] The Most Hackable Cars on the Road, August 19, 2015.

[7] C. Fitzgerald, The Most and Least Hackable Cars, August 4, 2014.

[8] Android Auto Developer Site. [Online]. Available: https://developer.android.com/auto/index.html

[9] Ogbo, Obaro. (August 15, 2015). Why Android Auto Scares Me.

[10] Currie, Rodrick. Development in car hacking, (December 5, 2015).

[11] On-board diagnostics. [Online]. Available: https://en.wikipedia.org/wiki/On-board_diagnostics

[12] B. Howard, Hack the Diagnostics Connector, Steal Yourself a BMW in 3 Minutes, July 10, 2012.

[13] OTA Software Update Technology for Vehicles – Highly Reliable and Quick Updates, April 27, 2017.

[14] OTA Updates Driving Connected Car Revolution? August 2, 2017).

[15] Your next car will update itself while you sleep, and maybe watch you too, January 2016.

[16] A Vulnerability in Modern Automotive Standards and How We Exploited It.

[17] The Crisis of Connected Cars: When Vulnerabilities Affect the CAN Standard, August 16, 2017.