



**SECURING CONTROLLER AREA NETWORKS IN VEHICLES VIA PACKET
SWITCHED NETWORK SEGREGATION**

THESIS

Eddie K. Caberto, Captain, USAF

AFIT-ENG-MS-17-M-009

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the United States Government. This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.

AFIT-ENG-MS-17-M-009

**SECURING CONTROLLER AREA NETWORKS IN VEHICLES VIA PACKET
SWITCHED NETWORK SEGREGATION**

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the
Degree of Master of Science in Computer Engineering

Eddie K. Caberto, B.S.

Captain, USAF

March 2017

**DISTRIBUTION STATEMENT A.
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.**

**SECURING CONTROLLER AREA NETWORKS IN VEHICLES VIA PACKET
SWITCHED NETWORK SEGREGATION**

Eddie K. Caberto, B.S.

Captain, USAF

Committee Membership:

Scott R. Graham, Ph.D.

Chair

Lt Col John M. Pecarina, Ph.D.

Member

Maj Samuel J. Stone, Ph.D.

Member

Abstract

As automobiles become increasingly connected via multiple wireless capabilities, the lack of security has become a substantial vulnerability. This growth in functionality and convenience has also increased access to a vehicle's Controller Area Network (CAN). CAN, the primary intra-vehicle network, allows time-sensitive communication between electronic control units (ECU's) that control one or many in-vehicle systems. Although CAN has proven very effective in data transfer, it was not designed for security. While some steps could be taken to add security layers and features to the existing CAN protocol, introducing security inevitably adds cost, data latency, and potentially reduces data throughput. There is a growing need to secure CAN networks without completely changing the protocol.

To improve the security within an automobile without an overhaul to the popular CAN protocol, this research developed the Secure CAN Architecture to provide security primitives at the data link layer. When combined with existing network security techniques, it introduces a number of possible security features. A flexible architecture such as this one provides vehicle manufacturers with an option to securely architect their CAN networks in current and future vehicle designs mitigating specific current, and possibly future, risks on an automotive CAN network. These methods apply to other applications with similar communication protocols.

Acknowledgments

I would like to thank Ryan Gordon; a computer-engineering student from Cedarville University, for the hard work and time spent implementing many of these concepts in our laboratory CAN network. I would like to thank my advisor, Dr. Scott Graham, for the continuous support, motivation, and guidance through the research process. And lastly I'd like to thank my wife for the unwavering support throughout this long and time intensive process.

Eddie K. Caberto

Table of Contents

Abstract	iv
Acknowledgments.....	v
Table of Contents	vi
List of Figures	ix
List of Tables	xi
List of Acronyms	xii
I. Introduction	1
1.1. Background and Motivation	1
1.2. Problem Statement.....	2
1.3. Research Objectives.....	3
1.4. Approach.....	4
1.5. Organization	5
II. Background.....	7
2.1. Controller Area Network	7
2.1.1. History and Application.....	7
2.1.2. Overview.....	8
2.1.3. Vulnerabilities.....	9
2.1.4. Other Related Protocols.....	9
2.2. Review of Vehicular Security Literature	11
2.2.1. Vehicular Security Analysis	11
2.2.2. Vehicular Security Improvement.....	13
2.3. Review of Vehicle Threats and Threat Models	15
2.3.1. Existing Threat Models.....	16
2.3.2. Post Access Activity	20
2.3.3. Attacker Motivation.....	20
2.3.4. Mitigations.....	23
2.4. Chapter Summary	24
III. Design and Implementation Methodology	25
3.1. Objective.....	25
3.2. Design Decisions and Constraints	25
3.3. Assumptions	28
3.4. Proposed Secure CAN Architecture	29
3.4.1. Functional Separation	29
3.4.2. Inspiration	30
3.4.3. 3 Functional Group Model.....	30
3.4.4. Required Hardware and Functionality.....	31
3.5. Core Functionality	34

3.5.1.	Switching and Forwarding.....	34
3.5.2.	Filtering.....	39
3.5.3.	Simulation.....	41
3.5.4.	Entertainment.....	43
3.5.5.	Non-Secure vs. Secure Mode	44
3.5.6.	Over-The-Air Update.....	45
3.6.	Putting it All Together	47
3.6.1.	Hardware.....	48
3.6.2.	Software.....	53
3.6.3.	BeagleBone Applications	55
3.6.4.	Experimentation and Demonstration	57
3.7.	Chapter Summary	59
IV.	Analysis	61
4.1.	Implications of Secure CAN.....	61
4.1.1.	Expandability.....	61
4.1.2.	Source Information	62
4.1.3.	Filtering and Access Control	62
4.1.4.	Improved Performance within sub-CANs	63
4.1.5.	Data Isolation.....	63
4.1.6.	CAN Network Management	64
4.1.7.	Fail Secure and Intrusion Prevention.....	64
4.1.8.	Benefits Over Cryptographic Methods.....	65
4.1.9.	Implications in Other Industries	66
4.2.	Secure CAN Architecture Security Assessment.....	66
4.2.1.	Ethernet Switch Review	66
4.2.2.	Access to a CAN Switch.....	69
4.3.	Application on 2015 Subaru WRX.....	71
4.3.1.	Network Mapping	72
4.3.2.	Assumptions	74
4.3.3.	Network Redesign	74
4.3.1.	CAN Switch Configuration	76
4.4.	Performance Impact.....	79
4.4.1.	Timing Impact	79
4.4.2.	Other Performance Impacts	85
4.5.	Vulnerability/Threat Assessment Framework	86
4.5.1.	Motivation and Approach.....	86
4.5.2.	CarVSS	88
4.5.3.	Results.....	97
4.6.	Chapter Summary	101
V.	Conclusions.....	102
5.1.	Summary.....	102
5.2.	Contributions	103
5.3.	Challenges in Vehicular Security	103

5.4. Future Work.....	105
5.4.1. Intrusion Prevention.....	105
5.4.2. Improved Simulation	105
5.4.3. More ECUs and Mechanical Systems	106
5.4.4. Tools	106
5.4.5. Secure Over The Air Updates.....	107
5.4.6. Potential Vulnerability Assessment Improvements.....	108
5.4.7. Real World Implementation	109
Bibliography	110

List of Figures

Figure 1: CAN Frame	8
Figure 2: Digital I/O channels in a modern vehicle.....	16
Figure 3: Basic CAN Architecture.....	32
Figure 4: Secure CAN Architecture.....	33
Figure 5: CAN Switch between primary and secondary CAN busses.....	36
Figure 6: Extended and Standard CAN Frame bit use.....	37
Figure 7 Example forwarding table	37
Figure 8: Inter sub-CAN communication path	38
Figure 9: HIL portion of the network	41
Figure 10: Access sub-CAN	44
Figure 11: Over-The-Air update process	46
Figure 12: Lab CAN network setup.....	48
Figure 13: BeagleBone Black	49
Figure 14: Raspberry Pi	49
Figure 15: Raspberry Pi 7" touchscreen display	50
Figure 16: CANtact v1.0.....	50
Figure 17: 2015 Subaru WRX combination meter	51
Figure 18: Fabricated transceiver boards.....	51
Figure 19: Nintendo Wii remote	52
Figure 20: NeoIV Fire 2 vehicle network adapter	52
Figure 21: OBD-II connector kit, vehicle side.....	53

Figure 22: Kodi home screen	54
Figure 23: TORCS screenshot	55
Figure 24: 2015 Subaru WRX Basic CAN Network	73
Figure 25: 2015 Subaru WRX Secure CAN network redesign	76
Figure 26: Teledyne Lecroy Wavepro oscilloscope	81
Figure 27: Delay measurement physical setup	82
Figure 28: Oscilloscope screenshot showing processing delay across a CAN Switch.....	84
Figure 29: CVSS Metric Groups.....	89
Figure 30 New Base Impact Sub-Score calculation	94
Figure 31 CVSS Base Scoring equations.....	96
Figure 32: Example Vector String	97
Figure 33: Basic CAN Jeep attack Vector String	99
Figure 34: Secure CAN Jeep attack Vector String	100

List of Tables

Table 1: Subaru WRX CAN communicating ECUs.....	73
Table 2: WRX ECU functional grouping	75
Table 3: Vehicle security categorization example	93
Table 4: CVSS Metric and Rating Chart	95
Table 5: CVSS Severity Rating scale	96
Table 6: Vector String values	97

List of Acronyms

ACL	Access Control List
ARP	Address Resolution Protocol
CAN	Controller Area Network
CDP	Cisco Discovery Protocol
CIA	Confidentiality, Integrity, and Availability
CM	Control Module
CVSS	Common Vulnerability Scoring System
DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
ECU	Electronic Control Unit
GPS	Global Positioning System
HIL	Hardware in the Loop
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
ISO	International Standard Organization
LIN	Local Interconnect Network
MAC	Media Access Control
MFD	Multifunction Display
MOST	Media Oriented System Transport
NHTSA	National Highway and Traffic Safety Administration

NIST	National Institute of Standards and Technology
ODBII	On Board Diagnostic System
OSMC	Open Source Media Center
RMF	Risk Management Framework
SCADA	Supervisory Control and Data Acquisition
SEVECOM	Secure Vehicle Communication
STP	Spanning Tree Protocol
sub-CAN	Sub-Controller Area Network
TDMA	Time Division Multiple Access
TORCS	The Open Racing Car Simulator
TPMS	Tire Pressure Monitor System
UART	Universal Asynchronous Receiver Transmitter
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
VANET	Vehicular Ad-Hoc Network
VLAN	Virtual Local Area Network

I. Introduction

1.1. Background and Motivation

Production automobiles today have advanced capabilities ranging from antilock braking systems to fully automated driving. These capabilities aim to enhance and safeguard the driving experience. Fuel efficiency, performance, luxury, convenience, and safety have all driven the increase in automotive modernizations, with computers controlling almost every function. They do so via nearly a hundred Electronic Control Units (ECUs) networked throughout the vehicle, executing upwards of 100 million lines of code [1]. Hence, vehicles have become increasingly dependent on software and microprocessor-based electronic control. This dependence comes with an associated set of potential vulnerabilities and threats that will continue to increase.

The Controller Area Network (CAN) is the primary intra-vehicle network allowing time-sensitive communication between ECUs. CAN offers benefits in simplicity, robustness, flexibility and speed making it extremely powerful in embedded applications like vehicle control systems. Although CAN has proven very effective in data transfer, it was not designed with security in mind. While some steps could be taken to add security layers and features to the existing CAN protocol, doing so may come at undue cost.

Effectiveness in the automotive sector has led to CAN use in a range of other industries and applications. Broadcast communication, as seen in CAN, is also utilized in various protocols, networks, and functions. While this thesis observes and improves upon vehicle implementations of CAN, the security methods utilized, and the resulting benefits, have very similar application and implication across various industries and broadcasted communication networks.

Vehicular CAN network security is of growing concern today, and can benefit from increased interest. As vehicles continue to advance, we are presented with the opportunity to include improved security from the beginning. The motive and drive to attack vehicles is currently limited, however, cyber security needs to be a forethought in today's critical system designs. CAN networks closely mirror that of supervisory control and data acquisition (SCADA), industrial control systems (ICS), and critical infrastructure networks; and merit the same attention and security focus.

1.2. Problem Statement

As automobiles have advanced technologically, they have become increasingly connected via multiple wireless capabilities (Bluetooth, Wi-Fi, cellular, RF, etc.). Although these wireless capabilities provide useful services, when combined with the high dependence on software and electronic control, the added convenience has also increased the risk of malicious access to a vehicle's CAN network. This provides access to many critical safety and control related systems, which, in this high-risk and high impact domain, has become a substantial vulnerability.

Current solutions to this problem are between cumbersome cryptographic techniques and nascent intrusion detection systems [2, 3, 4, 5, 6, 7, 8]. While there is certainly a place for such methods, introducing security inevitably adds cost, data latency, and potentially reduces data throughput. Maintaining CAN's effectiveness is paramount to a compelling solution. This research explores one possible architectural solution that can improve security, while minimizing overhead and impact to current performance standards within CAN networks.

1.3. Research Objectives

A large majority of today's research focuses on the lack of security and exploitability of modern vehicles, with few researchers dedicated to immediate improvements. The objective of this research effort was to understand the problem, formulate a solution, and evaluate the implications of such a solution as enumerated below:

- Understand the CAN protocol, determine key traits that contribute to its effectiveness in vehicles, and identify weaknesses leading to system vulnerabilities.
- Understand the motivation behind vehicular attacks, discuss possible threat models and scenarios to better guide the specific needs of the proposed solution.
- Analyze the current climate of vehicular security research so as to pinpoint specific areas that are lacking, acknowledge current mitigating solutions, and learn from other findings to potentially incorporate methods that have already proven effective.
- Develop a security solution that not only improves the security of current CAN, but also maintains its effectiveness and limits potential penalties in performance.
- Build the proposed solution so as to provide actual proof of concept.
- Demonstrate all possible capabilities of the proposed solution.
- Evaluate the implications of the proposed solution, its effectiveness, and any penalties that were otherwise unavoidable in its implementation.

Accomplishment of these objectives resulted in the development of a CAN network segregation architecture that utilized custom configured hardware to control the cross flow of messages between otherwise segregated networks. Successful implementation of this architecture motivated the creation of an evaluation framework capable of evaluating the impact of these changes to CAN and perhaps other cyber physical systems. Its application to the

proposed architecture allowed the quantifiable comparison between other less secure CAN implementations.

1.4. Approach

The research presented in this thesis blended aspects from both a qualitative and quantitative approach. It did not solely rely on the review of previous works, nor did it include rigorous collection of data and statistical analysis. Instead it incorporated traits that are commonly found in both respects. Combined it could best be described as a theoretical analysis exploring the implications and possibilities of a newly proposed architecture, and an evaluation of the quantitative measures of its successes and failures. This mixed methods approach aimed to provide a full and comprehensive presentation of the investigation and its results.

This research considered several constraints in safety, logistics, resources, and time that greatly influenced the methods utilized and approach taken. These constraints motivated critical decisions throughout the research process; such as the decision to develop, model, and simulate the necessary functionality of a modern vehicles CAN network. It was safer, more flexible, and equally as effective as demonstration on an actual vehicle. This approach placed nearly all variables under direct control simplifying demonstration, testing, and experimental capabilities.

In some cases, simple observation is all that was necessary to confirm the theories in question. Others would be purely theoretical presenting hypothetical data in textual form for discussion and reflection on the problem. Where applicable, experiments were conducted to capture quantifiable data for further comparison and analysis. This combination of qualitative and quantitative tactics allowed an extremely flexible and forgiving approach that ultimately supported the defined research objectives.

1.5. Organization

The organization of this thesis is as follows:

Chapter II begins by introducing CAN, its history and origins, basic functionality, vulnerabilities, and significance in today's society. It then brings light to the current climate of vehicular security research discussing notable works and recommended solutions in this field. Finally, it presents several threat models and threat scenarios to help describe the alarming capabilities and motives of an attacker.

Chapter III then describes the overall research objective in detail. It discusses critical constraints and limitations that impacted the methodology and approach taken. Based on such limitations, it also discusses any critical assumptions made. Network segregation, functional separation, and other key components of the proposed solution are then described revealing the required CAN Switch hardware that enabled the desired security enhancement. Finally, Chapter III puts it all together by describing all hardware and software utilized to accomplish the overall objective.

Chapter IV presents an analysis of the proposed solution. It introduces implications of the proposed security enhancements. With an idea of all advantages, it then evaluates several different areas that may be seen as disadvantageous. A review of the current solution and its potential vulnerabilities is presented. All methods are then theoretically applied to a real vehicle to determine feasibility. Lastly, it evaluates performance impacts, and presents a vulnerability assessment framework aimed at quantifying the security of the proposed solution.

Chapter V concludes the thesis with a summary of all research efforts and findings; possibilities for future work; challenges in this field; and enumerates the contributions of this research. It shows that this thesis provides a new method of securing CAN networks both

current and future, CAN Switches and the concept of packet switching a CAN network, a method of functionally separating and physically partitioning a CAN network, improved and configurable filtering capabilities, and finally, a mechanism of which might allow a viable vehicle intrusion prevention system.

II. Background

This chapter presents information critical to a comprehensive understanding of the vehicle threat and security environment. It begins with a description of the CAN protocol, its origins and current influence, and identifies weaknesses. With an understanding of the vulnerabilities and weaknesses in CAN, a review of current literature is presented highlighting the climate and significance of today's research contributions, or lack thereof. Lastly, this chapter summarizes the nature of the threats that exist today. It discusses the motives of an attacker, possible avenues of access, activity on the CAN bus, and finally, any mitigation techniques in consideration today. This information provides a realistic view of this evolving field of research setting the stage for the detailed discussion and analysis to follow.

2.1. Controller Area Network

2.1.1. History and Application

The CAN protocol is the primary communication protocol utilized in the automotive industry. It was developed in 1980s by Bosch GMBH to provide a field-bus communication system connecting ECUs in automotive vehicles. Prior to CAN, control functions within a vehicle required dedicated wires between system components. As vehicles advanced, the number of ECUs in a vehicle increased, and the associated point-to-point wiring became expensive, complex, and heavy. In addition to savings in cost and weight, CAN offered benefits in simplicity, robustness, flexibility and speed. Such benefits have made it the de facto standard for intra-vehicle communication. As of 2008 it is mandatory for all vehicles sold in the U.S [9].

In addition to its abundant use in the automotive world, CAN is found in other industries where distributed control is desired. Its diverse applications include agricultural machinery,

production machinery, medical instrumentation, elevator controls, building automation, fairground rides, transportation systems, and industrial automation control components [9][10].

2.1.2. Overview

CAN itself is a two-wire serial communication protocol that provides distributed control and multiplexing. The two wires are often referred to as CAN_H (CAN Hi) and CAN_L (CAN Lo), and the voltage difference between them is used to determine the logical state of the two-wire medium or bus. The bus will be in one of two states, dominant (0) or recessive (1), representing the binary data in transmission. When a node transmits a 0 bit, the bus will result in a positive differential voltage and the dominant state. If a 1 bit is transmitted, the bus will result in a negative differential voltage and the recessive state [11]. If two nodes attempt to transmit simultaneously, the bus will reflect the state of the dominant bit.

Transmitted data is broadcast in a CAN frame, shown below in Figure 1. Frames include arbitration, control, data, and CRC fields. The arbitration field consists of either an 11 or 29 bit message identifier (ID) indicating the type of data being transmitted. The identifier also serves as the frame or message priority. When two nodes try to transmit simultaneously, they will go through arbitration and contend for the bus. The node transmitting the lower ID will maintain the bus in the dominant state longer, and will thus win contention of the bus [11][12], without data loss, (i.e., there is no collision). Following the arbitration phase, the bus will reflect the transmitted data of the winning node while all other competing nodes are now listening.

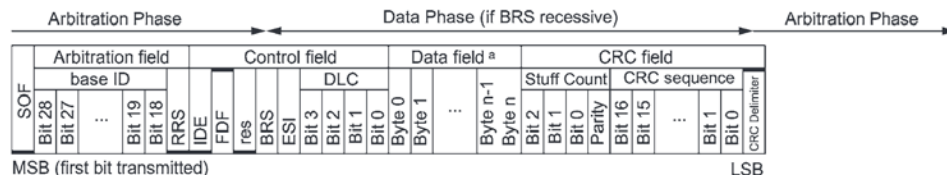


Figure 1: CAN Frame
[12]

The protocol provides configuration simplicity and flexibility, as nodes simply need access to both CAN lines. It provides robustness through a fault tolerant two-wire physical transmission medium lenient to most interference present in vehicles. And lastly, its speed comes from a multi-master priority-based bus access that is non-destructive [11].

2.1.3. Vulnerabilities

These properties offer great benefits to intra-vehicle communication, however they also come with several vulnerabilities. First, CAN frames don't contain the source or destination information and thus the protocol itself does not provide data generally used to authenticate the legitimacy of a message or its sender. Second is the broadcast nature of CAN. Although it can provide network-wide data consistency with a single message, when combined with the lack of authentication, it allows an attacker unrestricted access and transmission capabilities. An attacker has the ability to transmit to all, as well as listen to all. Lastly, the priority message ID scheme that allows uninterrupted transmission of higher priority (lower ID) messages also provides an easy avenue for an attacker to deny messages of lower priority. Sending the highest priority possible provides an easy denial of service (DoS) attack. There will of course be vulnerabilities that exist within the ECUs themselves, however these are the primary vulnerabilities of CAN that an attacker would try to leverage to compromise a vehicles network. They are the factors driving the need for improved security within CAN.

2.1.4. Other Related Protocols

While CAN is the de facto standard for intra-vehicle communication, there are additional network communication protocols often found in modern vehicles. Examples are LIN, FlexRay, and MOST. Each protocol has different strengths, and a vehicle may employ several protocols to accomplish different objectives. A brief description of each follows:

- *Local Interconnect Network (LIN)*: LIN is a single wire Universal Asynchronous Receiver Transmitter (UART) based serial communication protocol. It handles data rates up to 20 kBit/sec and is typically used where a large data rate is not required. Examples of its use are in body related ECUs for communication between sensors and actuators [13].
- *FlexRay*: FlexRay is a bus protocol with a maximum data rate of 10Mbit/s. It uses a Time Division Multiple Access (TDMA) method for access to its dual transmission channels (fault tolerance and redundant) [13]. FlexRay is the most expensive of the protocols presented here, typically utilized for high-end demanding applications such as high-performance powertrain, drive-by-wire, active suspension, or adaptive cruise control [14].
- *Media Oriented System Transport (MOST)*: MOST is yet another serial bus communication protocol. It can offer speeds of up to 14 Mbit/s (asynchronous) or 24 Mbit/s (synchronous) making it the highest data rate protocol presented here. It's typically used in automotive multimedia systems (entertainment, navigation, or telematics) transmitting audio, video, voice, and control data. Interestingly, unlike the other protocols, to include CAN; MOST messages encapsulate both sender and receiver address information providing management features not seen elsewhere [13].

While the research presented in this thesis is primarily focused on CAN, it's important to understand that CAN is not the only network in the automotive environment. For overall security, these protocols and networks must also be considered. Modifications which enhance CAN security must properly coexist with other interacting networks.

2.2. Review of Vehicular Security Literature

The field of vehicular security has grown immensely in the past decade. Advancements in modern automobiles have provided unforeseen opportunities. Newer technologies designed with convenience in mind leave vehicles vulnerable, and critical systems exposed, with little or no defensive capabilities. As a result, much research focuses on the lack of security and exploitability of modern vehicles, with little treatment of security improvements and possible solutions. This section discusses the most recent and relevant findings that have shaped and will continue to influence the field of vehicular security. It discusses works as they relate to two categories, Security Analysis, and Security Improvement. Those that heavily influence the research presented in this thesis are discussed further in future sections.

2.2.1. Vehicular Security Analysis

In 2010 a team of researchers from the University of Washington and University of California San Diego conducted an experimental security analysis on an undisclosed vehicle setting the stage for vehicle security related research. They demonstrated that once access is gained to a vehicle's CAN network; an attacker could manipulate a number of different control systems. Specific examples include disabling the brakes, selectively braking individual wheels and even stopping the engine [15]. This research was inappropriately criticized for the assumption that an attacker already had access to the vehicle's internal CAN network. The argument is that an attacker who was connected to the internal network already had physical control of the vehicle, so electronic control was moot. What the critics failed to recognize is that there are growing ways to reach the internal network, and that upon achieving this, they would have almost complete control of the vehicle, remotely. In 2011 they went on to examine wireless attack vectors in [16] where they demonstrated the feasibility of remote compromise via some

subtle avenues, including a malformed file on a compact disc played in the car's radio system. Their findings in external attack vectors led to the categorization of possible threat models, an analysis of the vulnerable systems, and a threat assessment evaluating the usefulness of a vulnerable system once compromised [16]. Their work is discussed more in section 2.3.

Vehicle security gained significant notoriety when a remote attack was successfully demonstrated on a 2014 Jeep Cherokee. Researchers Charlie Miller and Chris Valasek were able to send commands via the “always on” cellular radio to the internal CAN network through the Jeep's infotainment system. This allowed remote control over the radio, air conditioning, windshield wipers; and physical functions such as steering, transmission, and brakes [17]. They provided an alarming view of the vulnerabilities and possibilities that exist.

Research in [6] and [8] focused heavily on wireless remote attacks, taking an exceptional approach conducting much of their work on a live CAN network in an actual vehicle. While these efforts generated significant publicity, there are many who paved the way providing practical and meaningful results in a laboratory environment. Hoppe et al. in 2011 [18] explored four different attack scenarios each targeting different vehicular systems. They successfully exploited the electric window lift, warning lights, airbag control system, and gateway ECU, demonstrating the compromise of these systems within a lab setup. They further analyzed these exploits in [18] and presented two possible countermeasures aimed at addressing the vulnerabilities exploited in their experiments. Like Hoppe, others have analyzed the security of individual components within the vehicle such as [19] focusing on the Tire Pressure Monitoring System (TPMS).

The primary focus of the work identified previously was to identify vehicle cyber security problems and advocate for solutions. The following section will discuss the efforts in search of possible solutions.

2.2.2. Vehicular Security Improvement

A majority of the research in this field analyzes existing security flaws. There is little focus on immediate and achievable solutions. Kleberger et al. in [2] provide a clear and comprehensive review of research in security improvements. Their work presented research proposing improvements in Architectural Security Features, the use of Intrusion Detection Systems (IDS), and honeypots. In addition to the research Kleberger discussed, this section will identify other sources and possible extensions. Lastly it will also discuss research aimed at securing inter-vehicle communication.

Cryptographic techniques are a principal component of nearly all security technology. This is also true in vehicular security. Most of the work identified in [2]’s Architectural Security Features involved some form of cryptography. Wolf et al., in [3], discussed various cryptographic primitives for data encryption, decryption, signature generation, verification, and protocols identifying the security services they can provide and their relevance to vehicular security applications [3]. In [13] they also proposed improvements by use of a security gateway that should authenticate controllers, encrypt communication, and filter based on Message Authentication Codes (MAC). This gateway acts as a central security manager for all communication between networks and protocols like those identified in 2.1.4, actively managing data confidentiality and authentication via symmetric/asymmetric encryption and MACs. Additionally [4], [16], and [6] each utilize cryptographic methods to accomplish some

combination of data confidentiality/integrity, and authentication. They all differ in their approach but aim to solve the same problem.

Another type of security enhancement often presented is Intrusion Detection Systems (IDS), offering solutions that monitor a vehicle network for malicious activity. Researchers at the University of Michigan suggested a possible solution for intrusion detection on CAN networks called Clock-based Intrusion Detection System (CIDS). Their proposed system fingerprints ECUs on a CAN network utilizing clock skew. Using those fingerprints, CIDS can identify if an ECU may have been compromised [7]. Other methods propose the use of time intervals between messages as the criteria to detect an intrusion [8]. Kleberger [2] identified similar methods such as one proposed by Hoppe et al., utilizing an anomaly-based approach. Such an IDS could observe traffic on the CAN network and track the frequency at which certain messages are sent on the bus. It could then “detect” an intrusion when this frequency differs from a predefined baseline [2]. An IDS has the potential to provide some security benefits. However, the question remains of what to do when an intrusion is detected. This concept is discussed further in section 4.1.7 as the research presented in this paper offers one possible solution to this specific problem.

Lastly, research aimed at securing the inter vehicle communication, Vehicle-to-Vehicle (V2V) or Vehicle-to-Infrastructure (V2I) communication, is expected in the near future. *Secure Vehicle Communication* (SEVECOM) [20] is one example dedicated to ensuring the security of future vehicle communication networks. That work addresses some of the inherent challenges with securing V2V and V2I communication to include tamper resistant hardware, cryptosystems, validation and testing, etc. Their objective is to define a security architecture for such communication networks [20]. Raya and Hubaux [21] introduced the concept of Vehicular Ad-

Hoc Networks (VANET), discussed the challenges in design and implications of securing them, and identify what security is in VANETS. They further analyzed threats, identified security requirements, reviewed relevant security technology, proposed an architectural solution and finally analyzed its robustness [21]. Given the uncertain security aspects of these future communication networks, security must be implemented in depth, hardening any and all elements that can be hardened, especially the vehicles components. These networks will only be as secure as the nodes communicating in them.

Many of the ideas in the works discussed here overlap, and in fact, complement each other. Collectively they aim to improve the security in vehicles. One common problem among many of these proposed solutions is the high volume of required overhead. This thesis proposes a solution with reduced overhead among critical and time-sensitive systems. This will be discussed further in section 4.1.8.

2.3. Review of Vehicle Threats and Threat Models

This section analyzes the current vehicle threat environment as well as introduces several concepts crucial to this research and the analysis to follow. It elaborates on attack related works introduced in the previous section and in doing so presents a realistic view of the possible threats that exist today.

While the work presented in this thesis does not pertain to any specific type of attack, it's extremely critical to consider all aspects of an attack. Much like Koscher et al., in [15], this work assumes access to and communication on the CAN bus thereby removing the actual attack from scope. Because this research does not directly involve the analysis of vulnerable systems and the exploitation of such, it's extremely important to address the range of possibilities.

2.3.1. Existing Threat Models

There are several different threat models categorizing the large set of potential I/O channels in current automobiles. These models were developed by Checkoway et al., and aim to present threats and vulnerabilities as they relate to the entire automotive platform [16]. When combined with the inherent flaws in CAN, these vulnerabilities shape the current vehicle threat environment.

Checkoway and his team conducted an experimental analysis of automotive attack surfaces where they considered all modern I/O capabilities present in today's automobiles (see Figure 2). They then placed the I/O capabilities into 3 categories; indirect physical access, short-range wireless access, and long-range wireless access; and identified a threat model for each. These threat models are summarized below. In addition to these three, a simple threat model not identified in [16] but addressed in earlier work of [15] is that of direct physical access.

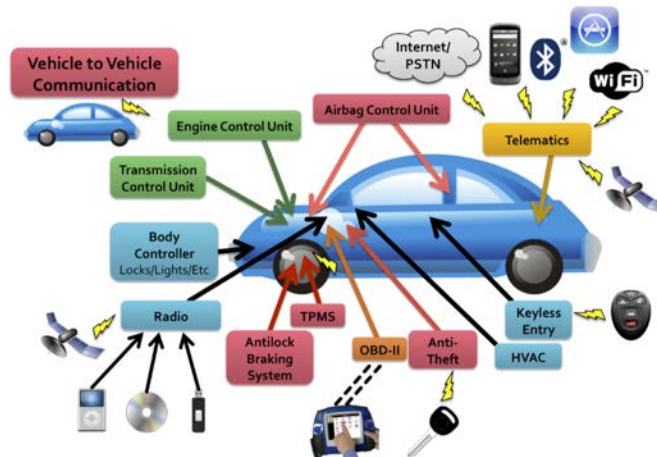


Figure 2: Digital I/O channels in a modern vehicle [16]

2.3.1.1. Direct Physical Access

Physical access here simply refers to accessibility gained directly via physical interfaces within the vehicle. This describes the situation where the attacker has physical access to (can

directly connect to electronic interfaces within) the vehicle. This would include momentary access allowing the attacker to place a malicious transmitter in the vehicles On Board Diagnostics (OBD-II) port later enabling remote communication. Another example of direct physical access is the ability to intervene in the component supply chain. This could place a malicious device within the vehicle legitimately [15]. Physical access is difficult to consider for the simple fact that if an attacker has physical access to the vehicle, the security of the network isn't the primary issue. However, it does present a valid model for attackers today and there is certainly still a need for network security measures.

2.3.1.2. Indirect Physical Access

Indirect physical access refers to second-order accessibility via physical interfaces in a vehicle. Unlike direct physical access, access is gained via devices that are legitimately granted physical access to standard physical interfaces in the vehicle. The two identified physical interfaces are the On Board Diagnostics (OBD-II) port and the Entertainment System. The OBD-II port provides direct access to a vehicle's CAN network and is federally mandated in the U.S [16]. The entertainment system is a familiar and standard component in nearly all late model vehicles.

These two interfaces are often physically connected to various different devices, cables, discs or potential transmission mediums in a variety of legitimate circumstances. The OBD-II port for example, is commonly accessed by service personnel during routine maintenance [16]. Such maintenance might be an oil change, tire replacement, tire alignment, 30,000-mile checkup, etc. A simple example requiring such access is switching from a winter set of tires to a summer set of tires. Each set of tires might contain different tire pressure monitor system (TPMS) sensors which require that a mechanic connect a diagnostic tool to the OBD-II port in the vehicle

and reprogram the ECU managing TPMS sensor input to allow the new set of TPMS sensors to communicate the appropriate tire pressure to your vehicle.

There are a wide variety of diagnostic tools intended to interact with the OBD-II port. Many are available to the public for reasons such as reading/clearing error codes or reflashing ECU's for performance reasons. Regardless of the tool itself, it's also common for these tools to require connection to a PC for full/proper functionality. Like the OBD-II port, the entertainment system might also physically interact with external mediums. Examples are CD's, thumb drives, iPods, or other multimedia devices. Like the diagnostic tools, these mediums may also require connection to a PC for full/proper functionality [16].

The second order access that Checkoway et al. refer to is an attacker's access to the diagnostic tool, CD, thumb drive, iPod; or even the PC that manages these devices. If any one of these are compromised, they can provide an attacker indirect physical access to the OBD-II port, entertainment system; and further, a vehicles CAN network. Checkoway et al. were able to demonstrate successful attacks via CDs and vehicle service visits.

2.3.1.3. Short-Range wireless Access

Short-range wireless access is considered to be remote access within a few hundred meters. ECUs utilizing short-range wireless communication offer potential entry points for attackers. Access in this form will leverage interfaces such as Bluetooth, Remote Keyless Entry, RFID keys, TPMS, and Wi-Fi [16]. Nearly all of these technologies are standard on many of today's vehicles. Checkoway et al. states that "For all of these channels, if a vulnerability exists in the ECU software responsible for parsing channel messages, then an adversary may compromise the ECU (and by extension the entire vehicle) simply by transmitting malicious

input within the automobile's vicinity" [16]. In this threat model, Checkoway and his team were able to successfully demonstrate attacks via Bluetooth.

2.3.1.4. Long-Range Wireless Access

Long-range access refers to remote access from distances generally over one kilometer. Access is accomplished over long distance digital channels of two different forms, broadcast channels and addressable channels. Among these channels are various technologies providing possible remote access to attackers.

Broadcast channels often provide services that individual vehicles can receive selectively. Examples include the Global Positioning System (GPS), Satellite Radio, Digital Radio, Radio Data System, and Traffic Message Channel. These services are usually received and processed by the vehicle's infotainment/entertainment system, which is the most likely compromised vehicular system [16].

Addressable channels are communication channels where nodes are individually addressable usually communicating over dedicated links via cellular networks. They are generally part of the telematics system in a vehicle. Some examples are Ford's Sync, GM's OnStar, Toyota's Safety Connect, Lexus' Enform, BMW's BMW Assist, and Mercedes-Benz' mbrace. These services, although convenient in a number of different ways, can be attractive to a determined attacker. Checkoway and his team were able to successfully demonstrate attacks via their experimental vehicle's telematics system. The exploit involved bypassing authentication, modifying the default timeout period allowing them to exploit a buffer flow and ultimately download and execute malicious code [16].

2.3.2. Post Access Activity

Literature has divided CAN threats into three categories that encapsulate all attacker activity on the CAN bus [22]. Once access has been gained via the many possible attack vectors, and according to intent, an attacker could engage in one, two, or all of the below CAN threat activities to accomplish nefarious objectives.

- ***Eavesdropping*** – Refers to listening to communication or traffic on the bus, easily done in a broadcasted network as all communication is broadcasted over the shared transmission medium, in this case the CAN bus.
- ***Unauthorized Transmission*** – With access to the bus, an attacker has the ability to transmit arbitrary messages. This refers to all forms of unwarranted communication on the bus, to include fuzzing, replays, injection, etc. All involve the injection of traffic that isn't a part of legitimate ECU communication and functionality.
- ***Denial of Service (DoS) attacks*** – Refers to an interruption in expected vehicle or ECU functionality. This is easily accomplished in a broadcast network and more so in CAN. An attacker would simply need to keep the bus in a dominant state, by repeatedly sending messages with a higher priority, or a lower identification. Once in the dominant state, all other ECU's attempting to communicate will no longer be able to.

2.3.3. Attacker Motivation

In light of the vulnerabilities in the CAN protocol and the many possible avenues to leverage those vulnerabilities, one might ask whether these pose a relevant threat today? To answer this question one must consider motivations for an attacker. What drives them to exploit these weaknesses and what do they stand to gain by doing so? Possible motives usually fall into one of four categories: financial gain, surveillance, disruption, and elimination. To better

describe and illustrate these, we present four different scenarios based on real world occurrences when available.

2.3.3.1. Scenario 1 – Surveillance

Surveillance offers benefit to an array of entities both good and bad (investigator or spy). Regardless of the entity and their intent, their primary focus is to monitor and/or gather information. Although no real world example exist to date (that we know of), Checkoway et al. found that it was possible for an attacker to record data from within the vehicle cabin. They accomplished this via a compromised telematics unit and an in-cabin microphone typically used for Bluetooth hands-free calling while driving [16]. Such information could be used in a number of different ways such as criminal conviction, blackmail, or espionage.

2.3.3.2. Scenario 2 – Financial Gain (Theft or Ransom)

There is a wide range of possible scenarios that could stem from financial motivation, most of which target a single vehicle. In the following paragraph, a simple example of theft via OBD-II port access, reported to be common in BMW vehicles, demonstrates the exploitation of the CAN bus [23]. Primary physical barriers to theft include door locks, the anti-theft security system, and the ignition system. Each of these systems interact with a key to grant access to and start the vehicle. Vulnerabilities in these systems allow thieves to circumvent these obstacles.

The attacker would first need to gain access into the vehicle, usually done in one of two ways. First by spoofing the key fob's RF communication with the keyless entry system; and second is by physical force, breaking a window or picking a lock. Once access is gained, the attacker would then have access to the OBD-II port located under the steering wheel in most vehicles. Utilizing a key/transponder programming device (commonly used for BMW and Mini

Cooper), program a new key via the OBD-II port [24]. The process takes no more than 15 seconds. Once programmed, push the start button to turn on the vehicle, and drive away [23].

2.3.3.3. Scenario 3 – Disruption

We relate this scenario to Hacktivism in which the true motivation is political or social. The attacker's primary focus is disruption, perhaps making a statement, or to cause harm or financial strain. It's likely that an attack of this form would focus on impacting many vehicles.

In 2010 a 20-year-old employee of Texas Auto Center in Austin Texas sought revenge after being laid off. The disgruntled employee, with access to company resources, utilized a vehicle immobilization system to disable 100 vehicles. Customers of these vehicles experienced uncontrolled horn honking or a vehicle that was undrivable [25].

The compromised system in this attack was a web based remote system called Webtech Plus used to motivate delinquent customers to make their car payments. It requires a small black box to be installed under the dash providing wireless access the vehicles CAN network [25]. Using this system, the disgruntled employee sent unauthorized commands to multiple CAN networks impacting much more than a single vehicle.

2.3.3.4. Scenario 4 – Elimination (Kidnapping or Worse)

Although attacks of this nature have not yet been reported, they are possible. The attacker focus here is a specific individual, using a vehicle as a means to access the human target, possibly a high value individual.

An attack such as this might begin similar to the Jeep hack discussed earlier, via wireless compromise of the infotainment system providing access to the CAN bus. The attacker could then inject messages to unexpectedly stop the vehicle restricting control as well as the ability to exit. This can further enable apprehension of the human occupant. If the intent is elimination of

the target, the attacker could inject commands to increase speed, reject brake pedal input, and if able, steer the vehicle off the road for a destructive outcome.

2.3.4. Mitigations

Section 2.2 presented recommended security measures found in current literature. Mitigation techniques range from data encryption and authentication to anomaly based intrusion detection. Among these security solutions, cryptographic techniques are the most common due to simplicity in implementation. Security is implemented at higher layers (application, within the ECU itself), and as a result, the security in CAN (a layer 1 & 2 protocol) is still left vulnerable. While these methods do offer enhancements in data confidentiality and limited authentication ability, they do not prevent access to the CAN bus. They don't prevent an attacker from sniffing, capturing, and replaying messages with the encrypted data. Although the data is encrypted and unknown to an attacker, injecting such traffic remains possible, and if done at the wrong time, can be disastrous.

The ability to identify an intrusion or malicious activity on the bus is certainly appealing, however, without response mechanisms, there is very little an IDS can do to halt an attack. The ability to pre-empt an attack requires the concept of an Intrusion Prevention System (IPS), which according to Kleberger et al., has yet to be presented in the vehicle setting [2].

Many of the security recommendations made in related works are not available in current vehicles. Checkoway et al. posed the question of why even simple and proven techniques are not already incorporated into some of today's high tech vehicles [16]. One answer is that vehicles have yet to be characterized as worthwhile targets. They have only recently become connected, and until now, have had no reason to incorporate security into what was designed to be an isolated network. Additionally there are challenges to incorporating some of these solutions.

One in particular is the lack of standards amongst vehicle manufacturers. What might secure one vehicle may not be effective on another. ECUs, network topologies, and network configurations differ greatly between manufacturers and can even differ between vehicle models from the same manufacturer. There is certainly a place for many of these mitigation techniques, however there is still work to be done, and solutions that might overcome some of these challenges.

2.4. Chapter Summary

There is no question that modern automobiles rely heavily on CAN for intra-vehicle communication. While CAN provides simple, robust, speedy and reliable communication, the protocol does not possess any inherent security mechanisms. With access to the CAN bus, an attacker has the potential to affect all computer controlled aspects of a vehicle that are connected to the CAN bus. Literature has shown that the increased connectivity in modern vehicles provides an array of different potential access points each with their own difficulties and advantages. Whether driven by financial gain or the urge to disrupt, an attacker, if truly motivated, will find a way to obtain access. Security measures to prevent such access are rather limited. Of the known mitigation techniques, many involve cryptographic methods, which come with associated challenges and costs. The industry is still in need of a feasible combination of security methods that provide sufficient protection and performance. The information presented here shows that these threats (physical and remote) are very real and the automotive industry can benefit from viable solutions.

III. Design and Implementation Methodology

3.1. Objective

The intent of this research effort was to demonstrate potential changes to the architecture of a vehicle CAN network that may offer improvements in the security of CAN implementations without a change to the protocol itself. In addition to demonstrating the possibilities of an alternate implementation, this research examined possible advantages and disadvantages of the proposed architecture.

Developing an entirely new protocol with security in mind would require years to engineer, standardize, and implement, leaving current and future designs vulnerable. Additionally, introducing security to CAN would add cost, data latency, and potentially reduce throughput, possibly degrading an already effective and reliable communication protocol. Still, there is an immediate need for improved security and maintained performance within CAN networks. This research provides one possible solution that minimizes security overhead and maintains current performance standards.

3.2. Design Decisions and Constraints

There are various different methods to approach and evaluate engineering solutions, each of which come with their advantages and disadvantages. For the vehicular security problem, four different evaluation methods were identified as feasible options. First is an analytical approach in which all concepts and architectural changes would be discussed and analyzed through a theoretical lens. This is the tabletop approach that while less accurate, requires less time, energy, and resources. Second, is the simulated approach, in which all elements of theory and design are implemented in software and run according to predetermined and programmed

specifications. This can improve the accuracy in findings, but is typically more difficult, time consuming, and expensive than a purely analytical approach. Third is a measurement approach in which all concepts are designed, implemented, and built exactly as they would exist in a real world application. For vehicular security, this would equate to architectural modification and full implementation on a production automobile. While this approach greatly increases credibility, it does so at great cost in time, energy, and resources. Lastly is an emulated approach. It represents a hybrid approach of simulation and measurement, involving the reproduction or imitation of key architectural components that are central to the demonstration capabilities. This approach provides sufficient accuracy at reasonable cost in terms of time, energy, and resources. This was the approach taken and presented here.

The decision was made to develop, model, and simulate the necessary functionality of a CAN network rather than investigate and implement the proposed architecture on an actual vehicle. While this decision certainly limits some capabilities, it comes with several advantages. First, it is much safer and cheaper than modifying an actual vehicle. Second, its development greatly enhanced the understanding of CAN networks and their operation in vehicular environments. Next, it allowed for unrestricted control of network functionality and behavior. With nearly all ECUs and network nodes developed in house, any required system modifications could simply be applied when and wherever needed. This level of control also simplified the experimental and testing phase. Simulation further limits the effects of real world factors reducing the overall complexity of the CAN network allowing research to focus on the essentials.

Emulation being the hybrid approach that it is, allows for the incorporation of real vehicle components into the network. Unfortunately, this proved to be rather difficult.

Incorporating real parts required knowledge of manufacturer proprietary vehicle design specifications, which generally, are not available to the public. This can include information such as the number of networks, network topologies, CAN message ID scheme, ECU specifications, ECU dependencies, timing constraints, etc. The lack of such information limited the parts worth pursuing as well as the ease of their incorporation. These details can be obtained through time-consuming reverse engineering, however the cost in time and effort to accomplish this outweighed the perceived benefit. Emulation removes the need for such data providing a low cost, low risk, and low data test environment.

While information may have been limited, several simple components were incorporated into the overall network. They were selected from a 2015 Subaru WRX and included a combination meter (instrument cluster), and ignition key module. They were primarily chosen for their ease of incorporation, and their physical and visual feedback in the network. The remainder of the network was created and developed using inexpensive microcontrollers. These developed ECUs were designed to model real working ECUs emulating their basic functionality and eliminating dependencies that normally exist in a real world CAN network. The result is a lab setup with limited but legitimate ECU interaction.

Without a complete vehicle, the lab network lacks representative interaction between ECU's that in most cases would be dependent upon real world conditions they're exposed to. In turn it also lacks the actual CAN traffic expected on an real CAN bus; traffic that is dependent upon the make/model of the vehicle, actual ECU's on the network as well as the driving conditions of the vehicle itself. This limits the actual CAN traffic seen in the network and future work includes a more advanced simulation capability allowing the simulation of much more advanced systems.

These design decisions and constraints have impacts that carry into the resulting research analysis. Many of these impacts center on the absence of information about the vehicle and its components, system dependencies, performance requirements, or system vulnerabilities. If these details were available, accuracy across all evaluation criteria could be improved. Many of these analysis specific effects were overcome with key assumptions that allowed the abstraction of uncontrollable factors and meaningful results to be obtained.

3.3. Assumptions

Many of the assumptions made are related to the fact that this work was conducted within a laboratory environment lacking the real world aspects of a CAN network. They are the direct result of the constraints identified in the previous section. They further allow the circumvention of conditions that were outside of laboratory control. Below are the key assumptions made.

- ***Basic CAN Network Configuration*** – With ECUs, network topologies and configurations differing greatly between modern vehicles, the assumption is made that critical systems of interest are accessible via a single primary CAN bus. To generalize the analysis of security a vehicle's network is assumed to be in a basic configuration. If a network consisted of multiple busses, bridgeable gaps exist as shown in [15]. For a particular make or model, the secure CAN configuration may look different than the one presented in this thesis including appropriate gateways, partitions, and protocols specific to that vehicle's functionality.
- ***Functional Dependencies*** – ECUs and sub-systems within a vehicle can be cleanly separated by functionality. In other words, there are no critical dependencies between functional areas within a vehicle. Explained further in Section 3.4.1.

- **Standard CAN Format** – Vehicles today primarily utilize standard CAN frame format with an 11 bit identifier field versus the extended 29 bit identifier field.
- **CAN Bus Access** – Removing the actual attack from scope, it is assumed that an attacker already has access via one of the possible vectors discussed in section 2.3.1. This research enhances the security of the bus but does not prevent an attacker from gaining access; prevention in that respect is out of scope of this research.

3.4. Proposed Secure CAN Architecture

The proposed CAN implementation is called the *Secure CAN Architecture*. It encompasses several key elements that enable the desired functionality and security improvement. Although conceptually simple, these architectural changes have the potential to transform automotive networks of the near future. To provide a basic understanding of the Secure CAN approach, the proposed architecture will be introduced conceptually before functionality; design, and implementation details are addressed later.

3.4.1. Functional Separation

In an analysis of vehicular CAN networks, a functional difference was observed between operational control of elements, and access to, a CAN network. Currently, the only required access point in a vehicle is the OBD-II port, included for diagnostics, configuration and testing purposes. All other potential access points are made through other systems or ECUs that incorporate wireless communication methods such as those identified in 2.3.1. In most cases, it's safe to assume that these ECUs are functionally different than those related to vehicle safety and control. Specifically, these systems are not required for basic vehicle functionality. That raises the question of why combine them? Thus, efforts were focused on the concept of “functional separation.”

Similar separations exist in some vehicles today where real-time control and safety related systems are isolated to a high-speed CAN while other less critical ECUs are on a low-speed CAN. If separated by an intelligent gateway rather than a hub like interconnect, it would offer limited control and filtering capabilities between CAN networks, purely filtering on the bits in transmission. This particular design choice appears to be for performance reasons rather than security. However, this concept can assist with security as well. So a different question is asked; why not separate these functions for both security and performance reasons? If done appropriately, benefits could be achieved in both performance and security.

3.4.2. Inspiration

Inspired by current network security techniques, efforts focused on a review of traditional network switches, which provide both physical and logical separation of nodes depending on configuration. Virtual Local Area Networks (VLANs) logically partition nodes into separate areas according to administrator-defined criteria such as function, location, owner, etc. Switches also provide basic filtering by use of Access Control Lists (ACL) selectively controlling traffic based on various conditions. These capabilities closely model the security effects desired in CAN. Interested in applying similar methods to a CAN Network, the focus transitioned to determining the feasibility of switching a CAN network. Is it possible and would it provide similar security enhancements in the CAN environment?

3.4.3. Three Functional Group Model

If functional separation were feasible, ECUs would first need to be categorized based on functional significance. They would be grouped with similar ECUs to help eliminate any potential security conflicts that might exist between non-similar systems. In a brief review of modern vehicle networks, it was hypothesized that most ECUs can successfully be categorized

into one of three functional groups; Access, Critical, and Non-Critical. The Access group would consist of all ECU's that have a potential entry point vulnerable to an attack. Such ECU's exist in the head unit (incorporating wireless protocols such as Bluetooth, Wi-Fi, and cellular), the on-board diagnostic system, or the keyless entry system for example. These are ECU's in systems such as those identified by Checkoway et al., and discussed in section 2.3.1. The Critical group would contain all critical ECU's that are safety and control related such as the engine control module, power steering system, or anti-lock braking system. Lastly, the Non-Critical group would contain ECUs that aren't directly in control of safety or control functionality such as the instrument cluster or multifunction display.

3.4.4. Required Hardware and Functionality

To accomplish these desired effects, physical hardware was developed to model the functionality of an Ethernet switch; called a "CAN Switch". Its development provides a mechanism that enables the desired layer 1 and 2 separation of ECU's much like switches do in traditional networks. To illustrate their utility and applicability in the Secure CAN Architecture, a comparison between a basic CAN configuration and that of the new proposed hierarchical approach is presented.

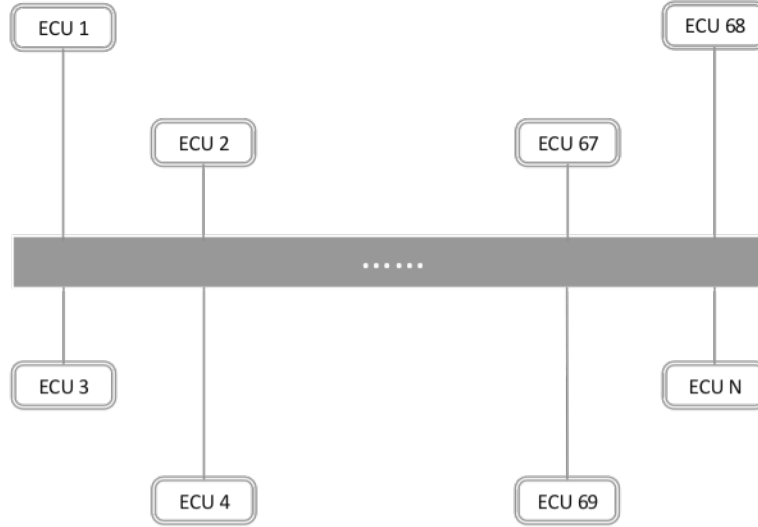


Figure 3: Basic CAN Architecture

Figure 3 above shows a basic CAN bus architecture found in modern vehicles, consisting of n ECU's communicating over a primary bus. All ECUs are also communicating using the standard CAN frame format and the associated 11-bit message ID.

In contrast to the basic CAN architecture, Figure 4 below shows the proposed Secure CAN Architecture, as configured in the laboratory. ECUs are separated into three functional areas corresponding to the three functional groups identified in the previous section; Access, Critical (control or safety), and Non-Critical (status and display). There are three CAN Switches each of which extend the primary bus (CAN1) to one of three secondary CAN busses (CAN0's). Each CAN switch and their respective CAN0 combine to create a new and separated sub-CAN Network (sub-CAN). These sub-CANs also correspond to the functional groups already identified. All ECUs would now connect via one of the secondary busses, located within the sub-CAN and area corresponding to a particular functional group.

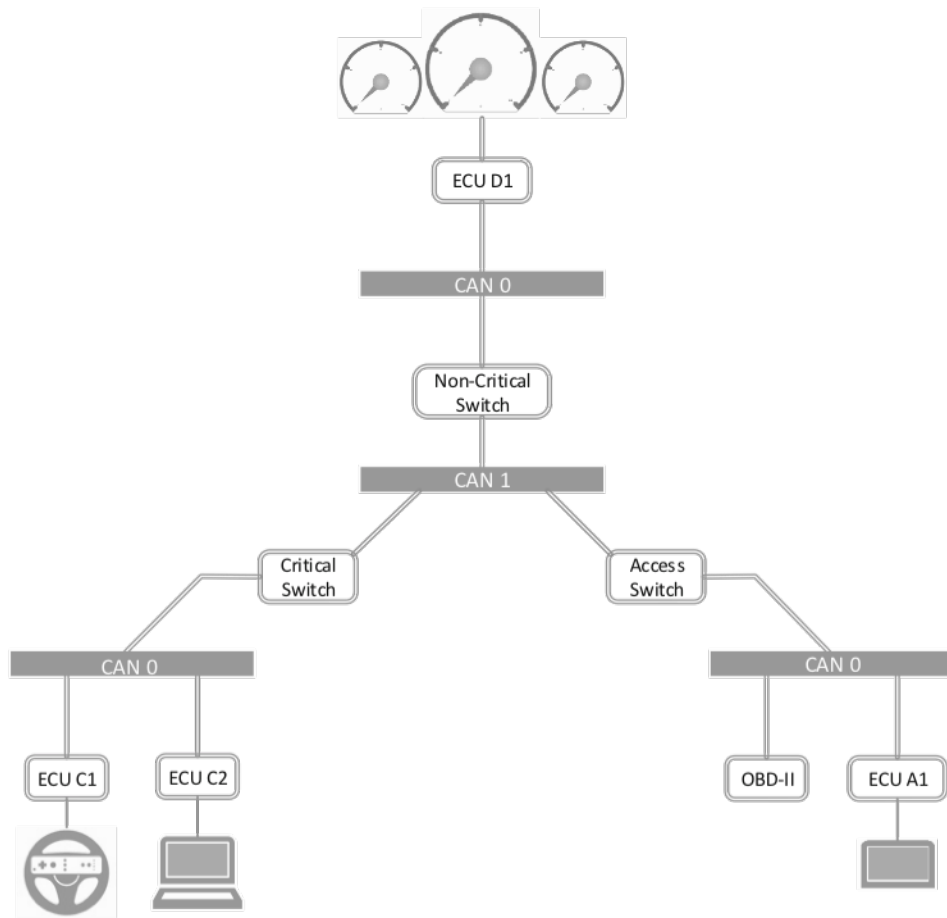


Figure 4: Secure CAN Architecture

With this hierarchical approach, and the introduction of sub-CANs, intelligent filtering logic can be applied at the CAN switches implementing ACLs to permit or deny traffic based on sub-CAN information. Because each CAN message would originate from one of the possible sub-CANs, CAN switches can provide details as to the source sub-CAN of each message. (Recall that CAN does not include source or destination information in each CAN packet. As CAN is a broadcast protocol, it is sufficient to include an identifier for the type of information being provided, rather than routing information.) This point is crucial to the security features of this architecture and is discussed in greater detail in Section 3.5.

3.5. Core Functionality

With a conceptual understanding of the secure CAN architecture, actual implementation is dependent on several core functions. These functions were absolutely necessary to implement a Secure CAN prototype and demonstrate its potential. Note that there are a number of different ways to implement these functions. The methods discussed here represent one solution set.

3.5.1. Switching and Forwarding

3.5.1.1. Ethernet Switching

Switching in traditional networks refers to the action of receiving, processing, and forwarding data frames according to destination information provided within the frame itself. Ethernet switches provide this functionality, making them an essential aspect in modern computer networks. They do so by first receiving an Ethernet frame on one of multiple input ports, processing the destination media access control (MAC) address contained within the frame, consulting a forwarding table to find the output port that corresponds to the appropriate MAC address, and finally forwarding the data frame to the switch port corresponding to that address.

There are several key elements that allow this process to occur. An Ethernet frame contains a source or transmission MAC address as well as a destination or target MAC address. The source information alone allows an Ethernet switch to actively fill their forwarding table, a lookup table containing all MAC addresses encountered and the port that provided that MAC address information. In conjunction with this lookup table, the destination MAC address allows the Ethernet switch to forward data frames out the switch port that corresponds to that address. Without this information or these capabilities, switches would function as simple hubs forwarding any received frames out every port.

3.5.1.2. CAN Switching

Switching on a CAN network differs largely in that CAN is strictly a broadcast network. One sender broadcasts a CAN frame and all nodes on the network receive it and process the information if designed to do so. As a result, there is no need for any source or destination information to be embedded within a CAN frame as they are in Ethernet frames. Without such information, a forwarding table cannot be generated and frames cannot be forwarded to any specific destination. Switching a CAN network would require the construction of equivalent information to accomplish similar functionality to that seen in Ethernet switches.

Recall that CAN frames contain a message ID that is either 11 or 29 bits in length depending on the networks use of standard or extended CAN frame format. Also recall that this identifier is typically used to identify the type of information being transmitted within a given message rather than any source or destination. When broadcasted, any ECU requiring the identified information will receive and act upon it.

Based on available automotive information, the assumption was made that most current vehicles utilize standard CAN frame format and have not fully adopted the extended frame format. With this assumption, an addressing scheme was developed to overcome the absence of any source or destination information. This particular scheme was designed to accommodate current production vehicles as well as future designs, the intent being that it would require very little cost and modification for any manufacturer to implement. There are no changes required to any ECUs already in use and they can continue to use the standard 11-bit ID as they always have.

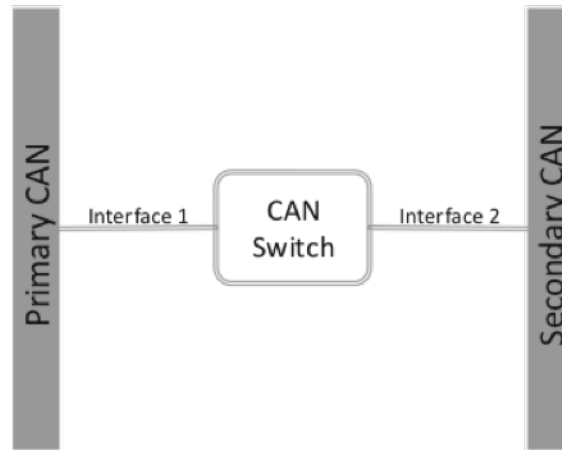


Figure 5: CAN Switch between primary and secondary CAN busses

Figure 5 above shows a CAN switch with 2 transmission interfaces as utilized in actual implementation. While Ethernet switches might range in their number of interfaces or ports; for simplification, CAN switches in this research only have 2 ports; one going to the primary CAN bus, and the other going to a secondary CAN bus. Note that CAN switches can be more complex involving more transmission ports if necessary. This would simply increase the number of secondary busses, as each additional port would correspond to a separate secondary bus.

As described in Section 3.4, the secondary busses would be the primary connection point of all ECUs within a vehicle. Thus all secondary busses (sub-CANs) use the standard CAN frame format and an 11-bit message ID. With CAN switches being the only devices connected to the primary CAN bus, the frame format utilized on that bus may be entirely different from the standard frame format. This allowed the use of the extended frame format and the associated 29-bit ID. Note that even when crossing a switch and forwarding occurs, the original 11-bit ID associated with any message must be maintained throughout the entire network. This ID is what identifies the type of data being transmitted, which will not change, even when forwarded. CAN messages forwarded between CAN switches utilize the extended CAN frame 29-bit ID leaving a

total of 18 additional bits for potential use. These 18-bits are used to identify a source sub-CAN of every message. Figure 6 below illustrates standard and extended CAN frames identifying the bits as described.

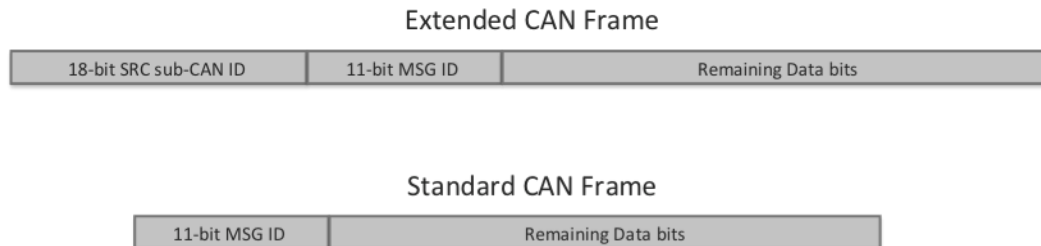


Figure 6: Extended and Standard CAN Frame bit use

With a viable addressing scheme, and the effective use of the message identifier bit space, CAN switches have the necessary information to generate a forwarding table and forward CAN frames accordingly. The forwarding table for each CAN switch could include an entry for every CAN message ID that is required within their respective sub-CAN as well as every ID generated within their sub-CAN requiring transmission to another sub-CAN. This table would need to be configured on the switch itself and highlights a key difference from traditional Ethernet switches. (Recall that Ethernet switches learn and automatically fill their forwarding tables. Without any source information available in a CAN frame, this is not possible in CAN networks.) CAN Switch table configuration could be accomplished a number of different ways, to include configuration files, update patches, etc., however, evaluating the implications of various configuration mechanisms is out of the scope of this research. An example forwarding table is shown in Figure 7.

11-bit MSG ID	Forwarding Interface
1A	INT1
2A	IN1, INT2
3A	INT2, INT3
⋮	⋮

Figure 7 Example forwarding table

To better illustrate the overall CAN switching process, the entire path of a message that requires forwarding is captured below in Figure 8. It is also described in detail here. Assume a message is transmitted from ECU1 in sub-CAN1 to ECU10 in sub-CAN2. A message is generated with an 11-bit ID of E1 from ECU1 and transmitted on the secondary bus within sub-CAN1. CAN Switch 1 receives that message on Interface 2, references its forwarding table, and identifies an entry for ID E1 with a forwarding interface of Int1. The CAN Switch then appends an 18-bit sub-CAN ID of C1 to the 11-bit ID of E1 to generate a new extended can frame message with a 29-bit ID. That new message is then transmitted on the primary CAN bus. CAN Switch 2 receives that message and immediately identifies the first 11 bits of the 29-bit ID. It then references its forwarding table and finds that the message ID E1 is utilized within sub-CAN2. It then strips the 18-bit sub-CAN ID and forwards the original standard CAN message out the appropriate interface. Finally, ECU10 receives the message originating from sub-CAN1 responds accordingly.

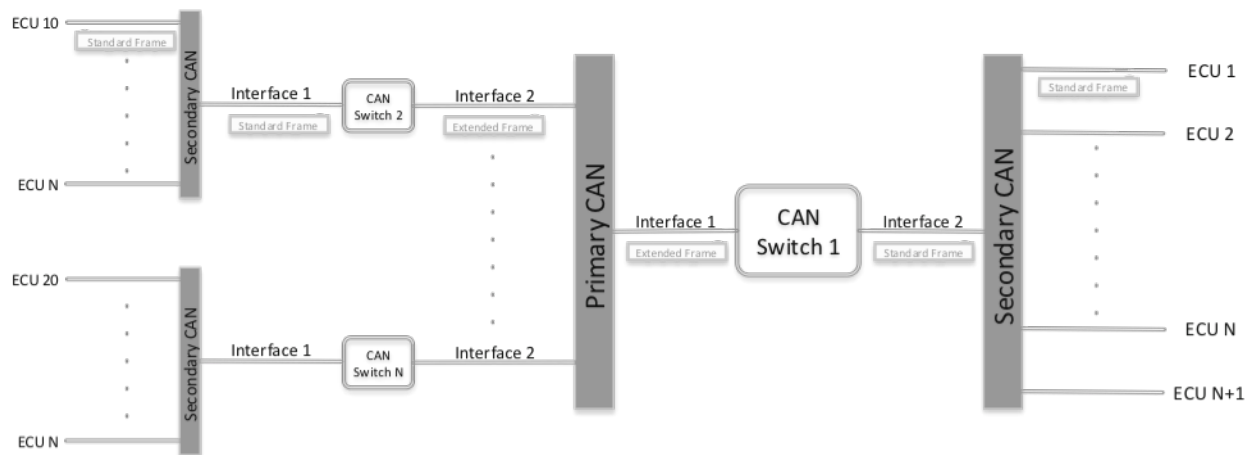


Figure 8: Inter sub-CAN communication path

This approach was designed to accommodate both current production vehicles as well as future designs. However, future vehicle designs might approach this differently as the

manufacturer is still in control of the desired CAN message ID scheme. A more advanced addressing scheme might involve some form of translation between CAN switches allowing an 11-bit ID to have different meaning depending on the sub-CAN it was transmitted from. If the entire vehicle utilized extended CAN frame format, it might even involve some form of variable length message ID's allowing the entire 29-bit space to be utilized more effectively improving functionality, filtering capabilities, and potentially increasing the possible ID space. The point here is that future designs have much more freedom in addressing. CAN switches provide a flexible approach to switching, allowing network designers to utilize them in a way that's most effective for their application.

3.5.2. Filtering

Within the Secure CAN Architecture, CAN Switches are strategically placed between all ECUs and the primary CAN bus. As seen in Section 3.5.1.2, if any communication is required between sub-CANs, a message must go through two CAN Switches before reaching its final destination; one when leaving the originating sub-CAN, and another when entering the destination sub-CAN. As a result, CAN Switches are perfectly situated to observe all inter-sub-CAN communication. Inter-sub-CAN communication also makes use of the extended CAN frame format that provides an additional 18 bits containing the source sub-CAN ID of any message. In conjunction with the standard CAN frame in transmission, a source sub-CAN ID enables intelligent filtering amongst CAN Switches. This greatly improves the filtering capability within a CAN network as CAN Switches have the ability to determine the source sub-CAN and also perform deep packet inspection if necessary.

Filtering could occur at either CAN Switch in the transmission path. Much like referencing a forwarding table, a switch would simply reference a white or black list or ACL (as

seen in Ethernet switches) depending on specific filtering logic. Additionally filtering could be implemented at each individual interface with the ability to filter both inbound and outbound traffic. The location of filtering would depend upon the desired security effects and resulting switch configuration.

If filtering were desired, the process described in Section 3.5.1.2 differs slightly. The differences are seen at the time of message delivery at each CAN switch. Regardless of the interface the message was received on, more processing would occur prior to referencing the forwarding table and forwarding the message. In reference to the example described and Figure 8 the first difference occurs when CAN Switch 1 receives the message from ECU 1 on Interface 2. Rather than reference its forwarding table, CAN Switch 1 would instead reference any filtering list or ACL associated with Interface 2. In this particular crossing, this would be considered an in-bound ACL. If one exists, it would search for an entry containing the 11-bit message ID E1. Assuming CAN Switch 1 finds an entry for ID E1, it would find instruction to either permit forwarding, deny forwarding, or inspect further conditions. At this point; CAN Switch 1 can forward the message continuing the process as described earlier; it can deny the message and halt transmission completely; or it could inspect the entire CAN frame for further details and more granular filtering.

Lastly it's important to note that filtering capabilities differ slightly depending on the placement of filtering rules and the direction of the traffic they are filtering. If filtering is desired at internal facing interfaces CAN Switches only have the ability to filter based on the 11-bit message IDs. However, CAN Switches are somewhat self-aware having been configured to function a certain way. Thus in most cases, a CAN Switch knows the significance of its sub-CAN and an 11-bit message ID is all that's required to determine that a filtered message is out of

place. Filtering traffic on an external (primary CAN bus) facing interface, as described above, has access to both 18-bit sub-CAN IDs as well as 11-bit message IDs.

3.5.3. Simulation

It was decided that simulation and modeling would serve as the appropriate approach in demonstrating the proposed Secure CAN Architecture. In an actual vehicle, the vehicle is responding to user input and the legitimate CAN interaction between ECUs is easily observed via proper vehicle operation. A similar effect was desired in the laboratory environment to improve realism and legitimize the demonstration. A vehicle simulator and hardware in the loop (HIL) system offered a visual representation of a vehicle and its proper operation, the generation of CAN traffic, and simulator-to-ECU as well as ECU-to-ECU interaction.

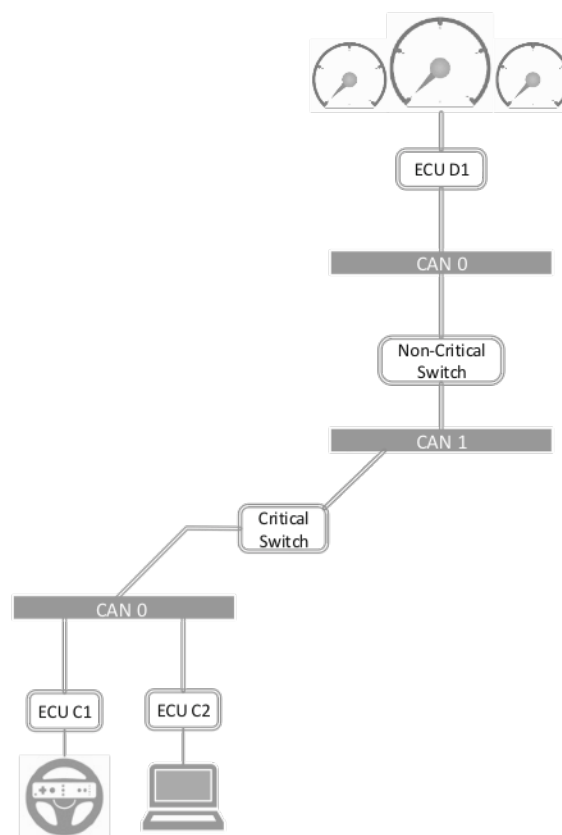


Figure 9: HIL portion of the network

Figure 9 above shows the entire HIL system within the network. Its functionality depends heavily on the simulator itself. To provide an effective simulator, an open source racing game was modified to incorporate CAN network connection, and communication. With slight modification it was capable of transmitting and receiving simple vehicle control information via CAN. In addition to the simulator, several CAN dependent systems were required to complete the hardware loop. To accommodate these needs, simple control and display ECUs were developed to produce hardware-generated traffic and consume simulator-generated traffic. The control related ECU, labeled C1, generates vehicle control information intended for simulator control. It makes use of a video game controller to generate acceleration, braking, and steering commands that in turn are transmitted via CAN. The display-related ECU, labeled D1, consumes vehicle status information generated by the simulator and intended for display. It receives speed and RPM information and translates it for display on the instrument cluster speedometer and tachometer. These subsystems are also isolated within different sub-CANs incorporating CAN switches and their capabilities into the overall simulation.

ECU D1 does not actually correspond to the Subaru instrument cluster itself, rather it refers to a component developed to translate implementation specific CAN messages to Subaru messages expected by the instrument cluster. It was implemented this way for two primary reasons; first, to allow the use of a laboratory controlled messaging scheme (standardized message contents) across the entire network; and second, to eliminate the extraneous CAN messages generated by the instrument cluster. When deprived of expected input, the instrument cluster repeatedly sends requests for updated information. ECU D1 reduced CAN traffic sent to and from the instrument cluster. As the instrument cluster is an actual vehicle component and Subaru CAN messaging information was unavailable, its use required some reverse engineering.

A brute force approach flooding the instrument cluster with various CAN frames revealed information regarding the message IDs and bit ranges corresponding to speedometer and tachometer control. Additional capabilities are possible via the Subaru instrument cluster and their inclusion is left to future work.

Once sufficient functionality was implemented, the end simulation involved a user driving the simulated vehicle with the game controller managed via ECU C1. ECU C1 communicates vehicle control commands on CAN0 of the Critical sub-CAN. The simulator, identified by ECU C2 is located within the same Critical sub-CAN and thus receives control commands immediately. The simulator interprets these commands and controls the simulated vehicle accordingly. As the simulation changes, the simulator also transmits vehicle speed and RPM information that makes its way to the Non-Critical sub-CAN and updated and displayed on the real instrument cluster via ECU D1. This describes the basic HIL functionality across the network.

3.5.4. Entertainment

Another capability desired in the laboratory CAN network was a component that modeled the head unit and entertainment system of an actual vehicle. It was desired primarily for its incorporation of wireless communication methods, and its particular advantage point in many known attacks. As a result of its increased access points, the entertainment system would exist within the Access sub-CAN of the network. This would provide a legitimate system to accompany the OBD-II port. In the overall simulation and demonstration, this entertainment system would be the launching point of all simulated attacks.

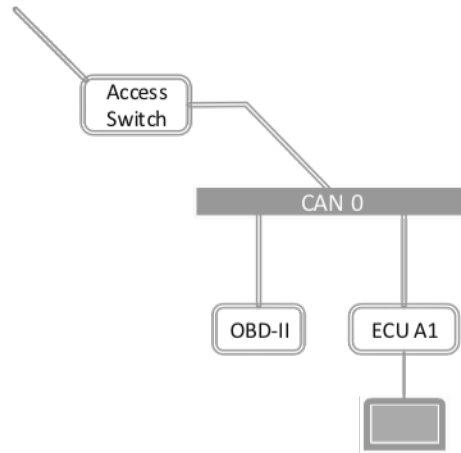


Figure 10: Access sub-CAN

Figure 10 above highlights the Access sub-CAN and entertainment ECU labeled A1. Like most other ECUs within the network, ECU A1 was also developed in house for similar reasoning. Developing this particular ECU allowed for easy simulation of attacker like capabilities on a CAN bus. With access to the CAN bus and the ability to communicate, all post access activities discussed in Section 2.3.2 can successfully be demonstrated from an actual ECU on the network. Its isolation to the Access sub-CAN enables successful demonstration of the security features provided by CAN Switches and Secure CAN Architecture. Additionally, the entertainment system also emulates the function of a real media center. With speakers, a touchscreen, and over-the-air control, it mirrors the entire automotive media experience seen in many modern vehicles.

3.5.5. Non-Secure vs. Secure Mode

Thus far, the specific configuration discussed has been that of the Secure CAN Architecture laboratory implementation. It includes CAN Switches for the desired functional separation and filtering that comes with. While these capabilities are the primary focus of this research, it is useful to demonstrate functionality of a network without the Secure CAN

modifications. This would allow the direct comparison to the less secure basic CAN architecture.

A secure/non-secure mode toggle switch was incorporated into the overall design to allow such comparison. When set to secure, the CAN Switches function as they were designed to and forward/filter CAN messages according configurations. When set to non-secure, the switches function like hubs or gateways simply passing all information received on one interface to the other. This is functionally equivalent to communication over a single primary bus (with minor delay). This feature greatly improves the analysis ability within the laboratory environment.

3.5.6. Over-The-Air Update

As vehicles have become increasingly dependent on software and microprocessor-based electronic control, vehicle manufacturers are extremely interested in methods to effectively update these systems remotely. While specific methods are out of the scope of this research, incorporating some remote update capabilities would allow for future work to include the direction of secure over-the-air updates. Additionally, within the laboratory environment itself, these remote connections greatly improved the workflow allowing easy modification and updating to specific ECUs or the entire network.

The approach taken for over-the-air updates closely followed what was observed in modern automobiles. It also incorporated practices and concepts proposed by Uptane, a solution implementing The Update Framework (TUF) to secure over-the-air updates for ground vehicles [26]. However, it does so in a limited form as secure updating was not yet a primary research objective.

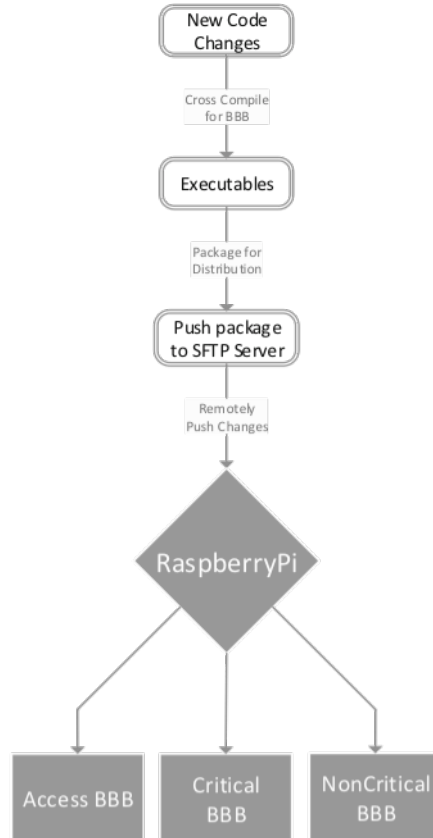


Figure 11: Over-The-Air update process

Figure 11 above outlines the entire update procedure as implemented. It starts with any code modification driven by desired changes or functionality in the network or demonstration. These changes could be for one or multiple ECUs to include the CAN Switches themselves. This code is then compiled for the appropriate platform to generate the necessary executable files. Once available, the executables are packaged for deployment to the update managing ECU within the network. In this particular configuration, the entertainment system served as the update manager, common in the automotive industry. Lastly, the entertainment system ECU distributes the appropriate updated executable files to their respective ECUs prompting all necessary action for a reliable update.

This process requires the network to be in non-secure mode forcing all switches to function as simple gateways and allowing all updated executables to successfully reach all ECUs. These update capabilities will certainly be a requirement within the Secure CAN Architecture through a switched CAN network, however, the method of securely doing so is left to future work. Implications of CAN Switches and what they might offer the secure update process are discussed in Chapter V.

3.6. Putting it All Together

The above key functions within a laboratory environment provide an outstanding implementation outline. With an understanding of the network functionality and resulting requirements, this section will “put it all together” addressing specific hardware and software and implementation decisions. The details presented pertain specifically to the current laboratory configuration presented in Figure 4. The physical components, including 3 CAN Switches, 4 CAN busses, 3 sub-CAN specific OBD-II ports, and 4 ECUs each controlling separate functions can be seen in Figure 12. Combined, these components support a complete, although limited, vehicle CAN network including a multimedia/entertainment center, a Hardware in the Loop (HIL) simulator to include vehicle control, and an instrument cluster for display of vehicle/simulator related information. Future work and improvements to this setup are presented in Chapter V.

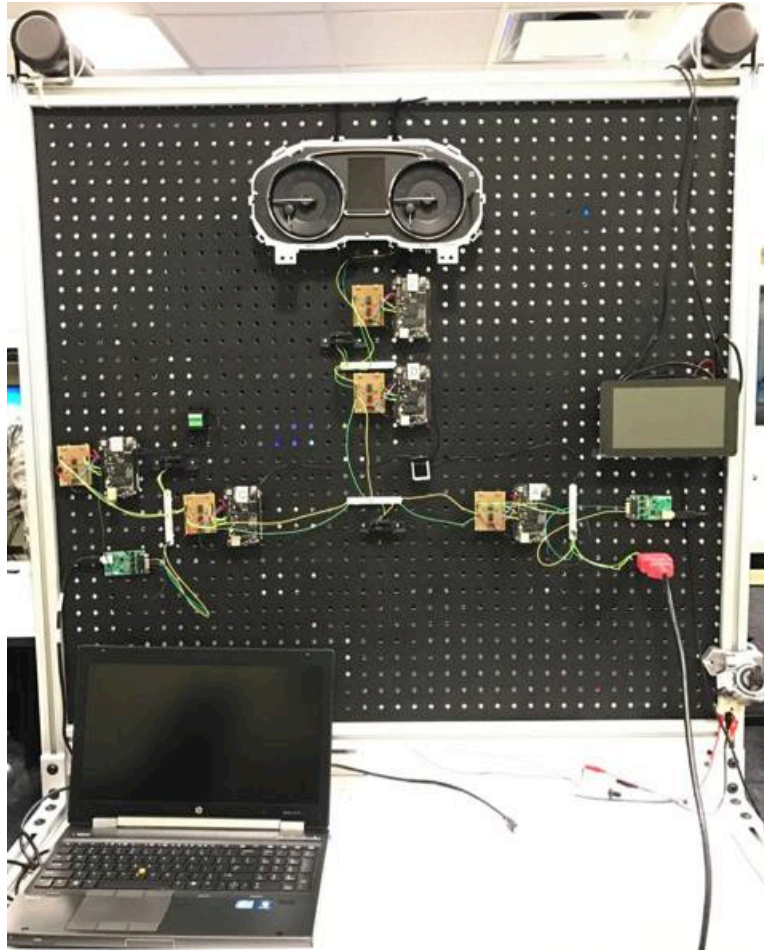


Figure 12: Lab CAN network setup

3.6.1. Hardware

Hardware elements in the lab CAN setup consist of the following components:

- **BeagleBone Black** – BeagleBones were utilized for both CAN Switches as well as general ECUs within the Secure CAN setup. Specifically, these were all CAN switches and ECUs, D1, C1 and C2 shown in Figure 4. Beagle Bones provided a platform with two CAN modules, a Linux operating system, and easy CAN access and communication capabilities. Once configured for CAN communication, they allowed the development of simple applications based on research specific CAN system needs. For these reasons, Beagle Bone Blacks were chosen as the primary CAN component on this CAN network.

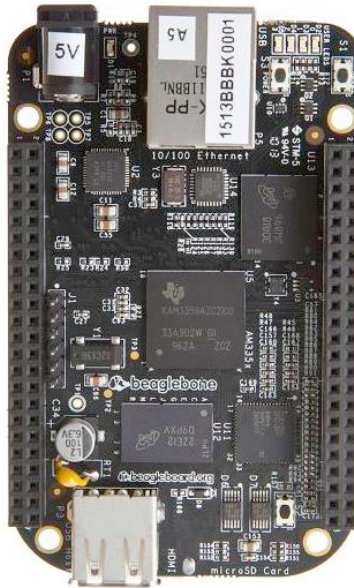


Figure 13: BeagleBone Black
[27]

- **Raspberry Pi** – A single Raspberry Pi was utilized in the representative entertainment system, element A1. Following a real world example of replacing the head unit with a Raspberry Pi touchscreen display [28], and some commercially available software, it closely modeled that of a vehicles in-dash head unit and multimedia system. The Raspberry Pi served as the launching point for representing over the air attacks, modeling what is expected in the real world.

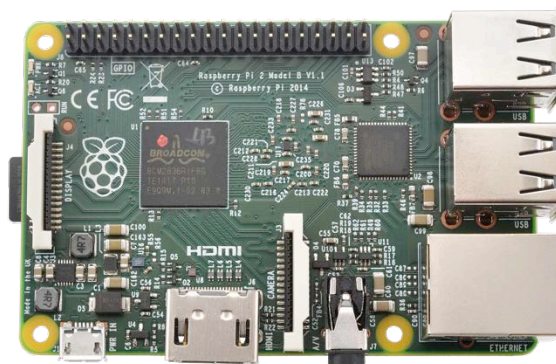


Figure 14: Raspberry Pi
[29]

- **Raspberry Pi 7" Touchscreen Display** – Utilized with the Raspberry Pi to provide a display and user input to the model entertainment system. It allowed for a very realistic and vehicle like experience in the lab setup.

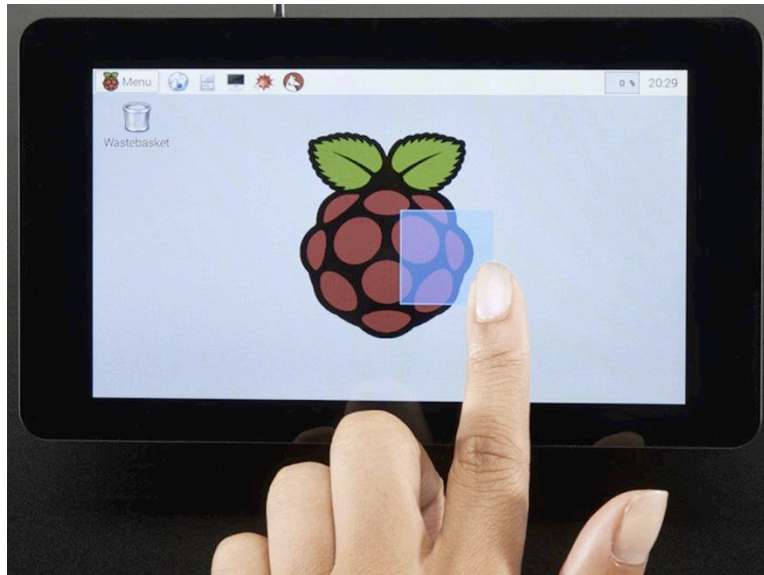


Figure 15: Raspberry Pi 7" touchscreen display
[30]

- **CANtact** – CANtact is a low cost open source CAN to USB interface. Using the SocketCAN protocol, it enabled CAN communication to various USB enabled devices in this network to include the Raspberry Pi, TORCS Simulator, and a PC for testing. The protocol is provided via CAN-utils on Linux [31].



Figure 16: CANtact v1.0
[31]

- **2015 Subaru WRX Combination Meter (Instrument Cluster)** – The combination meter from a 2015 Subaru WRX is utilized to display real time data generated by the TORCS simulator. Information is provided via the Secure CAN network and displayed accordingly. The setup currently handles basic speed and RPM data updating the speedometer and tachometer according to simulator provided input.

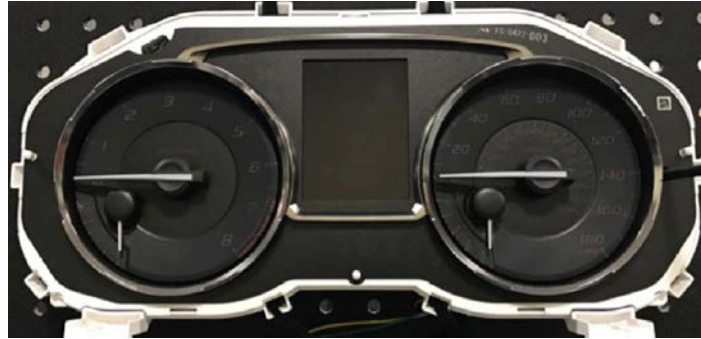


Figure 17: 2015 Subaru WRX combination meter

- **CAN Transceiver Boards** – In order for the BeagleBones to communicate over CAN, they required custom-made CAN transceivers. The boards were fabricated in-house to host two TI ISO1050 CAN Transceivers, one for each of the CAN communication channels on the BeagleBones. Thus, there is one transceiver board for each BeagleBone in use. The fabricated boards are shown below.

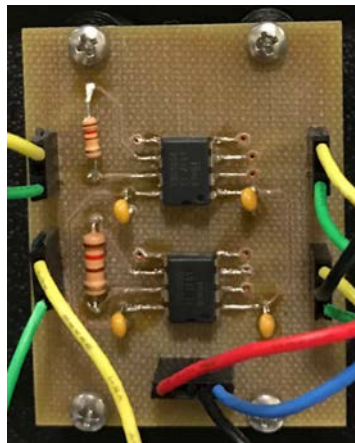


Figure 18: Fabricated transceiver boards

- **Nintendo Wii Remote** – The Nintendo Wii game controller allowed simple and reliable control of the vehicle simulator. It connected via Bluetooth to the control ECU C1 providing the necessary steering, acceleration, and braking input.



Figure 19: Nintendo Wii remote

- **NeoVI FIRE 2 OBD-II Vehicle Network Adapter** – This vehicle interface worked with the Vehicle Spy software providing the required direct access to a CAN network. In addition to its use in this network, it also interfaces with LIN and Ethernet; and provides standalone logging, scripting and simulation capabilities [32].



Figure 20: NeoIV Fire 2 vehicle network adapter [32]

- **OBD-II Connectors** – Each sub-CAN network within the secure CAN architecture had an OBD-II port available providing easy access to the sub-CAN busses for testing and

experimentation. In conjunction with the Vehicle Spy software, these ports allowed data collection via the NeoFire OBD-II access cable (Intrepid Control Systems, Inc., n.d.). The ports themselves were custom made to match the NeoFire cable pinout given it was the primary cable utilized for packet captures.



Figure 21: OBD-II connector kit, vehicle side
[33]

3.6.2. Software

Software elements and languages utilized consist of the following:

- **Ubuntu** – Provided a well-supported OS platform that further allowed use of built in CAN utilities (CAN-utils) and C++.
- **SocketCAN** – The SocketCAN package is available on Linux operating systems and provides access to CAN-utils, which offers a range of functions simplifying CAN communication. Functions used were candump, cansend, cansniffer, and cangen [34].
- **C++** – This was the primary programming language used to implement the ECUs. It was chosen for its simplicity, familiarity, and compatibility with the BeagleBones. Specifically, the ECU bootloader and all applications were written in C++.
- **Python** – This is the primary programming language used to script our test attacks. Its selection was driven by the requirements of the Raspberry Pi, the OSMC operating system and Kodi application. As part of configuring the Raspberry Pi, addons were created for use with Kodi. These addons use the python API provided with Kodi.

- **OSMC and Kodi** – The Open Source Media Center (OSMC) is a Linux distribution chosen for the Raspberry Pi for its access to the Kodi software media center. This provided a platform similar to that found in an actual vehicle’s head unit or multimedia center [35] [36]. Additionally, it provided access to Kodi Add-ons and their development. They allow the addition of functions not standard within Kodi itself. These Add-ons have a lot of potential in this network, but are primarily used to script test attacks for demonstration purposes.

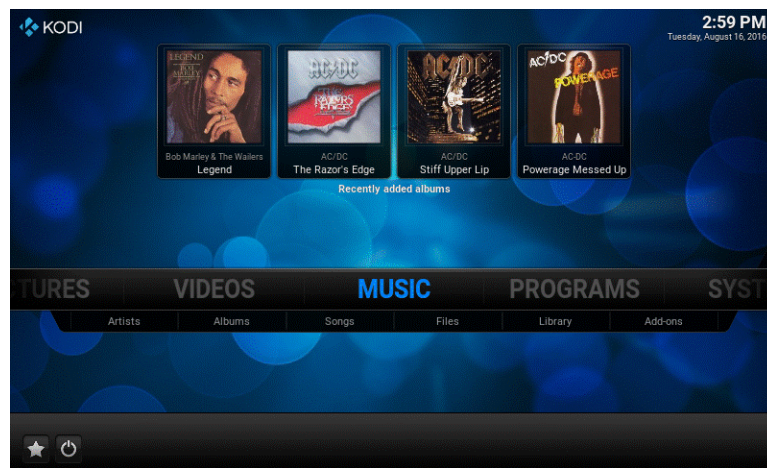


Figure 22: Kodi home screen

- **Vehicle Spy Professional** – This is software that, in conjunction with the NeoFire OBD-II vehicle network adapter and access cable, is utilized for all data collection. It provided a simple easy to use interface that enables bus monitoring and data acquisition. It allowed the viewing, saving, analyzing of message traffic; and in addition to CAN-utils, provides an easy avenue to inject CAN traffic [37].
- **The Open Racing Car Simulator (TORCS)** – TORCS is a very basic and portable multi-platform car racing simulator. Its serves as an excellent research platform providing an extremely modular design that simplifies project specific usage [38]. A great example of

such research is found at [39] which inspired its use in this network. It provided a very basic HIL solution to incorporate CAN ECU traffic and interaction. Unfortunately TORCS does not provide a lot of vehicle related data, limiting the actual simulation capabilities. However, it served as a fantastic starting point for this research.

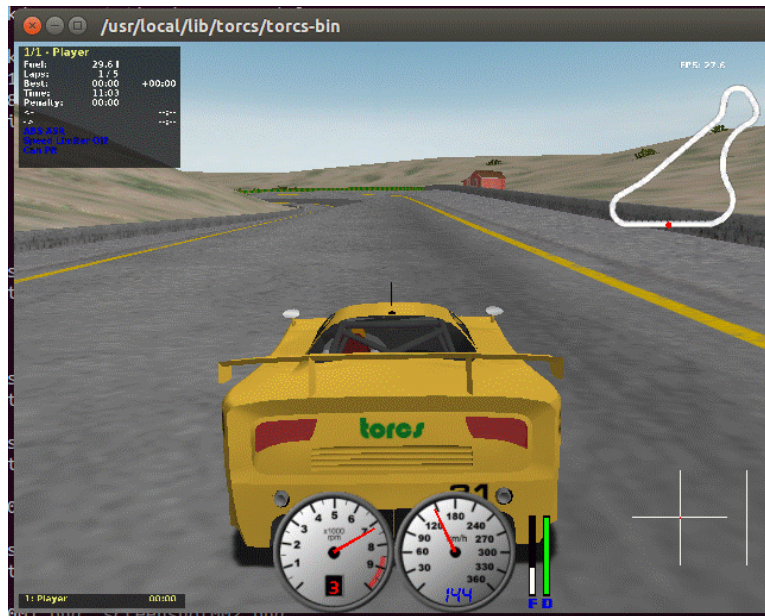


Figure 23: TORCS screenshot

3.6.3. BeagleBone Applications

BeagleBones were used to implement all ECUs and CAN switches within the network. To support multiple purposes, their design was extremely modular allowing easy implementation and programming of any ECU regardless of their role. To separate ECU functionality, several programs or applications were developed to govern each. These applications and their relationships are described here.

- **Bootloader** – A bootloader in traditional computing is a program that manages the booting process, loading the operating system and initializing the necessary system parameters. This bootloader application performed the latter half. Each BeagleBone in

its core implementation ran an identical OS and bootloader program. The bootloader itself acted as software manager for all ECUs controlling the programming and update process as well as the secondary (ECU specific) application running. The secondary application is what dictated the ECUs purpose controlling ECU specific functions. As a result of the modular approach, simply changing the secondary application running on any BeagleBone effectively changed the functionality of the ECU. The remaining applications discussed below are all secondary applications that run in conjunction with the bootloader.

- **Display** – The Display application governed the functionality of ECU D1 whose primary job is translation of implementation specific CAN messages to Subaru’s specific format. It provides the plug and play ability to external components such as Subaru’s combination meter. Without this program, display of simulated vehicle information would not be possible.
- **Wii Remote** – The Wii Remote application managed all logic required for proper Wii Remote control. It translated data provided by the Wii Remote’s accelerometer and buttons to steering, acceleration, and braking input. It further translated such input into usable commands to be transmitted via CAN. As a result, these commands and generated messages effectively controlled the vehicle simulator via CAN communication.
- **Switch** – The Switch Application executed on ECUs A, C, and D, managing all CAN switching and filtering functions. Other than the bootloader, this is the largest application developed. It was designed in a way that allows its implementation across all CAN Switches. In conjunction with the bootloader, this application has the ability to morph to accommodate the specific CAN Switches needs. Simply specifying the CAN Switch

desired at the time of programming, this BeagleBone automatically configures the Switch application appropriately. This, of course, is specific to this lab implementation. This application is also where the Secure/Non-Secure toggle switch is routinely checked allowing a simple change in switch functionality.

3.6.4. Experimentation and Demonstration

With a completed and functioning CAN network following the methods of the Secure CAN Architecture, experimentation consisted of various demonstrations of effectiveness. There are a total of 3 basic demonstrations providing examples of known attacker activities on a CAN bus (eaves dropping, unauthorized transmission, and denial of service). These attacks can be demonstrated during simulator operation. As a direct result of the of the HIL simulation capabilities, in each of these demonstrations it is possible to observe the effects of the attacker as well as the resulting increase in security provided by the CAN Switches and the Secure CAN Architecture.

The filtering logic required to demonstrate the filtering capabilities of the CAN switches is extremely simple. In its current implementation, CAN Switch C is the only switch filtering CAN traffic. CAN Switch C checks the message ID of every CAN frame it receives on interface 1; and if deemed critical, it also checks the source sub-CAN ID to verify that it was not transmitted from CAN Switch A. In other words, there are inbound filtering rules on interface 1 denying any traffic originating from the Access sub-CAN. While CAN Switches are capable of much more complex filtering, complexity was not necessary for a successful and impactful demonstration.

Each of the scripted attack examples are launched from the entertainment system or Raspberry Pi within the Access sub-CAN. They were written in python and developed as Kodi

Add-ons to function specifically as needed. Their functionality and resulting effects are described below.

- ***Eavesdropping*** – This attack sniffed all CAN traffic and dumped it to the entertainment system screen. Although not what would be done in the real world, in the laboratory environment it effectively demonstrated an attacker's ability. In this particular example, the effects were not as alarming as the latter two examples. Eavesdropping itself is a passive attack conducted primarily to gain information from the target network. Thus, no resulting effects can be seen except those displayed on the entertainment system.
- ***Unauthorized Transmission*** – This attack involves the direct transmission of vehicle control commands. Depending on the desired effect, the steering, gas, and brake commands can all be illegitimately transmitted directly from the entertainment system. This of course is improper functionality and if done on a single broadcasted system, the effects would be extremely disruptive. If done during simulator operation, these commands would interfere with the legitimate commands being transmitted from the correct control ECU, C1. The result is the loss of vehicle control.
- ***DoS attacks*** –When triggered, this attack repeatedly transmitted a CAN frame with the message ID of 1 (ID of 0 is not allowed). Recall that the CAN message ID also serves as the message priority for use in arbitration. The lower the ID, the higher the message priority. Thus, a CAN frame sent with the ID of 1 always has priority to transmit on the bus, thereby restricting any other CAN frame from being sent. The result is a complete rejection of ECU communication and vehicle operation.

With the overall simulation capabilities, to include non-secure and secure mode; the above scripted attack examples; and the basic filtering currently in place, the effects of these attacks were all observable directly on the simulator. As the effects were quite disruptive, particularly in the latter two attacks, it was nearly impossible to maintain control of the simulated vehicle during the attacks. This effectively demonstrated the realistic effects these attacks might have on an actual vehicle. Similarly, the security enhancements can be observed just as easily. At any point in time during attack execution, simply switching the network to Secure mode immediately halted the attack, eliminating any disruptive effects. At the push of a button, the CAN Switches effectively partitioned the networked ECUs into functional areas, forwarded legitimate CAN frames accordingly, and denied all illegitimate traffic originating from the Access sub-CAN. In these simple simulated scenarios, the proposed methods and architectural changes have been successfully demonstrated.

3.7. Chapter Summary

The overall objective of this research was to effectively demonstrate the proposed architectural changes as well as the potential improvement they might offer. Given the nature of vehicle security, there were challenges and limitations that drastically impacted the decided methods of implementation and experimentation. To overcome these challenges, it was decided that modeling and simulation would provide a low cost, low risk platform that required very little proprietary knowledge for successful production. This approach required the development of multiple ECUs and an HIL simulator to provide a network that functions under the new architectural design. Once obtained, replicating known attacker activity on the network allowed easy observation and demonstration of the security improvements provided. With functioning CAN Switch prototypes, a network configuration that allowed the desired functional separation

of ECUs, and the ability to simulate both ECU communication and attacker activity, the Secure CAN Architecture has successfully shown its effectiveness.

IV. Analysis

With a functioning CAN network and a prototype of the CAN Switching Concept the resulting analysis aims to evaluate 5 different elements of the proposed Secure CAN Architecture. First the *implications* of the proposed architectural changes. Second, a *security assessment* in which the CAN Switches are evaluated for any security risks and vulnerabilities they might have introduced into a CAN network. Third, their *applicability*, where Secure CAN methods are conceptually applied to an actual vehicle to determine feasibility in the real world. Fourth, *performance impact* on timing, where the incurred transmission delay of crossing a CAN Switch is measured and evaluated to determine acceptability. And fifth, overall *effectiveness*, where a vulnerability/threat assessment framework is proposed to quantify the resulting security improvements. These five elements address critical concerns regarding the use of Secure CAN Architecture.

4.1. Implications of Secure CAN

The proposed Secure CAN Architecture incorporates a CAN Switch to enable certain security enhancements. Recall that the desired effects were to provide the ability to separate and partition CAN nodes based on functional significance and offer more intelligent filtering capabilities. Analysis of the prototyped network revealed that these effects were achieved. In addition, improvements were seen in both performance and security. These improvements and implications of the architecture are described here.

4.1.1. Expandability

To meet functional separation and performance requirements, CAN switches and sub-CANs can easily be added to the primary bus as well as secondary busses within each sub-CAN.

Adding more than one secondary bus within a sub-CAN allows further separation of ECUs within a given sub-CAN. There could be ECUs in sub-CAN1 on bus 1, in sub-CAN1 on bus 2, in sub-CAN1 on bus 3, and so on for each sub-CAN. Finally, CAN switches would be configurable with a number of ports allowing the connection of multiple busses.

4.1.2. Source Information

Standard CAN does not inherently provide source or destination information. Hence, any filtering requires the entire frame. If the entire frame met the filtering criteria, it would be discarded, regardless of where it came from. Unfortunately, there is currently no way to determine valid messages (from a legitimate ECU) from invalid messages (from an attacker).

In the Secure CAN Architecture, all ECUs belong to a particular sub-CAN. As a result, filtering algorithms have access to at least the source sub-CAN of any message sent between sub-CANs. If a message is intended for an ECU in the same sub-CAN, there is no need to forward the message to an external sub-CAN. Messages sent between sub-CANs must be forwarded according to forwarding rules determined at design or configure time. With the ability to identify which sub-CAN a message was forwarded from, CAN switches are empowered with at least a minimal set of filtering capabilities.

4.1.3. Filtering and Access Control

With access to the source sub-CAN of a message, switches can implement filtering logic based on the origin. As ECUs are separated into sub-CANs based on functionality, CAN switches possess the ability to filter pertinent messages based on functional area. Filtering logic (both inbound and outbound) can be implemented at each CAN switch allowing different filtering logic for each sub-CAN.

With this capability, an attacker would need to do more than simply compromise the infotainment system. Rather, they would need to compromise an ECU that requires legitimate communication with the system they are trying to affect, or an ECU on the same sub-CAN as the targeted system, each of which is much more difficult; especially because all possible access points are isolated to the Access sub-CAN removed from any safety or control related systems.

4.1.4. Improved Performance within sub-CANs

To better understand the performance impact of sub-CANs, recall that in Standard CAN, all ECUs would communicate over a single CAN bus. Separating ECUs into functional areas and physically separated sub-CANs in the new Secure CAN Architecture, will theoretically distribute the original traffic across multiple busses, reducing the number of ECUs communicating on each sub-CAN bus and the volume of traffic within any given sub-CAN. The only traffic traversing the primary CAN bus is the legitimate traffic between sub-CANs, expected to be a fraction of the original network traffic. As a result of this architectural separation, it follows that ECUs will experience fewer priority collisions, reduced message delays and average system latency.

4.1.5. Data Isolation

With physical separation of ECUs into sub-CANs and grouping of access related ECUs into the Access sub-CAN, an attacker is limited in the information obtainable through eavesdropping. The most probable avenue to gain that access is via an ECU within the Access sub-CAN, and that attacker can only listen on the Access sub-CAN bus. To capture traffic elsewhere, an ECU within another sub-CAN, which does not have outside access, must be compromised, or the attacker must obtain physical access. If the network is partitioned appropriately, the information available to an attacker has very little impact if exposed.

In addition to data isolation, all attacker activity would be isolated as well. Unauthorized transmission and denial of service attempts would also be isolated to the Access sub-CAN limiting attacker capabilities and overall impact.

4.1.6. CAN Network Management

CAN switches are configurable, allowing network management similar to traditional networks. Network designers would design and configure networks based on performance and security needs using existing ECUs together with multiple identical CAN switches, each configured for the appropriate functional area. Switches would simply need forwarding tables (easily uploaded) and sub-CAN specific filtering logic if filtering is necessary. The initial network configurations would be set prior to the release of a vehicle, and any adjustments to the network configuration once in the hands of the consumer, could be modified via network patches or updates. This does however introduce a potential attack vector, discussed further in Section 4.2.2.

4.1.7. Fail Secure and Intrusion Prevention

In the Secure CAN Architecture, CAN switches are the connecting point of all sub-CANs to the primary CAN bus, They enable a safety or panic button feature which “disconnects” all sub-CANs from the primary bus by simply halting all forwarding, eliminating any communication between sub-CANs.

If an operator suspected a problem, and triggered a “limp mode” severing communication between sub-CANs, a compromised CAN network could enter a state of limited communication allowing the safety and control related functions (partitioned into their own sub-CAN) to continue while isolating most, if not all, malicious traffic. Provided the network design ensured

there were no critical systems with inter-sub-CAN dependencies, all systems within a vehicle should continue to function in limited capacity, without cross-sub-CAN communication.

Cutting communication between sub-CANs would have some adverse side effects. Display related ECUs would cease to display accurate information, generally sourced from the Critical sub-CAN. However, these ECUs merely reflect the status of critical systems. Control related ECUs would receive the required data as they always have, while the display related ECUs are left with outdated information. Regardless of what is displayed on the instrument cluster or multifunction display, the driver would remain in control of critical systems.

This capability could be coupled with security features such as an IDS, like those referenced in Section 2.2.2. Typically, an IDS is used to passively identify the existence of unwanted behavior within a network. When combined with the Secure-CAN architecture, the system could detect an attacker's presence, notify the driver that the vehicle has been compromised, and transition to a limp or safety mode by cutting off the attacker's access, while maintaining the basic vehicle control. A system such as this could not only identify potential threats, but respond to them, providing a feasible intrusion prevention system. According to Kleberger et al., in [2], there have been no IPS solutions proposed for the vehicle domain.

4.1.8. Benefits Over Cryptographic Methods

Cryptographic primitives such as encryption, decryption, message authentication codes, etc. all require either modification at higher layers in the communication stack (i.e. within the ECUs) or a tap in the wire between encrypting and decrypting ECUs and the primary CAN bus. The Secure CAN Architecture offers improvements without ECU modification.

Including encryption, decryption and authentication into the communication chain of an ECU will also impact timing performance for systems that use CAN. As hypothesized earlier in

Section 4.1.4, it's expected that Secure CAN will improve timing performance within sub-CANs adding little to no latency. Secure CAN incurs transmission delays across the primary CAN bus where, due to functional partitioning, delays are believed to be acceptable. Timing impact is discussed later in Section 4.4.1.

4.1.9. Implications in Other Industries

While the platform for this research is an automobile, Secure CAN has the potential to influence security practices in other fields. CAN is found in a wide range of industries outside of the automotive sector. Further, broadcast protocols that closely mirror the functionality of CAN are found in various SCADA systems, industrial control, critical infrastructure networks, etc. The switched CAN approach implemented in the Secure CAN Architecture can likely be utilized to achieve most if not all of the previously mentioned implications in the various other comparable applications.

4.2. Secure CAN Architecture Security Assessment

While it's claimed that the Secure CAN Architecture offers some security improvement over current CAN implementations as well as the many potential benefits seen in the last section, the question remains, does the introduced hardware (i.e. CAN Switches) add any new security risk to the network? The following sections explore answers to this question from several perspectives.

4.2.1. Ethernet Switch Review

Most layer 2 Ethernet switch attacks leverage one of the many intelligent protocols utilized by traditional switches. Examples of such protocols are Spanning Tree Protocol (STP), Cisco Discovery Protocol (CDP), Address Resolution Protocol (ARP), or Dynamic Host

Configuration Protocol (DHCP). Attackers typically influence improper switch functionality by using these protocols to their advantage.

CAN Switches do not rely on these dynamic intelligent protocols for proper functionality, negating such vulnerability. The automotive environment accommodates a much simpler design, with a smaller attack surface for exploitation as compared to more dynamic or intelligent protocols. For example, MAC flooding, a known attack on Ethernet switches, involves flooding the switch with data frames sourced from many different MAC addresses. As a result of the switches duty to fill and update their forwarding table, this would in turn rid the table of valid MAC entries replacing them with spoofed MACs. In response, the switch is forced to broadcast all received traffic out all interfaces (because it does not know where valid hosts are) giving the attacker access to all traffic, as well as the ability to transmit traffic out other interfaces. CAN Switches on the other hand, do not dynamically learn network paths to generate and update their forwarding tables. Rather, their tables are manually configured and updated according to network design requirements, eliminating the need for switches to learn which interface a given node is attached to, and ultimately overcoming the need to broadcast frames.

Other examples of Ethernet switch attacks include MAC spoofing, DoS, and ARP poisoning, each of which differ or don't apply in the Secure CAN environment. For instance, ARP poisoning involves the spoofing of ARP responses to trick a switch into forwarding frames to the spoofing attacker. This will not apply in CAN because CAN switches don't require the translation of IP address to MAC addresses.

In Secure CAN, MAC spoofing could translate to sub-CAN spoofing, which differs primarily in attacker intent. MAC spoofing in a traditional sense involves an attacker spoofing their MAC to pose as another node to receive private data or bypass any network security

obstacles. In Secure CAN, spoofing a source sub-CAN ID would be done strictly to bypass ACLs. This poses the question, does an attacker have the ability to spoof their source sub-CAN ID? In its current implementation, no known method exists. ECUs have no mechanism by which to obtain knowledge of their sub-CAN ID; they only utilize 11-bit standard CAN frame format. ID's are strictly managed and shared amongst CAN switches via the additional 18 bits provided by the use of extended CAN frame format. An attacker would need to directly compromise the switch itself to obtain and possibly modify this information.

On a CAN network, DoS attacks involve repeatedly sending a CAN frame with the highest priority eliminating all lower priority messages from being transmitted on the bus. CAN Switches in the Secure CAN Architecture break up DoS domains restricting the possible denial to ECUs isolated within the sub-CAN. In other words, flooding the sub-CAN bus (which an attacker might have access to) will certainly DoS the sub-CAN and all contained ECUs, but the rest of the network will function in its separation. Note that this will also DoS the switch associated with that sub-CAN but the attack will not impact the primary bus or other sub-CANs.

After Review and consideration of Ethernet switch vulnerabilities, it was hypothesized that the compromise of a CAN Switch will require physical access to the switches themselves or the ability to alter their configuration or firmware. Further, it will require much more than simple bus access and the ability to communicate; an attacker must do one of the following to successfully exploit a CAN Switch:

1. Physically access the switch to potentially modify its configuration and ultimately its functionality within the network.

2. Alter the configuration or firmware during a system patch or update where switch communication involves much more than inter-sub-CAN traffic; specifically switch control related information.

If the CAN Switch configuration and functionality can be modified, an attacker would have potential access to the primary bus in the Secure CAN Architecture. With such access an attacker would have view of other sub-CANs as well as the messaging between them. They would even have the ability to spoof sub-CAN IDs on messages to other sub-CANs allowing the circumvention of potential filtering rules in place. While this is certainly possible, it would require significant reverse engineering, technical skill, and access to accomplish such tasks.

4.2.2. Access to a CAN Switch

Modifications of CAN switch functionality will require some form of access to the switch itself. If physical access is gained, the security of a vehicles network is no longer the primary issue. This can be accomplished via physical force or within a legitimate setting such as routine service and maintenance where access to the vehicle is granted to a third party. To help mitigate such access, engineers might consider hardening the casing of a CAN Switch, the number of switches utilized, and carefully choosing their physical location and placement within the vehicle.

Vehicle maintenance often requires access by a mechanic or third party using diagnostic tools for troubleshooting and reprogramming. Thus, diagnostic protocols should require very strict and specific access rules to allow the legitimate reflashing and reprogramming of ECUs while restricting invalid attempts.

The reflashing of ECUs on the network provides manufacturers the ability to update them as needed while also appealing to a separate market of vehicle enthusiasts who modify their

vehicle's ECUs for performance improvements. This capability also provides a mechanism for patching future security vulnerabilities, and is expected to be a critical feature for future vehicles. Therefore, the process by which access is granted to ECUs isolated within separate sub-CANs must be secure. It could require user interaction to ensure the requested action is legitimate. This process might involve an unsecure (programming) mode enabled by flipping a switch or conducting a sequence of events requiring user specific interaction (perhaps with the key, brake pedals, etc.) ensuring that a remote party cannot achieve non-secure mode or "reflash mode". Despite the user often being considered the single biggest vulnerability in any system, their involvement does have the potential to reduce risk to secure access.

Management of CAN Switches will also require some form of access allowing administrators to modify systems via system patches or updates, eliminating the need for an administrator to remote in or physically connect to manage a CAN Switch. This process presents potential risk as well. To limit potential access here, the process should involve authentication, preventing an attacker from replicating the process. CAN Switches need the ability to validate current as well as proposed configurations. Additionally if something were to go wrong during the update process, switches should have the ability to enter a limp mode resorting to some default configuration allowing basic functionality to continue.

Another access avenue to consider is the supply chain, which has been seen in other domains and is certainly a possibility. Depending on where CAN Switches are produced or the standardized use of them, supply chain management of components or the entire switch itself should be available.

There are valid reasons to allow access to CAN Switches. Such access presents opportunity to alter proper functionality. Improper functionality enables network compromise

and reduces security. Thus, measures must be taken to either prevent or complicate any attacker attempts. As a result, secure accessibility will certainly need to be considered in CAN Switch design and implementation.

4.3. Application on 2015 Subaru WRX

The work and experimentation presented thus far, was conducted within a laboratory environment. While this proved effective for proof of concept, demonstration capabilities, and basic analysis, the question remains, can this actually be applied to a real vehicle? It's certainly possible to incorporate Secure CAN concepts and methods into a new network design, but is it possible on current production models? If network design freedom was available, what would a Secure CAN Network look like in a modern vehicle? Taking elements from a current model, is it possible to modify it to accommodate Secure CAN methods? Further, with knowledge gained from the current vehicle, is it possible to design a new Secure CAN network that would likely incorporate the same vehicle components? All of these questions aim to determine the feasible applicability of the Secure CAN Architecture.

To answer these questions work was done to analyze and map the current implemented CAN network in a 2015 Subaru WRX. Utilizing publically available information [40][41], vehicle systems were identified, and dependencies were determined where possible. With the knowledge gained of the current network, feasibility was observed through the re-design of the WRX network incorporating the proposed techniques; switched CAN, functional areas, logical and physical separation, and filtering. With actual application requiring much more time and resources, this was done hypothetically as a tabletop exercise.

4.3.1. Network Mapping

Network mapping involves the identification of network nodes, and in this case specifically, CAN network nodes. Without an actual vehicle, this process made use primarily of the Subaru WRX service manual, owners manual, wiring diagrams, and Mitchell's Pro-demand website access [40][41]. These sources provided enough information to accurately identify all CAN specific ECUs within the vehicle and map the direct connections. Table 1 below presents the 16 identified ECUs, and Figure 24 presents the associated network topology within the WRX CAN network. These ECUs are the primary control units within their respective vehicle sub-systems. This topology does not show the vehicle in its entirety but only captures CAN connected ECUs and their direct system connections. In some cases, these direct connections correspond to entire sub-systems, which at times include other communication networks like those identified in section 2.1.4. Mapping these networks was not necessary and would have resulted in an extremely complex network topology. The intent of this mapping was to identify network nodes and any system dependencies further enabling the re-architecture utilizing Secure CAN methods.

WRX CAN ECUs
Engine Control Module
Airbag CM
Transmission CM
Steering Angle Sensor
Vehicle Dynamics Control
Power Steering CM
Driver Control Center Differential CM
Data Link Connector
Remote Engine Start CM
Keyless Access CM
Body Integrated Unit
Impact Sensor
Multi-Function Display
Combination Meter
A/C Control Panel

Auto Headlight Beam Lever CM

Table 1: Subaru WRX CAN communicating ECUs

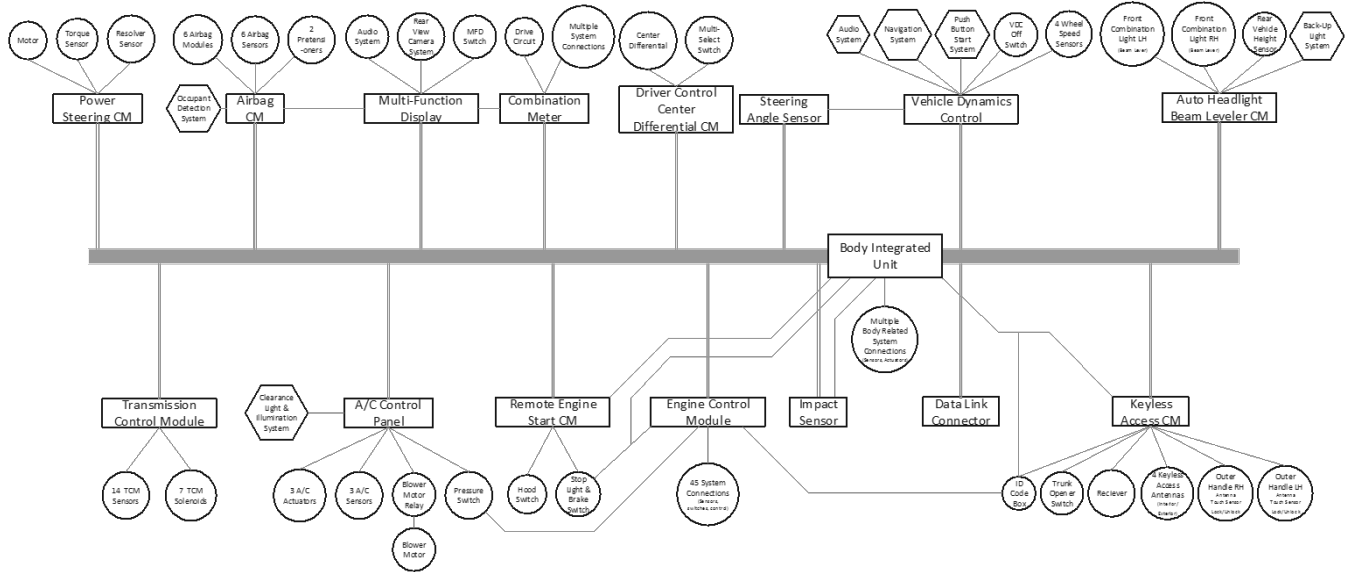


Figure 24: 2015 Subaru WRX Basic CAN Network

In the above diagram, rectangle components identify CAN communicating ECUs; circles identify direct connections to these ECUs such as sensors, relays, solenoids, actuators and switches; and hexagons identify entire sub-systems involving multiple other connections. This network map does not capture all sub-systems, just those directly connected to a CAN networked ECU. Lastly, the lines between CAN communicating ECUs and the CAN bus represent direct CAN connections while all other lines represent other system connections.

As can be seen, some CAN ECUs rely on direct connections to other CAN ECUs. This shows that information is transmitted via direct connection rather than through CAN. An example of this can be seen in the direct connection between the Airbag Control Module (CM) and the Multifunction Display (MFD). System specifications were not available to identify the function of such a connection, and without such knowledge, there is no explanation as to why this information is not communicated over CAN. The Airbag CM might provide an airbag

on/off status for display on the MFD, or the MFD could provide an on/off button for user input to the Airbag CM. Regardless of purpose, these connections exist throughout the WRX network. Where necessary, assumptions were made regarding their purpose allowing further speculation regarding any dependencies that might exist between ECUs.

4.3.2. Assumptions

Prior to redesigning the WRX CAN network it's necessary to first determine the purpose and functionality of all CAN ECUs. With the limited information available regarding Subaru ECU specifications, assumptions must be made where documentation is lacking. These assumptions enabled the intelligent redesign of the network driving key design decisions such as number of partitions, location of ECUs, filtering rules, and placement of rules.

- ***Full network design freedom*** – It's assumed that the redesign of the 2015 Subaru WRX network is done from the perspective of a Subaru engineer. While the knowledge base might not be equivalent regarding Subaru design specifications, assumptions were made to overcome such differences.

4.3.3. Network Redesign

Utilizing knowledge gained and assumptions made, redesigning the WRX CAN network was a relatively simple process. It involved a few basic steps that allowed the inclusion of the proposed Secure CAN techniques; switched CAN, functional areas, logical and physical partitioning, and filtering logic. The process began by identifying all CAN communicating ECUs via network mapping. Prior to redesign, it's also necessary to determine and understand the basic functionality of all identified ECUs to further separate them based on functional significance. ECUs with potential access points were grouped with other access related ECUs. This could be one or multiple groups. With separated and functional groups identified, ECU-to-

ECU dependencies were investigated. If any dependencies do exist, they would be deconflicted by regrouping dependent ECUs. If alternate grouping is not possible, the legitimate communication required between dependent ECUs will require specific filtering rules to allow communication across groups.

Each of the functional groups identified will correspond to sub-CANs in the new Secure CAN network. Thus the number of groups will also correspond to the number of required CAN Switches for proper Secure CAN implementation. To keep things simple in the new network design, the 3 sub-CAN model discussed in Section 3.4.3 was utilized to effectively separate and group all 16 ECUs. These three functional groups separate ECUs first by accessibility, and second by criticality to safety, control, and convenience. Of the 16 identified ECUs, 3 were considered access vulnerable and assigned within the access group. The remaining 13 ECUs were split between Critical and Non-critical groups. While this might be simple, it provides an effective partitioning scheme that simplifies the filtering that will follow. These three groups are outlined in Table 2.

Group 1 – Access	Group 2 – Critical	Group 3 – Non-Critical
Data Link Connector	Engine Control Module	Body Integrated Unit
Remote Engine Start CM	Airbag CM	Impact Sensor
Keyless Access CM	Transmission CM	Multi-Function Display
	Steering Angle Sensor	Combination Meter
	Vehicle Dynamics Control	A/C Control Panel
	Power Steering CM	Auto Headlight Beam Lever CM
	Driver Control Center Differential CM	

Table 2: WRX ECU functional grouping

Designs can certainly be more complex including more sub-CANs if necessary. A four sub-CAN system was considered isolating the body integrated unit and all body related functionality to a single sub-CAN, however, this was rejected because the body integrated unit is connected to many different systems as shown in Figures 24 and 25. With the limited

information available, it was difficult to accurately portray this fourth sub-CAN with the appropriate inter-sub-CAN dependencies.

Figure 25 below shows the redesigned network following the above considerations and groupings. It involves three CAN Switches providing the separation of the three identified functional groups. This redesigned topology includes the addition of CAN Switches, and maintains the symbols and direct connections from the original network shown in Figure 24.

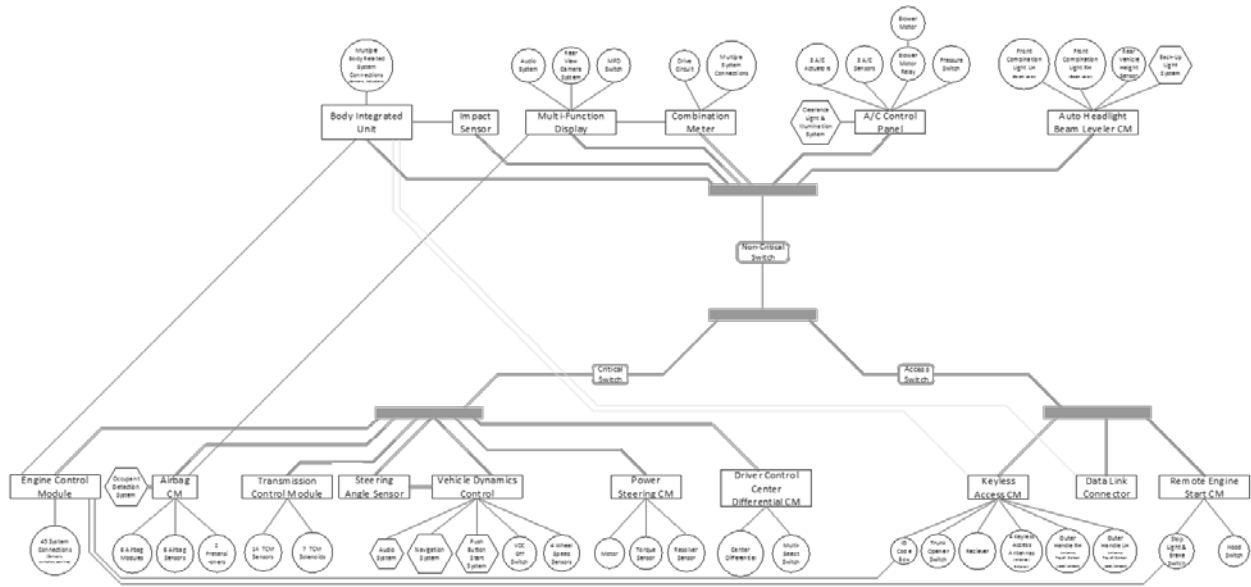


Figure 25: 2015 Subaru WRX Secure CAN network redesign

4.3.1. CAN Switch Configuration

With the above design and network topology, all that remains to enable Secure CAN communication is the configuration of CAN switches. Again, due to limited availability of vehicle specifications, actual switch configuration will be discussed hypothetically. Actual configuration would require detailed knowledge on network nodes and expected communication. It would be accomplished by Subaru network engineers with access to such information.

The two primary elements that need to be configured on each of the CAN Switches are forwarding tables and access control lists. Both of these will enable communication between

sub-CANs as well as improved filtering capabilities. Forwarding tables are simple and would include the message IDs of all messages expected to pass each switch and the corresponding interface or interfaces a message should be forwarded on.

Filtering requires knowledge of the source of a message as well as the legitimacy of that source. In other words, ACL configuration must restrict unauthorized combinations of source and message while allowing those that have been deemed acceptable. If source and message is not granular enough, and further inspection of the message contents is required, then knowledge of the message contents is also a necessity.

When configuring ACLs it's a necessary to consider the traffic that is expected to be leaving and entering all sub-CANs. It's essential to understand genuine reasons and situations that require inter-sub-CAN communication. To better understand how filtering rules might be developed, a few ECU communication scenarios are discussed below as well as the potential rules that might permit or deny such communication to occur. Further, they are presented in relation to each of the three desired sub-CANs (Access, Non-Critical, and Critical).

It's expected that an attacker will likely gain access via ECUs within the Access sub-CAN and thus any communication originating there should be considered carefully. Further, any communication going specifically to the Critical sub-CAN should be handled cautiously given the nature of the systems within.

Filtering rules to prevent such malicious traffic should reference the sub-CAN ID, message ID, or combination of both, of every message. Depending on the implementation of a white or black list, rules should either search for known IDs in the allowable whitelist or deniable blacklist. If not found in the whitelist, transmission is denied; and if not found on the blacklist, transmission is permitted. It's often recommended that a whitelist be utilized since

authentic traffic is known while malicious traffic is a bit unclear. Identifying a set of malicious attacks and methods is not effective or realistic in this situation. A whitelist approach is preferred in terms of security and simplicity where the actions and behaviors are well bounded and understood. Very effective and efficient IDS-type activity can be carried out with a well-constructed and strict whitelist.

Control and system update messages should only originate from the Critical sub-CAN, never from any ECU within the Access sub-CAN. If originating within the Access sub-CAN, transmission to the primary CAN bus and other sub-CANs should be prevented via a whitelist on the internal facing interface of the Access Can Switch. Recall that filtering traffic from within a sub-CAN only uses the 11-bit ID of a message. The internal placement of this whitelist will allow the switch to identify and reject any malicious traffic before transmission on to the primary CAN bus.

There may be legitimate communication between ECUs within the Access sub-CAN and those in other sub-CANs. Diagnostics, maintenance, and reflashing require external access to all possible ECUs, which can be filtered and controlled via a whitelist.

Legitimate communication may be required between other sub-CANs. For example the Critical sub-CAN regularly transmits system status information for display on ECUs within the Non-Critical sub-CAN. Conversely, ECUs requiring real time updates will often request said information if not regularly provided. E.g., when deprived of updated information, the WRX combination meter sent requests for updated information. This communication should occur strictly between Critical and Non-Critical sub-CANs. There should be no possible input provided by the Access sub-CAN. Thus, filtering rules can ensure spoofing is not possible by placing a blacklist on the external facing interface of each of the Critical and Non-Critical CAN

Switches. CAN Switches would simply check and ensure that a message of this type, identified by the 11-bit message ID, was not transmitted from the Access sub-CAN or any other sub-CAN specified by the blacklist.

Although not required for normal vehicle operation, the Secure CAN Architecture enables a range of security/filtering possibilities that might help mitigate any anticipated vehicle cyber threats.

4.4. Performance Impact

In addition to enhancing security, the proposed Secure CAN Architecture, or any other architecture, must meet and maintain various performance requirements to ensure continuous and effective operation. These include timing, size, weight, power, development time, and cost. Of these requirements, timing is one of the most crucial.

4.4.1. Timing Impact

Vehicular CAN networks enable communication between extremely critical and time sensitive systems. While section 4.1.4 presented possible performance improvements from using the Secure CAN Architecture, the addition of CAN Switch hardware will necessarily introduce delay between components that exist in separate sub-CANs, separated by two CAN Switches. As a result, it may be important to understand how much delay is expected for data crossing a CAN Switch. Should inter-sub-CAN dependencies be unavoidable in a Secure CAN network design, this delay can be considered and properly addressed.

4.4.1.1. Experiment

Given the lab configuration described in sections 3.5 and 3.6, it was straightforward to measure the processing delay across a CAN switch. The goal was to identify and measure the delay across a switch that is filtering and also one that is not. These two measurements, which

are based upon performance of the BeagleBone microcontrollers, which are not optimized for this role, should provide an upper bound on delays expected as a result of the inclusion of a CAN Switch in an existing architecture.

This experiment measured delay across 2 different CAN Switches, the Critical CAN Switch, and the Non-Critical CAN Switch. The Critical switch is forwarding CAN traffic and incorporates ACLs and active inbound filtering while the Non-Critical switch is simply forwarding traffic. As the Critical CAN switch is only filtering inbound traffic, this experiment will measure the processing delay of an inbound message to both of the Critical and Non-Critical Switches.

Equipment required to perform this experiment included the CAN Switches; the NeoVI FIRE 2 OBD-II Vehicle Network Adapter; the corresponding Vehicle Spy software; OBD-II connectors on each of the Critical, Non-Critical, and Primary CAN busses; and the addition of a Teledyne Lecroy Wavepro 7 Zi-A Oscilloscope shown below in Figure 26. The oscilloscope has the ability to accurately capture various serial communication protocols to include CAN. It also allows the trigger, decode, measurement, and graphing of CAN simplifying the timing measurement process.



Figure 26: Teledyne Lecroy Wavepro oscilloscope
[42]

The physical setup for this experiment is illustrated below in Figure 27. This measurement required the use of two oscilloscope probes tapped into the CAN busses on either side of the switch in question. This corresponds to the Primary CAN bus and the Secondary CAN bus within the sub-CAN of that switch. Across all measurements, Channel 1 on the oscilloscope was connected to the primary bus, while Channel 2 was connected to a secondary CAN bus. Thus the only difference between CAN Switch processing delay measurements setups in Figure 27 is the actual CAN Switch in question and the secondary bus connected to that switch.

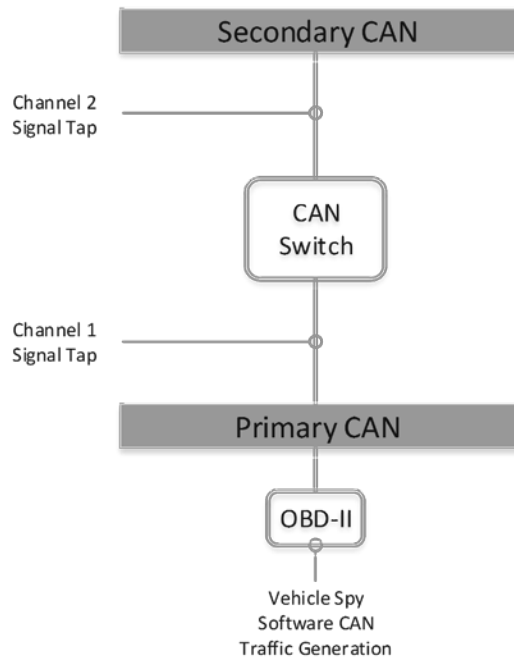


Figure 27: Delay measurement physical setup

Measurement required the presence of legitimate CAN traffic to cross CAN Switches. The Vehicle Spy software generated messages for transmission into the Primary CAN Bus. These messages were injected via the OBD-II connector attached to the primary bus. Recall that the Primary CAN bus communicates using the extended CAN frame format. Thus messages injected at this OBD-II port must be in extended frame format including an appropriate source sub-CAN ID. With traffic being successfully generated and processed by the desired CAN switch, the oscilloscope probes were tapped into the appropriate locations to receive the injected traffic both before and after CAN Switch processing. Each probe and respective channel triggered on the rising edge of the received signal, allowing measurement of the arrival time of each CAN frame, identified by each trigger. The desired delay was calculated by observing the difference between channels 1 and 2. The resulting value corresponds to the physical measured processing delay across the CAN switch. Due to high precision in measurement capabilities, the number of repeated measurements was 50 points across each switch (100 total); at which point,

variance was observed to determine statistical significance and evaluate the need for further measurement.

4.4.1.2. Results

The resulting measurements were as expected. The oscilloscope capture in Figure 28 shows channels 1 (bottom) and 2 (top) and their respective CAN signals. The decode capabilities of the oscilloscope verified that each field of the messages received matches that of the message generated by the Vehicle Spy software. This showed that the messages were in fact the correct messages on either side of the CAN Switch. Additionally, the oscilloscope correctly identified both CAN signal formats (extended on channel 1 and standard on channel 2). On the bottom of the screen capture, information about Channels 1 and 2 were listed in table format to include Time, Format, each field of the CAN Frame, and detected bitrate. The time shown here reflected the time of the specified oscilloscope trigger. Calculating the difference between this value revealed the difference between channels and the desired CAN Switch processing delay.

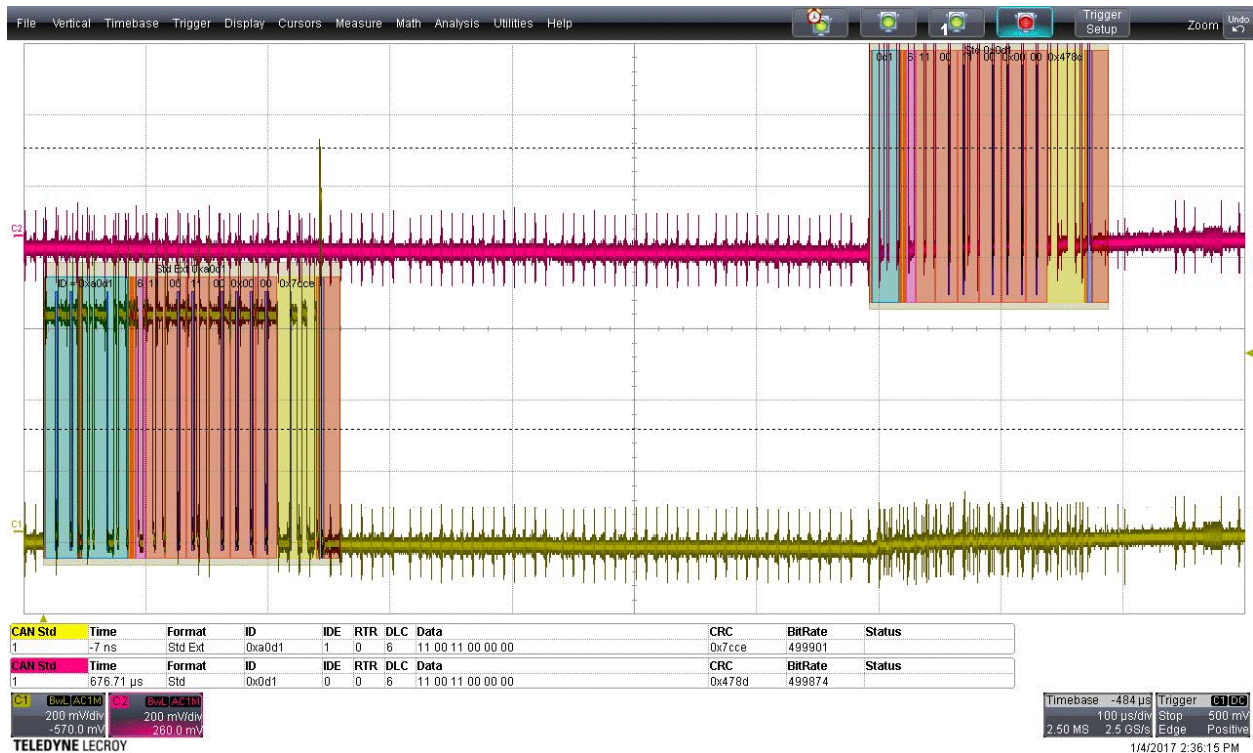


Figure 28: Oscilloscope screenshot showing processing delay across a CAN Switch

The signals captured in Figure 28 were the result of a CAN message sent from the Primary CAN bus to the Non-Critical sub-CAN and thus showed the measured delay across the Non-Critical CAN Switch. In this particular measurement, the time of trigger on channel 1 (Primary CAN) was at **-7 ns**, channel 2 was at **676.71 μs**. The measured delay across the Non-Critical, non-filtering switch was therefore **676.717 μs**. When measured across the Critical and filtering switch the measured delay was **772.355 μs**. As expected, this measurement was slightly higher to account for the additional check of an ACL to filter traffic. After these measurements, the expected processing delay for a CAN Switch appears to range from approximately **670 μs** to **770 μs**, depending upon the filtering in place.

This measurement was repeated a total of 50 times resulting in very similar measurements. In the set of messages observed, there was very little variance (less than 10 microseconds) between timing measurements. This was expected because there was no other

traffic on the bus that might impact contention time. CAN traffic was generated at a rate guaranteeing bus access and transmission of the generated message. Thus CAN bus load was not an issue resulting in very stable measurements. As a result, the 50 trials were deemed sufficient.

The challenge in reviewing and analyzing these results was that there are no known timing constraints or acceptable delay times to compare to. The primary focus of this experiment was to simply identify the worst-case delay of a CAN Switch. At less than 1 ms, this unoptimized delay is likely to be acceptable. For comparison, the timing of human visual conception, or the time between an observer's visual stimulus detection in the brain and that observer's manual response to the stimulus is approximately 150 – 250 ms [43]. Given that the human eye and response time is on the order of hundreds of milliseconds, a delay of less than 1 ms is considered to be insignificant.

Because ECUs are functionally separated, we assert that the delay experienced between sub-CANs and across CAN Switches will not impact the overall system functionality, and the expected delay is acceptable so long as there are no critical or safety related dependencies between sub-CAN networks. If this is not possible and inter-sub-CAN dependencies are unavoidable, an approximate delay of **700 μ s** should be expected for each CAN Switch in the path of a given message transmission.

4.4.2. Other Performance Impacts

While timing is the most crucial performance requirement and metric impacted by Secure CAN, there are others that should be considered as well. Vehicle manufacturers and network designers will rightfully question whether the increase in security has any negative impact on other performance criteria. With many improvements altering the current state of their systems, they will need to address and weigh potential penalties in size, weight, power, development time,

and cost. These metrics must be measurable and tradeoffs will be necessary to balance both security and performance.

In the emulated stage of development, the inclusion of Secure CAN methods and CAN Switches will inevitably increase measures of size, weight, power, development time, and cost. To account for such increases, CAN Switches should be properly designed and optimized for their specific platform allowing minimal penalties and reduced impact on performance. If designed in conjunction with other system components, impact could be minimized. Regardless of approach, if security is valued and enhancement is necessary, these performance metrics should be addressed to ensure a healthy balance between safety, security, and performance.

4.5. Vulnerability/Threat Assessment Framework

Throughout this research, it has been claimed that the Secure CAN Architecture could improve the security in modern automobiles. CAN Switches have demonstrated the ability to filter and deny unwanted traffic from certain areas, and thus certain ECUs, on a CAN network. Simply showing these capabilities via an HIL simulation is meaningful, but does not provide quantitative measures of security improvement. How exactly do you measure security, and more specifically, how do you measure security in vehicles?

4.5.1. Motivation and Approach

Security measurement in itself is an entire field of research. With the growing importance of cyber security specifically, the ability to measure just how secure a system or asset is, can be extremely beneficial for a number of different reasons. Security metrics help to communicate performance and effectiveness. This in turn allows the verification of quality, creation of and compliance in policy, comparison to competition, drive of improvement, increased manufacturer accountability, and the ability to effectively allocate resources [44].

For the purpose of this research a security metric serves two critical functions. First, it provides the ability to quantify, compare, and communicate the effectiveness of the proposed Secure CAN Architecture. Second, if Secure CAN and CAN Switches were entertained as a potential defense, a security metric provides a mechanism through which manufacturers can enumerate and prioritize potential threats and resources.

Advancing the field of security measurement is out of scope for this research. However, there are various security measurement standards, regulations, guidelines, and frameworks available that aim to quantify security in various different respects, which might be applicable in the domain of vehicle cyber security. General cyber security measurement might reference metrics such as number of vulnerabilities, number of vulnerable systems, defensive measures, response/remediation time, number of attacks, and frequency of attacks to attempt to quantify overall security of a given entity. They also generally aim to measure some level of compliance in adherence to standards, often based on known and developed defensive measures and security practices. Metrics, policy, and defenses such as these are not well defined for vehicles. In addition, there are relatively few documented cyber attacks on vehicles to use as threat models.

To overcome challenges in measuring vehicle security, the proposed security assessment approach instead looked towards vulnerability impact and severity to obtain a meaningful metric. Security measures are established to minimize the impact of a given vulnerability and the severity of its effects if exploited. If the security measure is successful or even improved, the overall impact and severity should decrease. This assessment will thus measure the overall severity and impact of a system vulnerability allowing the quantifiable observation of differences between CAN implementations. It is expected that networks under Secure CAN influence will result in a lessened vulnerability impact.

4.5.2. CarVSS

To obtain such a metric, a simple scoring system was created which combines two well-developed and proven evaluation methods to effectively capture the severity and impact of a vehicular vulnerability. The proposed assessment framework, called the Car Vulnerability Scoring System (CarVSS), closely follows the Common Vulnerability Scoring System (CVSS), a framework designed to communicate the characteristics and severity of software vulnerabilities. It differs primarily in its calculation of several impact related metrics, modified in a way to more accurately captures the impact an attack will have on a vehicle rather than a traditional IT system. Instead it makes use of step two of the National Institute of Standards and Technology (NIST) Cyber Security Risk Management Framework (RMF) applied to vehicles. Together, these two resources offer a unique view of vehicular vulnerabilities and their corresponding impact.

4.5.2.1. Common Vulnerability Scoring System

CarVSS utilizes methods, metrics, and calculations defined by the CVSS v3.0. CVSS is an open framework developed to help measure and quantify the impact of IT vulnerabilities. It makes use of three different metric groups; Base, Temporal, and Environmental; each of which contain metrics that encapsulate characteristics of modern vulnerabilities. These groups are shown below in Figure 29. The Base metrics describe certain characteristics of the vulnerability itself while the temporal metrics capture any changes in these characteristics that might occur over time. The Environmental metrics incorporate factors specific to the environment of the vulnerable component. These metrics combine to provide a standard and repeatable measurement process allowing those in IT security to speak a common language. The Base metrics alone generate a severity score ranging from 0 to 10. This Base score can then be

modified depending on the Temporal and Environmental metrics producing a final CVSS vulnerability score [39][40].

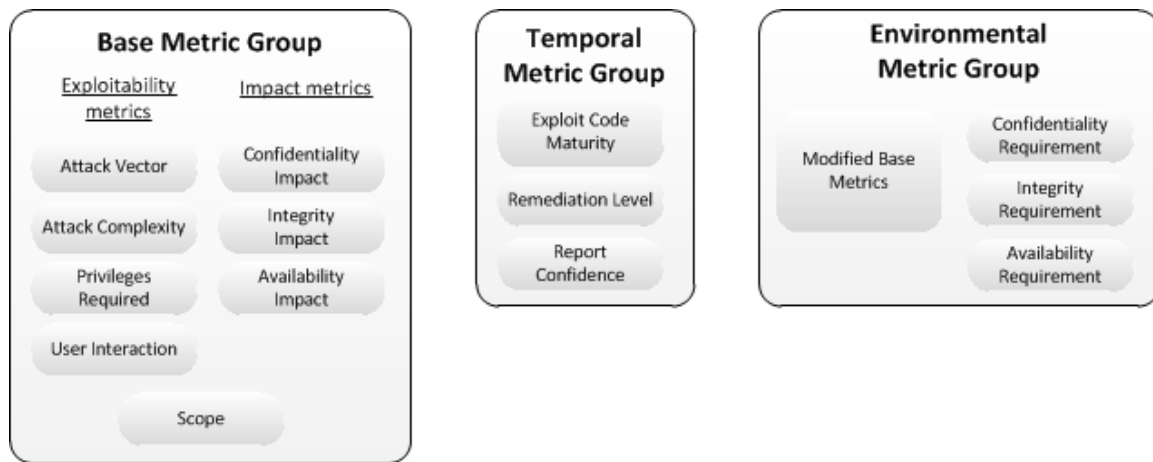


Figure 29: CVSS Metric Groups [45]

For the purpose of CarVSS, the Base numerical score alone will be utilized as the metric of interest. According to the CVSS specifications, the Temporal and Environmental scores are not required and their omission will simplify the calculation process as well as eliminate the need to evaluate their impact as they relate to vehicles specifically. The Base metric score gets calculated using two separate sub-scores; the Exploitability sub-score, and the Impact sub-score. Each of these are calculated by separate equations incorporating the Exploitability metrics and Impact metrics respectively [45].

The Exploitability metrics represent characteristics of the attack and exploit side of the vulnerability. These particular characteristics are traits that are also associated with vehicle system vulnerabilities. While CVSS was intended for software vulnerabilities, these metrics will also nicely apply to vehicular platforms. In CarVSS they are defined and calculated exactly as outlined in the CVSS specification document.

The Impact metrics, however, endeavor to capture the effects of successfully exploiting a particular vulnerability. They do so by quantifying the impact on confidentiality, integrity, and availability (CIA). Here is where CVSS is an imperfect model for vehicular platforms. On a vehicle, the impact and resulting effects if exploited are typically physical in nature, and in most cases more severe. Rather than utilize three different metrics for each CIA element, this assessment proposes a single metric that not only captures all three CIA elements, but does so across various vehicle states. It's believed that this approach will more accurately capture the potential impact an exploited vulnerability has on a vehicle under the various environmental conditions and vehicle states. The impact of a vehicular vulnerability changes drastically depending on operating state of the exploited vehicle. For example, the impact on a car exploited while parked in an empty parking lot is quite different than while going 60 mph transporting a family. This approach should result in a more accurate consideration of the impact a vulnerability might have.

4.5.2.2. NIST Risk Management Framework

NIST's Risk Management Framework provides an excellent mechanism to achieve such an impact metric. In October of 2014, the National Highway and Traffic Safety Administration (NHTSA) reviewed the NIST RMF guidelines from an automotive safety standpoint. In that review, NHTSA applied NIST RMF methods to current vehicles in an attempt to establish a baseline understanding and introduce risk management concepts to the automotive sector. Their goal was to increase knowledge and awareness in automotive cyber security, introduce the potential risks involved, and help support new and standardized security guidelines [47].

Although risk assessment and management is crucial to the security profile of any organization, it is not the focus of this research. Instead this effort focused primarily on step two of the NIST RMF and its role in the proposed vulnerability assessment framework, CarVSS.

Step two consists of a “Categorization of Vehicle Systems” [47]. This step involves the categorization of all systems within a vehicle, and is expected to be performed by the vehicle system owner or manufacturer. This categorization considers each systems impact if successfully exploited across potential states of a vehicle. Once each system has been considered individually, the overall system impact is presented as a final high, medium, or low overall system impact level. This overall system impact level provides an applicable metric and rating value to replace the standard CVSS Base impact CIA metrics.

The example categorization of vehicle systems presented in [47] is shown below in Table 3. If utilized in its current state, it should provide an effective and meaningful approach over the standard CVSS Base Impact metrics.

NHTSA Vehicle Data Sensitivity Analysis (DAST) Tool								Volpe National Transportation Systems Center						Cambridge MA										
Phases of Vehicle Trip								Vehicle Parked (e.g. Garage)			Vehicle at Rest (Traffic Light)			Vehicle Maint. (Dealer/Local Garage)			Vehicle at 65MPH (highway)			Vehicle at 20 MPH on a major highway (stop/go traffic)				
								C	I	A	C	I	A	C	I	A	C	I	A	C	I	A		
FIPS 199 Confidentiality - Integrity - Availability	CONFIDENTIALITY - A loss of confidentiality is the unauthorized disclosure of information.																							
	INTEGRITY - A loss of integrity is the unauthorized modification or destruction of information.																							
	AVAILABILITY - A loss of availability is the disruption of access to or use of information or an information system.																							
Powertrain																								
Throttle Valve Data									L	L	L	L	L	L	L	L	L	L	H	H	L	M	M	
CAN Bus Data message for the PCM (Powertrain Control Module)									L	L	L	L	L	L	L	L	L	L	H	H	L	M	M	
Apdaptive Cruise Control Data									L	L	L	L	L	L	L	L	L	L	H	H	L	M	M	
Local Interconnect Network (LIN) Steering Wheel data									L	L	L	L	L	L	L	L	L	L	H	H	L	M	M	
Antilock Brake System (ABS) Brake-by-Wire data (via FlexRay)									L	L	L	L	L	L	L	L	L	L	H	H	L	M	M	
Vehicle Safety																								
Onboard Diagnostics (OBD II) emissions data									L	L	L	L	L	L	L	M	M	L	H	H	L	M	M	
TPMS data (via Bluetooth)									L	L	L	L	L	L	L	M	M	L	H	H	L	M	M	
Firmware-Updates-Over-The-Air (FOTA) Remote Diagnostics Data									L	L	L	L	L	L	H	H	H	L	L	L	L	L	L	
Airbag Control Unit Data									L	L	L	L	L	L	L	L	L	L	H	H	L	M	M	
GPS Data									M	M	M	M	M	M	L	L	L	L	H	H	H	H	H	H

Comfort System																
Local Interconnect Network (LIN) Climate Control Data		L	L	L	M	M	M	L	L	L	L	M	M	L	M	M
Door Control Unit Data (Power doors)		L	L	L	M	M	M	L	L	L	L	M	M	L	M	M
Infotainment																
MOST (Media Oriented Systems Transport) audio amplifier data		L	L	L	M	M	M	L	L	L	L	M	M	L	M	M
Personal Identity Info (PII)		L	L	L	H	L	L	L	L	L	H	H	H	H	M	M
Internal connectivity data (for mobile devices)		L	L	L	M	M	M	L	L	L	L	M	M	L	M	M
Telematics																
GPS Data for Emergency Response		M	M	M	M	M	M	L	L	L	H	H	H	L	H	H
Map Navigation data		L	L	L	M	M	M	L	L	L	L	H	H	L	H	H
	FIPS 199 Confidentiality - Integrity - Availability	L	L	L	H	M	M	H	H	H	H	H	H	H	H	H
	FIPS 199 Overall System Impact Level:	L				H		H			H			H		

Table 3: Vehicle security categorization example
[47]

4.5.2.3. Combined Measurement and Scoring

Recall that the standard CVSS Base score utilized Exploitability and Impact metrics. The impact metrics were composed of three individual metrics, one for each C, I, and A. Each of the CIA metrics respectively had potential ratings of high, medium, or low. In CarVSS, these three metrics are replaced with a single high, medium, or low Impact metric generated by step 2 of the NIST RMF. This modification would alter the Impact Score calculation utilizing this single metric rather than 3 separate ones. In CVSS, the 3 separate CIA impact metrics apply to the ISC_{Base} calculation as seen in the equation in Figure 30. Each metric is accounted for with a (1-

Impact) in the overall equation. Simply removing two of the metrics simplifies the equation while maintaining the mathematical significance of the calculation itself. See Figure 30 below.

$$\begin{aligned}
 ISC_{Base} &= 1 - [(1 - Impact_{Conf}) \times (1 - Impact_{Integ}) \times (1 - Impact_{Avail})] \\
 &= 1 - [(1 - Impact_{Conf}) \times (1 - Impact_{Integ}) \times (1 - Impact_{Avail})] \\
 &= 1 - [(1 - Impact_{New})]
 \end{aligned}$$

Figure 30 New Base Impact Sub-Score calculation

All of the Base metrics, including the combined Exploitability and proposed Impact, are shown below in Table 4. Also included are brief descriptions, corresponding ratings and numerical values for each. Each rating correlates to a different numerical value that plays a role in one or multiple calculations. The numerical values for the proposed Impact metric were derived from the standard CVSS Base Impact metric values. The result will be one metric in the final equation rather than three.

CVSS Significance Factor	Metric	Metric Value	Description	Numerical Value
Scope	Scope (CVSS)	(U) Unchanged	An exploited vulnerability can only affect resources managed by the same authority. In this case the vulnerable component and the impacted component are the same. (Numeric Value is a multiplier)	6.42
		(C) Changed	An exploited vulnerability can affect resources beyond the authorization privileges intended by the vulnerable component. In this case the vulnerable component and the impacted component are different. (Numeric Value is a multiplier)	7.52
Exploitability	Attack Vector (CVSS)	(N) Network	A vulnerability exploitable with network access means the vulnerable component is bound to the network stack and the attacker's path is through OSI layer 3 (the network layer).	0.85
		(A) Adjacent	A vulnerability exploitable with adjacent network access means the vulnerable component is bound to the network stack, however the attack is limited to the same shared physical (e.g. Bluetooth, IEEE 802.11), or logical (e.g. local IP subnet) network, and cannot be performed across an OSI layer 3 boundary (e.g. a router).	0.62
		(L) Local	A vulnerability exploitable with Local access means that the vulnerable component is not bound to the network stack, and the attacker's path is via read/write/execute capabilities.	0.55
		(P) Physical	A vulnerability exploitable with Physical access requires the attacker to physically touch or manipulate the vulnerable component.	0.2
	Attack Complexity (CVSS)	(L) Low	Specialized access conditions or extenuating circumstances do not exist. An attacker can expect repeatable success against the vulnerable component.	0.77
		(H) High	A successful attack depends on conditions beyond the attacker's control. That is, a successful attack cannot be accomplished at will, but requires the attacker to invest in some measurable amount of effort in preparation or execution against the vulnerable component before a successful attack can be expected.	0.44
	Privileges Required (CVSS)	(N) None	The attacker is unauthorized prior to attack, and therefore does not require any access to settings or files to carry out an attack.	0.85
		(L) Low	The attacker is authorized with (i.e. requires) privileges that provide basic user capabilities that could normally affect only settings and files owned by a user. Alternatively, an attacker with Low privileges may have the ability to cause an impact only to non-sensitive resources.	0.62 (0.68 if Scope is Changed)
		(H) High	The attacker is authorized with (i.e. requires) privileges that provide significant (e.g. administrative) control over the vulnerable component that could affect component-wide settings and files.	0.27 (0.50 if Scope is Changed)
	User Interaction (CVSS)	(N) None	The vulnerable system can be exploited without interaction from any user.	0.85
		(R) Required	Successful exploitation of this vulnerability requires a user to take some action before the vulnerability can be exploited. For example, a successful exploit may only be possible during the installation of an application by a system administrator.	0.62
Impact	System Categorization and Criticality (RMF)	(H) High	High is the overall criticality considering the vulnerabilities impact across all phases of a vehicles trip as well as affects in Confidentiality (C), Integrity (I), Availability (A). (Value derived from CIA Impact values in CVSS)	0.56
		(M) Medium	Medium is the overall criticality considering the vulnerabilities impact across all phases of a vehicles trip as well as affects in Confidentiality (C), Integrity (I), Availability (A). (Value derived from CIA Impact values in CVSS)	0.22
		(L) Low	Low is the overall criticality considering the vulnerabilities impact across all phases of a vehicles trip as well as affects in Confidentiality (C), Integrity (I), Availability (A). (Value derived from CIA Impact values in CVSS)	0

Table 4: CVSS Metric and Rating Chart

With all numerical values identified, Base scoring for CarVSS involves several different equations. They are defined below exactly as outlined in the CVSS specification document. The Base Score is a function of the Impact and Exploitability sub score equations.

The Base score is defined as:

```

If (Impact sub score =< 0) 0 else,
Scope Unchanged, Round up (Minimum [(Impact + Exploitability),10])
Scope Changed, Round up (Minimum [1.08 × (Impact + Exploitability),10])

```

The Impact sub-score (ISC) is defined as:

```

Scope Unchanged 6.42 × ISCBase
Scope Changed 7.52 × [ISCBase-0.029] - 3.25 × [ISCBase-0.02]15

```

Where:

$$ISC_{Base} = 1 - [(1 - Impact_{Conf}) \times (1 - Impact_{Integ}) \times (1 - Impact_{Avail})]$$

And the Exploitability sub-score is:

$$8.22 \times AttackVector \times AttackComplexity \times PrivilegeRequired \times UserInteraction$$

Figure 31 CVSS Base Scoring equations
[45]

The resulting Base Score will be an overall severity rating between the numerical values 0 – 10, 0 being no severity at all and 10 being extremely critical. In addition to the numerical severity rating, textual ratings are also available if desired and shown in Table 5.

CVSS Score	Rating
0.0	None
0.1 – 3.9	Low
4.0 – 6.9	Medium
7.0 – 8.9	High
9.0 - 10	Critical

Table 5: CVSS Severity Rating scale
[47]

Lastly, for readability and reproducibility, CVSS provides a textual representation of all metrics and ratings used to generate any one given score. This representation is referred to as a

CVSS Vector String. It provides a concise and simple way to communicate the details of a severity rating. A vector string begins with the label “CVSS:” and a numerical version being utilized such as “3.0.” This is followed by each metric abbreviation, colon “:”, and the corresponding rating value for that metric. Each “metric:value” is then separated by a “/”. The vector string values for the metrics utilized in CVSS and CarVSS are captured in Table 6.

Metric Group	Metric Name, Abbreviation	Possible Values
Base	Attack Vector, AV	[N,A,L,P]
	Attack Complexity, AC	[L,H]
	Privileges Required, PR	[N,L,H]
	User Interaction, UI	[N,R]
	Scope, S	[U,C]
	Impact, I	[H,L,N]

Table 6: Vector String values

For example, a vector string for a severity score calculated using CVSS v3.0 with a Network Attack Vector, a Low Attack Complexity, High Required Privileges, No User Interaction, Unchanged Scope, and Low Impact rating, would result in the following vector string shown in Figure 30 [47]. This example was modified from the CVSS documentation example to fit our specific combined metric approach.

CVSS: 3.0/AV: N/AC: L/PR: H/UI: N/S: U/I: L

Figure 32: Example Vector String
[47]

4.5.3. Results

With Base scoring procedures understood, the severity and impact of known vehicle vulnerabilities can now be evaluated. Even further, they can be evaluated in both Basic and Secure CAN Architectures. Where technical vulnerability information was lacking, assumptions were made to assign metric ratings and allow the desired quantifiable comparison between CAN implementations.

The process of evaluating a single vulnerability is now described in detail, under both Basic and Secure CAN conditions. This evaluation will thus reveal two different CarVSS severity scores, one for each CAN implementation.

4.5.3.1. Standard CAN CarVSS Score

The most well-known attack on a vehicle to date is the successful remote Jeep exploit conducted by Charlie Miller and Chris Valasek. When evaluating this particular type of attack and vulnerability, consider the vulnerable system itself when rating all required metrics. [17] describes the general approach and exploitation chain in remotely exploiting a Jeep and other similar vehicles. The exploit began with remote network access to the vehicle's OMAP chip within the head unit. This drove the rating of the first metric, the Attack Vector. The rating was assigned Network(N) given the vulnerable system was exploitable remotely via network access. Next, actual exploitation of the OMAP chip was achieved via upload of an SSH key and configuration file allowing the start of the SSH service. While repeatable, this would require the attacker to invest a reasonable amount of effort in preparation for successful execution. As a result, Attack Complexity would be given a rating of High(H). Elevated privileges do not seem to be a requirement for the attacker to complete the attack, however it does require the modification of firmware on the Renesas V850ES/FJ3 chip, accessible via the OMAP chip. This process does not include cryptographic signatures for verification in firmware legitimacy. As a result, the Privileges Required metric was given the rating of None(N). The vulnerable system can be exploited with no user interaction at all and thus User Interaction was assigned the rating of None(N). Successful compromise of the OMAP chip in the head unit provided access to the V850ES/FJ3 chip which in turn provided access to the CAN busses and the rest of the vehicle. As a result, the vulnerable component and the impacted component are different, leading to the

rating of Changed(C) for the Scope metric. Lastly, the modified Impact score was investigated using NIST's RMF. The OMAP chip was considered a part of the head unit, and the V850 chip was what allowed the head unit to communicate over the required CAN busses. According to the RMF rating chart shown in Table 3, the infotainment system is primarily rated at medium and low across the various states of a vehicle. However, if the OMAP chip were to be successfully compromised, and the V850 chip firmware could be modified, this particular exploit chain has the potential to impact all systems in the vehicle, to include control and safety specific ones. As a result, the overall impact rating will be High(H). This analysis evaluates the Jeep or a similar vehicle in its current CAN implementation. It would result in a CarVSS severity score of **6.8** and the following vector string and rating values.

CVSS: 3.0/AV: N/AC: H/PR: N/UI: N/S: C/I: H

Figure 33: Basic CAN Jeep attack Vector String

4.5.3.2. Secure CAN CarVSS Score

With Secure CAN methods in place, several crucial metric ratings would differ. Most however, would remain the same. The Attack Vector would remain the same as access would still be gained via networked methods. Attack Complexity would naturally increase, however this attack was already rated at its highest possible rating, thus it too would remain the same. User Interaction would also remain the same requiring none at all. Privileges Required would also remain the same as the vulnerable system itself hasn't changed. Scope, however, would differ significantly; with Secure CAN methods in place, complexity of the network would be different and accessibility of certain ECUs has diminished. CAN Switches and filtering prevent Scope moving from one exploited system to another system. This would result in the improved rating of Unchanged(U). Lastly, for the same reasoning as the Scope metric, the Impact rating

will also improve, as compromise of the vulnerable system no longer provides access to safety or critical specific systems. The attacker would still have access to the head unit and any potential information that it contains, so this would result in the improved Impact rating of Low(L). Thus, under the Secure CAN Architecture the Jeep Attack would result in an overall CarVSS severity rating of **4** and the below vector string.

CVSS: 3.0/AV: N/AC: H/PR: N/UI: N/S: U/I: L

Figure 34: Secure CAN Jeep attack Vector String

Here we see a lower severity and impact rating on one of the most alarming attacks to date. Under CarVSS criteria this can be considered a quantifiable improvement over the Basic CAN implementation. This single evaluation has implications across all other possible vulnerable vehicular systems. In most cases, it appears that Attack Complexity, Scope, and Impact, will all improve under Secure CAN methods regardless of the vulnerability being evaluated. CAN switches and the resulting filtering capabilities will increase Complexity of the network and thus, the attack; it will also restrict any potential impacts to the sub-CAN the vulnerable system is isolated to. The differences seen between Basic and Secure CAN Architectures in this one example would be very similar across all vehicular system vulnerabilities. As a result, it's expected and shown that the Secure CAN Architecture will decrease the severity of vehicular vulnerabilities, and thus improve a vehicle's defense capabilities and overall security.

4.6. Chapter Summary

The analysis conducted in this chapter addressed several key concerns regarding the implications of the proposed Secure CAN architecture including: any introduced security vulnerabilities, applicability in today's automotive sector, performance impact of the added hardware, and lastly the effectiveness of the proposed architectural changes. These concerns were evaluated to ultimately show that Secure CAN offers a viable solution for vehicle security. It potentially offers improvements in growth, filtering, access control, transmission latency, and data restriction. It also introduces capabilities such as partitioning and functional separation, network management, fail secure and intrusion prevention. All of these benefits are achievable with minimal impact to current vehicle architectures.

CAN Switches in their current design, are remarkably simple, with no dependence on network protocols for proper functionality. As a result, penetrating and compromising the CAN Switches themselves is difficult, limiting introduced vulnerabilities.

Application to a modern vehicle was theoretically demonstrated to show that Secure CAN methods could likely be incorporated into modern designs, providing examples of several design decisions, and addressing concerns that may arise with the inclusion of Secure CAN. Further, processing delay across a CAN Switch was measured to provide an idea of the worst-case latency to be incurred when communicating between sub-CANs.

Finally, a method of quantifying the overall security improvement of Secure CAN was proposed. CarVSS provides a metric to allow the qualitative comparison between Secure CAN and other CAN Implementations. This analysis showed that the Secure CAN Architecture offers quantifiable improvement over Basic CAN implementations.

V. Conclusions

5.1. Summary

After careful investigation, it's clear that modern vehicles rely heavily on CAN for in-vehicle communication. While CAN provides simple, robust, speedy and reliable communication, it does come with some drawbacks and vulnerabilities. Literature has shown that these vulnerabilities realistically threaten the automotive industry today and that we are limited in our defenses and response capabilities.

The research in vehicle networks led to an architecture that, although very simple, offers several improvements over current implementations of CAN. This architecture could easily be incorporated into future vehicle designs, and with a little more effort, potentially be applied to current production models. Modeling and simulation provided a low cost, low risk, and flexible platform of which further exploration could be carried out. In house developed ECUs and HIL simulation capabilities combined to provide a functional network under the proposed architectural changes. Considering the overall objective of demonstrating the potential security improvement of an architectural solution, the developed CAN Switches and network allowed for such demonstration. Although presented on a vehicle, the proposed architectural modifications would likely be beneficial in other comparable protocols and applications.

The analysis addressed critical concerns regarding the proposed architectural changes and claimed improvements. It helped highlight several advantageous implications of the recommended solution. Further it showed that along with security improvement; CAN switches are also fairly secure in their inherent design, and as a result add minimal risk. While performance penalties are expected to be limited, considerations and tradeoffs may be necessary

to balance all network requirements. Lastly, this analysis provided a simple scoring system, CarVSS, to show that Secure CAN improves the security of a vehicle while also reducing the overall severity and impact of vehicular vulnerabilities. All of these concerns were evaluated and discussed; illustrating that Secure CAN offers a meaningful contribution to vehicle security.

5.2. Contributions

As claimed in Chapter I, this thesis research has provided several different contributions to the field of vehicular security. Through the development of CAN Switches, the notion of packet switching a CAN network was introduced. The hardware provided a mechanism to physically and functionally separate nodes on a CAN network; improving aspects in security and some in performance. These CAN Switches also improve filtering capabilities, providing the ability to do more than just deep packet inspection. In combination with an intrusion detection system, the Secure CAN Architecture itself has the ability to act on possible detections; and thus enables the first potential vehicle Intrusion Prevention System. Lastly, the overall Secure CAN Architecture was evaluated using a blended vulnerability assessment framework, CarVSS, the first example of a security evaluation framework specifically designed for vehicles. An assessment framework such as this could provide a common language and the ability for the vehicle security community to enumerate and prioritize potential threats and resources.

5.3. Challenges in Vehicular Security

This thesis introduced several constraints and challenges specific to Secure CAN that influenced the direction of the work conducted. As it relates to vehicular security as a whole, there are many more challenges and limitations involved.

Security in a vehicle will either be added to the ECUs themselves or incorporated via newly introduced hardware. Regardless of where security is implemented, computing resources will be limited and all systems will be held to hard requirements in complexity, memory size, and runtime efficiency. Further systems must also meet several physical requirements ensuring that they are able to withstand the extreme conditions they are often exposed to in the vehicle environment [3]. If additional hardware is a part of the security solution, that hardware must also have a product lifetime equal to that of a standard vehicle, which can be upwards of 20 years. Security must also accommodate a wide range of vehicles and vehicle owners. Solutions must largely function autonomously and independent of most user interaction, but also offer restricted control to users whom wish to be more involved in their vehicles security.

Vehicle security remains expensive in terms of researcher time, and the cost of acquiring representative vehicles as well as simulating and emulating either the environment or its components.

Lastly, vehicle security is a relatively young and evolving field. As a result, standards in security are rather limited and manufacturers are left to implement proprietary solutions. This can be extremely challenging from the perspective of the researcher. As a result of the lack of standards, the security community is also limited in their ability to accurately measure and communicate the security of a vehicle. This makes it extremely difficult to compare solutions, evaluate vulnerabilities, and effectively manage resources. If standards were developed manufacturers and researchers could work in unison to further communicate and improve the security of all vehicles.

5.4. Future Work

While this thesis offers one possible solution to an ever-increasing problem in vehicular security, there are several avenues to continue to expand the scope of this research. This research suggests that future work will be most valuable if efforts are dedicated to security improvements in two primary focus areas: Autonomy and Over the Air Updates. Based on trends in today's automotive evolution, this seems to be the future of vehicular technology. These two categories are rather broad and can include a number of ideas. Here are just a few, specific to the work presented in this thesis.

5.4.1. Intrusion Prevention

It was briefly described that Secure CAN has the potential to offer a viable vehicle intrusion prevention system. It's recommended that work be done to first implement one possible method of intrusion detection and tie that detection system into the Secure CAN kill switch. Beyond this, efforts might also go towards a Secure CAN unique detection method or improved response options to allow a fine tuned reaction dependent upon the actual detection triggered.

5.4.2. Improved Simulation

Currently the TORCS simulation software is rather limited in its capabilities. For the purpose of this thesis and the desired demonstration capabilities, simulating gas, brakes, and steering, served its purpose. However, for future work, it's recommended that these capabilities be expanded to include more realistic data. This would include simulation of nearly all ECUs, sensors, switches, actuators, etc. to more accurately replicate the CAN data seen on a typical CAN bus. This would bring more realism to the demonstration as well as lead to potential

analysis of Secure CAN Performance under realistic bus loads. BeamNG software, available at [48], offers one possible solution to accomplish these goals.

5.4.3. More ECUs and Mechanical Systems

Like the need for improved simulation capabilities, it might also be beneficial to incorporate more real world vehicle ECUs and their respective systems. If able, it might also be nice to incorporate mechanical aspects of such systems. These additions would aid to provide more realistic CAN traffic further enabling the capabilities within the laboratory environment. This option is certainly more costly in the required resources to acquire such parts and the time/expertise to accurately reverse engineer them. If an off the shelf simulation/HIL solution is available, it might provide the answer to both 5.4.2 and 5.4.3.

5.4.4. Tools

The tools utilized in this research were very basic and limited in their capabilities. With the exception of the NeoVI FIRE 2 OBD-II Adapter and the Vehicle Spy software, all were free and open source options. It's recommended that time be spent further investigating additional tools that might simplify or amplify CAN understanding and capabilities. A great example can be seen in [17] where Miller and Valasek made use of the Jeep's mechanics diagnostic tool. While extremely costly, this tool was crucial to their active tests and ECU unlocking.

If any particular vehicular platform is of interest, other mechanics diagnostic tools might reveal valuable information about that vehicle and its components. For example, the Jeep diagnostic tool was developed by wiTECH, and can offer several extremely helpful features. It's simple to use and even displays various aspects of the vehicle, to include a graphical representation of the Jeep's network topology and architecture. A feature such as this would greatly assist the uneducated researcher in redesigning a vehicle's network to include Secure

CAN methods. Tools such as these could provide a valuable insight allowing a wider evaluation of other vehicle architectures and CAN implementations.

5.4.5. Secure Over The Air Updates

Section 3.5.6 briefly mentions Uptane, the effort towards secure software updates for modern automobiles. Based on trends thus far, it's safe to assume that regular patching and updating of vehicle components is a necessary function. Ensuring that the process can occur as securely as possible would be extremely valuable.

It is thus recommended that future work involve investigating the potential role that CAN Switches and the Secure CAN Architecture might play in secure over the air updates. Uptane describes a tiered process to verifying TUF metadata on vehicles. This involves a primary ECU (1 per vehicle) that downloads, verifies and further distributes updated ECU images. The secondary ECU would depend on the primary to receive such updates [26]. Perhaps Secure CAN could assist in controlling access to a critical ECU such as the primary ECU in this process. Would isolating the primary ECU to its own sub-CAN provide any further security in this sensitive sequence of events?

Following investigation of the possibilities, it's also recommended that Uptane's method of securing over the air updates be implemented in the current laboratory environment. The combination of ideas could lead way to an overall defense in depth approach to securing vehicles.

5.4.6. Potential Vulnerability Assessment Improvements

While the developed vulnerability assessment framework serves its purpose in this analysis of Secure CAN, there are many potential ways to improve it for vehicles. The proposed assessment utilizes only the Base scoring calculations. There may be some benefit to further analyzing the possibility around including both Environmental and Temporal score modifiers. For example, some elements of security that aren't captured in the current assessment framework are Magnitude and Lifespan of the vulnerability. Magnitude is intended to capture the spread of the vulnerability. In other words, it would capture the number of vehicles affected by the potential vulnerability. Is it a vulnerability, that when exploited affects a single car? Or is it one that affects an entire fleet? Lifespan, on the other hand, is somehow meant to capture the life of the vulnerability. Things to consider when scoring this metric are the expected response time or time to mitigate the effects of the attack itself; and an appropriate fix time, or the time to patch and permanently mitigate the vulnerable system.

For magnitude, it's possible that CVSS might provide a possible solution. CVSS v3 incorporates an Environmental score that would be an appropriate place to capture a metric such as this. However, this score only offers a slight modification to that of the Base metrics should they differ in the vulnerability analyst's particular environment. With this in mind, the current environmental score does not capture the magnitude of vulnerability. This may need to be captured in a separate score or some how combined into one of the 3 provided scores.

CVSS v3 also incorporates a Temporal Metric score which measures the current state of a vulnerabilities exploit techniques, code availability, and the existence of any patches or workarounds. Rather than try to incorporate our own metric of lifespan into the current scoring algorithms, it may be more effective to utilize the Remediation Level metric within the Temporal

Score. The other metrics Exploit Code Maturity and Report Confidence can either be incorporated if knowledge is known, but it's expected that they will be left undefined so that they do not impact the overall Temporal Score.

5.4.7. Real World Implementation

Following the work conducted thus far in developing and analyzing Secure CAN, it is natural to question actual application in a real vehicle. While section 4.3 theoretically presents Secure CAN network designs of a 2015 Subaru WRX, it would be interesting to attempt to implement such a design or similar on an actual vehicle. Efforts in this area might lead to concerns that were not apparent in the table-top discussion presented in this thesis.

There are several limitations in safety, logistics, and time that might make this a very difficult venture. It would first be extremely costly to obtain a vehicle simply to take apart. Successful implementation would require detailed knowledge of the specific vehicle and its components. This could be obtained with assistance from the vehicle manufacturer if improved security is a shared desire. However it will likely be the case that system components require reverse engineering to accurately understand functionality.

Bibliography

- [1] R. N. Charette, "This car runs on code," *IEEE Spectr.*, vol. 46, no. 3, p. 3, 2009.
- [2] P. Kleberger, T. Olovsson, and E. Jonsson, "Security aspects of the in-vehicle network in the connected car," *IEEE Intell. Veh. Symp. Proc.*, no. Iv, pp. 528–533, 2011.
- [3] M. Wolf, A. Weimerskirch, and T. Wollinger, "State of the art: Embedding security in vehicles," *Eurasip J. Embed. Syst.*, vol. 2007, 2007.
- [4] A. Groll and C. Ruland, "Secure and authentic communication on existing in-vehicle networks," *IEEE Intell. Veh. Symp. Proc.*, pp. 1093–1097, 2009.
- [5] D. K. Nilsson, U. E. Larson, and E. Jonsson, "Efficient in-vehicle delayed data authentication based on compound message authentication codes," *IEEE Veh. Technol. Conf.*, pp. 1–5, 2008.
- [6] M. L. Chávez, C. H. Rosete, and F. R. Henriquez, "Achieving confidentiality security service for CAN," *Proc. - 15th Int. Conf. Electron. Commun. Comput. CONIELECOMP 2005*, vol. 2005, no. Conielecomp, pp. 166–170, 2005.
- [7] K.-T. Cho and K. G. Shin, "Fingerprinting Electronic Control Units for Vehicle Intrusion Detection," in *USEC*, 2016.
- [8] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," *Int. Conf. Inf. Netw.*, vol. 2016–March, pp. 63–68, 2016.
- [9] M. Farsi, K. Ratcliff, M. Barbosa, and K. Ratcliff, "An overview of Controller Area Network," *Comput. Control Eng.*, vol. 10, no. June, pp. 113–120, 1999.
- [10] K. H. Johansson, T. Martin, and L. Nielsen, "Vehicle Applications of Controller Area Network," *Handb. Networked Embed. Control Syst.*, pp. 741–765, 2005.
- [11] International Standard Organization, "Road vehicles — Controller area network (CAN) — Part 3: Low-speed, fault-tolerant, medium-dependent interface," *Norm*, vol. 3, no. ISO/DIS 11898-3, 2006.
- [12] International Standard Organization, "Road vehicles — Controller area network (CAN) — Part 4: Time-triggered communication Véhicules," *Norm*, vol. 4, no. ISO/DIS 11898-4, 2004.
- [13] M. Wolf, A. Weimerskirch, and C. Paar, "Security in Automotive Bus Systems," *Proc. Work. Embed. Secur. Cars*, no. July, pp. 1–13, 2004.
- [14] National Instruments, "FlexRay Automotive Communication Bus Overview," 2016. [Online]. Available: <http://www.ni.com/white-paper/3352/en/>. [Accessed: 14-Nov-2016].

- [15] K. Koscher *et al.*, “Experimental security analysis of a modern automobile,” *IEEE Symp. Secur.*, 2010.
- [16] S. Checkoway *et al.*, “Comprehensive Experimental Analyses of Automotive Attack Surfaces,” *System*, pp. 6–6, 2011.
- [17] C. Miller and C. Valasek, “Remote Exploitation of an Unaltered Passenger Vehicle,” *Blackhat USA*, vol. 2015, pp. 1–91, 2015.
- [18] T. Hoppe, S. Kiltz, and J. Dittmann, “Security threats to automotive CAN networks Practical examples and selected short-term countermeasures,” *Reliab. Eng. Syst. Saf.*, vol. 96, no. 1, pp. 11–25, 2011.
- [19] I. Rouf *et al.*, “Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study,” *Proc. USENIX Secur. Symp.*, vol. 39, no. 4, pp. 11–13, 2010.
- [20] T. Leinmüller *et al.*, “SEVECOM - Secure Vehicle Communication,” *Sci. York*, vol. 2006, no. 3, pp. 1–5, 2006.
- [21] M. Raya and J. Hubaux, “Securing vehicular ad hoc networks,” *J. Comput. Secur.*, vol. 15, pp. 39–68, 2007.
- [22] T. Matsumoto, M. Hata, M. Tanabe, K. Yoshioka, and K. Oishi, “A Method of Preventing Unauthorized Data Transmission in Controller Area Network,” *2012 IEEE 75th Veh. Technol. Conf. (VTC Spring)*, pp. 1–5, 2012.
- [23] B. Howard, “Hack the diagnostics connector, steal yourself a BMW in 3 minutes,” *ExtremeTech*, 2012. [Online]. Available: <http://www.extremetech.com/extreme/132526-hack-the-diagnostics-connector-steal-yourself-a-bmw-in-3-minutes>.
- [24] EDILock, “BMW and Mini Cooper key/transponder programming device via OBD II,” *EDILock Ltd.*, 2016. [Online]. Available: <http://www.vag-info.com/category/preview/BMW?scroll=51>.
- [25] K. Poulsen, “Hacker Disables More Than 100 Cars Remotely,” *WIRED*, 2010. [Online]. Available: <https://www.wired.com/2010/03/hacker-bricks-cars/>. [Accessed: 12-Nov-2016].
- [26] T. Karthik *et al.*, “Uptane : Securing Software Updates for Automobiles.”
- [27] Beagleboard.org, “BeagleBone Black.” [Online]. Available: <https://beagleboard.org/black>. [Accessed: 27-Nov-2016].
- [28] ARDUinoautoMOTIVE, “Raspberry Pi2 Car PC System.” [Online]. Available: <http://www.ardumotive.com/raspberry-car-pc.html>. [Accessed: 26-Jan-2017].

- [29] RasPi.TV, “Raspberry Pi 2 – the New Quad-core A7 Pi2.” [Online]. Available: <http://raspi.tv/2015/raspberry-pi-2-the-new-quad-core-a7-pi2>. [Accessed: 27-Nov-2016].
- [30] adafruit, “Pi Foundation Display - 7" Touchscreen Display for Raspberry Pi.” [Online]. Available: <https://www.adafruit.com/products/2718>. [Accessed: 27-Nov-2016].
- [31] E. Evenchick, “CANTact.” [Online]. Available: http://store.cantact.io/products/cantact-v1-0?source_app=embed&referer=http%3A%2F%2Flinklayer.github.io%2Fcantact%2F. [Accessed: 27-Nov-2016].
- [32] Intrepid Control Systems Inc, “neoVI FIRE 2,” 2016. [Online]. Available: <https://www.intrepidcs.com/products/vehicle-network-adapters/neovi-fire-2/>. [Accessed: 27-Nov-2016].
- [33] OBD2 Diagnostic Inc, “OBD2 Diagnostic Connector Kit.” [Online]. Available: <http://www.obd2allinone.com/products/obd2dlc.asp>. [Accessed: 27-Nov-2016].
- [34] debian, “Linux-CAN / SocketCAN user space applications.” [Online]. Available: <https://packages.debian.org/sid/can-utils>. [Accessed: 27-Nov-2016].
- [35] KODI, “OSMC,” 2016. [Online]. Available: <http://kodi.wiki/view/OSMC>. [Accessed: 27-Nov-2016].
- [36] KODI, “About Kodi,” 2016. [Online]. Available: <https://kodi.tv/about/>. [Accessed: 27-Nov-2016].
- [37] Intrepid Control Systems Inc, “Vehicle Spy Professional,” 2016. [Online]. Available: <https://www.intrepidcs.com/products/software/vehicle-spy-professional/>. [Accessed: 27-Nov-2016].
- [38] B. Wymann, “TORCS.” [Online]. Available: <http://torcs.sourceforge.net/>. [Accessed: 27-Nov-2016].
- [39] D. Chandrashekar, H. Mehta, K. Mohta, U. Drolia, and V. Kumar, “AutoNet.” [Online]. Available: <https://github.com/mlab-upenn/AutoPlug-2.0/blob/master/Documentation/AutoNet2010.pdf>. [Accessed: 26-Jan-2017].
- [40] Subaru, “WRX & STI Factory Service Manuals.” [Online]. Available: http://www.wrxinfo.com/service_manuals/.
- [41] Mitchell1, “ProDemand.” [Online]. Available: <https://prodemand.com/>. [Accessed: 22-Feb-2017].
- [42] Teledyne Lecroy, “WavePro 7 Zi-A Oscilloscopes.” [Online]. Available: <http://teledynelecroy.com/oscilloscope/oscilloscopeseries.aspx?mseries=341>. [Accessed: 22-Feb-2017].

- [43] K. Amano, “Estimation of the Timing of Human Visual Perception from Magnetoencephalography,” *J. Neurosci.*, vol. 26, no. 15, pp. 3981–3991, 2006.
- [44] P. E. Black, K. Scarfone, and M. Souppaya, “Cyber security metrics and measures,” *Sci. Technol.*, vol. 5, pp. 1–9, 2008.
- [45] FIRST, “Common Vulnerability Scoring System v3.0: Specification Document,” pp. 1–21, 2015.
- [46] NIST, “CVSS v3 Information.” [Online]. Available: <https://nvd.nist.gov/cvss.cfm>. [Accessed: 29-Dec-2016].
- [47] NHTSA, “National Institute of Standards and Technology Cybersecurity Risk Management Framework Applied to Modern Vehicles,” 2014.
- [48] BeamNg, “BeamNG.” [Online]. Available: <http://www.beamng.com/>.

REPORT DOCUMENTATION PAGE				Form Approved OMB No. 074-0188	
The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.					
1. REPORT DATE (DD-MM-YYYY) 23-03-2012		2. REPORT TYPE Master's Thesis		3. DATES COVERED (From – To) July 2015 – March 2017	
TITLE AND SUBTITLE Securing Controller Area Networks in Vehicles Via Packet Switched Network Segregation				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Caberto, Eddie K., Captain, USAF				5d. PROJECT NUMBER	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/ENG) 2950 Hobson Way, Building 640 WPAFB OH 45433-8865				8. PERFORMING ORGANIZATION REPORT NUMBER AFIT-ENG-MS-17-M-009	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Intentionally left blank				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT DISTRIBUTION STATEMENT A. APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.					
13. SUPPLEMENTARY NOTES This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States.					
14. ABSTRACT As automobiles become increasingly connected via multiple wireless capabilities, the lack of security has become a substantial vulnerability. This growth in functionality and convenience has also increased access to a vehicle's Controller Area Network (CAN). CAN, the primary intra-vehicle network, allows time-sensitive communication between electronic control units (ECU's) that control one or many in-vehicle systems. Although CAN has proven very effective in data transfer, it was not designed for security. While some steps could be taken to add security layers and features to the existing CAN protocol, introducing security inevitably adds cost, data latency, and potentially reduces data throughput. There is a growing need to secure CAN networks without completely changing the protocol. To improve the security within an automobile without an overhaul to the popular CAN protocol, this research developed the Secure CAN Architecture to provide security primitives at the data link layer. When combined with existing network security techniques, it introduces a number of possible security features. A flexible architecture such as this one provides vehicle manufacturers with an option to securely architect their CAN networks in current and future vehicle designs mitigating specific current, and possibly future, risks on an automotive CAN network. These methods apply to other applications with similar communication protocols.					
15. SUBJECT TERMS Controller Area Network, CAN, Security, Safety, Vehicle					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT UU	18. NUMBER OF PAGES 128	19a. NAME OF RESPONSIBLE PERSON Scott Graham, AFIT/ENG
a. REPORT U	b. ABSTRACT U	c. THIS PAGE U			19b. TELEPHONE NUMBER (Include area code) (937) 255-6565, ext 4581 (NOT DSN) (scott.graham@afit.edu)

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18