UNIVERSITY OF WESTERN AUSTRALIA

FINAL YEAR PROJECT

Evaluating Attack Risk on Connected Vehicles

Terence Leong

School of Electrical, Electronic & Computer Engineering

Supervised by

Dr. Jin HONG
Faculty of Engineering, Computing & Mathematics

18 October 2020

5290 Words

Abstract

Cybersecurity has become a very necessary research topic for modern vehicular systems due to the reliance on external network systems for communication. However, developing strong defences against attack threats have been proven difficult due to a variety of restrictions in terms of software and hardware. Many of these issues could stem from the limited information and clarity that is known about these attacks. Therefore an attempt to address this problem came in the form of presenting a method of capturing the relationship between attack and defence, whilst providing a greater insight into potential cost benefits.

The goal of this research focuses on researching widely abused security threats that targets the ECU and developing a method to quantify and classify these attacks. This was accomplished through analysing the current industry standard of vulnerability classification and other proposed methods of classification over the last few years. This information is then used to suggest a systematic approach to quantify these attack with the end goal of calculating the return on attack for each security threat. This quantification method then influenced the metrics that were selected for classifying attacks with the key features identified to be methods of performing attack and the components that they affect. An optimisation algorithm was then developed to minimise the implementation cost of defences, whilst maximising the coverage of security threats. the optimisation algorithm was then evaluated through the use of sensitivity analysis.

Contents

1 Introduction

In recent decades, modern vehicles have been rapidly increasing their number of in-vehicle electronic devices and ever increasing complexity of in-vehicle systems. Therefore it has been imperative to rely upon external network systems for efficient communication services. However, this development in technology has exposed them to new types of internal and external threats. As a result, cyber security has become one of the most significant challenges in connected vehicular systems [1], [2]. The biggest security concern relates to the in-vehicle network composed of Electronic Control Units (ECU) and various buses such as the Control Area Network (CAN) bus, which are primarily used to monitor and control the state of vehicular systems [3]. A variety of attack types against the ECU and buses have been discovered over the years and a number of defence techniques have been developed to combat them.

In 2010, Koscher et *al.* were the first group of researchers to perform an attack on a real car [4]. They were able to successfully control a range of modules in two 2009 automobiles of the same make and model by taking advantage of the vulnerabilities presented by the CAN bus through frame injection. In 2015, Chrysler was forced to recall 1.4 million vehicles due to a pair of hacker demonstrating that it was possible to remotely hijack a Jeep's digital system over the internet [5]. As those examples demonstrate, a more pro-active approach has to be taken to efficiently react to these attacks. In a recent survey study, eight common attack patterns were identified and classified including Denial of Service Attacks (DoS), black hole attacks and replay attacks [3].

Extensive research attention has been paid to develop solid defences against these attacks. Defences can be categorised based on their approach: crytographic defence, network security defence, software vulnerability detection defence and malware detection defence [2]. The most robust and highly researched area involves cryptography with defence techniques covering all the eight common attack patterns. However, limitations still exist for these techniques. Some of these limitations include high overhead, difficult implementation and low processing speed. Even though there exist a long-established computer security policy, the industry standard does not follow it due to hardware constraints and differences in network configuration for these in-vehicle systems [6], [7], which further increases the difficulty to develop an all-encompassing defence.

The main motivation is to optimise the ability to select the most cost efficient defences against a wide variety of attacks. The current approach to determining the risk and impact of security threats rely upon the Common Vulnerability Scoring System (CVSS). However, CVSS only analyse threats based on vulnerabilities and nuances presented by attacks are usually overlooked. Therefore in that regard, a novel threat modelling framework is proposed. The proposed model will be able to capture details presented by attacks to capture a larger amount of information, allowing a better understanding on security threats. In this paper, first, a novel security threat modelling framework is proposed to extend the capabilities of CVSS. Second, I developed a heuristic-based optimisation algorithm called Ekko-OPT to solve the attack cover problem.

The contribution of this paper is summarised as follow:

- I propose a novel threat modelling framework taking into account the widely accepted CVSS;
- I develop a novel heuristic-based optimisation algorithm to optimise the attack cover problem
- I demonstrate the application and feasibility of the threat model framework
- I evaluate the effectiveness of the optimisation algorithm through metrics such as time comparison and scalability

The rest of the paper is organized as follows. Section II presents the background and literature review on vehicular security.Section III presents the proposed security modelling framework and heuristic-based optimisation algorithm. The demonstration of the framework and results of the experiments conducted are presented in Section IV. In Section V, the findings and limitations are discussed, whilst the paper is concluded with future work in Section VI.

2 Literature Review

2.1 Vehicular Security Technology

2.1.1 Vehicular System Architecture

The central communication point for an in-vehicle system is the Electrical Control Unit (ECU). The ECU is an embedded device that was designed to monitor vehicle state and control multiple systems within the overall architecture [8], [9]. Due to the number of increasing features in a vehicular system, the ECU was developed as multiple components each controlling a different set of functions within the system. This meant that a network had to be developed for seamless communication between ECU components.

Nowadays, various networks exist which provide efficient communication amongst control modules within a vehicle. Each of these networks help provide efficient communication and strengthens the overall performance of the vehicle. The most commonly found networks are the Controller Area Network (CAN), FlexRay, Media Oriented System Transport (MOST) and Local Interconnect Network (LIN) [8], [10], [11]. The networking type that I'll be focusing on is the CAN bus. This is because the CAN bus is the most widely used protocol for in-vehicle networks, allowing different ECU components to communicate with each other through a wired CAN bus. In addition, the CAN bus is heart of a connected vehicular system, therefore securing it would greatly improve a vehicular system's defence capabilities.

2.1.2 Controller Area Network

The CAN bus is a robust linear-type network designed to allow devices to communicate with each other without a central master that controls all the devices.Cabling for a CAN bus is low in cost and due to the linear nature of this network, it makes it easy for new nodes to connect to it [8]. However, the CAN bus was designed as a closed network and has several vulnerabilities such as:

- Multicast Messaging: Messages are broadcasted to everyone. Passive attackers are capable of listening in on this broadcast with ease.
- No authentication: There is no way to ensure that messages received are from a credible source. Attackers are able to perform attacks by pretending to be a CAN bus.
- No encryption: Data on a CAN bus is not encrypted. Attackers can easily analyse a CAN frame.
- Common point of entry: Once a CAN bus has been compromised, it is possible to access all parameters on it without limit.

Despite that, the CAN bus only defines the protocols for the physical and data link layers within an OSI model. Hence, error checking and addressing are not implemented for it. The network layer is not included within a CAN protocol because it would require more infrastructure for the routing process in addition to the higher cost investment compared to a data link layer approach. Lower latency is also highly preferred for a vehicular system, therefore a data link layer solution is more pragmatic [12].

2.2 Security Attacks

As the prominence of connected cars rise throughout the years, so has the variety of attacks that can be performed on them. These attacks can be analysed from two different perspectives: the attack surface and attack vector [1]. The attack surface is described as the total sum of vulnerabilities present in the network that is accessible to attackers. In this scenario, there exist an attack surface through the On-Board Diagnostic II (OBD-II) and firmware of the media player manufacturer. Whereas the attack vector is seen as the pathway taken by attackers to achieve a malicious outcome. This could include the potential exploitation of web applications designed for vehicles or spoofing certain commands to gain access to on-board devices. The following is the different attack vectors that are typically taken to exploit a vehicular system:

- Denial of Service Attacks (DoS): This form of attack involves flooding the network with large amount of packets in an attempt to overload it. This would prevent the transmission and processing of legitimate incoming packets.
- Distributed Denial-of-service Attacks (DDoS): The basic concept of flooding the network is exactly the same as a DoS attack. However, due to the high strain placed on the attacker's resources in a DoS, DDoS was developed. As opposed to using a single attacking node, DDoS utilises multiple IP

addresses to perform an attack. Thereby reducing resource strain dramatically.

- Black-Hole Attacks: This form of attack creates a metaphorical 'black hole' where no packets are able to move through the network. This type of attack causes serious damage to network performance and routing. A well placed black-hole node on a critical path could cripple the entire network.
- Replay Attacks: This form of attack attempts to capture information before relaying the packet to its destination. Replay attacks allow the attacker to use the intercepted information later to authenticate themselves or attack the network.
- Impersonation Attacks: This form of attack allows the attacker to gain access to information and resource that is otherwise restricted. Impersonation attacks are typically seen following a replay attack, as a replay attack would provide login details and passwords.
- Fuzzy Attacks: This form of attack is achieved by iterative injection of random packets. This allows the attacker to find vulnerabilities through unexpected values or random data entered into the network.
- Malfunction Attacks: This form of attack target specific functions within the network and cause it to stop working.

These large number of security threats can be performed through internal and external attacks. In terms of internal attacks, the attacker would typically exploit the vulnerabilities found in the OBD-II port. This would allow for eavesdropping on bus traffic or even frame sending. In addition, if an attacker is capable of controlling a single node in the network, current protocols make it possible for him to gain control over every ECU in the network. External attacks mostly refers to the exploitation of a vehicle's communication interface, therefore it would not require physical access to the embedded network. These attacks are performed through indirect physical access, short range wireless access or even long range wireless access [13].

2.3 Defence Against the Attacks

An increase in connectivity for vehicular systems has proven to expose vehicles to more security and safety dangers, making safety critical features a huge concern in modern day research. Despite all research efforts, there is currently no singular conclusive method to defend against attacks.

Based on a recent survey study, security defences can be categorised into: cryptography, network security, software vulnerability detection and malware detection [2]. The most immensely researched of which is the cryptographic approach to authentication and integrity. Message authentication code (MACs) and digital signature are some prime examples of this. As a CAN frame can only accommodate up to eight bytes, it is difficult to implement a crytographic approach that is robust enough to prevent all attacks [14]. CAN+ attempts to solve this issue by using out of band channels to transmit bits instead. However a large drawback of this method and all similar ones remain in the form of backward compatibility [15]. Some have attempted to address this problem, but none of which provide the security necessary for this method to be reliable enough for practical use. Therefore it can be concluded that methods that rely solely on cryptography will be vulnerable to brute force attacks. Table 1 shows the mapping of the application for existing effective defence against common security attacks; whilst table 2 contains a list of defences that have been proposed over the course of the last decade and the attacks that they are implemented against.

	Defence Categories			
Cyber-Attacks	Cryptography	Network Security	Software Vulnerability Detection	Malware Detection
DoS	x	x		
DDoS	x	x		
Black-hole	x	x		
Replay	x			
Sybil	x	x		
Impersonation	x	x		
Malware	x		x	x
Falsified Information	x	x		
Timing	x			

Table 1: Application of Defence Type to Cyber-Attack Type [2]

2.3.1 Intrusion Detection System (IDS)

An IDS is a system capable of monitoring traffic moving on networks and through systems for suspicious activity and issues alerts when such activity is discovered [64]. An example of this is an anomaly intrusion detection algorithm based on survival analysis proposed by Kim et *al.* [1] Anomaly detection is used to detect suspicious patterns that do not conform to the expected behaviour, whereas survival analysis is a statistical model used to discover survival rate and survival duration of measurement objects [1]. The proposed method can be split into two major components: chuck-based threshold measurement and the detection algorithm.

By applying survival analysis in this manner, the IDS is not restricted to searching for specific patterns, it is also capable of detecting unknown attacks and doesn't require regular updates of signature concerning attack patterns. However, the major disadvantage posed by this method is the fact that it is incapable of checking the condition of the vehicle or perform safety diagnostics either. It is also incapable of determining the impact and risk that these attack threats pose on the system, hence the requirement for a classification and quantification attacks.

Category	Security Solutions	Main Mitigated Attacks
Cryptography	2FLIP [16]	DoS
	TESLA [17]	DoS, Packet injection
	RAISE [18]	Impersonation
	PACP [19]	Eavesdropping, Replay, Impersonation
	ECDSA [20]	All malicious attacks
	SA-KMP [21]	DoS, Replay, Impersonation
	GKMPAN [22]	Eavesdropping, Replay
	PPAA [23]	Sybil
	Calandriello et al. [24]	DoS, Jamming
	PPGCV [25]	Collusion
	TACKs [26]	Eavesdropping, Sybil, Correlation
	GSIS [27]	DoS
	SRAAC [28]	Unauthorised access
	DABE [29]	Collusion
	ABACS [30]	Collusion
	Xia et al. [31]	Collusion, Replay
	Bouabdellah et al. [32]	Black-hole
Network Security	Bißmeyer et al. [33]	Sybil
	REST-Net [34]	Impersonation, Falsified information
	CIDS [35]	DoS, Masquerade
	Martynov et al. [36]	DoS
	IDFV [37]	Selective forwarding, Black-hole
	Song et al. [38]	Message injection
	Zaidi et al. [39]	Sybil, falsified information
	OTIDS [40]	DoS, Impersonation, Fuzzy
	PES [41]	Sybil
	AECFV [42]	Black-hole, Worm hole, Sybil
	Markovitz et al. [43]	Falsified information
	PML-CIDS [44]	DoS, Probing, Unauthorised access
Software Vulnerability Detection	Tice et al. [45]	Control-flow
	Dahse and Holz [46]	XSS, Remote code execution
	PITTYPAT [47]	Control-flow
	DFI [48]	Buffer overflow
	FindBugs [49]	Buffer overflow
	Generational search [50]	Malware (Bug)
	TaintCheck [51]	Overwrite
	Dytan [52]	Control-flow, Data-flow, Overwrite
	GenProg [53]	DoS, Overflow
	Shin et al. [54]	Malware (bug)
	Perl et al. [55]	Malware (Bug)
	Zhou and Sharma [56]	DoS
	Shar et al. [57]	Injection, File inclusion
	VDiscover [58]	Malware
Malware Detection	MSPMD [59]	Malware
	MRMR-SVMS [60]	Malware
	Huda et al. [61]	Malware
	CloudIntell [62]	Malware

Table 2: Proposed Defence Solutions[63]

3 Methodology

The aim was to develop an attack classification method capable of capturing as much information regarding attack methodology and impact as possible. The proposed classification method was shaped around fulfilling key conditions such as easy to use, reliable and capable of capturing all or most threats. The methodology was centered around developing a novel method that incorporated and expanded on the current industry standards of evaluating vulnerabilities called the Common Vulnerabilities Scoring System, CVSS. A new systematic approach to classification called Path-based Attack Classification Tree (PACTree) is developed based on the knowledge gathered. PACTree is then tested against CVSS to evaluate its reliability and effectiveness.

Finally, an optimisation algorithm is developed to minimize the cost of defence whilst defending against all or as many potential attacks as possible. Arbitrary cost are then assigned to the list of current potential defences in order to evaluate the effectiveness of the algorithm through sensitivity analysis.

3.1 Developing a Method to Classify Attacks

Prior to developing a novel attack classification method, it was imperative to evaluate the effectiveness of the current industry standard, CVSS. This was accomplished to provide a better understanding on what is currently being used and is functional for the purpose of evaluating attacks. Additional information was gathered through surveying other proposed classification method to better incorporate unique features. All the information was utilised to guide the development of PACTree.

The current industry standard CVSS is used to assess the severity of computer system security vulnerabilities by assigning severity scores to them and allowing responders to prioritise responses and resources according to threat levels. However, CVSS only evaluates vulnerabilities and lacks the ability to assess impact and risk from the point of view of the attack. Therefore when looking at the larger picture, a lot of information regarding attacks are lost when utilising only CVSS. Based on a proposed classification system [65], there are a number of metrics that can be deployed to further expand on CVSS in order to maximise the amount of information obtainable from a perceived threat. The proposed metrics were used as guiding material in order to help construct metrics and relationships between them that could be applied in the context of connected cars. In addition, the selected metrics need to be capable of directing the ability to quantify these attacks.

3.2 Quantification Metrics

When selecting the desired metrics to quantify these security attacks, it was important to determine the level of details which we would like to assess these attacks. Considering the fact that the classification method should be easily accessible and simple to use, generic metrics were selected for the quantification. These include attack cost, attack impact, probability of attack, attack risk and return on attack.

3.2.1 Attack Cost

The native metric used to determine the cost of an attack is attack cost. It is defined as the cost spent by an attacker to successfully exploit a vulnerability (i.e., security weakness) on a host. It is achieved by determining the summation of expected values from the requirements of performing an attack. It is denoted as a_C. The attack cost can reveal how much effort an attacker has to commit to successfully execute an attack. This metric will help determine the expected return an attacker will get from an attack.

3.2.2 Attack Impact

This is the quantitative measure of the potential harm caused by an attacker to exploit a vulnerability. In other words, The maximum potential loss suffered caused by an attacker who successfully compromises the system. It is denoted as a_I and defined as the base impact component of CVSS. The impact metric reveals the damage associated to any possible attack path. A security administrator can use this metric to determine which attack to defend against first. For instance, attack that have the highest attack impact value would be prioritised over everything else and a defence would be implemented for it first.

3.2.3 Probability of Attack

Probability of attack is defined as the likeliness of an attacker to successfully compromise the system. It is denoted as p_A and defined as the overall CVSS score divided by ten. The closer p_A value is to 1, the higher is the likelihood that an attacker will succeed in exploiting the target.

3.2.4 Attack Risk

Attack risk is defined as expected value of the impact on an attack path. It is computed as the product of the probability of attack success p_A and the amount of damage a_C belonging to an attack path. The metric is denoted as a_R and calculated using the equation 1. This metric determines the level of risk associated with an attack.

$$a_R = p_A * a_R \tag{1}$$

3.2.5 Return on Attack

The return on attack is defined as the gain the attacker expects from successful attack over the losses he sustains due to the countermeasure deployed by his target. This security metric is typically used from the attacker's perspective and it is by organisations to evaluate the effectiveness of a countermeasure in discouraging a certain type of intrusion attempts. From a defender's point of view, the network administrator can use this metric to reduce the attacker's benefit by implementing more vigorous defences against attacks with high return on attack. The metric is denoted as ROA and is calculated using the equation 2.

$$ROA = \frac{a_R}{a_C} \tag{2}$$

By calculating ROA, it is possible to determine if implementing certain defences against high cost attack is cost effective. Idealistically, a defence should be available for every possible attack that could be attempted against the system. However, realistically speaking this is near impossible as factors such as cost and physical space is considered.

3.3 PACTree

When designing the classification system, it was important to determine the desired perspective to analyse an attack from. The perspective defined here will act as the foundation of the classification system. The candidates for selecting the foundation included:

- Threat class - This concerns the type of attack it is defined by the six categories in STRIDE: spoofing, tampering, repudiation, information disclosure, denial of service and elevation of privilege.
- Violated property - This concerns personal and private data or information that is subject to data protection regulations and intellectual property.
- Access point - This concerns the attack vector that an attack is being introduced to the system through.
- Impact

The final design and progression of the classification system was called PACTree and can be seen by Figure 1. The access point was ultimately chosen as the foundation of the new classification system. This was the final decision due to the fact that more clarity is shown when separating attacks by access point as opposed to the other metrics. It also provided a clear direction as to the metrics that will be explored next, which is defined by the attack threats that will exploit the current access point. Each attack threat were defined by the class of attack separated using STRIDE and the type of attack it is.

The methods to perform these attacks are dissociated from the attack themselves and constitute a section of its own. The decision to split them up was implemented because the method an attack is performed can

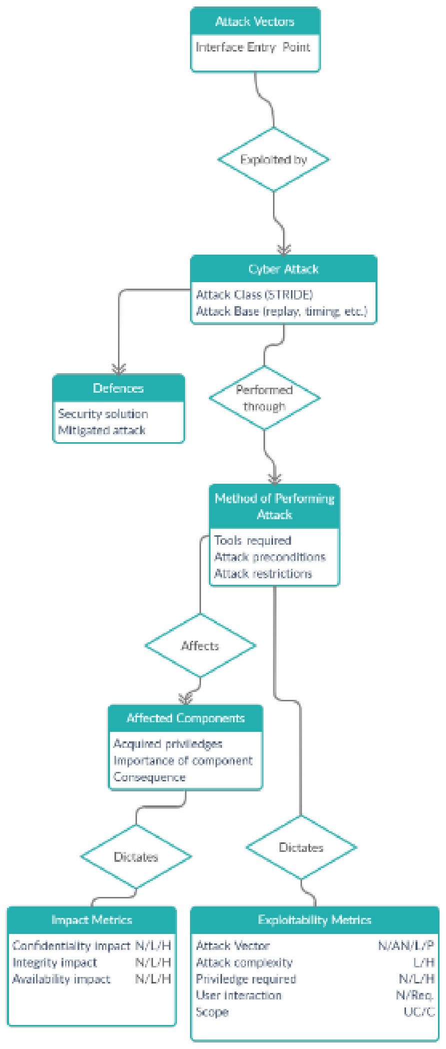
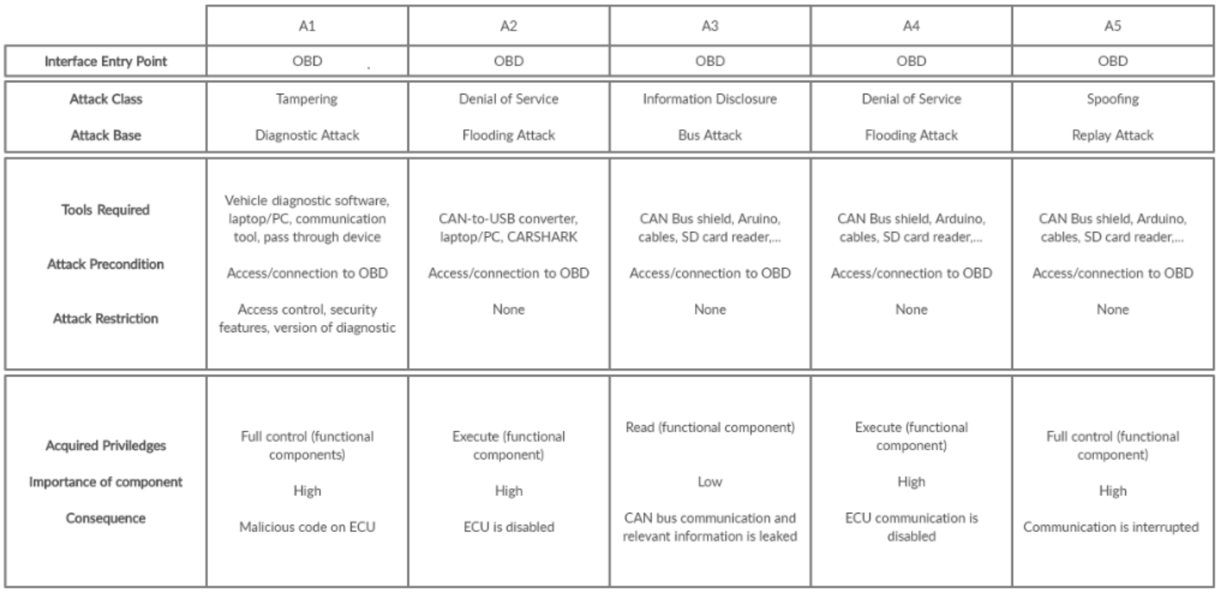


Figure 1: ER diagram illustrating the flow of PACTree

heavily influence the quantification of the threats presented. This is because different methods exist for the execution of a single type of attacks and metrics such as precondition, restrictions and tools required will determine the risk of an attack. Therefore by generalising an attack to its highest impact method of execution or simplest method of execution for example will very unlikely provide the most the necessary insight as to the level of threat an attack can pose.

The components in the vehicle that are affected by an attack are also influenced by the method of attack. Hence it is important to define the affect of an attack by the method of execution as opposed to the actual attack itself. Finally, the quantification of impact and risk is given by the respective sections in CVSS.

3.3.1 Using PACTree



	A1	A2	A3	A4	A5
Interface Entry Point	OBD	OBD	OBD	OBD	OBD
Attack Class	Tampering	Denial of Service	Information Disclosure	Denial of Service	Spoofing
Attack Base	Diagnostic Attack	Flooding Attack	Bus Attack	Flooding Attack	Replay Attack
Tools Required	Vehicle diagnostic software, laptop/PC, communication tool, pass through device	CAN-to-USB converter, laptop/PC, CARSHARK	CAN Bus shield, Aruino, cables, SD card reader,...	CAN Bus shield, Arduino, cables, SD card reader,...	CAN Bus shield, Arduino, cables, SD card reader,...
Attack Precondition	Access/connection to OBD	Access/connection to OBD	Access/connection to OBD	Access/connection to OBD	Access/connection to OBD
Attack Restriction	Access control, security features, version of diagnostic	None	None	None	None
Acquired Priviledges	Full control (functional components)	Execute (functional component)	Read (functional component)	Execute (functional component)	Full control (functional component)
Importance of component	High	High	Low	High	High
Consequence	Malicious code on ECU	ECU is disabled	CAN bus communication and relevant information is leaked	ECU communication is disabled	Communication is interrupted

Figure 2: classifying attacks using PACTree

A list of possible attacks can be seen in Figure 2 to illustrate how PACTree can be used. First determine the interface point of entry that should be examined, which in all these scenario are the OBD-II. Other entry points could include WiFi, Bluetooth or even the Passive Keyless Entry and Start (PKES) system. From there, begin listing all likely attacks that can exploit the OBD-II. The table contains five example attacks that can most likely occur against them. Next, determine the different methods attackers can initiate the attack through this interface. Any variance in tools, precondition or restrictions will result in another column with the same attack as can be seen by attack A2 and A4. Conduct the same process with the affected components, drafting a new column if there is any variance that may occur. Finally, for each column, calculate the CVSS rating, which will be used to determine probability of attack, impact and risk.

To better understand the significance for this method of classification, take attack A2 and A4 from table (next to diagram) as an example. Both attacks are initiated through the OBD-II port and are denial of service attacks. The CVSS ratings of both attacks are identical as well. The slight variation in executing the attack doesn't affect the outcome for the vulnerability rating and can't be captured by CVSS. The proposed method attempts to capture this slight variation that can affect attack cost and as a result return on attack to determine the best approach to implementing a defence.

3.4 Optimisation Algorithm

After achieving a suitable classification and quantification method, it was imperative to develop an algorithm to minimise the cost of implementing defence, whilst maximising the amount of damage we could mitigate. This problem was tackled from the perspective of a weighted set cover optimisation problem, where given a set of defences and their respective defence cost(cost of implementing defence, similar to attack cost), we wanted to determine the number of defences to include in a collection so that the total defence cost is as low as possible, whilst the total value of attack covered as large as possible. A heuristic-based optimisation algorithm called Ekko-OPT was suggested. The foundation of this algorithm was built upon a similar swapping mechanism to 2-OPT in conjunction with a number of heuristics to improve upon an existing local optimal set of defences. The pseudo-code for Ekko-OPT can be found in Algorithm 1.

Defence	**DefenceCost**	**DoS**	**Packet Injection**	**Impersonation**	**Eavesdropping**	**Replay**	**Sybil**	**Jamming**	**Collusion**	**Black-hole**
2FLIP [16]	4	1	0	0	0	0	0	0	0	0
TESLA [17]	2.5	1	1	0	0	0	0	0	0	0
RAISE [18]	6.4	0	0	1	0	0	0	0	0	0
PACP [19]	7.2	0	0	1	1	1	0	0	0	0
SA-KMP [21]	3.4	1	0	1	0	1	0	0	0	0
GKMPAN [22]	3.9	0	0	0	1	1	0	0	0	0
PPAA [23]	5.1	0	0	0	0	0	1	0	0	0
Calandriello et al. [24]	4.8	1	0	0	0	0	0	1	0	0
PPGCV [25]	4.2	0	0	0	0	0	0	0	1	0
TACKs [26]	6.1	0	0	0	1	0	1	0	0	0
GSIS [27]	4.2	1	0	0	0	0	0	0	0	0
DABE [29]	3.5	0	0	0	0	0	0	0	1	0
ABACS [30]	5.7	0	0	0	0	0	0	0	1	0
Xia et al. [31]	4.9	0	0	0	0	1	0	0	1	0
Bouabdellah et al. [32]	4.1	0	0	0	0	0	0	0	0	1

Table 3: Realistic Example Dataset for Ekko-OPT

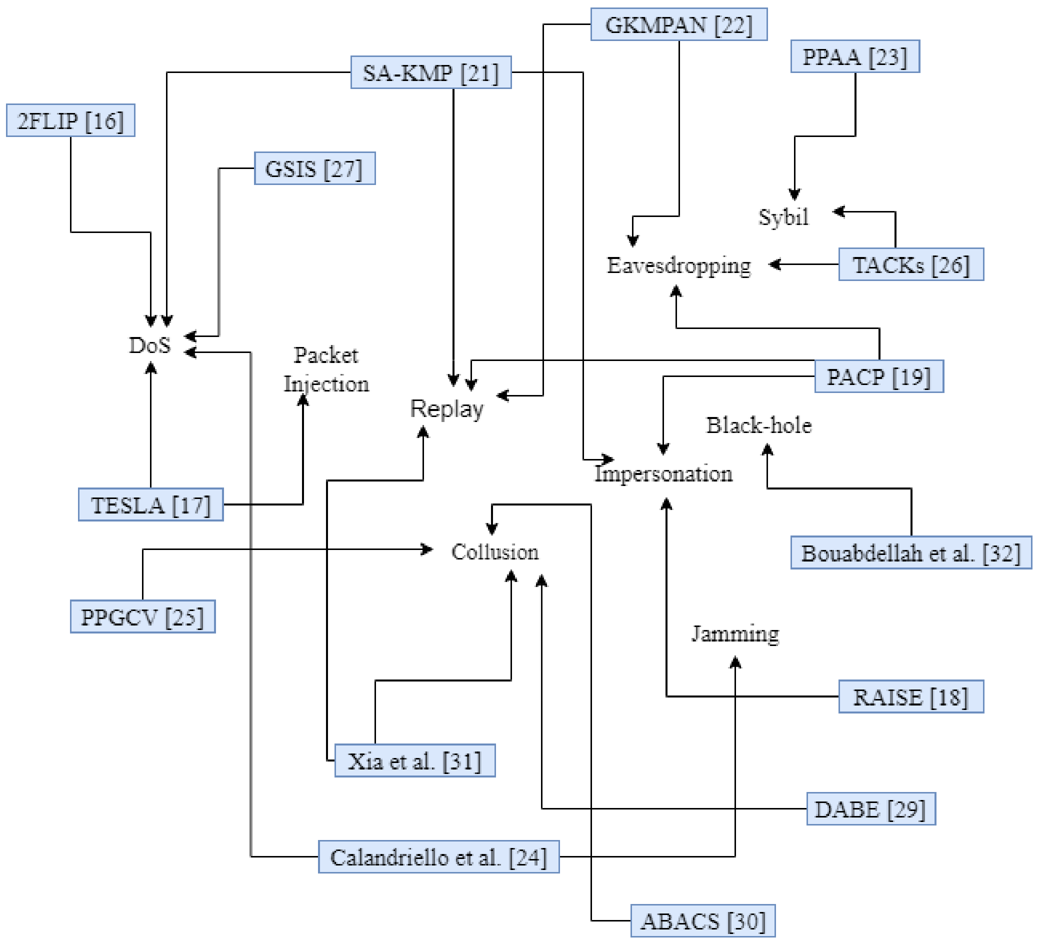


Figure 3: Mapping Realistic Example Defences to Attacks

The implementation of this algorithm began with the arrangement of a suitable dataset consisting of a number of defence, their associated cost and the set of attacks they cover. The defences listed in Table 3 is a subset of proposed defence solutions discussed previously. Due to the inability of accurately calculating defence cost for these proposed solutions, arbitrary cost was assigned to each defence to illustrate the functionality of Ekko-OPT. Figure 3 maps the defences of the realistic example dataset to their respective attacks to provide more visibility. A simple greedy algorithm was initially implemented as a baseline to find a local optimal collection of defences. The output of this greedy algorithm was then taken as an input for Ekko-OPT. There were two heuristics that were important in shaping Ekko-OPT:

- A heuristic to remove defences with unique attacks, where unique attack is described as an attack that is only covered by one defence
- A heuristic for during defence swapping where we ignore attacks that are already being covered by the current collection and only search for defences to cover the remaining ones

Algorithm 1: Ekko-OPT

```
Input: currentCollection, currentCost, table
Output: newCollection
nDef = list of defences;
Remove defences with unique attacks from nDef;
Queue = currentCollection;
while queue is not empty do
    $D_i$ = pop(queue);
    oldCost = cost(swapAttempt);
    $A_{atk}$ = attacks covered by $D_i$ ;
    for i in nDef do
        newCost = cost(i);
        if i covers anything in $A_{atk}$ then
            $S_d$.append(i);
            $A_{atk}$.remove(attacks covered by $S_d$);
            for j in nDef do
                if i covers anything in $A_{atk}$ then
                    $S_d$.append(j);
                    newCost = newCost + cost(j);
                    $A_{atk}$.remove(attacks covered by $S_d$);
                end
                if $A_{atk}$ is empty then
                    if newCost < oldCost then
                        currentCost = currentCost - oldCost + newCost;
                        currentCollection = currentCollection - $D_i$ + newCombination;
                        add newCombination to search queue;
                    end
                end
            end
        end
    end
end
```

Ekko-OPT begins by checking every defence in the local optimal collection for any unique attacks. Since these defences will always be included in the final collection, it was computationally more efficient to remove them at the start. The remaining defences were then added into a queue to perform the swapping sequence. The swapping sequence can be split into three stages: determining what needs to be covered, search for new better local optimal, update. A defence, D_i is first popped out of a queue and attacks they covered are then placed into an array,A_{atk}. The array is then checked against all attacks that are covered by the current defence collection excluding D_i and any overlap between them is removed from A_{atk}. The local optimal search then iterates through all remaining defences that are not in the current collection and adding defences that cover anything in A_{atk} into a swap array, S_d. The attacks covered are then removed from A_{atk}. When $Aatk$ is empty, it means that a new combination of defence that covers all attacks that D_i was responsible for has been found. The cost of this new combination is then compared to the cost of D_i. If the new cost is lower, the collection and cost is then updated with the new one and appended onto the queue as well. This process is repeated until all the defences in the queue is checked.

Ekko-OPT was tested using the dataset in Table 3 to demonstrate its capabilities. When the dataset was fed to the greedy algorithm, a local optimal collection was outputted with the following defences: ['TESLA', 'Calandriello et al.', 'Bouabdellah et al.', 'PACP', 'PPAA', 'PPGCV'] and a defence cost of 27.9. Whereas when Ekko-OPT was given this information, a more optimised path of ['TESLA', 'Calandriello et al.', 'Bouabdellah et al.', 'PACP', 'PPAA', 'DABE'] was outputted with a defence cost of 27.2. This proves that Ekko-OPT does indeed improve upon the initial greedy algorithm and provide a better solution. However, Ekko-OPT still doesn't produce a global optimal solution, but a much better local optimal solution in a reasonable time.

4 Analysis & Results

4.1 Quantification Metric Analysis

In order to justify the need for quantification metrics when classifying attacks, it is necessary to investigate the relationship between these metrics. One such relationship is between return on attack and attack cost. As illustrated by Figure 4, it can be identified that of ROA exponentially decreases when attack cost is increased. From a defence point of view, it is possible to assume that the attacks with high cost would less likely occur due to attackers gaining much less return on them. Therefore the urgency to implement a defence specifically for that attack would be much lower than one for attacks with lower cost. Since implementation of defence for all attacks would be difficult, focus on defending against attacks with higher ROA would be a more practical approach. However, only considering ROA is not enough to determine which attacks are the most important to defend against. An attack with low cost and risk could potentially have the same ROA as an attack with high cost and risk. Therefore it is imperative to take into account other metrics such as attack cost and attack risk when selecting defences, as opposed to just ROA.

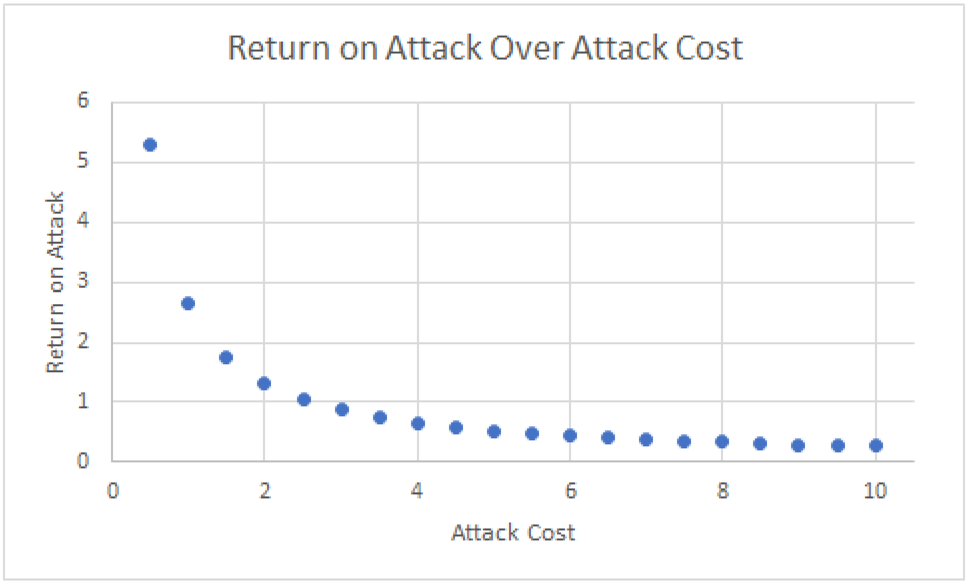


Figure 4: Relationship between ROI and Attack Cost

4.2 Optimisation Algorithm Analysis

A sensitivity analysis was conducted on Ekko-OPT to demonstrate the changes that can occur when small adjustments are made to defence cost. Initial experiments were conducted whereby the defence cost of a singular defence was altered, whilst all others defence cost remained the same. This was carried out by changing the cost of different defences such as: TESLA [17] which covered unique attacks, PPAA Based on Figure 5, it was immediately established that when the cost of a defence that has a unique attack changes, the associated total cost of implementation linearly changes with it. Since a defence that covers a unique attack will always be included in the final collection, this linear progression in defence cost was expected to occur. Observation dictates that when changing the cost of a defence that is included in the final collection, a linear progression will initially occur. However, an equilibrium will be reached when the defence cost exceeds that of implementing other possible combination of defences. This can be seen in the two other scenarios illustrated in Figure 5 given by defence PACP [19] and PPAA [23]. It can be assumed that every non-unique attack defence that exist in a collection will eventually be replaced if their defence cost is increased enough.

5 Conclusion

Cybersecurity in connected cars has become a highly researched topic in recent years. There is a need to improve the defensive technology deployed against security threats on the CAN bus. Quantification and classification can help bridge the gap between what can be done and what needs to be addressed to produce a working solution for these security threats. The current approach of determining risk and impact of secur-

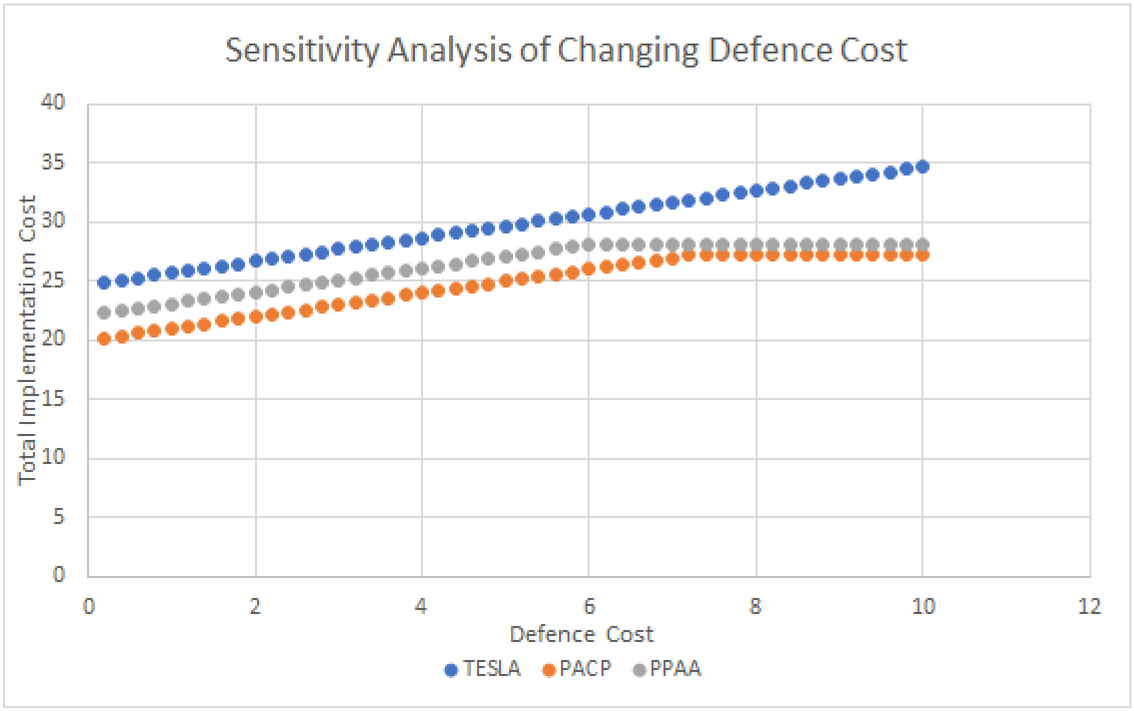


Figure 5: Sensitivity analysis showing changes in implementation cost as defence cost changes

ity threat is CVSS. However, CVSS can only capture the vulnerabilities and not the attacks themselves. The goal of this project was to develop a quantification and classification system to better illustrate and capture details in attacks. This information can then be used to better understand security threat and develop more effective means of combating them. In addition, an optimisation algorithm was developed to minimise the cost of implementing defence against a wide variety of attacks.

Through research into CVSS and other proposed classification system for general security scenarios, a set of quantification metrics were developed. These were attack cost, attack impact, probability of attack, attack risk and return on attack. These metrics were then used to shape a novel classification method for attacks called PACTree. The optimisation algorithm, Ekko-OPT was then built on the foundation of the swapping mechanism in 2-OPT and a variety of heuristics.

The findings in this research shows that it is important to consider the relationship between quantification metrics as opposed to just the raw numbers they present. The findings from the optimisation algorithm show that small changes in defence cost can cause drastic difference in the selection of optimal defence. Ekko-OPT was also able to demonstrate that it can improve upon the local optimal collection that a greedy algorithm finds.

5.1 Future Works

The current quantification metrics can be expanded upon to increase the accuracy of determining the risk a security threat can pose on the system, whilst improvements can be made to the optimisation algorithm by adding additional heuristics and attack cost. One such heuristic would involve a threshold whereby defences that do not have a good return on defence based on the amount of attack cost they are covering will be ignored or swapped out completely. Heuristics such as this can greatly improve the performance of Ekko-OPT, thus greatly improving the ability to minimise defence cost in the future.

References

[1] M. Han, B.-I. Kwak and H. K. Kim, 'Anomaly intrusion detection method for vehicular networks based on survival analysis', *Vehicular Communications*, vol. 14, Sep. 2018. DOI: 10.1016/j.vehcom.2018.09.004.

[2] M. Dibaei, X. Zheng, K. Jiang, S. Maric, R. Abbas, S. Liu, Y. Zhang, Y. Deng, S. Wen, J. Zhang, Y. Xiang and S. Yu, *An overview of attacks and defences on intelligent connected vehicles*, Jul. 2019.

[3] M. Wolf, A. Weimerskirch and T. Wollinger, 'State of the art: Embedding security in vehicles.', *EURASIP J. Emb. Sys.*, vol. 2007, Jan. 2007.

[4] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham and S. Savage, 'Experimental security analysis of a modern automobile', in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 447–462.

[5] G. A., *Jeep hackers are back to prove car hacking can get much worse*, 2016. [Online]. Available: https://www.wired.com/2016/08/jeep-hackers-return-high-speed-steering-acceleration-hacks/.

[6] C. Patsakis, K. Dellios and M. Bouroche, 'Towards a distributed secure in-vehicle communication architecture for modern vehicles', *Computers & Security*, vol. 40, Jan. 2013. DOI: 10.1016/j.cose.2013.11.003.

[7] M. Cheah, S. Shaikh, J. Bryans and P. Wooderson, 'Building an automotive security assurance case using systematic security evaluations', *Computers & Security*, vol. 77, Apr. 2018. DOI: 10.1016/j.cose.2018.04.008.

[8] C. Blommendaal, *Information security risks for car manufacturers based on the in-vehicle network*, 2015. [Online]. Available: http://essay.utwente.nl/68169/.

[9] J. Liu, S. Zhang, W. Sun and Y. Shi, 'In-vehicle network attacks and countermeasures: Challenges and future directions', *IEEE Network*, vol. 31, no. 5, pp. 50–58, 2017.

[10] P. Kleberger, T. Olovsson and E. Jonsson, 'Security aspects of the in-vehicle network in the connected car', in *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011, pp. 528–533.

[11] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham and S. Savage, 'Experimental security analysis of a modern automobile', in *2010 IEEE Symposium on Security and Privacy*, 2010, pp. 447–462.

[12] *Bosch. osi layers in automotive networks*, 2013. [Online]. Available: http://www.ieee802.org/1/files/public/docs2013/%20new-tsn-diarra-osi-layers-in-automotive-networks-0313-v01.pdf.

[13] I. Studnia, V. Nicomette, E. Alata, Y. Deswarte, M. Kaaniche and Y. Laarouchi, 'A survey of security threats and protection mechanisms in embedded automotive networks', Jun. 2013, pp. 1–12. DOI: 10.1109/DSNW.2013.6615528.

[14] K. B. Kelarestaghi, M. Foruhandeh, K. Heaslip and R. Gerdes, 'Vehicle security: Risk assessment in transportation', Apr. 2018.

[15] H. Guan, W. Chen, H. Li and J. Wang, 'Stride-based risk assessment for web application', *Applied Mechanics and Materials*, vol. 58-60, Jun. 2011. DOI: 10.4028/www.scientific.net/AMM.58-60.1323.

[16] F. Wang, Y. Xu, H. Zhang, Y. Zhang and L. Zhu, '2flip: A two-factor lightweight privacy-preserving authentication scheme for vanet', *IEEE Transactions on Vehicular Technology*, vol. 65, pp. 896–911, 2016.

[17] A. Perrig, R. Canetti, J. Tygar and D. Song, 'The tesla broadcast authentication protocol', *RSA CryptoBytes*, vol. 5, Nov. 2002. DOI: 10.1007/978-1-4615-0229-6_3.

[18] C. Zhang, X. Lin, R. Lu and P. .-.-. Ho, 'Raise: An efficient rsu-aided message authentication scheme in vehicular communication networks', in *2008 IEEE International Conference on Communications*, 2008, pp. 1451–1457.

[19] D. Huang, S. Misra, M. Verma and G. Xue, 'Pacp: An efficient pseudonymous authentication-based conditional privacy protocol for vanets', *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 736–746, 2011.

[20] M. Knezevic, V. Nikov and P. Rombouts, 'Low-latency ecdsa signature verification—a road toward safer traffic', *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 24, pp. 3257–3267, 2016.

[21] H. Tan, M. Ma, H. Labiod, A. Boudguiga, J. Zhang and P. H. J. Chong, 'A secure and authenticated key management protocol (sa-kmp) for vehicular networks', *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9570–9584, 2016.

[22] S. Zhu, S. Setia, S. Xu and S. Jajodia, 'Gkmpan: An efficient group rekeying scheme for secure multicast in ad-hoc networks', in *The First Annual International Conference on Mobile and Ubiquitous Systems: Networking and Services, 2004. MOBIQUITOUS 2004.*, 2004, pp. 42–51.

[23] P. P. Tsang and S. W. Smith, 'Ppaa: Peer-to-peer anonymous authentication', in *Applied Cryptography and Network Security*, S. M. Bellovin, R. Gennaro, A. Keromytis and M. Yung, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 55–74, ISBN: 978-3-540-68914-0.

[24] G. Calandriello, P. Papadimitratos, J.-P. Hubaux and A. Lioy, ‘Efficient and robust pseudonymous authentication in vanet’, Jan. 2007, pp. 19–28. DOI: 10.1145/1287748.1287752.

[25] A. Wasef and X. Shen, ‘Ppgcv: Privacy preserving group communications protocol for vehicular ad hoc networks’, in *2008 IEEE International Conference on Communications*, 2008, pp. 1458–1463.

[26] A. Studer, E. Shi, F. Bai and A. Perrig, ‘Tacking together efficient authentication, revocation, and privacy in vanets’, in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 1–9.

[27] X. Lin, X. Sun, P. Ho and X. Shen, ‘Gsis: A secure and privacy-preserving protocol for vehicular communications’, *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3442–3456, 2007.

[28] L. Fischer, A. Aijaz, C. Eckert and D. Vogt, ‘Secure revocable anonymous authenticated inter-vehicle communication (sraac)’, Dec. 2.

[29] N. Chen, M. Gerla, D. Huang and X. Hong, ‘Secure, selective group broadcast in vehicular networks using dynamic attribute based encryption’, in *2010 The 9th IFIP Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)*, 2010, pp. 1–8.

[30] L. Yeh, Y. Chen and J. Huang, ‘Abacs: An attribute-based access control system for emergency services over vehicular ad hoc networks’, *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 3, pp. 630–643, 2011.

[31] Y. Xia, W. Chen, X. Liu, L. Zhang, X. Li and Y. Xiang, ‘Adaptive multimedia data forwarding for privacy preservation in vehicular ad-hoc networks’, *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2629–2641, 2017.

[32] M. Bouabdellah, F. E. Bouanani and H. Ben-azza, ‘A secure cooperative transmission model in vanet using attribute based encryption’, in *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*, 2016, pp. 1–6.

[33] N. Bißmeyer, C. Stresing and K. M. Bayarou, ‘Intrusion detection in vanets through verification of vehicle movement data’, in *2010 IEEE Vehicular Networking Conference*, 2010, pp. 166–173.

[34] A. Tomandl, K. Fuchs and H. Federrath, ‘Rest-net: A dynamic rule-based ids for vanets’, in *2014 7th IFIP Wireless and Mobile Networking Conference (WMNC)*, 2014, pp. 1–8.

[35] K.-T. Cho and K. Shin, ‘Fingerprinting electronic control units for vehicle intrusion detection’, in *USENIX Security Symposium*, 2016.

[36] D. Martynov, J. Roman, S. Vaidya and H. Fu, ‘Design and implementation of an intrusion detection system for wireless sensor networks’, in *2007 IEEE International Conference on Electro/Information Technology*, 2007, pp. 507–512.

[37] H. Sedjelmaci and S. M. Senouci, ‘A new intrusion detection framework for vehicular networks’, in *2014 IEEE International Conference on Communications (ICC)*, 2014, pp. 538–543.

[38] 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’, in *2016 International Conference on Information Networking (ICOIN)*, 2016, pp. 63–68.

[39] K. Zaidi, M. B. Milojevic, V. Rakocevic, A. Nallanathan and M. Rajarajan, ‘Host-based intrusion detection for vanets: A statistical approach to rogue node detection’, *IEEE Transactions on Vehicular Technology*, vol. 65, no. 8, pp. 6703–6714, 2016.

[40] H. Lee, S. H. Jeong and H. Kim, ‘Otids: A novel intrusion detection system for in-vehicle network by using remote frame’, *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, pp. 57–5709, 2017.

[41] S. Hamdan, A. Hudaib and A. Awajan, ‘Detecting sybil attacks in vehicular ad hoc networks’, *CoRR*, vol. abs/1905.03507, 2019. arXiv: 1905.03507. [Online]. Available: http://arxiv.org/abs/1905.03507.

[42] H. Sedjelmaci and S.-M. Senouci, ‘An accurate and efficient collaborative intrusion detection framework to secure vehicular networks’, *Computers Electrical Engineering*, vol. 43, Apr. 2015. DOI: 10.1016/j.compeleceng.2015.02.018.

[43] M. Markovitz and A. Wool, ‘Field classification, modeling and anomaly detection in unknown can bus networks’, *Veh. Commun.*, vol. 9, pp. 43–52, 2017.

[44] T. Zhang and Q. Zhu, ‘Distributed privacy-preserving collaborative intrusion detection systems for vanets’, *IEEE Transactions on Signal and Information Processing over Networks*, vol. 4, no. 1, pp. 148–161, 2018.

[45] C. Tice, T. Roeder, P. Collingbourne, S. Checkoway, Ú. Erlingsson, L. Lozano and G. Pike, ‘Enforcing forward-edge control-flow integrity in gcc amp; llvm’, in *Proceedings of the 23rd USENIX Conference on Security Symposium*, ser. SEC’14, San Diego, CA: USENIX Association, 2014, pp. 941–955, ISBN: 9781931971157.

[46] J. Dahse and T. Holz, ‘Static detection of second-order vulnerabilities in web applications’, in *USENIX Security Symposium*, 2014.

[47] R. Ding, C. Qian, C. Song, B. Harris, T. Kim and W. Lee, ‘Efficient protection of path-sensitive control security’, in *26th USENIX Security Symposium (USENIX Security 17)*, Vancouver, BC: USENIX Association, Aug. 2017, pp. 131–148, ISBN: 978-1-931971-40-9. [Online]. Available: https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/ding.

[48] M. Castro, M. Costa and T. Harris, ‘Securing software by enforcing data-flow integrity’, in *Proceedings of the 7th Symposium on Operating Systems Design and Implementation*, ser. OSDI ’06, Seattle, Washington: USENIX Association, 2006, pp. 147–160, ISBN: 1931971471.

[49] N. Ayewah, W. Pugh, D. Hovemeyer, J. D. Morgenthaler and J. Penix, ‘Using static analysis to find bugs’, *IEEE Software*, vol. 25, no. 5, pp. 22–29, 2008.

[50] P. Godefroid, M. Y. Levin and D. A. Molnar, ‘Automated whitebox fuzz testing’, in *NDSS*, 2008.

[51] J. Newsome and D. Song, ‘Dynamic taint analysis for automatic detection, analysis, and signature-generation of exploits on commodity software.’, Feb. 2005.

[52] J. Clause, W. Li and A. Orso, ‘Dytan: A generic dynamic taint analysis framework’, Jan. 2007, pp. 196–206. DOI: 10.1145/1273463.1273490.

[53] C. Le Goues, T. Nguyen, S. Forrest and W. Weimer, ‘Genprog: A generic method for automatic software repair’, *IEEE Transactions on Software Engineering*, vol. 38, no. 1, pp. 54–72, 2012.

[54] Y. Shin, A. Meneely, L. Williams and J. A. Osborne, ‘Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities’, *IEEE Transactions on Software Engineering*, vol. 37, no. 6, pp. 772–787, 2011.

[55] H. Perl, S. Dechand, M. Smith, D. Arp, F. Yamaguchi, K. Rieck, S. Fahl and Y. Acar, ‘Vccfinder: Finding potential vulnerabilities in open-source projects to assist code audits’, in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, New York, NY, USA: Association for Computing Machinery, 2015, pp. 426–437. [Online]. Available: https://doi.org/10.1145/2810103.2813604.

[56] Y. Zhou and A. Sharma, ‘Automated identification of security issues from commit messages and bug reports’, in *Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering*, ser. ESEC/FSE 2017, Paderborn, Germany: Association for Computing Machinery, 2017, pp. 914–919, ISBN: 9781450351058. [Online]. Available: https://doi.org/10.1145/3106237.3117771.

[57] L. K. Shar, L. C. Briand and H. B. K. Tan, ‘Web application vulnerability prediction using hybrid program analysis and machine learning’, *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 6, pp. 688–707, 2015.

[58] G. Grieco, G. L. Grinblat, L. Uzal, S. Rawat, J. Feist and L. Mounier, ‘Toward large-scale vulnerability discovery using machine learning’, in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, ser. CODASPY ’16, New Orleans, Louisiana, USA: Association for Computing Machinery, 2016, pp. 85–96, ISBN: 9781450339353. DOI: 10.1145/2857705.2857720. [Online]. Available: https://doi.org/10.1145/2857705.2857720.

[59] Y. Fan, Y. Ye and L. Chen, ‘Malicious sequential pattern mining for automatic malware detection’, *Expert Systems with Applications*, vol. 52, Jan. 2016.

[60] S. Huda, J. Abawajy, M. Alazab, M. Abdollalihian, R. Islam and J. Yearwood, ‘Hybrids of support vector machine wrapper and filter based framework for malware detection’, *Future Generation Computer Systems: the international journal of grid computing: theory, methods and applications*, pp. 376–390, Feb. 2016, ISSN: 0167-739X.

[61] S. Huda, M. S. Miah, M. Hassan, M. R. Islam, J. Yearwood, M. Alrubaian and A. Almogren, ‘Defending unknown attacks on cyber-physical systems by semi-supervised approach and available unlabeled data’, *Information Sciences*, vol. 379, Sep. 2016.

[62] Q. K. Ali Mirza, I. Awan and M. Younas, ‘Cloudintell: An intelligent malware detection system’, *Future Generation Computer Systems*, vol. 86, Jul. 2017. DOI: 10.1016/j.future.2017.07.016.

[63] M. Han, B.-I. Kwak and H. K. Kim, ‘Cbr-based decision support methodology for cybercrime investigation: Focused on the data-driven website defacement analysis’, *Security and Communication Networks*, vol. 2019, pp. 1–21, Dec. 2019. DOI: 10.1155/2019/1901548.

[64] M. Pratt, *What is an intrusion detection system? how an ids spots threats.* 2018. [Online]. Available: https://www.csoonline.com/article/3255632/what-is-an-intrusion-detection-system-how-an-ids-spots-threats.html.

[65] J. D. Florian Sommer and R. Kriesten, ‘Survey and classification of automotivesecurity attacks’, 2019. [Online]. Available: https://www.researchgate.net/publication/332530687_Survey_and_Classification_of_Automotive_Security_Attacks.