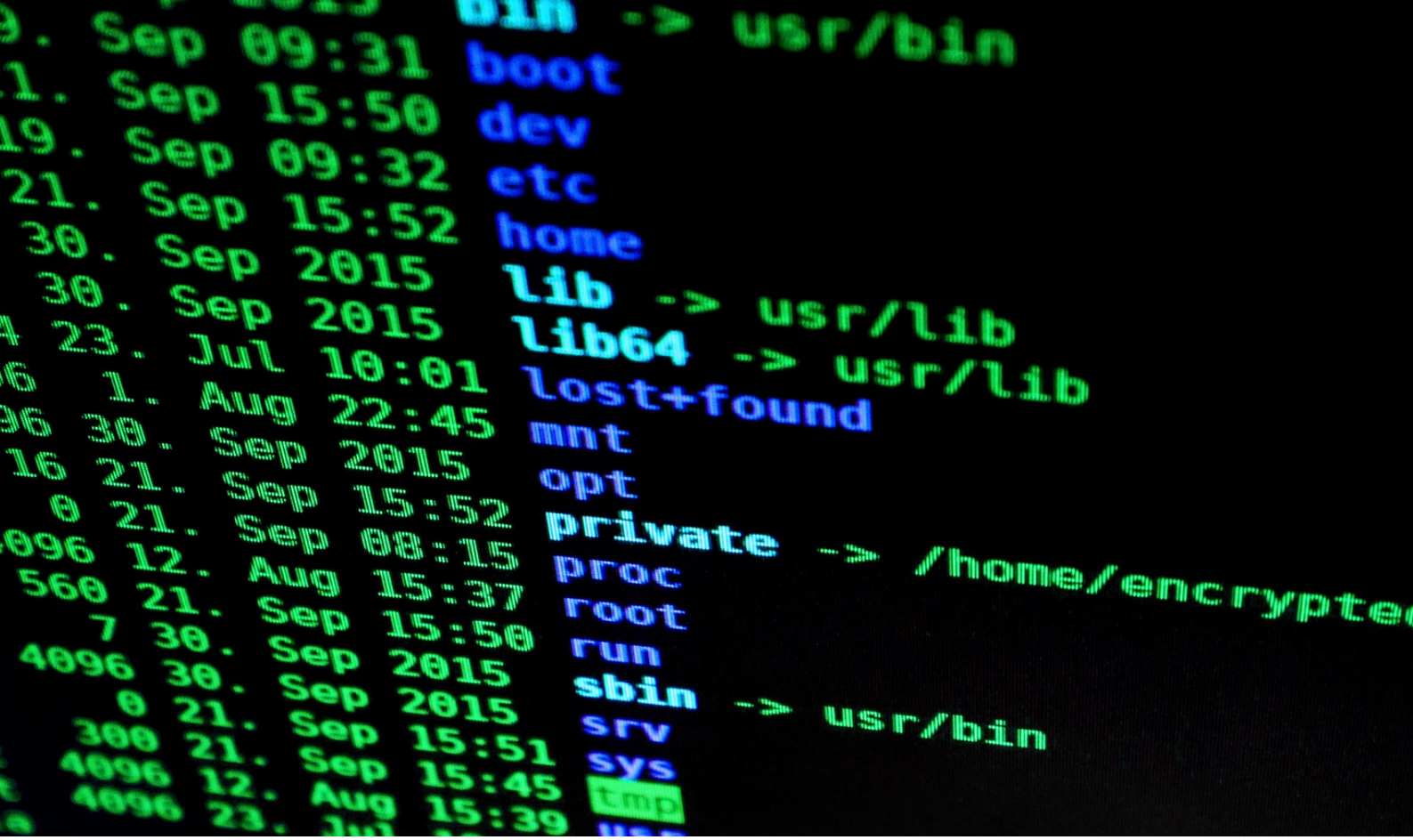


VEHI QILLA

Hacking Connected Autonomous Vehicles and Electric Vehicles



INTRODUCTION

Vehiqilla Inc. was founded to meet the cybersecurity challenges of today's & tomorrow's Connected & Autonomous Vehicles (CAVs). This transformation in Mobility is changing the Cyber Threat Landscape, and there is a need to reassess the Risk Impact of Cyber to the Connected & Autonomous Vehicles (CAV). Vehiqilla Inc. aims to address all areas of this changing threat landscape, including In-Vehicle Security, V2X Security & Cybersecurity of the Supply Chain.

ABSTRACT

This white paper aims to provide the reader with an understanding of the importance of CAV security and how CAVs are tested. This paper outlines a typical testing setup to perform penetration tests on vehicles as well as how to prepare before starting a pentest. The architecture of CAVs and EVs is then broken down to provide an understanding of key components, followed by relevant threat modeling. Steps for reconnaissance and vulnerability analysis are then highlighted and lastly, examples of exploitations are provided.

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

TABLE OF CONTENTS

EXECUTIVE SUMMARY

This white paper is intended to highlight the importance of automotive security, specifically for CAVs and EVs. As technology continues to advance, vehicles are more connected than ever which has introduced a plethora of cybersecurity concerns. Manufacturers are failing to prioritize security when developing new vehicles, leaving vehicles and their infrastructure vulnerable to high-risk threats.

An overview of how to set up a testing environment is first provided, highlighting various components within a typical testing environment. Recommendations of tools are given including both software and hardware.

A breakdown of planning and scoping for CAV and EV hacking is then outlined. Emphasis is put on understanding the target fleet organization before beginning any testing. Once this has been done, it is crucial to then define the scope of the test to determine what assets are allowed to be hacked as well as what third-party applications may be affected during testing.

Before testing begins, it is important to have a foundational understanding of the target CAV or EV architecture. A high-level breakdown is provided alongside explanations of key components such as the CAN bus, J1939, and other IVNs. V2X technology and its various applications are also defined.

Threat modeling is a key component that needs to be included in the CAV and EV development process. Security testing is often a result of threat modeling. This paper highlights the ISO21434 standard, TARA, and the threat modeling framework STRIDE in relation to vehicle specific threats.

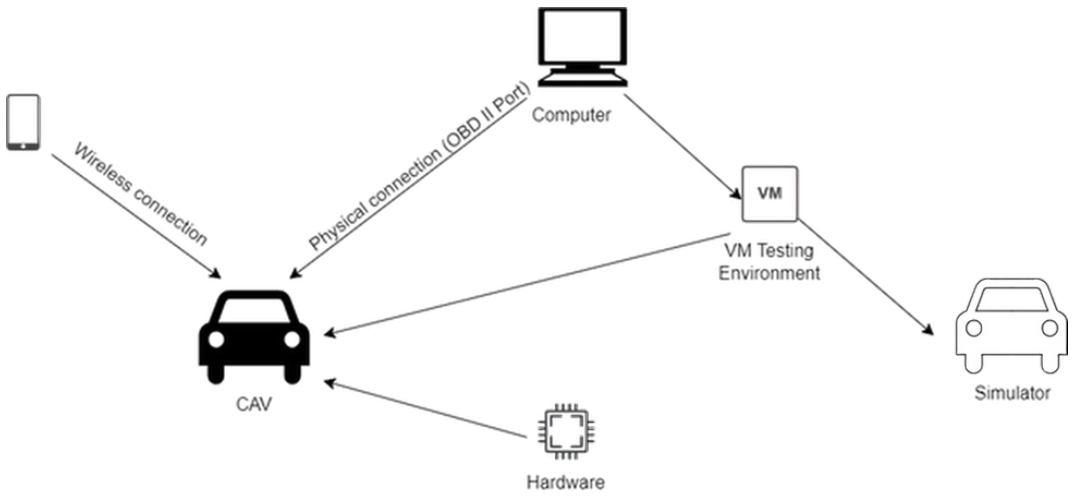
Reconnaissance and vulnerability analysis is the last key component of the testing process before exploitation takes place. At this point both passive and active reconnaissance are required to gain information on the CAV or EV through indirect or direct engagement, respectively. General IVN and V2X insecurities are highlighted that could be abused during exploitation.

CAVs have a plethora of attack vectors, however this paper outlines just several possible exploitation techniques via the CAN Bus, Bluetooth, Wi-Fi, and cellular network. An overview of post-exploitation activities is then provided, highlighting the extent to which an attacker can leverage their foothold.

CAV/EV HACKING SETUP

CREATING A TESTING ENVIRONMENT

When performing a penetration test it is crucial to set up a testing environment. Typically, this is done by creating a VM using an ISO image of your desired operating system. Linux is most often used and is recommended for CAV hacking as it supports CAN drivers. Two of the most popular security-focused Linux distributions are Kali Linux and Parrot OS, both of which are designed for security testing and come with a plethora of tools installed. A popular option for virtualization that can be used to run these operating systems is VirtualBox, a free open-source type-2 hypervisor developed by Oracle.



TOOL RECOMMENDATIONS

Depending on the intended attack vector, different tools are required. This section highlights some of the best tools used in industry for pentesting that can be applied to CAV hacking.

DATA CAPTURE & EXPLOITATION SOFTWARE

METASPLOIT

One of the most popular pentesting frameworks used by security professionals, Metasploit is a crucial tool for exploitation. This framework's database contains a plethora of common exploits and provides a simple interface for users to make a selection, configure settings, and begin exploiting targets.
Current version: 6.2.34

WIRESHARK

Wireshark is the most popular network protocol analyzer used today. It allows for live capturing of packets for analysis or analyzing PCAP files. Various network types can be captured such as Bluetooth and Ethernet, making it possible to capture OBD II data.
Current version: 4.0.3

SOCKETCAND

This daemon can be used to access CAN interfaces on a machine using the network interface.
Current version: v0.6.1

CARINGCARIBOU

This module-based tool is a catch-all for automotive testing. Its features include diagnostics such as service discovery, sending and dumping CAN packets, fuzzing and more.
Current version: GitHub Repository

SOCKETCAN & CAN-UTILS

Linux has its own CAN subsystem implementation known as the SocketCAN, which allows for a collection of tools known as can-utils which can interact with the CAN protocol. There are several categories of tools within this utility such as

- tools for creating, capturing, and replaying CAN traffic;
- CAN access via IP sockets;
- CAN bus measurement and testing;
- ISO-TP tools;
- J1939/ISOBus tools; and
- Log file converters;

Current version: v2021.08.0

SIMULATORS

ICSIM

This instrument cluster simulator is a popular tool used to replicate CAN traffic, making it a great tool for POCs, practice, or testing without a vehicle.
GitHub Repository

UDSIM

From the same creator of ICSim, this unified diagnostic services simulator is capable of emulating various modules within a vehicle and can respond to UDS requests. It was designed to run alongside ICSim.
GitHub Repository

BLUETOOTH HACKING

BLUEZ

BlueZ is the official Linux Bluetooth protocol stack and is an open-source project. All of its associated utilities can be found in Kali Linux which includes tools for monitoring, configuring devices, and more.
Current version: 5.66

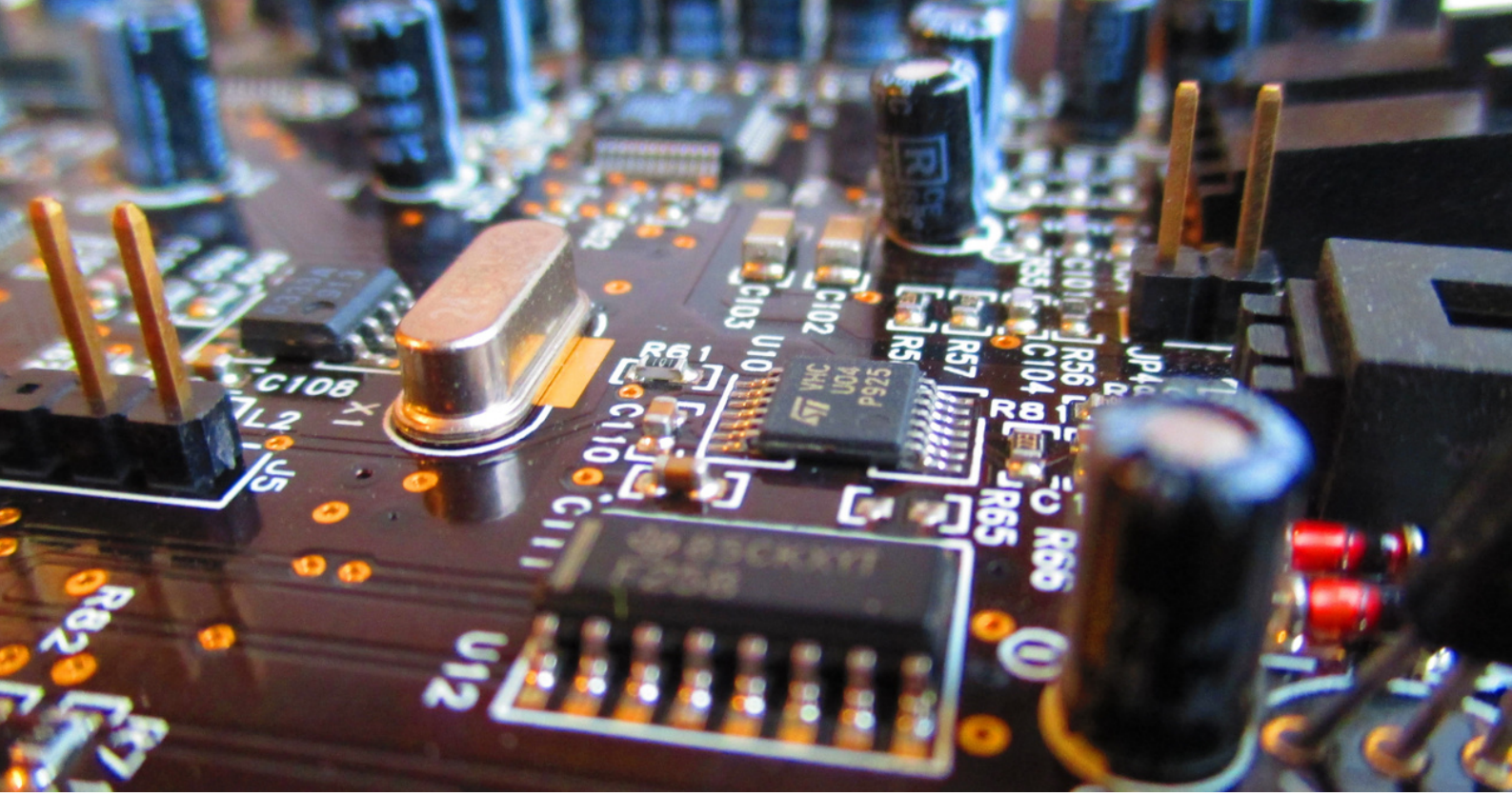
WIFI HACKING

KISMET

Kismet is an open-source wireless sniffer that comes preinstalled in Kali Linux. It is most commonly used for radio frequency monitoring which allows for monitoring and identifying networks without needing an associated access point. Other uses include wardriving and wireless intrusion detection. Kismet is also compatible with Bluetooth interfaces.
Current version: 2022.08.R1

AIRCRACK-NG

Aircrack-ng is a popular tool used for cracking 802.11 a/b/g WEP/WPA that can be used to recover WEP keys . It can also be used to attack WPA1/2 networks using advanced methods or brute force.
Current version: 1.7



HARDWARE

OBD II ADAPTER

To hack the OBD II port of a vehicle a physical connection is required. A basic OBD II Adapter is relatively inexpensive making it an easy piece of equipment to obtain.

RASPBERRY PI

A custom hardware approach would be the Raspberry Pi, a compact and inexpensive computer that retails for as little as $60 CAD. This price goes up depending on how much RAM is needed.
Latest model: Raspberry Pi 4

AUTOPI TMU

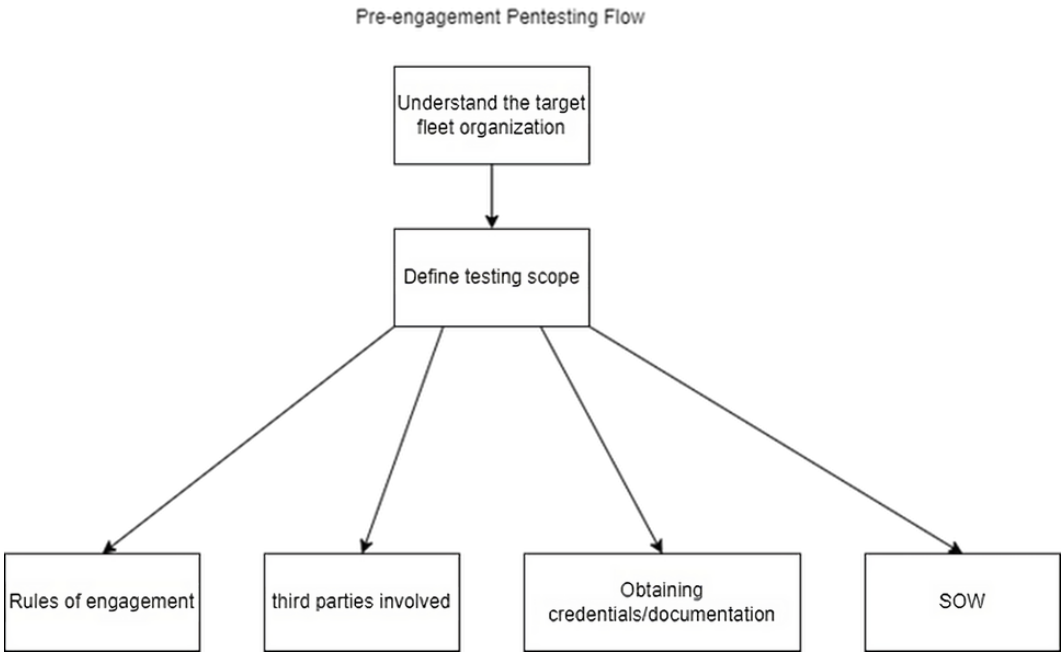
The AutoPi TMU is a telematics device powered by a Raspberry Pi. This device allows users to take advantage of a vehicle's TCU, the embedded system on board. Though not intended as a security tool, its extensive features such as an OBD chipset would be a beneficial addition to any testing arsenal.
Latest model: AutoPi TMU CM4

PLANNING & SCOPING FOR CAV/EV HACKING

UNDERSTANDING THE TARGET FLEET ORGANIZATION

When preparing for a penetration test of a CAV, it is important to start off with understanding the organization behind the vehicle. Questions to consider are:

- Where is the organization located?
- Do they have a website?
- How many trucks do they have if fleets are involved?
- What are the possible communications between the vehicles and external elements?
- How many employees are there? Will social engineering be possible?
- What third parties are involved in enabling the software of the vehicle?
- What geographical area is covered by the fleets?



DEFINING SCOPE

Another important step in the pentesting process is to understand the scope of the test. Outlined below are several crucial areas to consider when planning a CAV test.

Black Box Test: No direct access to the vehicle and no documentation is provided, testing is done remotely.

White Box Test: Documentation such as source code is provided, and direct access to the vehicle is given. Testing it is done with knowledge of how exactly the vehicle's systems operate.

Grey Box Test: The middle ground of having some detail of the inner workings of the vehicle, may have limited access to the vehicle.

3rd-Party Applications: 3rd-parties may be used for different software features of the vehicle. Certain attacks may affect endpoints owned by these 3rd parties.

UNDERSTANDING THE CAV/EV TARGET ARCHITECTURE

To begin a CAV penetration test, it is important to first understand the vehicle's architecture in order to determine the scope of your engagement. At a high level, the vehicle's architecture includes the following:

- Smartphone & 3rd party apps
- Vehicle Access System ECU
- Airbag ECU
- OBD II
- Bluetooth
- V2X technologies
- Remote key
- Passive Key-less Entry
- TPMS
- ADAS System ECU
- Lighting System ECU
- Engine and Transmission ECU
- Steering and Braking ECU

For all these various components, especially the ECUs, to communicate with one another there are several in-vehicle networks in place. Most of these networks use what is known as the CAN bus protocol.

CAN BUS

CAN, or Controller Area Network, is the bus protocol used within vehicles that provides the framework for components to communicate. CAN is a broadcast protocol, meaning all nodes on the bus can pick up traffic as messages are being sent. Though this protocol inherently has a single point of failure, it is the most used IVN.

J1939

This SAE standard is an extended version of CAN used in trucks and other heavy-duty vehicles. J1939 is a higher layer protocol, meaning it uses higher layers of the ISO model such as the network and transport layer which traditional CAN does not.

OTHER IVNS

Beyond CAN there are several other IVNs used depending on the use case.

LIN

The Local Interconnect Network (LIN) is another protocol that uses the bus topology and was created as a replacement for CAN in low-bandwidth applications. LIN is typically used for motorized operation in the vehicle such as chair adjustments, mirror adjustments, and controlling the windows.

FLEXRAY

Flexray provides high bandwidth and high reliability communication and is used in instances where CAN is not fast enough such as x-by-wire systems. However, it is a rather expensive protocol to implement.

CAN-FD

CAN-FD is essentially the CAN protocol but with a flexible data rate. This is achieved by increasing the rate of transfer after message ID arbitration is complete.

MOST

The media-oriented systems transport is a serial network that follows the ring topology. It is used for in-vehicle media systems and can have up to 64 devices per ring configuration.

V2X

V2X is the technology that allows CAVs to communicate with other entities. There are several different types of V2X applications, several of which are highlighted below.

V2V

Vehicle to vehicle communication is what allows autonomous vehicles to communicate with each other. This communication occurs in real time and is crucial in helping to reduce accidents and improve road safety.

V2N

Vehicle to network communication allows CAVs to use cellular networks and the dedicated short-range communications standard to receive and send data. V2N makes it possible for other V2X communications to occur, such as V2V, V2I, and V2P.

V2I

Vehicle to infrastructure is the communication between CAVs and road infrastructure such as traffic lights and road signs.

V2C

Vehicle to cloud communication uses V2N to access the cloud. V2C makes OTA updates possible, allows for remote vehicle diagnostics, and allows for communication with other cloud connected IoT devices.

V2P

Vehicle to pedestrian communication allows CAVs to detect and promptly react to potential collisions with pedestrians.

V2D

Vehicle to device communication allows CAVs to communicate with smart devices such as smartphones. A popular application of this type of communication is connecting smart devices to the infotainment system, such as Android Auto from Google.

V2G

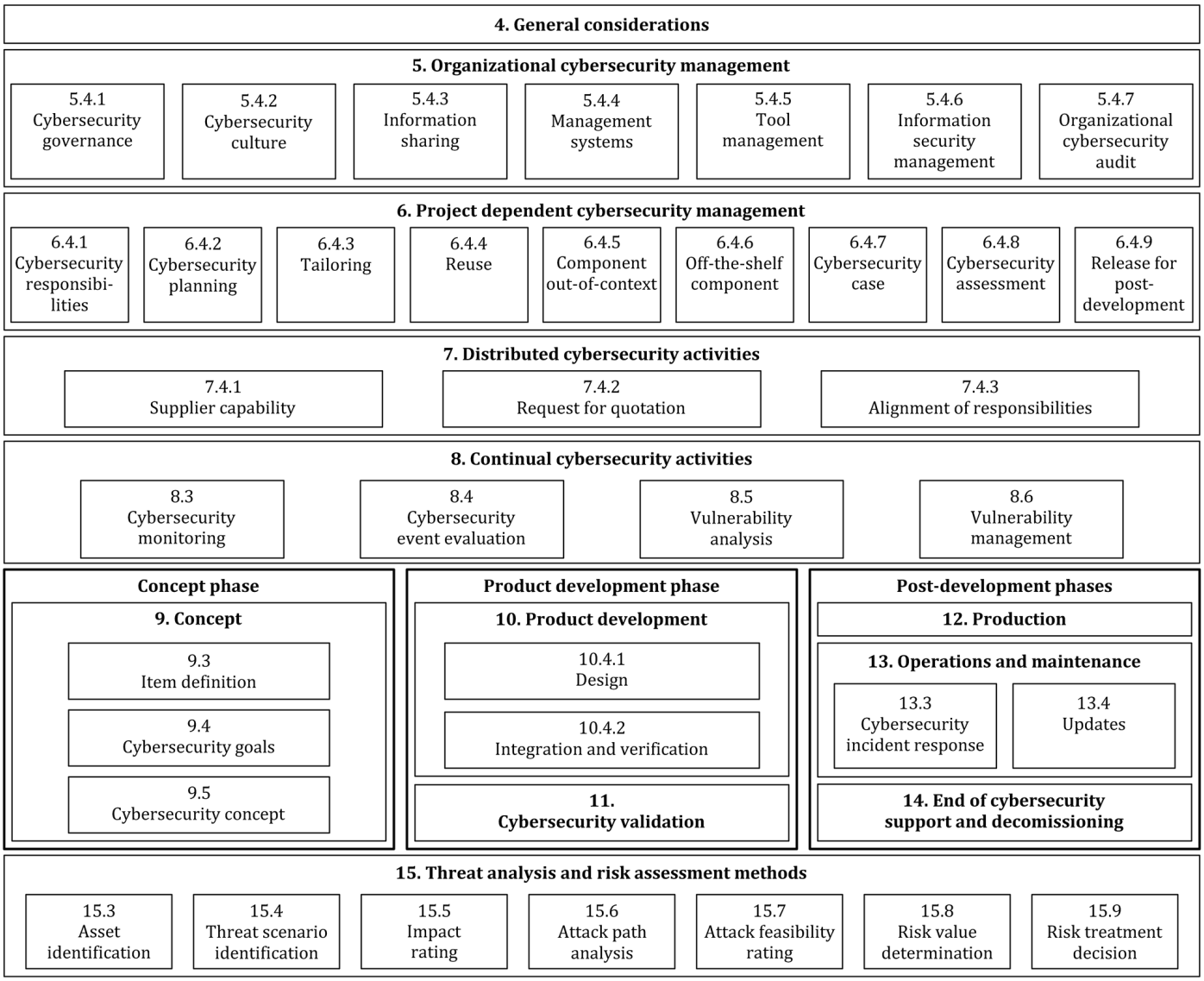
Vehicle to grid communication is what gives CAVs the ability to return excess energy to a power grid.

THREAT MODELING FOR CAVS/EVS

Threat modeling is an important process in the creation and implementation of new technologies. A threat model provides a framework for identifying potential threats and allowing for a discussion on mitigation strategies and validation of applied solutions.

ISO21434 & TARA

ISO21434 is the Road Vehicles Cybersecurity Engineering standard that integrates risk management into the development of vehicles. The standard specifies cybersecurity risk management for the conception, development, production, operation, maintenance and decommission of CAVs. The image below outlines these various stages and includes both vulnerability analysis and management which is why pentesting of CAVs is required. Both of these points fall under section 8, continual cybersecurity activities, which emphasizes the requirement of identifying weaknesses caused by cybersecurity events. TARA falls under section 15, that includes steps such as attack path analysis which results in pentesting to occur.



STRIDE

STRIDE is a threat modeling framework used to identify threats in the following six categories:

- Spoofing - Impersonation of users or other vehicles
- Tampering - Data modification
- Repudiation - The denial of an action without the ability to prove otherwise
- Information Disclosure - Attackers gain access to information they should not have permission to view
- Denial of Service - Denying service of valid users
- Elevation of Privilege - An unprivileged user is able to elevate their privileges in order to access and compromise the victim's system

Penetration testing happens because of threat modeling to verify potential vulnerabilities in the system.

Type of Threat	Actual Threat	Vulnerability
Spoofing	The messages and data the vehicle receives are spoofed	Spoofing of messages via impersonation or Sybil attack
Spoofing	Mobile application used pretends to be a trusted publisher	A fake application steals information when paired with a vehicle
Spoofing	Key fob spoofing	Threat actor captures RFID signal to gain access to the vehicle
Tampering	The data and/or code of the vehicle is manipulated	Vehicle's driving data can be falsified such as the speed, mileage, fuel, etc.
Tampering	The data and/or code of the vehicle is erased	Unauthorized deletion/manipulation of system event logs
Tampering	Manipulation of vehicle parameters	Messages sent to the ECU regarding speed, breaks, etc. are manipulated to alter the vehicle behavior
Repudiation	Introduction of malware	User's may install unofficial third-party applications on the vehicle system or attempt to jailbreak the vehicle system
Information Disclosure	Extraction of vehicle data or code	Unauthorized access to the owner's private data such as PII, location information, and vehicle ID
Denial of Service	Disruption of systems or operations	DoS through attacks such as flooding the CAN bus
Elevation of Privilege	An unauthorized user gains admin privileges in the vehicle system	Maintenance personnel escalating their access level to reach higher level ECUs

RECONNAISSANCE AND VULNERABILITY ANALYSIS

Before exploitation can take place, reconnaissance and vulnerability analysis is required to gain insight into the vehicle being tested and its potential weaknesses. OSINT is a common method used to gather this information.

PASSIVE RECONNAISSANCE OF CAVS/EVS

Passive reconnaissance is performed to gain information on the CAV and its associated company without directly engaging with the vehicle. This can be done by,

- Analyzing the company's website;
- Analyzing company socials such as LinkedIn and Twitter; and,
- Leveraging google dorking to search for documentation, diagrams, etc;
- Determine the physical location of the vehicle by observation

ACTIVE RECONNAISSANCE OF CAVS/EVS

Active reconnaissance involves engaging with the CAV to gather further information for the pentest. Common methods for this step include:

- Scanning the Wi-Fi/Bluetooth/cellular network of the CAV and capturing traffic
- Capturing and analyzing V2X traffic
- Analyzing accompanying mobile applications using SAST
- Social engineering of employees/drivers

IN-VEHICLE NETWORK (IVN) INSECURITIES

The CAN protocol within vehicles is inherently insecure due to its lack of encryption and authentication. The protocol was designed to be a broadcast network, giving threat actors a way to capture unencrypted data. Furthermore, it is susceptible to data injection and its priority-based messaging can be abused to perform DOS attacks.

V2X INSECURITIES

The V2X network's reliance on wireless communication makes it vulnerable to the CIA triad being compromised. The network also contains various entry points that threat actors could abuse, such as cloud platforms, Bluetooth, Wi-Fi, and LTE.

EXPLOITATION OF CAVS/EVS

EXPLOITING THE VEHICLE'S KEY FOB

Cars used to be unlocked using a single code shared with the key fob, which was extremely easy to exploit using replay attacks. However, manufacturers now use rolling codes. The key fob and vehicle both share the same algorithm to generate one time use codes. However, security researcher Samy Kamkar developed an exploit for this new model known as a roll jam attack. This attack is carried out by,

1. Jamming a portion of the vehicle's receiving frequencies window and creating a capturing frequency window in the same range as the key fob.
2. When the victim tries to unlock the vehicle using the key fob the signal will be jammed preventing the car from using the code. The code is however caught using the capturing frequency window.
3. After 2 codes have been captured, the first code is replayed so the victim assumes the key fob has no issues. The second code is still valid and can be used by the attacker to unlock the vehicle.

EXPLOITING THE CAN BUS VIA OBD II

To capture CAN bus traffic on a computer, a direct connection to OBD II port using hardware such as the USB2CAN is required. Cansiffer from can-utils can then be used to sniff packets and analyze them. A replay attack can be done by capturing packets while actions are being done in the vehicle. The packets are then dumped using candump into a log file and replayed using canplayer. A simulation of this exploit using ICSim is shown below.

```
┌──(kali㉿kali)-[~/ICSim]
└─$ ifconfig vcan0
vcan0: flags=193<UP,RUNNING,NOARP>  mtu 72
        unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00  txqueuelen 1000  (UNSPEC)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

Ifconfig is used to confirm the CAN interface is available.

```
┌──(kali㉿kali)-[~/ICSim]
└─$ ./icsim vcan0
Using CAN interface vcan0
```

ICSim started using the CAN interface.

```
┌──(kali㉿kali)-[~/ICSim]
└─$ ./controls vcan0
Warning: No joysticks connected
```

Vehicle controls are started for the CAN interface.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ candump -l vcan0
Disabled standard output while logging.
Enabling Logfile 'candump-2023-04-05_220122.log'
```

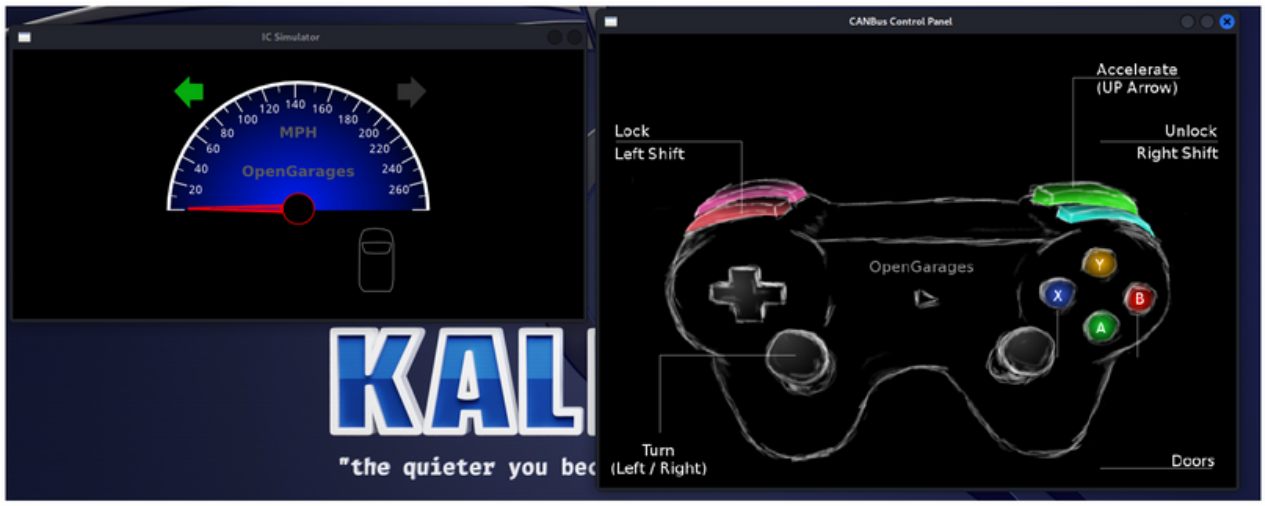
The -l flag can be used to dump packets into a log file for later analysis.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ candump vcan0
  vcan0  1D0   [8]  00 00 00 00 00 00 00 0A
  vcan0  166   [4]  D0 32 00 09
  vcan0  158   [8]  00 00 00 00 00 00 00 0A
  vcan0  161   [8]  00 00 05 50 01 08 00 0D
  vcan0  244   [5]  00 00 00 01 E8
  vcan0  188   [4]  00 00 00 00
  vcan0  191   [7]  01 00 90 A1 41 00 30
  vcan0  133   [5]  00 00 00 00 98
  vcan0  136   [8]  00 02 00 00 00 00 00 1B
  vcan0  13A   [8]  00 00 00 00 00 00 00 19
  vcan0  13F   [8]  00 00 00 05 00 00 00 1F
  vcan0  164   [8]  00 00 C0 1A A8 00 00 31
```

Candump is used to display the packets being captured.

```
┌──(kali㉿kali)-[~/Desktop]
└─$ canplayer -I candump-2023-03-31_184753.log
```

Canplayer can be used to replay packets on the CAN interface. In this case the left indicator packet was replayed.



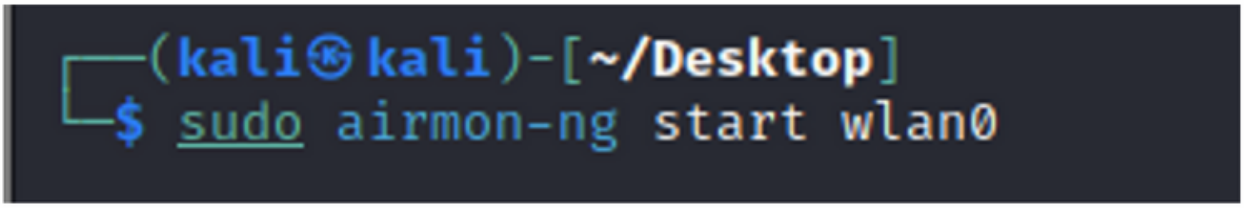
The simulated vehicle's left indicator is on after the packet dump was replayed.

EXPLOITING THROUGH BLUETOOTH

To compromise a vehicle using Bluetooth, the target vehicle's MAC address is needed. This can be found by sniffing the car's Bluetooth traffic when the vehicle is started near a previously paired device. Alternatively, the Bluetooth traffic of a known device can be sniffed without needing to be near the vehicle. Next the PIN used during pairing is required and can be captured by sending a pairing request to the vehicle if the vehicle is known to respond to all requests. Using the MAC address and PIN, attacks such as bluesnarfing can now be performed on the vehicle.

EXPLOITING THROUGH WI-FI

The kismet tool alongside a wireless network adapter can be used to exploit Wi-Fi networks. The wireless network adapter will first need to be put into monitor mode. In Kali this can be done in the terminal with the following command if the adapter was named wlan0.



The name of your adapter can be found using the ifconfig or ip a commands. You can then launch kismet and begin capturing information on wireless networks in the area. Kismet allows you to see clients associated with a network, providing a list of devices connected to a particular network. This can be used to identify what Wi-Fi network a CAV may be connected to and perform targeted de-authentication attacks.

EXPLOITING THROUGH CELLULAR

Knowing the target CAV's IP address, the vehicle's cellular network can be accessed via a smartphone plugged into a laptop. This device's cellular connection can then be used to extract information from the vehicle as well as possibly override crucial code such as firmware that would allow for physical components of the CAV to be compromised.

BEYOND EXPLOITATION

Once a CAV has been exploited, a threat actor could be capable of,

- Controlling physical components such as the breaks, lights, and engine;
- Geolocation tracking;
- Extracting sensitive data from the vehicle;
- Compromising devices connected to the vehicle such as over Bluetooth; and
- Installing malicious software in the vehicle's system.

Abbreviation	Definition
ADAS	Advanced Driver Assistance Systems
CAN	Controller Area Network
CAV	Connected Autonomous Vehicles
CIA	Confidentiality, Integrity, and Availability
DoS	Denial of Service
ECU	Engine Control Unit
EV	Electric Vehicle
ISO	International Organization for Standardization
IP	Internet Protocol
IoT	Internet of Things
IVN	In-Vehicle Network
LIN	Local Interconnect Network
LTE	Long-Term Evolution
MOST	Media Oriented Systems Transport
OBD	On-Board Diagnostics
OSINT	Open-Source Intelligence
OTA	Over-the-air
PII	Personal Identifiable Information
RAM	Random Access Memory

Abbreviation	Definition
RFID	Radio Frequency Identification
SAE	Society of Automotive Engineers
SAST	Static Application Security Testing
STRIDE	Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, and Elevation of privilege
SOW	Statement of work
TARA	Threat Assessment and Remediation Analysis
TCU	Telematics Control Unit
TPMS	Tire Pressure Monitoring System
V2C	Vehicle-to-cloud
V2D	Vehicle-to-device
V2G	Vehicle-to-grid
V2I	Vehicle-to-infrastructure
V2N	Vehicle-to-network
V2P	Vehicle-to-pedestrian
V2V	Vehicle-to-vehicle
V2X	Vehicle-to-everything
VM	Virtual Machine

REFERENCES

Amblard, M. (2022, January 23). Landscape of the Autonomous Vehicle Ecosystem (10th edition). Medium. Retrieved April 7, 2023, from https://medium.com/@mamblard75/landscape-of-the-autonomous-vehicle-ecosystem-10th-edition-341639812d53

Automotive Security Research Group. (2020, August 8). Automotive in-vehicle networks. YouTube. Retrieved April 7, 2023, from https://www.youtube.com/watch?v=EtDjyBPfMjw&list=PLpfIV1bSSmK-oERoJW_2gAXbOOAyoA2ay&index=1&ab_channel=AutomotiveSecurityResearchGroup

Automotive Security Research Group. (2023, January 12). V2X Valhalla. YouTube. Retrieved April 7, 2023, from https://www.youtube.com/watch?v=i-fx_3HWP4A&ab_channel=AutomotiveSecurityResearchGroup

Bozdal, M., Samie, M., & Jennions, I. (2018). A survey on CAN bus protocol: Attacks, Challenges, and potential solutions. 2018 International Conference on Computing, Electronics & Communications Engineering (ICCECE). https://doi.org/10.1109/iccecome.2018.8658720

Bozdal, M., Samie, M., Aslam, S., & Jennions, I. (2020). Evaluation of CAN bus security challenges. Sensors, 20(8), 2364. https://doi.org/10.3390/s20082364

Car hacking is the new carjacking. Packetlabs. (2023, January 9). Retrieved April 7, 2023, from https://www.packetlabs.net/posts/car-hacking-is-the-new-carjacking/

Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., & Kohno, T. (2011). Comprehensive Experimental Analyses of Automotive Attack Surfaces. Centre for Automotive Embedded Systems Security. Retrieved April 7, 2023, from http://www.autosec.org/pubs/cars-usenixsec2011.pdf

Controller Area Network (can bus) protocol. Kvaser. (2022, September 29). Retrieved April 7, 2023, from https://www.kvaser.com/can-protocol-tutorial/

Coward, C. (2023, April 7). Hacking a car's key fob with a Rolljam attack. Hackster.io. Retrieved April 7, 2023, from https://www.hackster.io/news/hacking-a-car-s-key-fob-with-a-rolljam-attack-7f863c10c8da

Drake, V. (n.d.). Threat modeling. Threat Modeling | OWASP Foundation. Retrieved April 7, 2023, from https://owasp.org/www-community/Threat_Modeling

Greenberg, A. (2015, July 21). Hackers remotely kill a Jeep on the highway-with me in it. Wired. Retrieved April 7, 2023, from https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/

Jegeib, barbkess, msmbaldwin, DavidCBerry13, & brittmsantos. (2022, August 25). Threats - microsoft threat modeling tool - azure. Threats - Microsoft Threat Modeling Tool - Azure | Microsoft Learn. Retrieved April 7, 2023, from https://learn.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats

Kathalu, J. (2022, February 9). Bluetooth hacks: How to hack a car's bluetooth? Geek Computer. Retrieved April 7, 2023, from https://www.geek-computer.com/wiki/bluetooth-hacks-how-to-hack-a-cars-bluetooth

Khatri, N., Shrestha, R., & Nam, S. Y. (2021, April 8). Security issues with in-vehicle networks, and enhanced countermeasures based on Blockchain. MDPI. Retrieved April 7, 2023, from https://www.mdpi.com/2079-9292/10/8/893

Kim, P., & Kim, K. (2018). The hacker playbook 3: Practical guide to penetration testing. Secure Planet LLC.

Kismet – wireless network hacking, sniffing & monitoring. Hackonology. (2021, May 1). Retrieved April 7, 2023, from https://hackonology.com/courses/kali-linux/lesson/kismet-wireless-network-hacking-sniffing-monitoring/

Mayoni, E. (n.d.). Security issues in can bus- attack scenarios & risks. LinkedIn. Retrieved April 7, 2023, from https://www.linkedin.com/pulse/security-issues-can-bus-attack-scenarios-risks-ella-mayoni/
Null Byte. (2018). Use Kismet to Find & Monitor Nearby Wi-Fi Devices [Tutorial]. YouTube. Retrieved April 7, 2023, from https://www.youtube.com/watch?v=3v_bwtHlToQ&ab_channel=NullByte.

Ojha, Y. (2019, November 24). Car hacking 101: Practical guide to exploiting can-bus using instrument cluster simulator-part... Medium. Retrieved April 7, 2023, from https://medium.com/@yogeshojha/car-hacking-101-practical-guide-to-exploiting-can-bus-using-instrument-cluster-simulator-part-ee998570758

Olmstead, K. (2022, August 2). Introduction to car hacking: The can bus. OffSec. Retrieved April 7, 2023, from https://www.offensive-security.com/offsec/introduction-to-car-hacking-the-can-bus/

Sachdev, M. (2022, November 2). 7 types of vehicle connectivity. Blog. Retrieved April 7, 2023, from https://blog.rgbsi.com/7-types-of-vehicle-connectivity

Samwcyo. (2023, January 11). Web hackers vs. the auto industry: Critical vulnerabilities in Ferrari, BMW, Rolls Royce, Porsche, and more. Sam Curry | Web Application Security Researcher. Retrieved April 7, 2023, from https://samcurry.net/web-hackers-vs-the-auto-industry/

Smith, C. (2016). The car hacking handbook. O'Reilly Media

Steve Mould. (2020, December 3). I hacked into my own car. YouTube. Retrieved April 7, 2023, from https://www.youtube.com/watch?v=5CsD8I396wo

Wang, J., Shao, Y., Ge, Y., & Yu, R. (2019). A survey of vehicle to everything (V2X) testing. Sensors, 19(2), 334. https://doi.org/10.3390/s19020334

ACKNOWLEDGMENTS

ALLIYAH MOHAMMED

Alliyah Mohammed is a 5th-year Computer Science student at Toronto Metropolitan University with a background in cybersecurity from past co-op placements and self-studying. As a Cyber Intern for the Rogers Cybersecure Catalyst, she worked alongside industry fellow AJ Khan, CEO of Vehiqilla, to develop this white paper on CAV hacking. This paper was a collaborative effort aimed at highlighting the importance of CAV security and the entire process of pentesting a CAV.

MARCUS SANTOS

My career is deeply rooted in Computer Engineering. I was born and raised in Brazil, and obtained my Phd in Artificial Intelligence at the Universidade de São Paulo. Since 1999, I have taught Computer Science to thousands of students at Toronto Met. My prime interest is Artificial Intelligence and Genetic Programming. In 2014, I was appointed Associate Dean, Undergraduate Programs and Student Affairs – a role that combines my passion for both the academic and human sides of education. In 2021, I was appointed Executive Academic Director, Cyber Studies and Rogers Cybersecure Catalyst - a role that combines my effective academic leadership and creativity in program development.

AJ KHAN

AJ Khan is the CEO and founder of the CyberStrategiez Inc. Group. AJ Is passionate about bringing together his two passions, cybersecurity and innovation, to add value to the connected world we live in. AJ believes that Cybersecurity is not one size fits all and focuses on enabling cyber innovation in new and challenging disciplines of our connected world.

VEHIQILLA

We take your fleet's cybersecurity seriously

Acquire a FleetCyber package with Vehiqilla today

CONTACT US

www.vehiqilla.com
info@vehiqilla.com
+1 226 674 0725
4510 Rhodes Drive, Suite 510, Windsor, ON N8W 5K5, Canada
333 West San Carlos Street, Suite 600, San Jose, CA 95110, USA