A Vulnerability Assessment Testing Methodology for V2X-Enabled Environments

by

Jason Howard Cuneo

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of
Master of Science

Auburn, Alabama
December 10, 2022

Keywords: Vulnerability Assessment, V2X Security

Approved by

David Umphress, Chair, Professor, CSSE
Daniel Tauritz, Co-Chair, CSSE
Mark Yampolskiy, Member, CSSE

Abstract

V2X systems and networks are unlike existing wireless technologies since more data, bandwidth, and speed is required when communicating over a highly dynamic and ever-changing vehicular network. Protocols supporting V2X environments must have more advanced hardware and software to process the volume of messages necessary to effectively transmit telemetry and safety data. The existing vulnerability assessment testing methodology is geared towards enterprise networks and protocols such as IEEE 802.3 and IEEE 802.11a/n/g and do not address the needs of more advanced vehicular networks like CAN, IEEE 802.11p, IEEE 1609.3, and IEEE 1609.4. This thesis provides an overview of the benefits of deploying V2X enabled systems and networks into the transportation infrastructure and provides an overview of V2X communication protocols. By evaluating the existing vulnerability assessment testing methodology, it is shown why it is insufficient for V2X environments and an improved testing methodology is proposed that meets the needs of those environments. The thesis concludes with results from two case studies that demonstrate the potential efficacy of the proposed improvements.

Acknowledgments

I would like to acknowledge the following people for their support during the development of this thesis. First, and foremost I would like acknowledge Jesus Christ for saving me from an eternity without him. I pray that those reading this document would consider a relationship with him. Next, I would like to thank my wife and children, Valerie, Samuel, and James for their dedicated support, I love you dearly! I would like to give a special thanks to Chris Parkman and Scientific Research Corporation for their interest and support of a V2X testing environment. Lastly, I would like to thank my thesis committee for their patience and support during the development of this document.

Table of Contents

Chapter 1

Introduction

On average some 40,000 people in the United States and 1.35 million people around the world die annually from vehicular-related accidents, which has prompted numerous research efforts to address this critical problem [1]. One method proposed to improve the safety outcomes on roadways throughout the world is the development of an Intelligent Transportation System (ITS) that allows vehicles and pedestrians to share data such as position, speed, and direction while in transit and at rest [2]. A system of this kind will provide real-time routing and communication that allows endpoints to generate messages based on changes in traffic conditions, weather, and emergency events. Additionally, other studies have considered the augmentation of ITS systems with autonomous vehicles to determine the efficacy of this technology and whether safety outcomes could also be improved [3].

Although utilization of ITS technologies has great potential to reduce vehicular-related deaths, this nascent technology also introduces new security concerns as well. Due to its heavy dependence on cellular and wireless technologies, ITS systems present adversaries with an attack surface not previously seen in transportation systems. By their very nature, ITS systems and networks are unlike existing wireless technologies since more data, bandwidth, and speed is required when communicating over a highly dynamic and ever-changing vehicular network.

Protocols supporting ITS environments must have more advanced hardware and software to process the volume of messages necessary to effectively transmit telemetry and safety data. Not only are location, speed, and heading information shared over the network, but information relating to safety system status and decision-making logic is distributed as well. Because of differences with existing wireless systems, attackers will have to develop new methods of detecting, collecting, and manipulating ITS systems and their associated networks.

Although ITS provides an umbrella term for frameworks used in advanced vehicular communications, the more common description is Vehicle-to-Everything (V2X) which includes four different communication paths: Inter-Vehicular Communications (IVC), Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), and Vehicle-to-Pedestrian (V2P) [4]. Unlike traditional enterprise networking protocols, V2X protocols will need to handle more complex and larger data packets including status such as position, speed, direction of travel, and safety conditions. These data items will then be shared over the network to allow endpoints to determine the most appropriate course of action based on those conditions.

Attackers armed with information about the function and operation of V2X-enabled systems will use an array of attack methods to degrade, deny, and disrupt inter-vehicular and extra-vehicular communications between vehicles, infrastructure, and pedestrians and several research efforts demonstrated successful attacks on physical, near-field, and far-field layers [5,6,7] as illustrated in Figure 1.

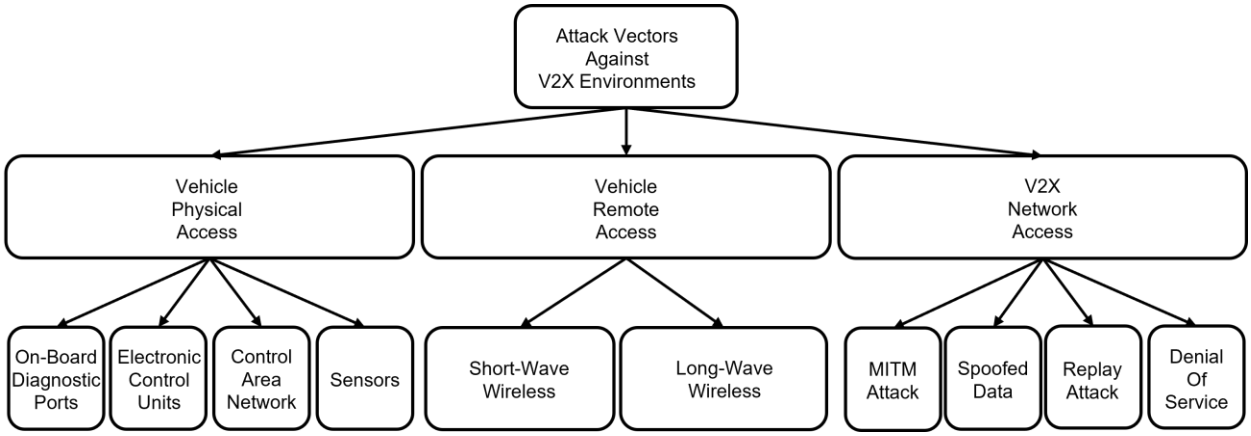


Figure 1. Attack Vectors Against V2X Environments [5,6,7].

As illustrated in Figure 2, V2X-enabled environments require extensive use of advanced communication protocols to support inter-vehicular and extra-vehicular communications including physical (e.g., CAN), short-range RF (e.g., Bluetooth, NFC), long-range RF (e.g., IEEE 802.11p), cellular (e.g., 4G-V2X, 5G-V2X), and GPS.

Figure 2. V2X Networked Environment [8].

Unlike existing vehicular traffic which depends heavily on driver observation, V2X-enabled environments increase safety outcomes by sharing sensor data between platforms communicating over the network. Instead of making safety decisions based only on driver decisions, V2X

networks will contain status updates from driver and sensor inputs for V2V, V2I, and V2P platforms.

Applying the benefits of a V2X-enabled environment, consider the case of a single vehicle stranded on the shoulder of an interstate. Although this case appears simple, each driver needs to evaluate several factors before deciding if the stranded vehicle is a threat to safe travel. To make an informed decision, each driver must consider factors such as: 1) The orientation of the stranded vehicle, 2) The location of pedestrians outside of the stranded vehicle, 3) The driver's vehicle speed, 4) Surrounding vehicles' location, orientation, and speed, 5) Current weather conditions, and 6) Location of roadway obstacles. These are by no means the only factors that a driver will consider for a safe outcome as it passes the stranded vehicle, but it highlights the need for an advanced traffic management system that could assist a driver to make effective decisions while traveling.

Due to the burgeoning development of V2X environments, a critical challenge facing security practitioners is how to effectively conduct vulnerability assessments against V2X systems and the networks that connect them. Traditional vulnerability assessments focus on enterprise protocols such as IEEE 802.3 and IEEE 802.11a/g/n, but the dynamic nature of V2X environments creates a capability gap that must be filled with new evaluation tools and techniques. As a result of these gaps, several questions must be answered before proposing an improved vulnerability assessment methodology for V2X environments including:

- What are the underlying protocols within V2X and why can't existing vulnerability assessment methods be applied to them?
- What testing capabilities are needed to generate attacks against V2X systems and environments?

- What are the phases of a traditional vulnerability assessment testing methodology and what modifications must be made to improve V2X vulnerability assessment outcomes?

The answer to each of these questions will be addressed in this thesis and we will conclude with a proposed vulnerability assessment testing methodology for use in future V2X vulnerability assessments.

Chapter 2

Overview of V2X Communication Protocols

Although many well-known protocols have been utilized for enterprise wired and wireless communication networks (e.g., IEEE 802.3, IEEE 802.11 a/g/n), they lack the data structures, power specifications, and advanced functionality necessary to operate within a dynamic V2X environment. This reality becomes apparent when reviewing the traditional OSI model and the functionality associated with each layer as shown Figure 3 [9].

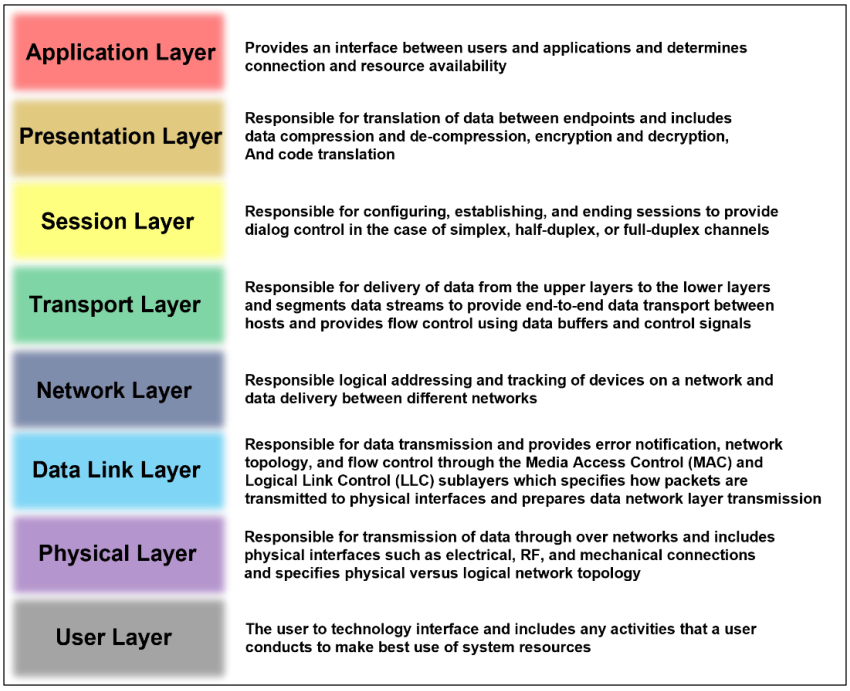


Figure 3. Traditional OSI Layers and Associated Functions.

To address the limitations within the traditional OSI model and enterprise protocols, several professional organizations including the Institute of Electrical and Electronics Engineers (IEEE) and the Society of Automotive Engineers International (SAE International) created several classes of vehicular-based protocols that improved the structure of the OSI model with the result shown in Figure 4 [10,11]. In comparison to the traditional OSI structure, the upper layers of the modified OSI model either combine functions or establish new layers to address V2X specific functionality needs. Additionally, the traditional OSI model does not directly address user-related activities, which is a critical interface within a V2X environment, so a user level is also added.

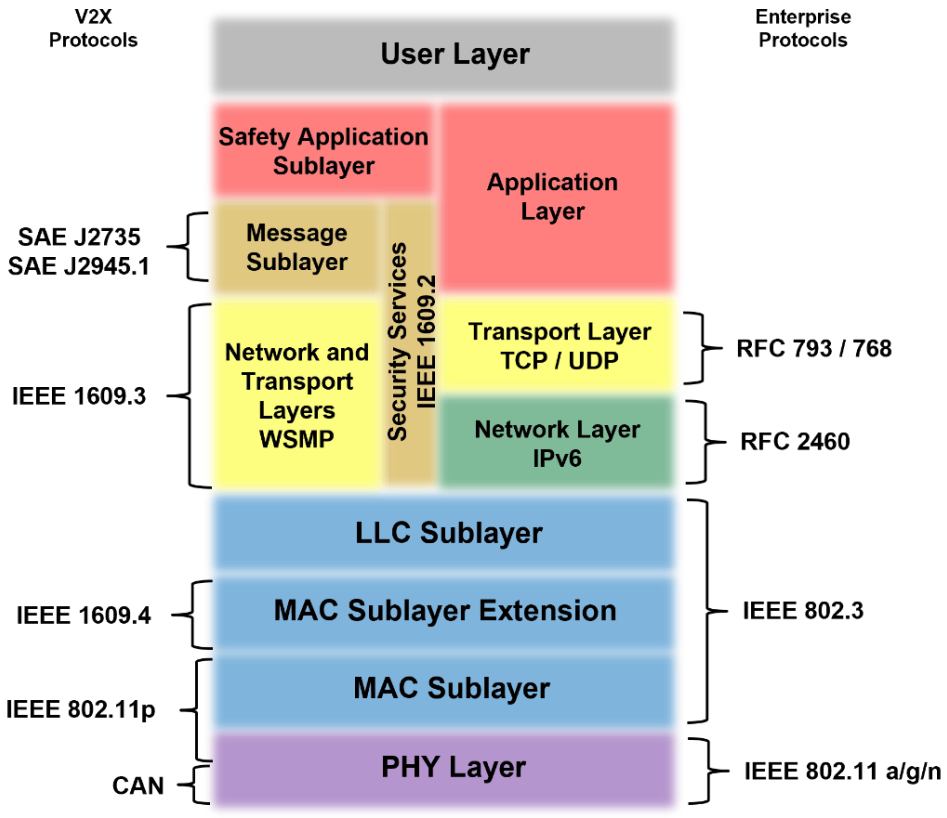


Figure 4. Modified OSI Model for V2X Protocols [10].

a) User Layer

Regardless of technological improvements, user actions and interactions must always be evaluated to determine the overall security posture of networks, systems, or applications. Even with security mitigation strategies integrated into a system, failure of a single user to abide by best security practices can result in loss of confidentiality, integrity, or availability of system resources.

b) Physical Layer – Controller Area Network (CAN), SAE J1939

The first interface that most drivers and passengers interact with is the physical layer. In a V2X environment, the protocols that fall into this layer are the Controller Area Network (CAN) and SAE J1939 known as "Recommended Practice for a Serial Control and Communications Vehicle Network" [12]. Although a casual observer might mistakenly think that these protocols are synonymous, a historical review shows that they are different in several areas. As highlighted in Table 1, the CAN protocol was developed to address several shortcomings of traditional point-to-point communication networks. With the increased complexity of Electronic Control Units (ECU) distributed throughout a vehicle combined with modern power distribution systems, traditional point-to-point (P2P) wiring and communication protocols could no longer support the growing capability needs within those vehicles.

Table 1. CAN – J1939 Historical Overview [13,14].

Year	Innovation
Pre-CAN	Electronic Control Units (ECU) communicate over complicated point-to-point wiring harnesses
1983	Robert Bosch conceives of the Controller Area Network (CAN) protocol to improve automobile quality and safety through improved messaging between ECUs
1986	Bosch published CAN 1.0 protocol
1991	Bosch publishes CAN 2.0 to extended message formatting
1993	OSI publishes ISO 11898 which standardized CAN for vehicular communication networks
1994	SAE publishes J1939-11, 21, and 31 to standardize heavy-vehicle communication networks
2000	SAE adds CAN to the J1939 standard
2001	SAE replaces all previous communication standards with J1939
2012	Bosch publishes CAN Flexible Data Rate (CAN FD 1.0)
2015	OSI publishes ISO 11898-1 which standardized CAN FD
2016	OSI published ISO 11898-2 which standardized 5 Mb/s CAN data rates

After publication of CAN 2.0, support grew across the automotive industry and led to protocol standardization by OSI in ISO 11898 and publication of SAE J1939 standard for heavy-vehicle communication networks [15]. Although the initial SAE J1939 standard did not originally include the CAN protocol, SAE International and the automotive Original Equipment Manufacturers (OEM) recognized CAN protocol innovations and they subsequently amended SAE J1939 to include CAN.

To gain an appreciation for the level of technical complexity contained within modern vehicles, and why the CAN protocol was so critical for efficient operation, Figure 5 illustrates the wide variety of ECU's common on a CAN bus network. Although some of the more common ECUs distributed throughout a vehicle include systems such as the engine, acceleration and braking sub-systems, suspension system, sensor arrays, and driver dashboard systems, more complex vehicle designs can have hundreds of ECU's operating within the Intra-Vehicular Network (IVN) [16].

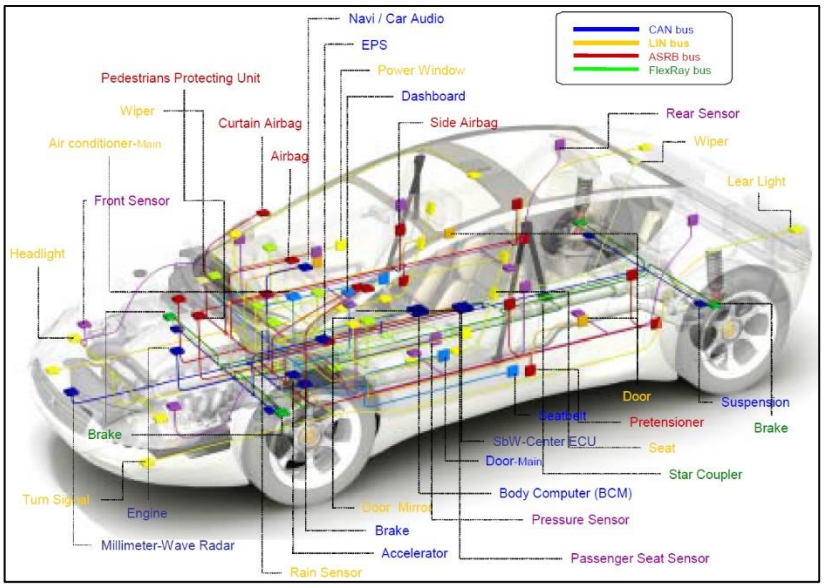


Figure 5. IV communications networks [17].

From a technical perspective, each CAN 2.0 message contains eight fields, and a maximum of 126 bits, which provides information about the communication bus and ECU's including the start and end of each message, source and destination addresses, frame types, message integrity checks, and ECU data [18,219]. Each CAN 2.0 message is broadcast over the IVN in an unencrypted and unauthenticated manner.

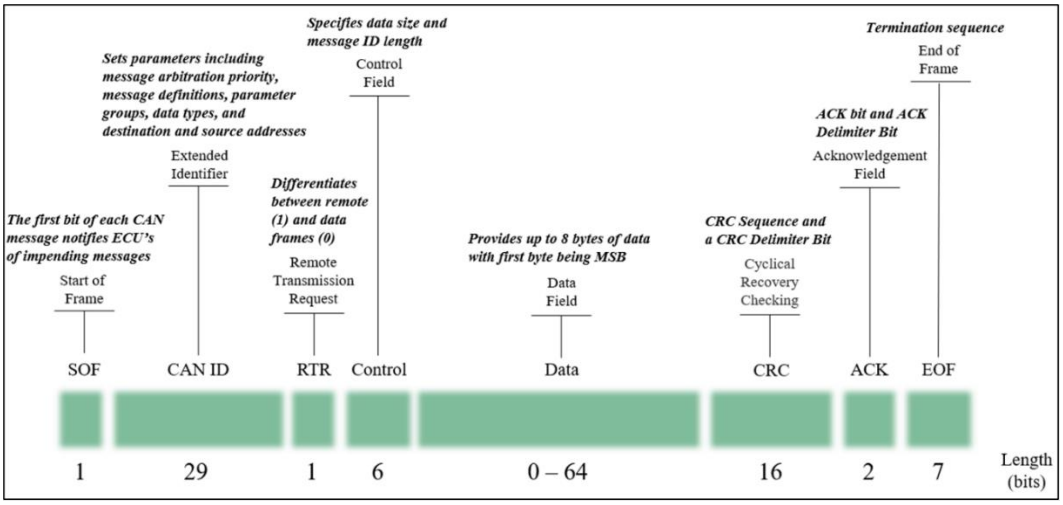


Figure 6. CAN 2.0 Message Composition.

Starting in 1996, all light vehicles and trucks were required to provide an Onboard Diagnostics (OBD) port to support emissions testing and in 2005 this requirement expanded to heavy vehicles [20]. Examples of light and heavy vehicle interfaces are seen in Figure 7.

Figure 7. CAN Network Interfaces [21].

c) Physical Layer / Media Access Control Sub Layer – IEEE 802.11p

Recognizing that earlier 802.11 standards for wireless access such as 802.11a, g, and n were inadequate for extra-vehicular communication, IEEE introduced IEEE 802.11p-2010 to establish communications across the Physical Layer (PHY) and Medium Access Control (MAC) sub-layers of the OSI model [22]. Table 2 compares the technical capabilities of legacy IEEE 802.11a versus IEEE 802.11p and shows improvements on several levels including increased channel capacity for the transmission of control and service data between vehicles and increases packet delivery ratio (PDR) in vehicular environments by over 80% as compared with the earlier standard [23].

Table 2. IEEE 802.11a/p Comparison [23].

Parameter	802.11a	802.11p
Channels	1	7
Frequency Range	5 GHz	5.86 - 5.92 GHz
Subcarriers	52	52
Data Rates (Mbps)	6, 9, 12, 18, 24, 36, 48, 54	3, 4, 5, 6, 9, 12, 18, 24, 27
Modulation Types	BPSK, QPSK, 16QAM, 64QAM	BPSK, QPSK, 16QAM, 64QAM
Coding Rates	1/2, 1/3, 3/4	1/2, 1/3, 3/4
OFDM Symbol Duration	4 µs	8 µs
Guard Time	0.8 µs	1.6 µs
Preamble Duration	16 µs	32 µs
Subcarrier Spacing	0.3125 MHz	0.15625 MHz

The increase in channels, frequency range, symbol duration, and the reduced subcarrier spacing make IEEE 802.11p a more robust and capable protocol when applied to vehicular environments. Arguably the greatest improvement that IEEE 802.11p provides is the greater channel capacity as highlighted in Figure 7. The center channel of the protocol is configured to provide control information between vehicles (CCH), while the remaining six service channels provide non-safety and information system messages (SCH) [26].

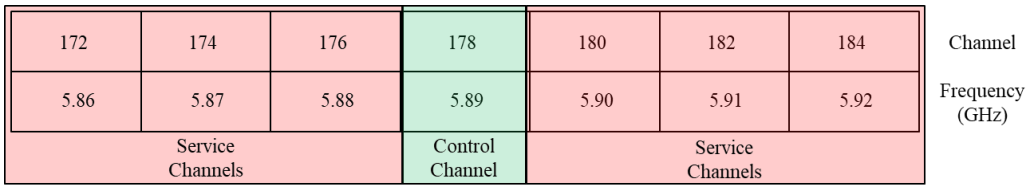


Figure 7. Channels and Frequency Ranges for IEEE 802.11p [24].

d) MAC Sublayer Extension – IEEE 1609.4

In 2016, the Intelligent Transportation Systems Committee of the IEEE Vehicular Technology Society published IEEE 1609.4-2016 which defined the functions and services required for multi-channel wireless connectivity between IEEE 802.11 Wireless Access in Vehicular Environments (WAVE) [25]. This standard specifies data plane and management services, where data plane services focus on functions such as channel coordination, channel routing, and user

priority while management services focus on functions such as multi-channel synchronization, channel access, management information base (MIB) maintenance, and MAC addressing changes [26]. When interfacing with other layers within the communication stack, WAVE utilizes control, management, and data frames depending on protocol requirements. For example, control frame information provides necessary values for IEEE 802.11 protocols below it, whereas management and data frames provide information for higher level protocols including Internet Protocol (IP) and WAVE Short Message Protocol (WSMP) [27].

e) Network and Transport Layers – IEEE 1609.3

IEEE 1609.3 consolidates network and transport layer functions into a structure known as the WAVE Short Message Protocol (WSMP) and its associated format known as the WAVE Short Message (WSM). Whereas the Internet Protocol (IP) has become the default method of routing network traffic over the Internet, vehicular applications do not require the level of routing required in enterprise applications. Since nodes on a vehicular network send packet information directly through RF transmissions, routing is not a primary concern which makes protocols in IEEE 1609.3 more streamlined than IP [28].

As per Figure 8, the first structure of the WSM is the WSMP version, which is a 1-byte value to specify the WSMP version used during the session. The version is followed by a 4-byte value that specifies the payload service identifier (PSID) which tells the message which process it is going to pass to for further processing. The PSID can be compared to the TCP / UDP port number which is specified in the transport layer and specifies which application the data will be used by [29].

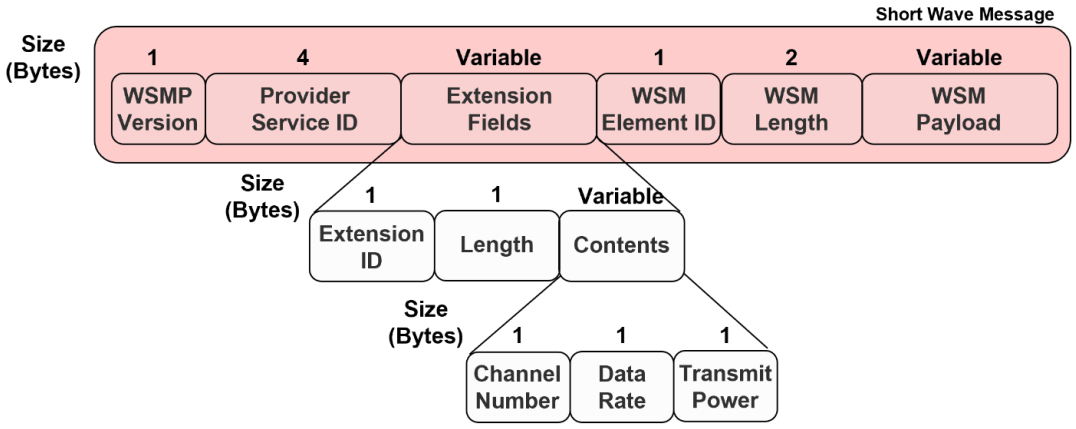


Figure 8. WAVE Short Message Format [29].

The next WSM structure, which is optional, is the extension field. If the WSM contains an extension field, it contains an extension ID, length, and contents that include transmission information such as channel numbers, data rates, and transmission power. Next, the WSM Element ID and WSM Length are mandatory and specify the WSM data field format and length of the WSM header respectively. The WSM data field format is important because it provides critical information about the destination of WSM data. Table 4 provides a summary of the WSM.

Table 4.* WSM *Description Summary.

Field	Size (Bytes)	Mandatory	Description
WSMP Version	1	Yes	4-bit WSMP Version Number / 4-bits Reserved
Provider Service Identifier	4	Yes	WSM Payload Service Identifier
WSM Extension Fields	Variable	No	Identifier, Length, and Content
WSM Element ID	1	Yes	Specifies the WSM Data Field Format
Length	2	Yes	WSM Data Field Size
WSM Data	Variable	Yes	WSM Payload

f) Safety Message Sublayer – SAE J2735 – SAE J2945

As with SAE J1939, SAE International also developed SAE J2735 – SAE J2945 which specifies the format of messages, data frames, and applications sent over a DSRC / WAVE environments

for heavy and light vehicles [30]. One of the critical improvements of this specification is the inclusion of Basic Safety Message (BSM) information that is generated and broadcast by vehicles within a V2X environment. From a technical perspective, SAE J2735 BSM's are composed of two parts which provide information about physical vehicle parameters such as heading, position, elevation, and steering wheel angle, whereas BSM Part II provides extension information such as safety event flags and path prediction to provide additional safety information. The definitions of each parameter of each part of the J2735 BSM are provided in Table 5 and Table 6 respectively.

Table 5. SAE J2735 Basic Safety Message (BSM) Part I Definitions [27].

Data item name, Element/Frame, and length	Description
`DSRC_MessageID` element, 1 byte	The first element in every message, used by the parser to determine how to parse the rest of the message
`MsgCount` element, 1 byte	A sequence number, incremented with each successive transmission of a `BSM` by a given vehicle, used primarily to estimate packet error statistics.
`TemporaryID` element, 4 bytes	A value chosen randomly and held constant for a few minutes, it helps a receiver correlate a stream of `BSMs` from a given sender.
`DSecond` element, 2 bytes	The current time, modulo one minute, with resolution 1 millisecond.
`Latitude, Longitude` 2 elements, 4 bytes each	Geographic latitude and longitude, with resolution 1/10 microdegree.
`Elevation` element, 2 bytes	Position above or below sea level, resolution 0.1 meter.
`PositionalAccuracy` frame, 4 bytes	Conveys the one-standard-deviation position error along both semi-major and semi-minor axes, and the heading of the semi-major axis.
`TransmisionAndSpeed` frame, 2 bytes	3 bits encode vehicle transmission (gear) setting. 13 bits convey unsigned vehicle speed, resolution 1 cm/second.
`Heading` element, 2 bytes	Compass heading of vehicle's motion, resolution 1/80 degree.
`SteeringWheelAngle` element, 1 byte	Current position of the steering wheel, resolution 1.5 degree. Clockwise rotation is a positive angle.
`AccelerationSet4Way` frame, 7 bytes	Provides longitudinal acceleration, lateral acceleration, vertical acceleration, and yaw rate.
`BrakeSystemStatus` frame, 2 bytes	Conveys whether or not braking is active on each of four wheels, also conveys the status of the following control systems: Traction Control, Anti-Lock Brakes, Stability Control, Brake Boost, and Auxiliary Brakes.
`VehicleSize` frame, 3 bytes	Vehicle length and width, resolution 1 cm.

Table 6. SAE J2735 Basic Safety Message (BSM) Part II Definitions [27].

Data item name, Element/Frame, and length	Description
`EventFlags` element, 2 bytes	An optional set of bit flags, each of which can convey the occurrence of a given "event." A given event may be flagged only if a set of minimum activation criteria are met. Examples include: Hard Brake, Hazard Lights, Emergency Response Vehicle, Stop Line Violation
`PathHistory` frame, variable length (typically on the order of 20 bytes for a straight path and less than 100 bytes for a curved path)	Used to convey where a vehicle has been, in the form of individual data structures sometimes called "Bread Crumbs." Each bread crumb includes a prior position, and optionally time and position accuracy. `PathHistory` is useful in identifying lane level information in the absence of map data. The number of bread crumbs in a frame is a function of the degree to which the actual path can be represented in piecewise linear fashion.
`PathPrediction` frame, 3 bytes	Indicates the path that a sender expects to traverse. 2-byte radius of curvature and 1-byte prediction confidence.
`RTCMPackage` frame, variable	Conveys GPS correction data in the RTCM style. Variable length depends on number of satellites in view.

Understanding technical specifications across each of the modified layers of the OSI and the associated V2X protocols will allow for a greater appreciation when attack methodologies are introduced.

Chapter 3

Survey of V2X Attack Vectors

Figure 9 highlights the most common interfaces used from a user, vehicle, and network perspective and maps them to attack methods that degrade, disrupt, and prevent proper operation of systems and users within a V2X environment [31,32].

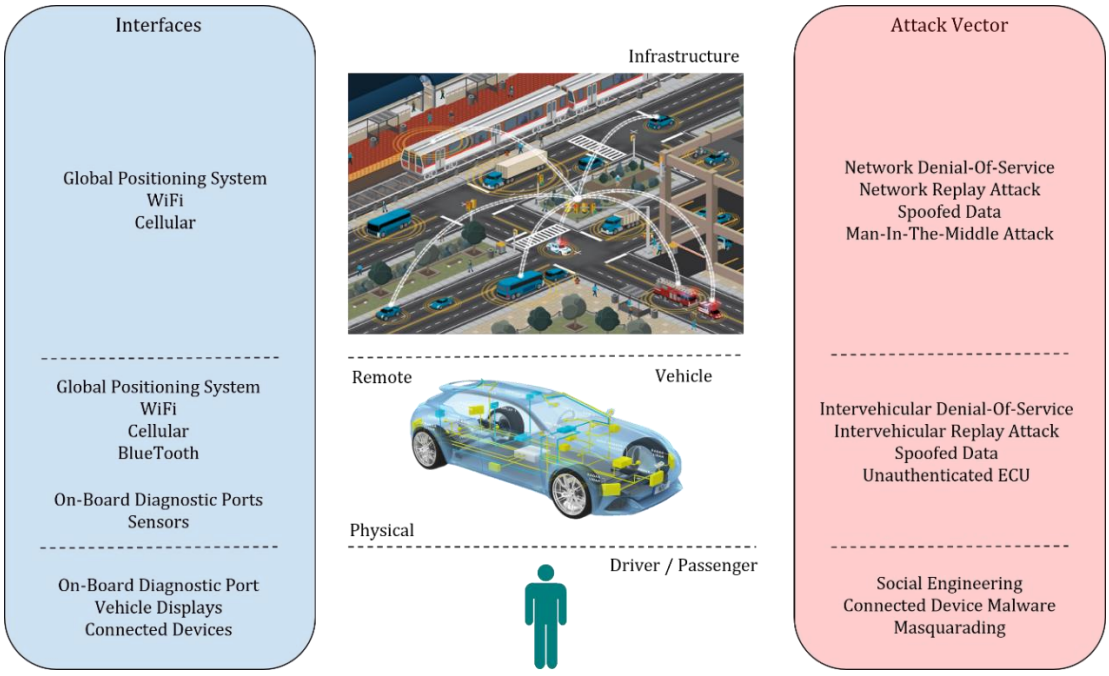


Figure 9. V2X Interfaces and Associated Attack Vectors.

a) User Layer Attacks

End users continue to be the weakest link in an organization's security program and provide attackers with the greatest possibility of system access [33,34]. Although users are generally categorized as drivers, passengers, pedestrians, and other human participants in a V2X network, automated systems that utilize and / or share network resources should also be considered users and potential targets of attack.

As with more traditional network configurations, users on V2X networks are not immune from social engineering attacks which attempt to deceive users into providing attackers with system access. Due to the extensive integration of external wireless and cellular systems into vehicular systems, attackers can use existing social engineering tools and techniques to connect to and pivot across V2X networks and connected systems [36]. For example, users that import system applications such as email and messaging services should consider the possibility that phishing attacks may be directed at gaining access to V2X systems and infrastructure. Due to the nascent nature of V2X technologies and users lack of experience with dealing with V2X security, phishing campaigns against these users will result in a greater threat to V2X security posture [35,36].

Another user-level attack to consider is a trusted device connected to a vehicle that initiates malware download, configuration, and initiation [37, 38]. Although there are several dependencies on the effectiveness of malware on vehicular systems, additional research has pointed to the possibility of malware spreading between V2V networks [39]. Due to the lack of intrusion detection and malware identification on mobile devices and V2X endpoints, the threat of vehicle infection from user connected devices will continue to increase until effective mitigation strategies are implemented.

b) Physical Layer Attack

Although physical access to information systems in an enterprise network is not always assumed, the same cannot be said for vehicular environments where physical access to vehicles, infrastructure, and pedestrians is a required condition for operation. The primary attack vectors at the physical layer of a V2X environment are against CAN and IEEE 802.11p.

Koscher et al. identified several attack vectors against the implementation of the CAN protocol that took advantage of the broadcast nature of the protocol and the relatively easy method of crafting messages over the CAN network [40]. Additionally, CAN networks do not authenticate nodes as they connect to the network which means that all nodes will automatically be given access to the network and be allowed to send and receive messages if their messages adhere to protocol formatting requirements. From a security perspective, this means that once an attacker finds an OBD-II or J1939 interface, they can unilaterally connect, observe, and send messages over the CAN network.

One last security consideration is that the CAN protocol is broadcast-enabled, meaning that all nodes on the network broadcast every message to every other node on the network. Coupled with the unauthenticated nature of CAN, an attacker can quickly connect and listen to all other messages sent through the network. A summary of all CAN protocol vulnerabilities and attack vectors are summarized in Table 7.

Table 7. CAN Network Vulnerabilities and Attack Vectors.

CAN Vulnerability	Potential Attacks
Unencrypted Messages	Network Sniffing
Unauthenticated Endpoints	Unrestricted Endpoint Entry Message Replay
Broadcast Messaging	Message Collection Malicous Message Generation

In addition to their original research effort, Koscher et al. expanded on their original findings and demonstrated several practical CAN network attacks that included: 1) Spoofing instrument panel readings, 2) Changing engine timing to disable engine operation, and 3) Manipulation of brake settings to prevent braking while in the vehicle was in motion [41]. Due to the nature of the CAN protocol, these types of attacks will continue until security enhancements are integrated into the specification.

c) Network Layer Attacks

A security survey conducted by Alnasser et al. noted that attacks against confidentiality, integrity, availably, authentication, and non-repudiation apply to V2X technologies and protocols and the idea was further expanded by demonstrating attack vectors against IEEE 802.11p, LTE-V2X cellular, and LTE-V2X device-to-device (D2D) technologies [42].

i) Confidentiality attacks

Attacks against confidentiality attempt to gain access to information only intended for legitimate users of a resource. An attacker conducting a confidentiality attack against V2X environments is trying to gather information about individuals, systems, and networks that can be used in a malicious manner.

- ✓ Network Sniffing

- If an attacker can connect to a network and listen to network communications between endpoints, it may be possible to capture authentication and protocol specific information. This type of attack can be thwarted if messages are encrypted prior to broadcast.

✓ Location Tracking

- Sharing location and safety information between endpoints within a V2X environment aids in increasing safety outcomes. Since location and safety information is provided openly to adjacent V2X nodes (i.e., V2V, V2I, V2P) attackers will be able to collect and analyze this information for potential attacks.

ii) Integrity attacks

The objective of integrity-based attacks is to introduce, manipulate, or question the validity of information sent over a V2X network and are categorized as either 1) Message injection, 2) Message manipulation, 3) Message replay, or 4) GPS spoofing.

✓ Message Injection

- Once an attacker has gained access to a V2X network, this attack vector requires technical knowledge of packet structures and how to broadcast messages once generated. The attacker's objective is to have legitimate endpoints on the network accept injected messages to impact the legitimate operation of the V2X network.

✓ Message Manipulation

- Sharing location and safety information between endpoints within a V2X technologies aids in increasing safety outcomes. Since location and safety information is provided openly to adjacent V2X nodes (i.e., V2V, V2I, V2P) attackers will be able to collect and analyze this information for potential attacks.

- ✓ Message Replay
    - Once an attacker gains access to a V2X network, traffic is collected and retransmitted over the same or different network. Objectives for a message replay attack include obfuscating attack objectives, invalidating user dashboards, or casting doubt on network trustworthiness.
- ✓ GPS Spoofing
    - Although GPS location and time information is provided from satellite constellations, if attackers can compromise orbiting nodes or broadcast over terrestrial nodes, it may be possible to change GPS information used by V2X systems.

iii) Availability attacks

Availability of a system means that legitimate users have access to a resource when they need it. Due to the dynamic nature of V2X environments, availability is a critical security principle that must be maintained to ensure both operational and safety requirements. Four types of availability attacks are used by attackers to reduce V2X resource availability, namely:

- ✓ Blackhole / Greyhole Attack
    - A key benefit of V2X networking protocols is their ability to provide real-time updates to vehicles, infrastructure, and pedestrians. A potential networking attack that can be used against V2X environments is where a node on the network takes messages sent over the network and drops those messages without broadcasting them to other endpoints [43]. Figure 10 illustrates a scenario where a vehicle experiences a safety critical event and broadcasts the event to the rest of the network, but a blackhole takes those updates and drops them before sharing the

information with other vehicles in the network. The only difference between blackhole and greyhole attacks is that all messages are dropped in the blackhole scenario, where only some strategic messages are dropped in a greyhole attack.

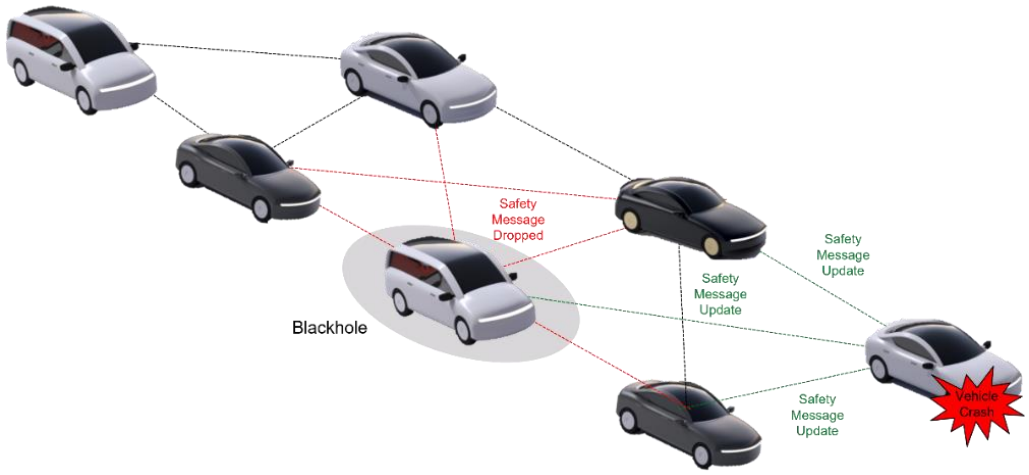


Figure 10. Blackhole Attack.

- ✓ Signal Flood
  - A signal flood is a denial-of-service attack used to prevent legitimate users on a V2X network from accessing critical resources. Due to the extensive use of RF communications used in V2X environments, an external signal flood would have to be extensive to be successful.

- ✓ Coalition Attack
  - Unlike blackhole and greyhole attacks that use a single attacker to collect and drop messages within a V2X environment, the scope of a coalition attack is extended to use multiple attackers. The objective of a coalition attack can be to create a general sense of confusion between endpoints, or to isolate a specific

endpoint. Figure 11 illustrates a coalition attack against an individual vehicle by sending specially crafted messages to isolate a vehicle [42].

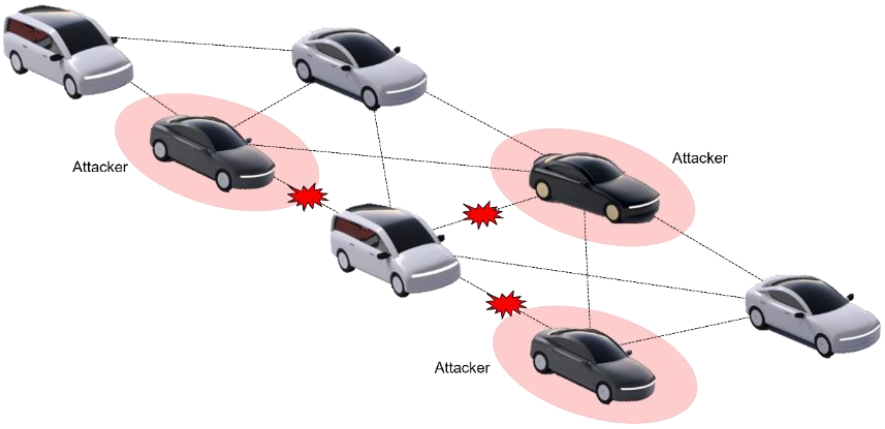


Figure 11. Coalition Attack.

iv) Authentication attacks

Authentication is the process of confirming the validity of a user prior to being given access to an information system. Authentication within a V2X environment is based on signed messages using a modified version of public key infrastructure where a public and private key pair are generated for confidentiality, integrity, and non-repudiation [44].

Figure 12 illustrates the steps that adhere to the European Telecommunications Standards Institute (ETSI) Intelligent Transportation System (ITS) Trust Model which is specified in the ETSI Technical Standard (TS) 102 940 [45].

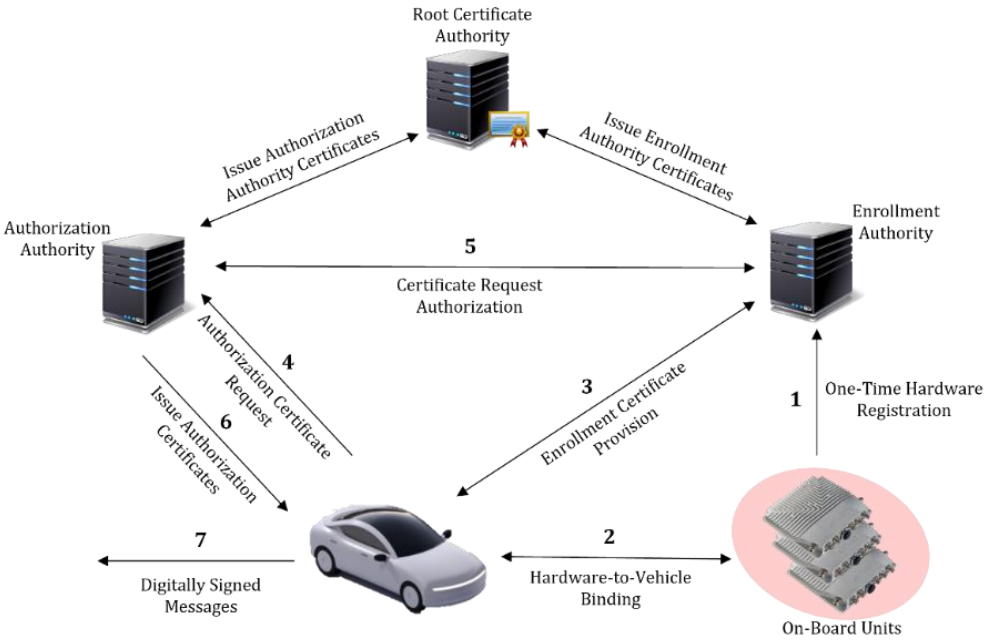


Figure 12. ETSI ITS Trust Model for V2X Communication [45,46].

Prior to sending digitally signed messages, each On-Board Unit (OBU) within a V2V or V2I node first needs to register according to the trust model specification:

1) Each OBU submits a one-time hardware registration request to the Enrollment Authority
2) Each OBU will bind to a specific vehicle or infrastructure system
3) Once bound, the system will request and receive an enrollment certificate provision
4) Concurrent with the enrollment certificate provision, the system will also request an authorization certificate from the authorization authority
5) The root certificate authority provides certificates to both the authorization authority and the enrollment authority which results in certificate request authorization
6) Authorization certificates are issued to the system
7) The system can send digitally signed messages across a V2X environment

Attackers with an understanding of the legitimate certificate registration process can then attempt several attacks against V2X authentication:

- ✓ Certificate Replication
    - A certificate replication attack occurs when an attacker compromises a V2X node and generates certificates from previously blacklisted certificates. A certificate replication attack works if the V2X environment does not have a certificate validation mechanism in place such as a Certificate Revocation List (CRL).
- ✓ Sybil Attack
    - If an attacker can compromise a V2X node, messages can be crafted with varying hardware identifiers making it difficult to trust the messages being received from that system [43]. An attack of this kind allows the attacker to generate different identification profiles.
- ✓ Masquerading
    - Unlike a Sybil attack, a masquerading attack attempts to impersonate a known V2X node so that messages destined for that system will be received, processed, and re-transmitted by the attacker.

Table 8 summarizes the possible attack vectors found by Alnasser et al. where an x indicates possible attack vectors, and a checkmark indicates non-likely attack vectors due to existing security controls.

Table 8. V2X RF Vulnerabilities and Attack Vectors [42].

Security Principle	Threat	IEEE 802.11p		LTE-V2X			
		[redacted]		Cellular-Based		D2D-Based	
		External	Internal	External	Internal	External	Internal
Confidentiality	Sniffing	✓	×	✓	×	×	×
	Location Tracking	×	×	×	×	×	×
Integrity	Message Manipulation	✓	✓	✓	×	✓	×
	Message Injection	✓	×	✓	×	✓	×
	Replay Attack	✓	✓	✓	✓	✓	✓
	GPS Spoof Attack	×	×	×	×	×	×
Availability	Blackhole Attack	✓	×	✓	×	×	×
	Signal Flood	✓	×	✓	×	×	×
	Signal Jamming	×	×	×	×	×	×
	Coalition Attack	✓	×	✓	×	✓	×
Authenticity	Certificate Replication	✓	✓	✓	✓	×	×
	Sybil Attack	✓	✓	✓	✓	×	×
	Masquerading	✓	✓	✓	✓	×	×
Non-Reputiation	User Distrust	✓	✓	✓	✓	×	×

Chapter 4

Traditional Vulnerability Assessment Testing Methodology[1]

Prior to suggesting an improved vulnerability assessment testing methodology for V2X-enabled environments, we must first consider activities commonly conducted during traditional vulnerability assessments. As Figure 13 shows, vulnerability assessments generally include coordination activities, system and network analysis, vulnerability and exploit analysis, and reporting of results.

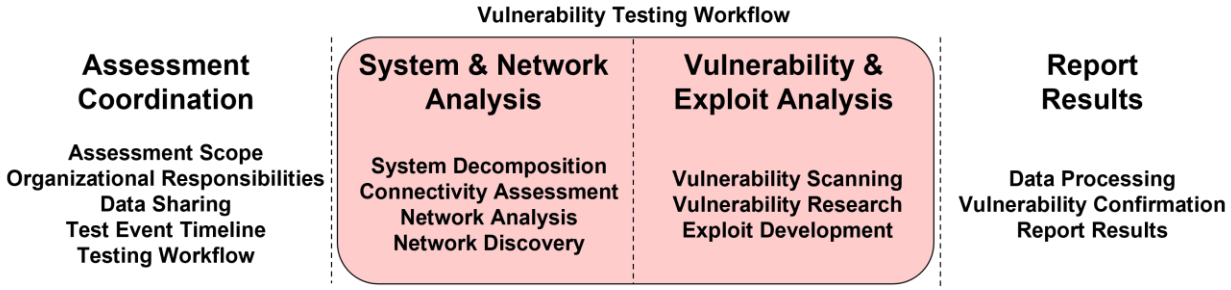


Figure 13. Traditional Vulnerability Assessment Testing Methodology.

[1] The vulnerability assessment testing methodology was used in support of prime contract number 22BRODBECK

a) Assessment Coordination

Prior to conducting vulnerability testing against a system or network several administrative and logistical activities that must be conducted first including a vulnerability assessment scoping review, assignment of organizational responsibilities, data sharing expectations, and a timeline for all events conducted during the assessment. Many contracts will require a kickoff meeting which will address each of these coordination activities.

1) Assessment Scope

The vulnerability assessment scope review brings together vendors and customers from across management and technical groups to ensure that expectations during the assessment are established. Prior to or during the kickoff meeting an assessment questionnaire will be answered by the customer to establish what events are being agreed to and the way those activities will be executed. Additionally, the project scope will include conflict resolution procedures in the event of a disagreement during contract execution. Key questions to ask during the assessment scope include:

- What systems / networks need to be assessed?
- Are there any exclusions during the assessment?
- Are there limitations on the types of security attacks that can be conducted during the assessment?
- Are there limitations on test tools that can be used during the assessment?
- Are there scheduled operational events that will prevent testing at certain times?
- What test results require immediate reporting?

2) Organizational Responsibilities

In addition to the assessment scope, the kickoff meeting also includes discussions and agreements on customer and vendor responsibilities throughout the lifecycle of the project. Contact information for all responsible parties during expected events and contingencies will be distributed to all parties.

- Who are the Points of Contact (POC) for the management and technical contingencies during the assessment?
- If a critical security finding is identified, what is the procedure for reporting?

3) Data Sharing

Data sharing between testing vendors and customers is critical to ensure system capabilities are understood and how test events may impact system function. Data sharing of this kind will include exchange of network diagrams, hardware and software interface control documents (ICD), and policy documentation that elaborates the anticipated operation of the system.

4) Test Event Timeline

Assessment coordination will also include development of a detailed schedule of events during the life of the assessment. A comprehensive schedule will ensure that personnel, facilities, and systems under test are allocated and available when the vendors arrive on site. In addition, test timeliness will include details about the environment configuration prior to conducting each test in support of the vulnerability assessment. For example, if a test event requires testing of onboard cellular systems, then the organization under test will ensure that cellular subscriptions and SIM cards are installed and active prior to test execution.

5) Vulnerability Testing Workflow

Another product provided to the organization under test during the assessment coordination phase is a draft of the vulnerability test workflow that the vendor will use during test events. The workflow is derived from the system and network analysis and vulnerability and exploit phases of the vulnerability assessment testing methodology. Figure 14 highlights the procedures for technical tests conducted on the system and will be explained further in the next section.

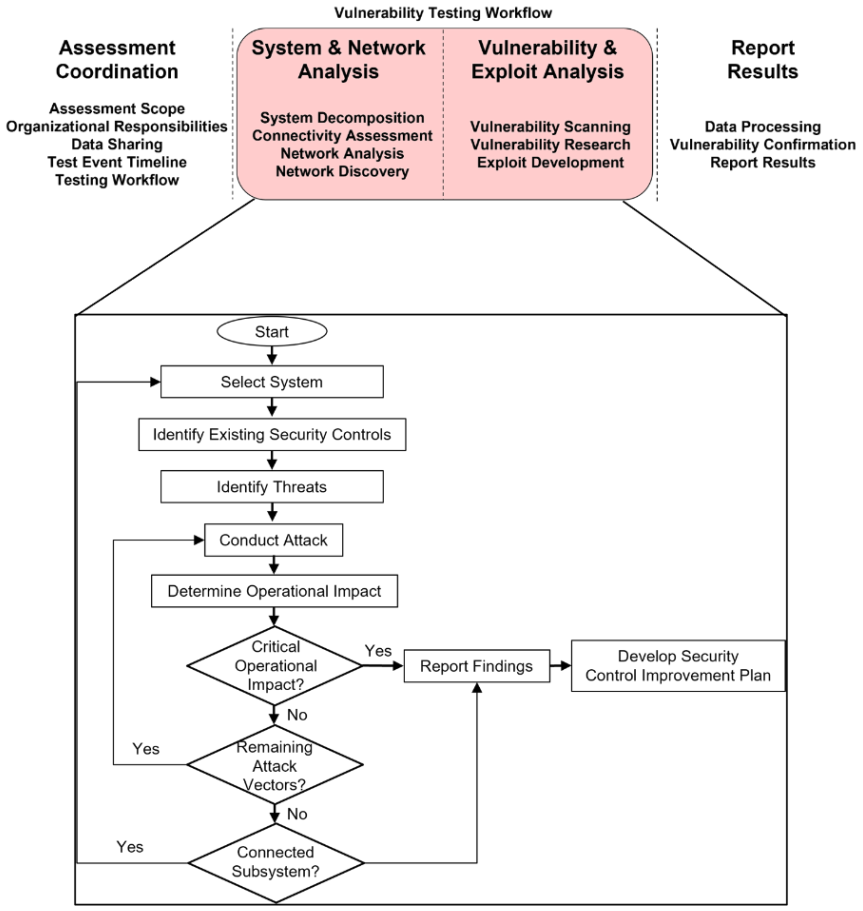


Figure 14. Traditional Vulnerability Testing Workflow.

b) System & Network Analysis

This phase of the vulnerability assessment methodology attempts to break down the constituent parts of the system and network under test to identify weaknesses that can be used by threats to negatively affect confidentiality, integrity, or availability.

1) System Decomposition

During data sharing collaborations vendors will gain an understanding of the underlying functions of systems and will evaluate the internal structure of systems and integrated software. System decomposition can assist will identifying security gaps that would not be otherwise identified because low-level analysis was never conducted. Due to supply chain security concerns, emphasis has been put on conducting more thorough system decomposition to identify weaknesses introduced during system development and deployment.

2) System Interconnection

A helpful consequence of system decomposition is that it can provide an electrical mapping of sub-systems and connected systems. Although understanding wired and wireless network connections is critical during a vulnerability assessment, system interconnection also looks at electrical connectivity to determine if there are existing attack paths into the system. Analysis of system interconnections may illuminate an unanticipated connection that can be used to access a system or a connected sub-system.

3) Network Analysis

Although network diagrams and interconnections are provided during the coordination phase of the vulnerability assessment methodology, it is likely that not all network connections are completely understood prior to security testing. During this step, networks are scanned to determine the actual network configuration and those results are compared to network diagrams to ensure continuity. Any deviations from expected results will be either reported immediately or added to the final report based on reporting procedures.

c) Vulnerability & Exploit Analysis

Phase 3 of the vulnerability assessment methodology evaluates systems to determine if system misconfigurations and vulnerabilities exist and if there are legitimate exploits that can take advantage of those misconfigurations and vulnerabilities.

1) Vulnerability Scanning

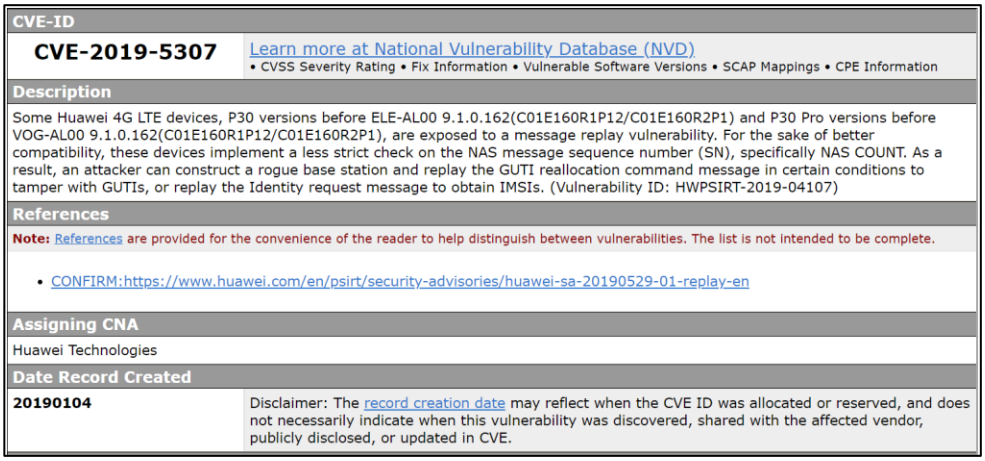
Vulnerability scans collect a wide array of network, system, and application data to determine potential vulnerabilities in a system under test by comparing scan responses with known vulnerability signatures.

2) Vulnerability Research

Although vulnerability scanning provides a small window into the overall vulnerability profile of a networked environment, it does not provide a full understanding of system vulnerabilities, so vulnerability research becomes a critical feature of an effective vulnerability assessment methodology. There are several critical resources that security assessors use to conduct proper vulnerability research.

- Common Vulnerabilities and Exposures
    - CVE provides a catalog of publicly disclosed cybersecurity vulnerabilities and establishes a common description for each vulnerability specified [47]. When a CVE report is generated, the submitting author will provide a description and public

reference for the finding and the database will generate a vulnerability identification number that will be used throughout the life of the finding.



CVE-ID

CVE-2019-5307 — Learn more at National Vulnerability Database (NVD)
• CVSS Severity Rating • Fix Information • Vulnerable Software Versions • SCAP Mappings • CPE Information

Description

Some Huawei 4G LTE devices, P30 versions before ELE-AL00 9.1.0.162(C01E160R1P12/C01E160R2P1) and P30 Pro versions before VOG-AL00 9.1.0.162(C01E160R1P12/C01E160R2P1), are exposed to a message replay vulnerability. For the sake of better compatibility, these devices implement a less strict check on the NAS message sequence number (SN), specifically NAS COUNT. As a result, an attacker can construct a rogue base station and replay the GUTI reallocation command message in certain conditions to tamper with GUTIs, or replay the Identity request message to obtain IMSIs. (Vulnerability ID: HWPSIRT-2019-04107)

References

Note: References are provided for the convenience of the reader to help distinguish between vulnerabilities. The list is not intended to be complete.

- CONFIRM:https://www.huawei.com/en/psirt/security-advisories/huawei-sa-20190529-01-replay-en

Assigning CNA

Huawei Technologies

Date Record Created

20190104 — Disclaimer: The record creation date may reflect when the CVE ID was allocated or reserved, and does not necessarily indicate when this vulnerability was discovered, shared with the affected vendor, publicly disclosed, or updated in CVE.

Figure 15. CVE Example [48].

- National Vulnerability Database
  - Although CVE reports provide an initial identification of vulnerabilities, those findings are augmented when they are fed into the National Vulnerability Database (NVD). The NVD will expand information about CVE entries by adding vulnerability severity, vulnerability mitigation recommendations, and overall impact ratings [49]. Another benefit of the NVD versus the CVE is the ability to search for vulnerabilities based on operating systems, vendor details, and vulnerability classes [50].

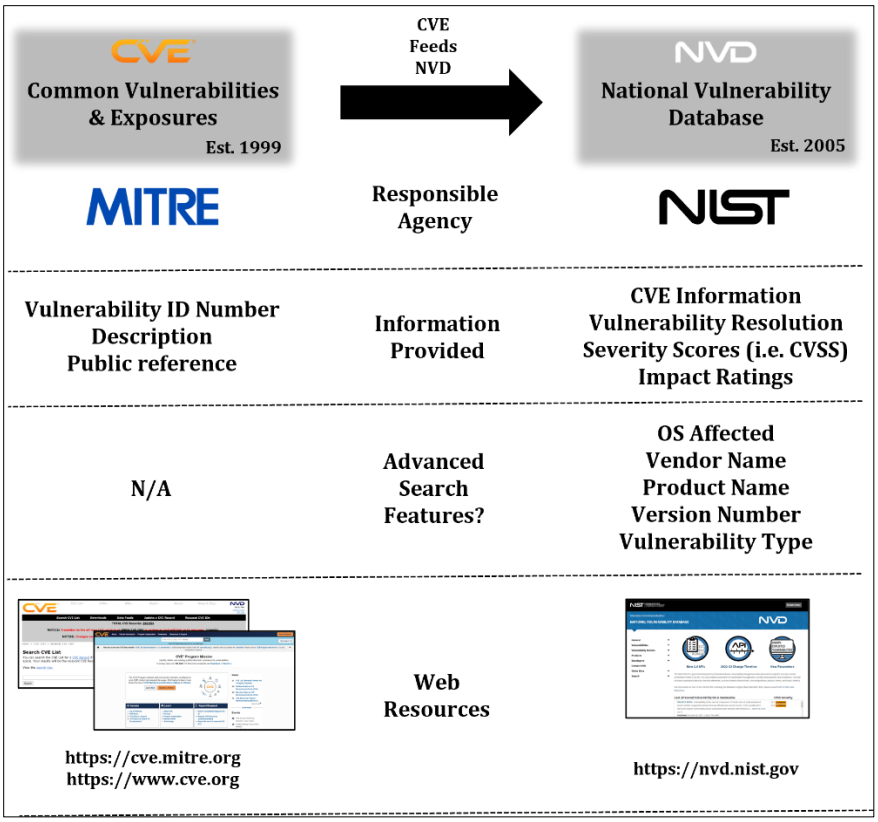


Figure 16. CVE / NVD Comparison.

- Rapid7 Vulnerability & Exploit Database
    - Outside of government and Federally Funded Research and Development Centers (FFRDC), such as MITRE, commercial security vendors provide additional resources to identify vulnerabilities. For example, Rapid7, which maintains the Metasploit Framework, provides a vulnerability and exploit database with “technical details for over 180,000 vulnerabilities and 4,000 exploits” [51] for security community use and an example of a vulnerability lookup can be seen in Figure 17.

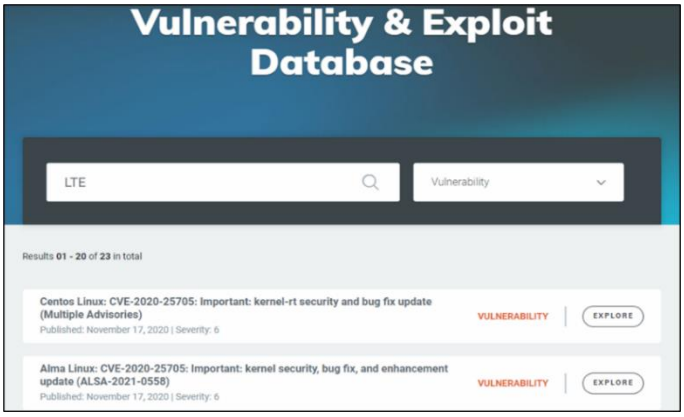


Figure 17. Rapid7 Vulnerability Lookup [55].

- CISA National Cyber Awareness System
    - The NCAS is a subscriber reporting system developed by the Department of Homeland Security (DHS) and the Cybersecurity & Infrastructure Security Agency (CISA) that provides information on current cyber activities through alerts, bulletins, and analysis reports to industry and government organizations [52]. This resource can provide vulnerability information for systems under test and daily reports can provide security vendors with additional tools during a vulnerability assessment.

3) Exploit Development

During a vulnerability assessment, a point may be reached where a direct exploit or proof-of-concept may be needed to demonstrate the validity of a vulnerability. Based on exploit criticality and system impact on system operation, a demonstration provides organizational leadership with confirmation of the findings. Although security testing organizations have personnel that can develop exploits internally for test events, there are several open-source resources that can be used as well.

- Exploit-DB

- Offensive Security, the creator of the Exploit Database, states "The Exploit Database is a CVE compliant archive of public exploits and corresponding vulnerable software, developed for use by penetration testers and vulnerability researchers. The database provides a repository for exploits and proof-of-concepts rather than advisories, making it a valuable resource for those who need actionable data right away" [53]. An example of a remote code execution exploit against an IP camera is shown in Figure 18 as an example of the resources available to security professionals conducting vulnerability assessments and penetration tests.

```
# Exploit Title: TP-Link Tapo c200 1.1.15 - Remote Code Execution (RCE)
# Date: 02/11/2022
# Exploit Author: hacefresko
# Vendor Homepage: https://www.tp-link.com/en/home-networking/cloud-camera/tapo-c200/
# Version: 1.1.15 and below
# Tested on: 1.1.11, 1.1.14 and 1.1.15
# CVE : CVE-2021-4045

# Write up of the vulnerability: https://www.hacefresko.com/posts/tp-link-tapo-c200-unauthenticated-rce

import requests, urllib3, sys, threading, os
urllib3.disable_warnings(urllib3.exceptions.InsecureRequestWarning)

PORT = 1337
REVERSE_SHELL = 'rm /tmp/f;mknod /tmp/f p;cat /tmp/f|/bin/sh -i 2>&1|nc %s %d >/tmp/f'
NC_COMMAND = 'nc -lv %d' % PORT # nc command to receive reverse shell (change it depending on your nc version)

if len(sys.argv) < 3:
    print("Usage: python3 pwnTapo.py <victim_ip> <attacker_ip>")
    exit()

victim = sys.argv[1]
attacker = sys.argv[2]

print("[+] Listening on %d" % PORT)
t = threading.Thread(target=os.system, args=(NC_COMMAND,))
t.start()

print("[+] Serving payload to %s\n" % victim)
url = "https://" + victim + ":443/"
json = {"method": "setLanguage", "params": {"payload": "';" + REVERSE_SHELL % (attacker, PORT) + ";'"}}
requests.post(url, json=json, verify=False)
```

Figure 18. ExploitDB IP Camera Exploit [54].

- GitHub Repositories
    - GitHub repositories can be used to find specifically crafted exploits against system vulnerabilities and have become an excellent resource for the security community.

4) Attack Trees

Prior to conducting a full-scale vulnerability assessment, developing a choreographed attack list can save time during a vulnerability assessment. Attack trees can provide a rigorous list of attack paths based on available system resources [55,56]. Attack trees can be either general or tailored where a general attack tree provides an exhaustive list of every kind of attack known by the security community. Tailored attack trees are modified versions of the general attack tree and only include those attacks relevant to the system under test.

d) Assessment Reporting Activities

The last phase of the vulnerability assessment testing methodology is vulnerability reporting where assessors provide a list of findings that classify how "bad" vulnerabilities within the system are. Although several vulnerability grading mechanisms exist, the Common Vulnerability Scoring System (CVSS) is frequently used to provide a prioritized list of vulnerability findings. First developed by the National Infrastructure Advisory Council (NIAC) in 2003, CVSS is now maintained by the National Institute of Standards and Technology (NIST). NIST defined CVSS as "an open framework for communicating the characteristics and severity of software vulnerabilities" and consists of three metric groups named Base, Temporal, and Environmental which are graded on a scale from 0 to 10 [57]. Unlike other vulnerability grading methods, CVSS uses seven objective metrics to evaluate vulnerability impact and exploitability and each of these is defined in Table 9.

Table 9. CVSS Metric Definitions [57].

Metric	Definition	Values
Confidentiality	Measures the ability to protect critical data within and between bus systems from unauthorized disclosure	No Impact – No loss of confidentiality Low Impact – Some loss of confidentiality, information loss is limited. High Impact - Total loss of confidentiality and all resources divulged to an attacker
Integrity	Measures the ability to protect critical data within and between bus systems from unauthorized modification	No Impact – No loss of integrity Low Impact – Possible data integrity breach, but not controlled by the attacker High Impact – Total loss of data integrity due to attacker actions
Availability	Measures the ability to provide services to authorized users and bus systems	No Impact – No loss of availability Low Impact – Some limited loss of system availability High Impact – Total loss of system availability due to attacker actions
Attack Vector	Measures the level of access an attacker requires to conduct an attack against bus systems and sub-systems	Physical – Physical access to system resources is required to conduct an attack Local – Keyboard or console access to a system is required to conduct an attack Adjacent – Limited adjacent network access is required to conduct an attack Network – System access through a network stack or internet connection
Attack Complexity	Measures the amount of time and effort that an attacker will need to successfully access bus systems	Low – An attacker has all resources needed to exploit a system High – An attacker requires additional resources and time to conduct an attack
User Interaction	Measures the involvement a legitimate user plays in assisting with a bus compromise	None – No user or user-controlled credential is required to conduct a system attack Required – A user or user-controlled credential is required to conduct a system attack
Privileges Required	Measures the level of privilege an attacker must possess before successfully exploiting a bus vulnerability	None – The attacker does not require privileged access to carry out an attack Low – The attacker requires user or local privileges to carry out an attack High – The attacker requires administrative privilege to to carry out an attack
Scope	Measures the degree to which compromises to one bus system effects other bus systems	Unchanged – An exploited vulnerability affects only a single system Changed – An exploited vulnerability affects systems outside of the compromised system

CVSS scoring is comprised of an impact and exploitability subscore that can then be mapped to a risk matrix. The Impact Subscore Base (ISC_{Base}), considers the impact of attacks against confidentiality, integrity, and availability and is scored as:

$$ISC_{Base} = 1 - [(1 - I_C) * (1 - I_I) * (1 - I_A)]$$

In the same manner, the Exploitability Subscore (E), considers the exploitability metrics of attack complexity, attack vector, user interaction, and privileges required:

$$E = 8.22 * AV * AC * PR * UI$$

Lastly, the scoring system also considers if an attack will affect connected systems or networks. If a connected system is impacted by the attack, this is defined as a scope change and is scored as:

$$ISC = 7.52 * [ISCBase - 0.029] - 3.25 * [ISCBase - 0.02]^{15}$$

If the scope is unchanged, then it is scored as:

$$ISC = 6.42 * ISC_{Base}$$

Taking the results of the CVSS grading method, Figure 19 plots notional systems of various vulnerabilities against impact and exploitability metrics. Systems with high impact and exploitability values, seen in the red area of the risk matrix, indicate systems that require higher priority for vulnerability mitigation and the final report will include a prioritized list of systems and recommended mitigation strategies.

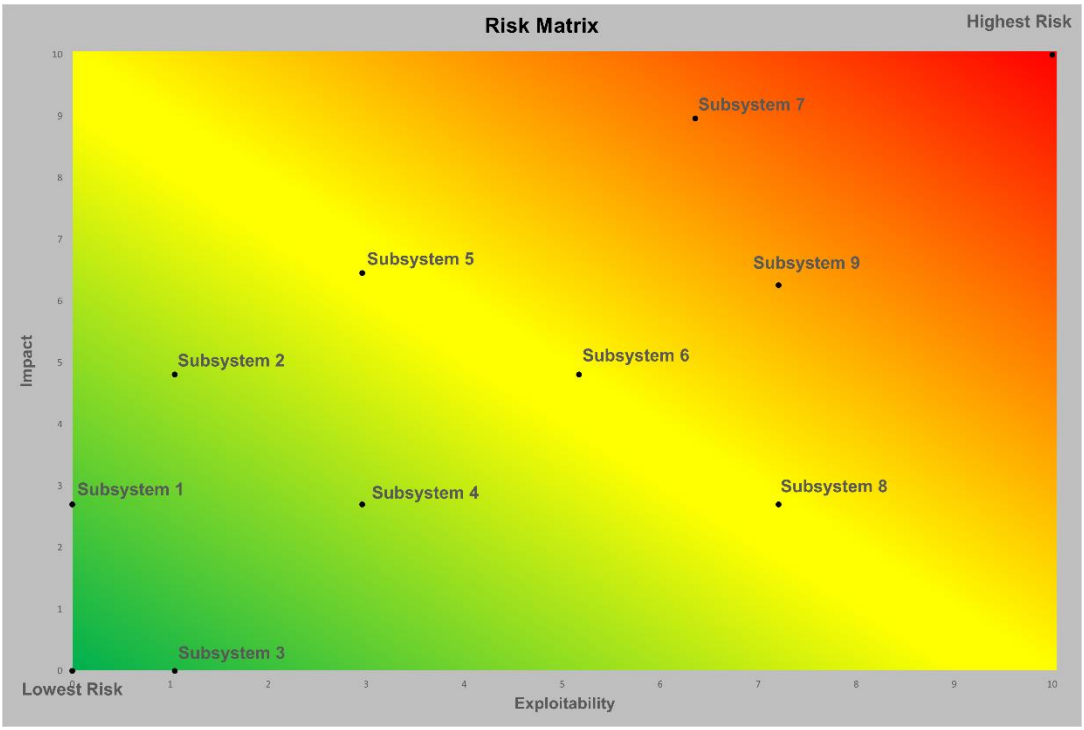


Figure 19. Example Vulnerability Assessment Risk Analysis.

Chapter 5

Vulnerability Assessment Testing Methodology for V2X-Enabled Environments

Now that we have introduced the traditional vulnerability assessment testing methodology, we will discuss the inherent limitations of the methodology regarding V2X environments, recommend a path forward to reformulate it to improve V2X vulnerability assessments and validate the proposed improved methodology with a case study.

I. Traditional Vulnerability Assessment Testing Methodology Limitations

Due to the complexity and dynamic nature of V2X protocols, hardware, and networks, the traditional vulnerability assessment testing methodology lacks the granularity needed to evaluate vulnerabilities across a V2X environment. The traditional vulnerability assessment testing methodology is geared towards enterprise networks and protocols such as IEEE 802.3 and IEEE 802.11a/n/g and does not address the needs of more robust V2X protocols such as CAN, IEEE 802.11p, IEEE 1609.3, and IEEE 1609.4.

One of the largest gaps observed when overlaying the traditional vulnerability assessment testing methodology against V2X environments is that it does not provide a mechanism to evaluate the dynamic changes that systems moving through a V2X environment will encounter. Unlike an enterprise environment in which connected systems are essentially static during their standard

mode of operation, this is not the case with V2X environments since V2V, V2I, and V2P will be in motion and require near real-time network updates. This gap within the traditional testing methodology could be addressed by establishing a traffic scenario framework that can model vehicles, infrastructure, and pedestrians interacting in a V2X environment along with the associated V2X network protocols. It is insufficient to test individual V2X components during a vulnerability assessment; rather one needs to test the impact that an attack on an individual component and protocol could potentially have on the entire V2X environment.

Another drawback of applying the traditional vulnerability assessment testing methodology against V2X environments is that legacy threat models are immature and do not provide the level of complexity needed to effectively stress systems connected to V2X networks. Many traditional threat models do not provide the technical capabilities needed to interact with V2X protocols and a model that integrates these into the testing methodology must be included. To address this gap in the traditional testing methodology, a reconfigurable threat platform that mimics attacker methods against V2X systems and networks should be created. A capability of this kind will allow security assessors to conduct a wide variety of realistic attack vectors against V2X systems.

II. Case Studies Supporting V2X Vulnerability Assessment Testing Methodology[2][3]

During this research effort, two vulnerability assessment projects were conducted to aid in the identification of vulnerabilities in vehicular systems with one focusing on identifying vulnerabilities within mass transit systems while the other aimed to establish a reconfigurable lab

[2] The vulnerability assessment testing methodology was used in support of prime contract number 22BRODBECK

[3] A V2X testing lab was built in support of contract number SR20192324-C2

environment for the testing of V2X enabled systems. Although the objectives of these projects were different, the lessons learned from each helped to identify gaps in the existing vulnerability assessment testing methodology and to establish capability improvements through the development of a reconfigurable V2X lab environment.

1) Traffic Scenario Generation

During the vulnerability assessment of mass transit systems, it became apparent that the Traditional Vulnerability Assessment Testing Methodology did not provide a capability to model vehicular traffic scenarios or the underlying V2X protocols that are seen during these assessments. As a result, the Vulnerability Testing Workflow was augmented as shown in Figure 20 by adding a traffic scenario generation between the selection of a V2X system and identification of existing security controls.

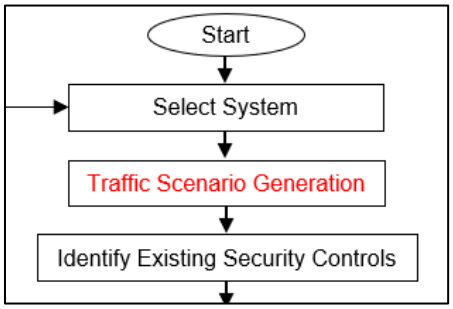


Figure 20. Traffic Scenario Generation Capability Improvement.

From experience gained during mass transit vehicle testing we recognized that traffic scenarios must be established after a V2X system is selected to ensure it is immersed in a realistic operational scenario. By selecting the correct V2X system and placing it into a realistic operational context can we start to evaluate the threats against that system.

To address the need for a traffic scenario capability, several V2X software and hardware packages were evaluated to determine if they could effectively be integrated into the traditional

vulnerability assessment testing methodology. Vector's CANoe framework was found to provide the greatest coverage for V2X testing and is described as a "comprehensive software tool for development, test and analysis of individual ECUs and entire ECU networks that supports network designers, development and test engineers throughout the entire development process" [58]. The use of the term simulation can be misleading, because CANoe provides not only simulated traffic and protocol events but can also be configured to integrate physical V2X systems into the vulnerability assessment process (hardware-in-the-loop).

In addition to the simulation software packages, the Vector VN4610 software defined radio (SDR) shown in Figure 21, is "a special solution for IEEE 802.11p and CAN based applications and extends the CANoe test tool and supports receiving and transmitting of IEEE 802.11p frames which are used for the implementation of V2X applications [59]."

Figure 21. Vector VN4610 V2X Software Defined Radio [60].

V2X traffic simulation provides an effective method of configuring the systems, protocols, and networks in the same way that they will be evaluated during physical testing during a vulnerability assessment. An additional benefit of using both software and hardware during a V2X vulnerability assessment is that simulations can be executed iteratively with differing traffic scenarios, network bandwidth, protocol utilization, and attack vectors. An example of a

simulation of this kind is shown in Figure 22 where two V2X-configured systems are communicating basic safety message (BSM) information on a specified course. Prior to broadcasting messages from physical VN4610 SDR's, the CANoe V2X application generated safety messages during the simulation. The simulation also provided extensive selection of transportation scenarios, vehicle databases, selection of network nodes and certificate repositories, and configuration of physical radios during reception and transmission between radios.

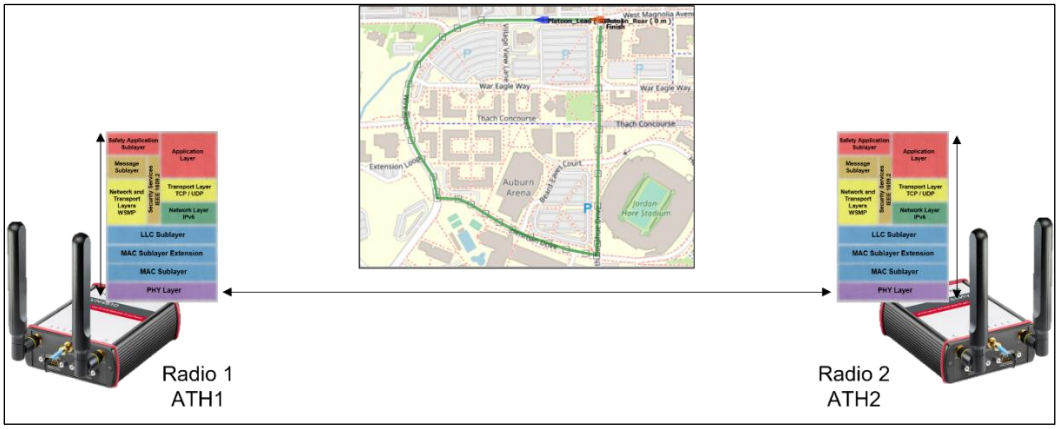


Figure 22. Traffic Simulation Overview.

In addition to configuration and deployment of traffic scenarios, the simulation framework also provides the ability to configure protocol definitions. For example, Figure 23 illustrates how an SAE J2735 Message Sublayer can be defined during a traffic scenario.

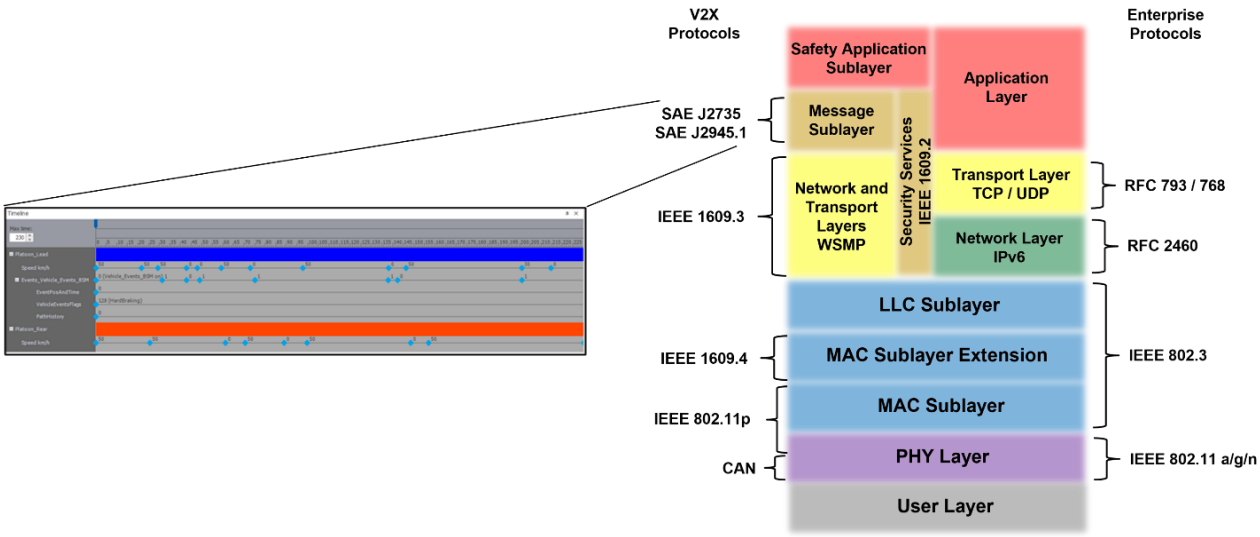


Figure 23. Basic Safety Message Configuration.

An expanded timeline view in Figure 24 shows each of the parameters specified during the traffic simulation and includes a vehicle conducting a hard braking event at different times and the associated response from a trail vehicle.

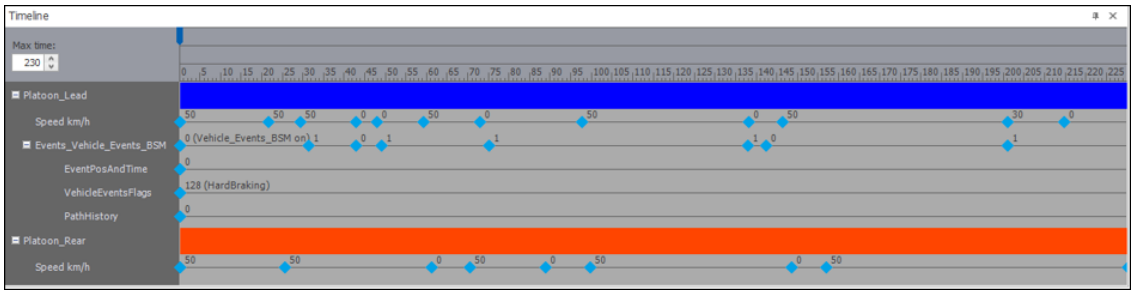


Figure 24. Basic Safety Message Parameters.

As a result from lessons learned during conduct of the vulnerability assessment against mass transit systems and deployment of the V2X lab testing environment we learned that traffic and protocol simulation can be effectively deployed.

2) V2X Threat Emulation

An additional capability improvement needed within the vulnerability assessment workflow for V2X vulnerability assessments is the creation of a reconfigurable threat model that can assist with deploying a realistic set of attack vectors against V2X systems under test. As shown in Figure 25, this capability is placed after threats are identified and before attacks are deployed during the test event.

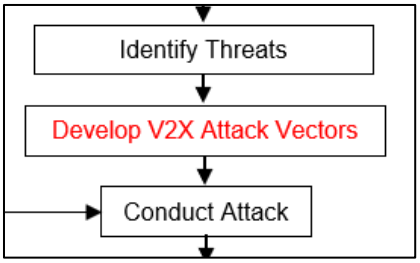


Figure 25. Deploying V2X Attack Vectors within a Reconfigurable Lab Environment.

To realize this capability, we took an iterative approach to build different threat models into the testing lab which included anticipated threat activities within a V2X environment including 1) Traffic Sniffing, 2) Man-In-The-Middle (MITM), and 3) Message Injection. Although research conducted by several groups identified attack vectors against CAN, IEEE 802.11p, and LTE-V2X, it was interesting to note the lack of reconfigurable platforms to generate V2X-specific test events, so we utilized several commercial and open-source tools to create an environment to model threat actor capabilities including signal sniffing, location tracking, traffic replay, and signal flooding attacks.

To create a traffic sniffing capability for a V2X environment, it was necessary to understand the underlying communication protocols and the technical details about signal reception and transmission. Using the technical information provided about IEEE 802.11p and higher layer protocols in previous chapters, we created a software defined radio application to capture transmissions between legitimate V2X nodes. As illustrated in Figure 26, a traffic scenario

generated with the CANoe application was sent between two VN4610 systems while a notional attacker used an open source HackRF system to collect and analyze signals sent between them.

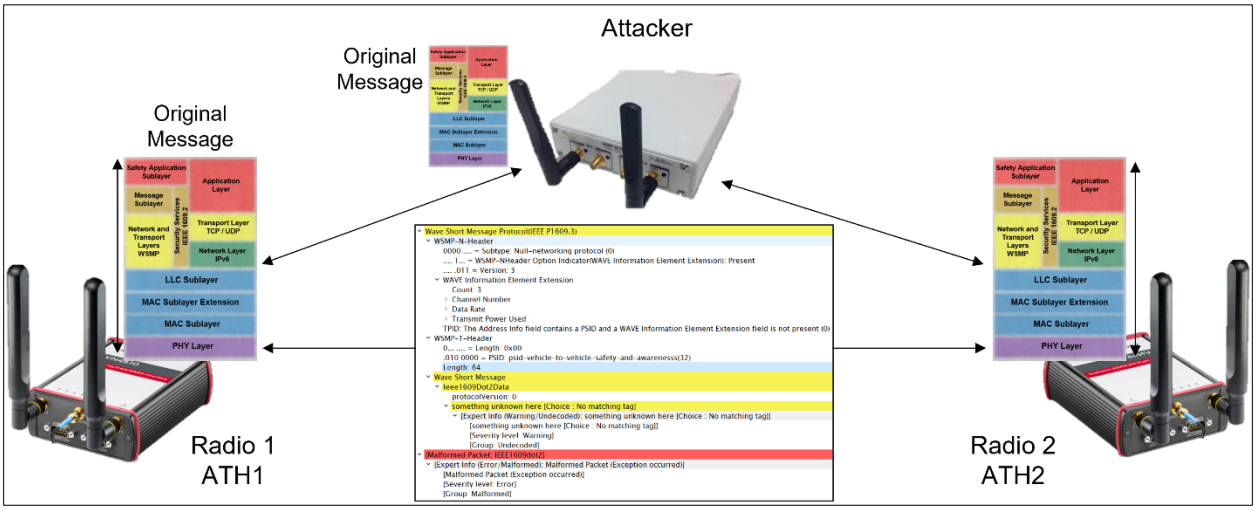


Figure 26. V2X MITM Traffic Collector.

Utilizing two open-source GitHub repositories, gr-foo and gr-ieee802-11 [61,62], GNURadio flowgraphs were used to capture, store, and replay transmissions intercepted between VN4610 radios and provided information about protocol structure and message timing information. Figure 27 shows the GNURadio flowgraph for the collection of 802.11p signals [63].

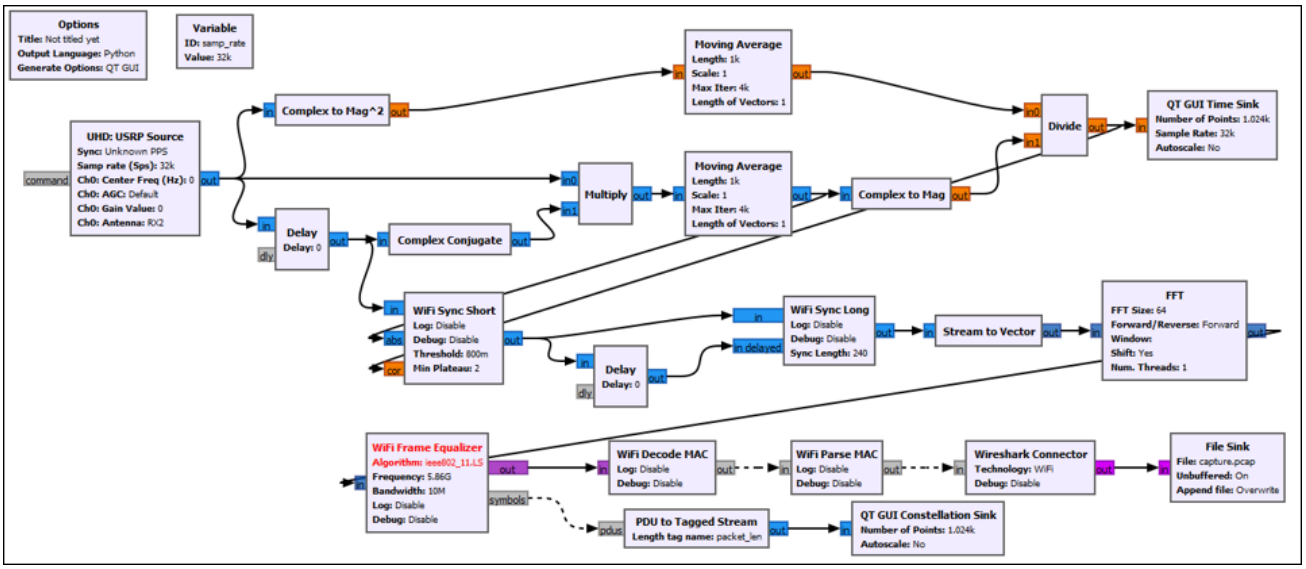


Figure 27. GNURadio Traffic Capture Flowgraph.

The iterative nature of the V2X testing environment led to installation of additional SDR's for traffic replay attacks. Due to reception and re-transmission considerations, it was realized that deployment of multiple SDR's would provide a multi-channel replay capability that could effectively test numerous channels within a V2X environment and is particularly helpful when testing the security of IEEE 802.11p, which has six service channels.

Figure 28 provides an overview of the combined traffic generation and threat model capability developed during this effort. The first part of the demonstration shows basic safety message information being sent between two legitimate V2X systems using VN4610 SDR's. At the same time, a notional attacker using an Ettus X310 SDR for collection and an Ettus B210 SDR for replay attempts to disrupt a common operating picture. To demonstrate the effectiveness of this attack an initial capture of the traffic scenario is run and shows correct breaking information and vehicle positions, but upon execution of the replay breaking information and vehicle positions are unreadable.

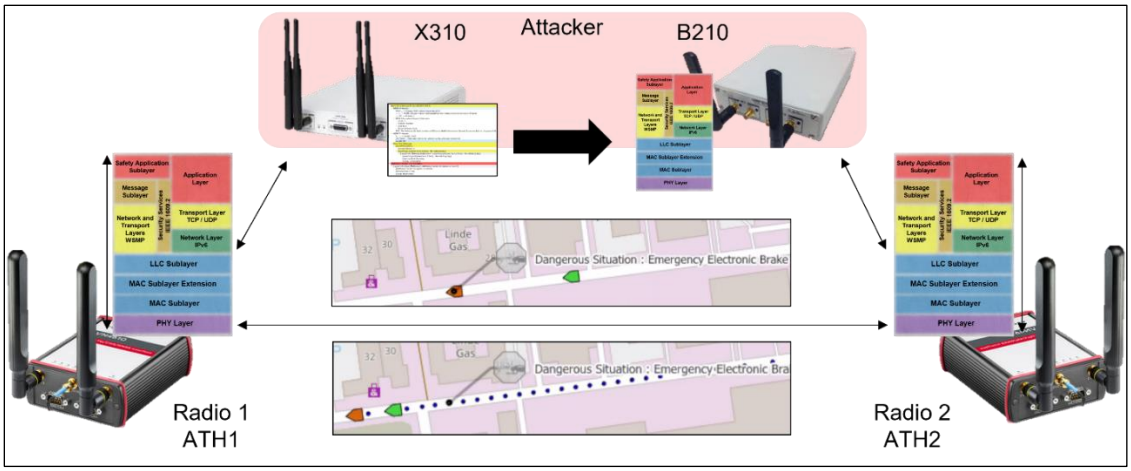


Figure 28. V2X Replay Attack.

With a capability of this kind, a security assessment team could generate hundreds of simulations and model realistic threats before physically deploying V2V, V2I, or V2P systems in a real-world context.

III. Benefits of the V2X Vulnerability Assessment Testing Methodology

The development and application of the V2X vulnerability assessment testing methodology resulted in several identified benefits:

1) Traffic and Protocol Prioritization

Integration of a simulation capability into the traditional vulnerability assessment testing methodology provides security assessors an opportunity to set conditions experienced during a system attack. Iterative traffic scenarios result in a prioritized list of traffic events that may be more likely to result in operational and security impacts and gives security assessors a better understanding of what to expect when a physical test event occurs.

2) Recognition of Advanced Training for Security Assessors

In addition to the technical improvements brought about by the updated testing methodology, an additional benefit is that it reveals a skills gap between the existing testing capability of security personnel and the skills required to conduct a V2X-focused vulnerability assessment. Although the traditional vulnerability assessment testing methodology has some well-known tools and techniques for enterprise systems and networks, they are not the same skills needed by security assessors on V2X vulnerability assessments. Consequently, additional training and skill development activities will be needed for security professionals to deploy the tools and techniques in V2X environments.

3) Physical testing validates simulation

The results of simulation tests must be validated by physical testing events to ensure that simulated traffic scenarios and protocol deployment can be realized in the real world. For example, a simulation may indicate that a denial of service against vehicles or infrastructure is possible, but when applying physical complexities of signal propagation, the simulated condition may not validate. It may be discovered that the test conducted provides only a limited attack capability and does not result in a complete denial of service condition.

4) A V2X-centric CVSS scoring mechanism

Utilization of the CVSS scoring system for traditional vulnerability assessments provides an objective means to specify the impact and exploitability of an information system. When applying the CVSS against V2X environments we recognized that a critical feature missing in the exportability evaluation is the lack of a mission criticality metric. Not every system communicating over a V2X network will have the same level of criticality necessary for operational and mission success, so there needs to be a metric that evaluates the criticality of each V2X endpoint.

Chapter 6

Conclusions and Future Work

The traditional vulnerability assessment testing methodology is not designed for testing of advanced communication protocols such as CAN, IEEE 802.11p, IEEE 1609.3, and IEEE 1609.4 and as a result, we proposed an improved vulnerability assessment testing methodology that includes both a traffic scenario generation and threat emulation capability. The two case studies conducted in support of this effort demonstrated the potential efficacy of the proposed methodology.

I. Conclusions

During this research effort we answered the questions posed in the introduction about how to extend the traditional vulnerability assessment testing methodology for use in V2X environments:

- What are the underlying protocols within V2X and why can't existing vulnerability assessment methods be applied to them?

Existing vulnerability assessment methodologies are geared towards enterprise networks and protocols such as IEEE 802.3 and IEEE 802.11a/n/g and do not address the needs of V2X-centric protocols such as CAN, IEEE 802.11p, IEEE 1609.3, and IEEE 1609.4. As a result, the use of traditional vulnerability assessment methods will not provide adequate tools to effectively evaluate V2X environments and need to be modified appropriately. Additionally, assessment teams conducting V2X-specific vulnerability assessments need to learn more about the underlying protocols that support V2X environments.

- What testing capabilities are needed to generate attacks against V2X systems and environments?

Although research conducted by several groups identified attack vectors against CAN, IEEE 802.11p, and LTE-V2X it was interesting to note the lack of reconfigurable platforms to generate V2X-specific test events. During this effort we utilized commercial and open-source tools to create a V2X-focused testing platform that provided a method of generating traffic scenarios while also providing an environment to model threat actor capabilities including signal sniffing, location tracking, traffic replay, and signal flooding attacks which are highlighted in green in Table 10.

Table 10. V2X RF Replicated Attack Vectors [42].

Security Principle	Threat	IEEE 802.11p		LTE-V2X			
				Cellular-Based		D2D-Based	
		External	Internal	External	Internal	External	Internal
Confidentiality	Sniffing	✓	×	✓	×	×	×
	Location Tracking	×	×	×	×	×	×
Integrity	Message Manipulation	✓	✓	✓	×	✓	×
	Message Injection	✓	×	✓	×	✓	×
	Replay Attack	✓	✓	✓	✓	✓	✓
	GPS Spoof Attack	×	×	×	×	×	×
Availability	Blackhole Attack	✓	×	✓	×	×	×
	Signal Flood	✓	×	✓	×	×	×
	Signal Jamming	×	×	×	×	×	×
	Coalition Attack	✓	×	✓	×	✓	×
Authenticity	Certificate Replication	✓	✓	✓	✓	×	×
	Sybil Attack	✓	✓	✓	✓	×	×
	Masquerading	✓	✓	✓	✓	×	×
Non-Reputiation	User Distrust	✓	✓	✓	✓	×	×

- What modifications to traditional vulnerability assessment testing methodology must be made for V2X vulnerability assessments?

Figure 27 highlights the recommended improvements necessary to conduct a more effective vulnerability assessment for V2X environments in the future. Although the traditional vulnerability assessment testing methodology provides general considerations for vulnerability assessments, we have identified several capability gaps that need to be addressed.

First, the V2X vulnerability assessment methodology requires a traffic simulation and stimulation framework that provides a realistic interface for the vehicles, infrastructure, and pedestrians that will participate in a V2X test event. It is insufficient to test individual V2X components during a vulnerability assessment; additionally, the impact that an attack on an individual component could potentially have on the V2X environment must be tested.

In addition, a reconfigurable threat platform is necessary to properly mimic attacker methods against V2X systems and networks. In this research effort we demonstrated how open-source hardware and software could be used to conduct both MITM and replay attacks against legitimate V2X systems.

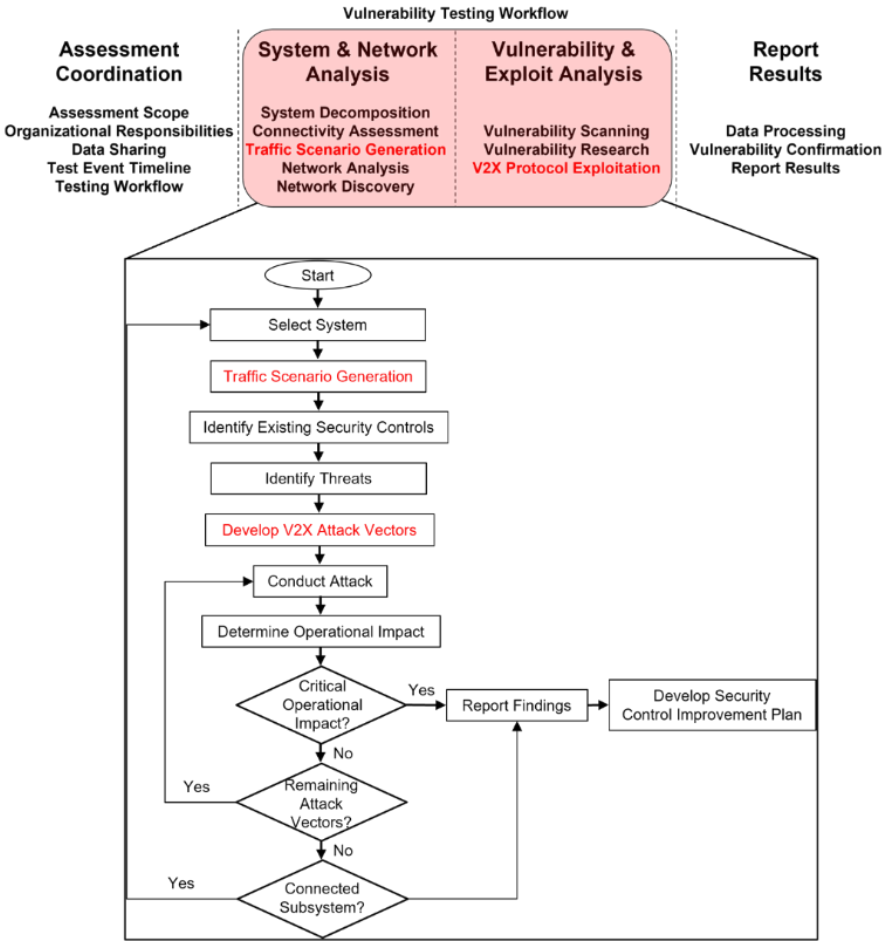


Figure 27. Improved Vulnerability Assessment Methodology for V2X Environments.

Future Work

The results of this research effort point to several follow-on investigations:

1) Testing the efficacy of the proposed testing methodology on broader communication technologies

A question that was raised during this investigation regarded the application of the proposed vulnerability assessment testing methodology against non-standard technologies such as Industrial Control System (ICS) and Supervisory Control and Data Acquisition (SCADA)

systems. A follow-on effort could investigate the ability to design a simulation capability to test systems using the proposed methodology.

2) Authenticity Attack Vectors

Attack vectors against authenticity were not addressed in this effort but are a natural progression for future research efforts. Since V2X transmissions are not encrypted due to overhead considerations and because the V2X community depends heavily on the integrity of messages based on certificate services, attacks such as certificate replication and sybil and masquerading attacks will continue to be a cause for concern. The next logical demonstration would be to show how the trust model introduced in Figure 12 can be manipulated. One of the benefits of the CANoe framework, which was used in this research effort, is that it provides a certificate generation capability so that certificate service security can be tested.

3) Jamming Attack Vectors

In addition to adding authenticity attacks into the threat model, a follow-on effort would be to integrate additional SDR's into the test environment that would allow for signal jamming, blackhole, and coalition attacks. One of the challenges for the blackhole and coalition attacks is the dynamic nature of vehicles, infrastructure, and pedestrians moving throughout a V2X environment. Design and deployment of a simulation for these availability attacks would need to be completed prior to physical testing during the vulnerability assessment.

4) Validation of Findings

Although the initial results of this effort indicate the efficacy of the proposed vulnerability assessment testing methodology, additional real-world testing will be necessary to validate the results. A possible follow-on effort would include a full-scale simulation event followed by a

physical testing event. A side-by-side comparison of assumptions, tests, and results would be necessary to validate the efficacy of the testing methodology.

5) CVSS Improvements for V2X Systems

Another observation that was made during this effort, albeit non-technical, was that the vulnerability scoring system that was used to evaluate systems during a vulnerability assessment does not account for the mission criticality of systems it evaluates. Although, from an exploitability standpoint, the CVSS evaluates attack vector, attack complexity, user interaction, and privileges required, it does not prioritize components based on their operational or mission importance. Without an objective measure of the importance of components within a system, it will be difficult to provide an effective prioritization for remediating vulnerabilities. A future research effort looking into the reconfiguration of the vulnerability scoring system could provide great value to the security community. Additional research in this area should consider how to best include a criticality metric into the CVSS scoring system when applied to V2X environments.

The expansion of V2X technologies will continue to grow as a greater percentage of vehicles are transitioned over to autonomous and driver aided capabilities. As a result, we will need to expand our testing capabilities for vulnerabilities in these environments and the proposed vulnerability assessment methodology will be step in that direction.

References

[1] Q, A. (2022, April 21). Road Safety Facts. Association for Safe International Road Travel. https://www.asirt.org/safe-travel/road-safety-facts/

[2] Intelligent Transportation Systems Using IEEE 802.11p, Application Note. (n.d.). https://scdn.rohde-schwarz.com/ur/pws/dl_downloads/dl_application/application_notes/1ma152/1MA152_5e_ITS_using_802_11p.pdf

[3] Lee, J., Lee, D., Park, Y., Lee, S., & Ha, T. (2019, October). Autonomous vehicles can be shared, but a feeling of ownership is important: Examination of the influential factors for intention to use autonomous vehicles. https://linkinghub.elsevier.com/retrieve/pii/S0968090X19301895

[4] El-Rewini, Z., Sadatsharan, K., Flora, D., Plathottam, J., & Ranganathana, P. (2020, June). Cybersecurity Challenges in Vehicular Communications. https://doi.org/10.1016/j.vehcom.2019.100214

[5] Alnasser, A., Sun, H., & Jiang, J. (2019, March 14). Cyber Security Challenges and Solutions for V2X Communications: A Survey. https://doi.org/10.1016/j.comnet.2018.12.018

[6] A Survey on Security Attacks and Defense Techniques for Connected and Autonomous Vehicles, https://doi.org/10.1016/j.cose.2021.102269

[7] Physical-Layer Security and Privacy for Vehicle-to-Everything. (2019, October 1). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/8875720/

[8] Morgan, L. (2020, June 24). What Impact Will The Full Implementation of 5G Technology Have on The Automotive Industry? Mobile Tutorials, Satellite & IPTV Tutorials, PC & Automotive Tips. https://www.lemmymorgan.com/5g-technology-and-the-automotive-industry/

[9] ISO - 35.100 - Open systems interconnection (OSI). (n.d.). Retrieved October 24, 2022, from https://www.iso.org/ics/35.100/x/

[10] Dedicated Short-Range Communications (DSRC) Standards in the United States. (2011, July 1). IEEE Journals & Magazine | IEEE Xplore. https://ieeexplore.ieee.org/document/5888501/

[11] Wikipedia contributors. (2022, September 15). Layer 8. Wikipedia. https://en.wikipedia.org/wiki/Layer_8

[12] J1939_200710: Recommended Practice for a Serial Control and Communications Vehicle Network - SAE International. (n.d.). Retrieved October 24, 2022, from https://www.sae.org/standards/content/j1939_200710

[13] SAE J1939: The Ultimate Guide (2022). (n.d.). Retrieved October 24, 2022, from https://www.autopi.io/blog/j1939-explained/

[14] Parikh, B. (2022, June 1). CAN protocol: Understanding the controller area network. Engineers Garage. https://www.engineersgarage.com/can-protocol-understanding-the-controller-area-network-protocol/

[15] SAE J1939 Standards Collection. (n.d.). Retrieved October 24, 2022, from https://www.sae.org/standardsdev/groundvehicle/j1939a.htm

[16] Stoltzfus, J. (2020, August 3). Your Car, Your Computer: ECUs and the Controller Area Network. Techopedia.com. https://www.techopedia.com/your-car-your-computer-ecus-and-the-controller-area-network/2/32218

[17] Rosu, C. (2015, September 8). EMC FLEX BLOG | CAN bus (Controller Area Network). http://www.flexautomotive.net/EMCFLEXBLOG/post/2015/09/08/can-bus-for-controller-area-network

[18] Hacking, T. (n.d.). CAN ID Explanation. Tractor Hacking. Retrieved October 24, 2022, from https://tractorhacking.github.io/IdExplanation/

[19] Controller Area Network (CAN Bus) Tutorial - Message Frame Format. (2021, June 16). Copperhill. https://copperhilltech.com/blog/controller-area-network-can-bus-tutorial-message-frame-format/

[20] Bolboceanu, V. (2021, January 3). What is OBD-II and how do we use it ? Mura Car Accessories. https://mca.electricmura.ro/en/blog-what-is-obd-ii/

[21] Ocstar SAE J1939 Type1 to J1939 Type2 Adapter Male to Female 9 Pin Black Connector to Green Truck Cable Diagnostic Cable GPS Trackers and Scan Tools : Amazon.in: Car & Motorbike. (n.d.). Retrieved October 24, 2022, from https://www.amazon.in/Ocstar-Adapter-Connector-Diagnostic-Trackers/dp/B07DC6C72B

[22] IEEE SA. (n.d.-a). IEEE SA - IEEE 802.11p-2010. IEEE Standards Association. Retrieved October 24, 2022, from https://standards.ieee.org/ieee/802.11p/3953/

[23] WLAN 802.11a vs 802.11p-Difference between 802.11a,802.11p. (n.d.). Retrieved October 24, 2022, from https://www.rfwireless-world.com/Terminology/WLAN-802-11a-versus-802-11p.html

[24] Sahoo, P. K. (n.d.). SVANET: A Smart Vehicular Ad Hoc Network for Efficient Data Transmission with Wireless Sensors. MDPI. Retrieved October 24, 2022, from https://www.mdpi.com/1424-8220/14/12/22230

[25] IEEE SA. (n.d.-c). IEEE SA - IEEE 1609.4-2016. IEEE Standards Association. Retrieved October 24, 2022, from https://standards.ieee.org/ieee/1609.4/6183/

[26] Program, T. I. S. (n.d.). ITS Standards Program | Fact Sheets | ITS Standards Fact Sheets. Retrieved October 24, 2022, from https://www.standards.its.dot.gov/Factsheets/Factsheet/80

[27] Dedicated Short-Range Communications (DSRC) Standards in the United States. (2011, July 1). IEEE Journals & Magazine | IEEE Xplore. http://dx.doi.org/10.1109/JPROC.2011.2132790

[28] Bouk, S., Kim, G., Ahmed, S., & Kim, D. (2015, February). Hybrid Adaptive Beaconing in Vehicular Ad Hoc Networks: A Survey. http://dx.doi.org/10.1155/2015/390360

[29] A Cooperative Communication Protocol for QoS Provisioning in IEEE 802.11p/Wave Vehicular Networks, http://dx.doi.org/10.3390/s18113622

[30] Hedges, C. (2007, October 29). Overview and Use of SAE J2735 Message Sets for Commercial Vehicles. https://trid.trb.org/view/1816482

[31] Intelligent Transportation Systems - Vehicle to Infrastructure (V2I) Deployment Guidance and Resources. (n.d.). Retrieved October 24, 2022, from https://www.its.dot.gov/v2i/

[32] Knight, A. V. (2018, June 15). The Hitchhiker's Guide to Hacking Connected Cars: ECUs Demystified. https://www.linkedin.com/pulse/hitchhikers-guide-hacking-connected-cars-ecus-alissa-valentina-knight

[33] Squires, S., & Shade, S. (2015, December). People, the Weak Link in Cyber-security: Can Ethnography Bridge the Gap? https://doi.org/10.1111/1559-8918.2015.01039

[34] Mittal, S. (2016, April 1). Understanding the Human Dimension of Cyber Security. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=2975924

[35] Salahdine, F. (n.d.). Social Engineering Attacks: A Survey. MDPI. Retrieved October 24, 2022, from https://www.mdpi.com/1999-5903/11/4/89

[36] Abbas, S. G. (n.d.). Identifying and Mitigating Phishing Attack Threats in IoT Use Cases Using a Threat Modelling Approach. MDPI. Retrieved October 24, 2022, from https://www.mdpi.com/1424-8220/21/14/4816

[37] Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges. (2017, November 1). IEEE Journals & Magazine | IEEE Xplore. https://doi.org/10.1109/TITS.2017.2665968

[38] Al-Sabaawi, A., Al-Dulaimi, K., Foo, E., & Alazab, M. (2021). Addressing Malware Attacks on Connected and Autonomous Vehicles: Recent Techniques and Challenges. SpringerLink. https://doi.org/10.1007/978-3-030-62582-5_4

[39] Vehicle Security: A Survey of Security Issues and Vulnerabilities, Malware Attacks and Defenses. (2021). IEEE Journals & Magazine | IEEE Xplore. https://doi.org/10.1109/ACCESS.2021.3130495

[40] Koscher, K., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Czeskis, A., Roesner, F., & Kohno, T. (2011). Comprehensive Experimental Analyses of Automotive Attack Surfaces. http://www.autosec.org/pubs/cars-usenixsec2011.pdf

[41] Experimental Security Analysis of a Modern Automobile. (2010, May 1). IEEE Conference Publication | IEEE Xplore. https://doi.org/10.1109/SP.2010.34

[42] Alnasser, A., Sun, H., & Jiang, J. (2019, March 14). Cyber Security Challenges and Solutions for V2X Communications: A Survey. https://linkinghub.elsevier.com/retrieve/pii/S1389128618306157

[43] Toubi, A., & Mazri, T. (2020, July). Attacks against security in the vehicular network and the impact of Sybil attack and Blackhole attack on the vehicular network performances: average Throughput as a study of case. https://www.researchgate.net/publication/344875278

[44] P2DAP — Sybil Attacks Detection in Vehicular Ad Hoc Networks. (2011, March 1). IEEE Journals & Magazine | IEEE Xplore. https://doi.org/10.1109/JSAC.2011.110308

[45] GEMALTO V2X. (2020, March 4). IoT Automotive News. https://iot-automotive.news/gemalto-v2x/

[46] Intelligent Transport Systems (ITS); Security; Pre-standardization study on pseudonym change management. (2018, April). https://www.etsi.org/deliver/etsi_tr/103400_103499/103415/01.01.01_60/tr_103415v010101p.pdf

[47] CVE - CVE. (n.d.). Retrieved October 24, 2022, from https://cve.mitre.org

[48] CVE - CVE-2019-5307. (n.d.). Retrieved October 24, 2022, from https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2019-5307

[49] CVE - CVE and NVD Relationship. (n.d.). Retrieved October 24, 2022, from https://cve.mitre.org/about/cve_and_nvd_relationship.html

[50] NVD - Home. (n.d.). Retrieved October 24, 2022, from https://nvd.nist.gov

[51] Vulnerability & Exploit Database. (n.d.). Rapid7. Retrieved October 24, 2022, from https://www.rapid7.com/db/?type=nexpose

[52] National Cyber Awareness System | CISA. (n.d.). Retrieved October 24, 2022, from https://www.cisa.gov/uscert/ncas

[53] Offensive Security's Exploit Database Archive. (n.d.). Retrieved October 24, 2022, from https://www.exploit-db.com

[54] TP-Link Tapo c200 1.1.15 - Remote Code Execution (RCE). (2022, September 23). Exploit Database. https://www.exploit-db.com/exploits/51017

[55] Salter, C., Saydjari, O., Schneier, B., & Wallner, J. (1998, January). Toward a secure system engineering methodolgy. https://doi.org/10.1145/310889.310900

[56] Lallie, H., Debattista, K., & Bal, J. (2020, February). A review of attack graph and attack tree visual syntax in cyber security. https://doi.org/10.1016/j.cosrev.2019.100219

[57] NVD - Vulnerability Metrics. (n.d.). Retrieved October 24, 2022, from https://nvd.nist.gov/vuln-metrics/cvss

[58] Vector CANoe Software. (n.d.). Retrieved October 24, 2022, from https://www.vector.com/int/en/products/products-a-z/software/canoe

[59] Vector Gallery. (n.d.). Retrieved October 24, 2022, from https://www.vector.com/int/en/products/products-a-z/software/canoe/#c60442

[60] Vector VN4610 Software Defined Radio. (n.d.). Retrieved October 24, 2022, from https://www.vector.com/us/en/products/products-a-z/hardware/network-interfaces/vn4610

[61] GitHub - bastibl/gr-foo: Some GNU Radio blocks that I use. (n.d.). GitHub. Retrieved October 24, 2022, from https://github.com/bastibl/gr-foo

[62] GitHub - bastibl/gr-ieee802-11: IEEE 802.11 a/g/p Transceiver. (n.d.). GitHub. Retrieved October 24, 2022, from https://github.com/bastibl/gr-ieee802-11

[63] Bloessl, B., & Segata, M. (2013, September). *Decoding IEEE 802.11a/g/p OFDM in software using GNU radio*. https://doi.org/10.1145/2500423.2505300