System Requirements Specification

for

EcoCar Mobility Challenge – Cybersecurity Team

Version 1.0 approved

Prepared by Nicholas Moline, Casey Ranft, Dustin Cribbs, Andrew Henderson, Joshua Palmer, Robert Duke, Jayson Tinsley, Kevin Pepin

Embry-Riddle Aeronautical University
ECSSE Department Senior Design
Daytona Beach, FL

2/13/2020

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Casey Ranft	2/12/2020	Document Creation	1.0
Robert Duke	2/13/2020	Grammar/Editing	1.1
Everyone	2/13/2020	Correcting/Adding in sections	1.2

1. Introduction

1.1 Purpose

To create a preliminary design and implementation of Cybersecurity for the EcoCar Chevy Blazer. This entails detecting when an attack is happening and early design or testing of V2X/V2V systems. When an attack is detected, the Decision-Making team will be notified so that proper action can be taken. The Cybersecurity team is also tasked with securing communication lines with both physical and virtual means. A hardware-in-the-loop encryption method will be used to obscure the information crossing the vehicle communication lines.

1.2 Document Conventions

Nested requirements inherit the priority of their parent requirements and are only meant to represent sub-requirements necessary to fulfill a parent requirement.

1.3 Intended Audience and Reading Suggestions

This document is meant for EcoCar members and customers to understand Cybersecurity's responsibilities in this project. Readers are expected to be familiar with the terms, concepts, and technologies, associated with Cybersecurity, autonomous vehicles, and EcoCar which will all be used throughout. This document will also serve as an overview of possible attacks on this autonomous car system. It should be read in order to fully understand Cybersecurity's role.

1.4 Product Scope

The software being used for this product is ROS (Robot Operating System), MATLAB, and Linux.

- ROS is the system being used by EcoCar 4 to control all the sensors and the messaging channels upon which the sub-systems communicate.
    - Data is packaged within a .bag file and will have to be processed on arrival
    - ROS currently runs on Unix-based platforms
- MATLAB Simulink is a graphical programming environment that is being used to simulate the car in a controlled environment.
    - More capable version of drag and drop programming.
    - Will require downloading of the ROS Toolbox and Simulink provided by the EcoCar MATLAB license.
- Linux/C will be used to run the Host-based Intrusion Detection System (HIDS) which will detect anomalies (malware, foreign software, virus) in the physical computer system.
- A security module running on Linux will differentiate between attacks and sensor issues.
    - A virtual world will be used to train a preliminary system until the car is further developed where data is accurate and abundant which can be used to train the system.
    - Create an encryption module that will decrypt and encrypt CAN signals going between each module and the sensors. This will be accomplished through the implementation of a custom developed CAN Encryption Module (CAN-EM).

2. Overall Description

2.1 Product Perspective

At a high level, the main CAV system follows this loop. The Data Acquisition team gathers raw sensor streams then pre-processes it into a workable format. Data Recognition processes the incoming streams to classify objects such as a dog, tree, sign, etc. Decision Making acts on this information by taking evasive actions such as swerving around a hazard in the road. Cybersecurity is part of this loop but has the separate functionality of detecting attacks and keeping data confidential. The cybersecurity module receives data from Data Acquisition and Data Recognition. After receiving the data and processing it the cybersecurity module then identifies whether that data is an attack and alerts Decision Making.

2.2 Product Functions

The main functions this system shall perform are:

- Encrypt all data traveling from sensors to CAV system and along the CAN bus.
- Detect attacks
    - Denial of Service (DOS) – Overloading the system to slow or halt processing
    - Replay – Reusing previous data to create a desired effect
    - Jamming – Sensors being prevented from receiving data
    - Spoofing – False sensor data being sent into the system
- Pass Decision-Making an assessment whether the data was an attack or failing sensor
- Detect active intrusions into processing system.

2.3 User Classes and Characteristics

No interaction is needed from the user to activate the cybersecurity features. Our features automatically start when the car is turned on, and never stop running.

2.4 Operating Environment

The Cyber Security "Module" will operate and run from the main tank of the Blazer. The computational demands of the analysis on the data as well as consistent hardware checks requires the module to have enough processing power. The software will be mainly Linux and ROS.

2.5 Design and Implementation Constraints

1. Guaranteeing a minimal impact on other modules needing to process information. Thorough testing under low-load environments, as well as high-load environments are required, so as to ensure that the bounds for processing requirements and data throughput do not exceed an allowable amount. Exceeding the allowable amount could have drastic effects on the abilities of other critical modules essential to the operation of the Blazer (autonomously).
2. Training the HIDS on V2X/CAVS specific data. Algorithms can be written to take in simulated data and detect an intrusion, but it will have to be reworked and retrained when the car is functional and the exact V2X transmission format is decided upon.

3. The constraints of the hardware systems include both timing buffer related requirements. Actions must be taken to ensure the proper response times of the data going through physical encryption modules to keep the system real-time. The buffer sizing and careful consideration must also be taken to ensure overflows do not occur within the processor of the encryption modules.
4. Training the security module without thorough and accurate data. Algorithms can be written now to take in simulated data and detect a deviation/anomaly, but it will have to be reworked and retrained when the car is up and running. To make the module work efficiently, the car would have to be driving frequently and regularly to establish patterns.

2.6 User Documentation

Our system does not involve any user documentation as it is all running the background of the system.

2.7 Assumptions and Dependencies

Assumptions

- To assume that the data transferring between modules works as intended.
- The tank will have no data corruption and will be in use for integrity checks.
- There will be enough accurate sensor data to verify if Mobileye is faulty/attacked.
- There will be enough accurate Mobileye data to verify if a sensor is faulty/attacked.
- There will be enough processing power on the tank to allow for real-time analysis.

Dependencies

- The current Ecocar CAN bus can only process 300kilobits per second.
- SAE j3016 – Levels of Driving Automation
- Processed data must be continuously passed to the security system by the Data Acquisition system.
- The security system must be notified to data inconsistencies by the Data Recognition system.
- Analysis results must be acted upon by the decision-making system.

3. External Interface Requirements

3.1 User Interfaces

Our program is designed to not directly interact with the user. Our data is passed off to Decision Making and is processed through them. From there if there is a warning, it is up to them to interact with the user about how to handle the issue.

3.2 Hardware Interfaces

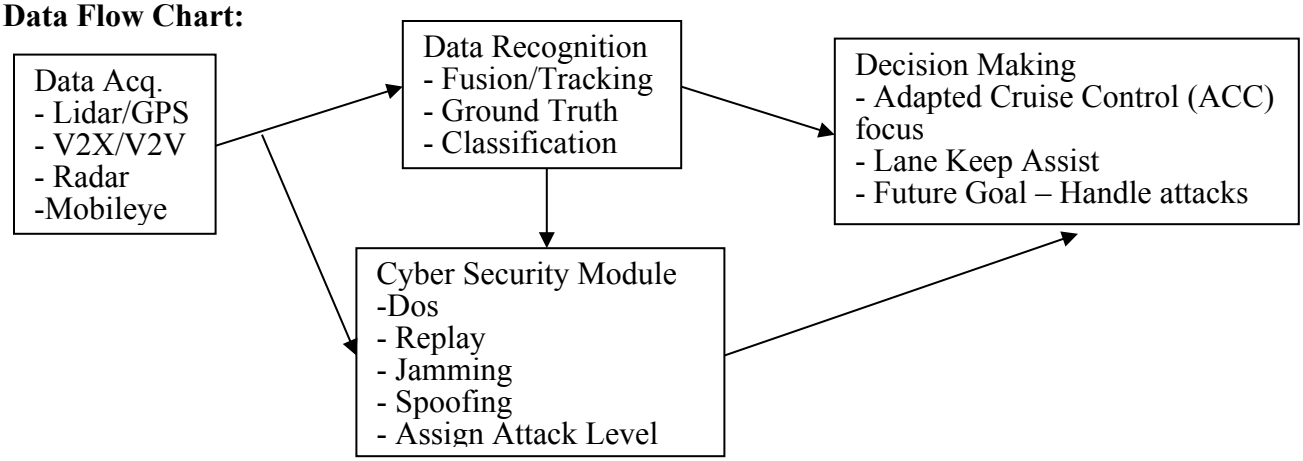
The hardware Interface of the cybersecurity subsystem consist of the interface between the CAN Encryption Module (CAN-EM) and the end device(s). These devices include CAN based sensors such as RADARs, MobileEYE cameras, and other CAN capable devices such as the AIoT Tank.

This Interface was designed to meet the ISO 11898 standard while complying with CANopen protocol for the physical pinout of the 9 pin Dsub connector. The Nominal V+ on this interface is specified to be +12V and shall draw no more than 2A.

3.3 Software Interfaces

The cybersecurity software module will interface with Data Acquisition, Data Recognition, and Decision Making. This language will all be in ROS, therefore no language barriers.

Data Flow Chart:



ROS can run on Unix systems and some Linux so the HIDS will run on Linux. The HIDS works by analyzing data coming in using statistics. This does not require thorough analysis of the original processed sensor data. Instead, this will study the ROS messages throughout the entire tank and discern patterns until enough data is studied that an inconsistency such as a virus will be detected.

3.4 Communications Interfaces

The CAN-EM contains two CAN-FD communication Interfaces for pass through data encryption as well as a USB interface for debugging and firmware updates from a host device. The CAN-FD interface supports a minimum bandwidth of 300kbps.

4. System Features

4.1 Encryption Module

4.1.1 Description and Priority

The encryption module is designed to encrypt any messages going over the CANBUS to allow information to be delivered securely. The module shall also be used to receive encrypted messages and decode them before they are received by the tank or any preceding module.

4.1.2 Stimulus/Response Sequences

If the module receives any information, then it will encode them. If the module receives and CAN messages, then it shall decode them.

4.1.3 Functional Requirements

REQ-1: The device shall provide security measures to CAN bus.
REQ-2: The device shall encrypt CAN data sent from the end device bus onto the vehicle bus.
REQ-3: The device shall add its own security header on the front of the CAN payload.
REQ-4: The security header shall contain a unique device ID corresponding to the encryption module.
REQ-5: The security header shall contain a sequence count field.
REQ-6: The device shall not interfere with the operation of other devices on the CAN bus.
REQ-7: The device shall require no modification of end devices.
REQ-8: The device shall appear "transparent" to the CAV System.
REQ-9: The device shall support a minimum data rate of 500 kbps.
REQ-10: The device shall have a through delay of <TBD>.
REQ-11: The delay shall vary no more than +/-5ms.
REQ-12: The device shall be integrated in-line (series) with vehicle cabling.
REQ-13: The device shall Interface with end devices via a DB-9 connector.
REQ-14: The DB-9 Connector shall be wired in accordance to CANopen pinout specification.
REQ-15: The device shall contain its own power protection circuitry.
REQ-16: The device shall be protected from overvoltage events.
REQ-17: The device shall cut power to itself when the input voltage exceeds 15V.
REQ-18: The device shall have overcurrent protection.
REQ-19: The device shall cut power is input currents exceed 2 amps (excluding inrush transient).

4.2 Cybersecurity Module

4.2.1 Description and Priority

Anomalies shall be detected and assigned as an attack level, e.g. “attack”, “possible attack”, “sensor misread”. The attack level will then be sent to Decision Making. This assignment needs to accurate and is considered a high priority because of the repercussion from a successful attack.

4.2.2 Stimulus/Response Sequences

The module will tap into the feed between Data Acquisition and Data Recognition. From that feed, the past 5 seconds of processed data will be continuously saved to a temporary buffer. In parallel, Data Recognition will have classified the same data and will then actively send suspicious data that they processed to the security module. When Data Recognition sends suspicious data, the security module will transfer the past 5 seconds of data from the temporary buffer to a separate buffer which will classify the attack level. The attack level is then passed to Decision Making.

4.2.3 Functional Requirements

REQ-1: The module shall detect a Denial of Service Attack
REQ-2: The module shall detect a Replay Attack

REQ-3: The module shall detect Sensor Jamming
REQ-4: The module shall detect Consecutive Sensor Spoofing
REQ-5: The module shall determine the confidence of an attack happening then assign an attack level
REQ-6: The attack level shall be classified in 3 levels from misread->possible attack->attack
REQ-7: The module shall notify Decision Making of the attack level
REQ-8: Module shall receive continuous processed data from Data Acquisition
REQ-9: Module shall receive suspicious classified data from Data Recognition
REQ-10: Module shall have 2 memory buffers
REQ-11: Module shall have 1 temporary memory buffers that holds the previous 5 seconds data from Data Acquisition
REQ-12: The temporary buffer shall take 1 second worth of new data at a time
REQ-13: The temporary buffer shall remove 1 second worth of the oldest data in buffer
REQ-14: The second buffer shall copy and hold the previous 5 seconds of data when suspicious data is sent from Data Recognition
REQ-15: The second buffer shall analyze the sensor data to assign an attack level
REQ-16: Mobileye will be used to detect sensor attacks
REQ-17: The sensors and the Data Recognition classification will be used to detect an attack against the Mobileye

4.3 Host-Based Intrusion Detection System

4.3.1 Description and Priority

Intrusions into the system via the V2X module or other vectors will be detected as deviations from the standard and classified as either malicious or within reasonable deviation. This assessment needs to be accurate and is considered high priority due to the potential damage that could stem from malware.

4.3.2 Stimulus/Response Sequences

This system will be actively monitoring activity on the tank and classifying it as either anomalous or within expected parameters. When detected anomalous activity, it will trigger an alert.

4.3.3 Functional Requirements

REQ-1: The module shall detect anomalous such as malware or data integrity corruption activity on the tank system.
REQ-2: The module shall classify anomalies as either attacks or as non-malicious activity.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The encryption module shall not delay data transmission and recognition by any more than 100 milliseconds. After this rate, the system congests the data flowing to the tank and slows down the response time of the vehicle.

5.2 Safety Requirements

The cyber security module put in place must be able to complete its necessary attack detection functions in a timely manner. Upon detection of an attack, ensure that the Decision Making team is notified of the attack level and can act accordingly.

5.3 Security Requirements

As the cybersecurity section of this project, security is of the most importance. The system will be able to operate against specified threats such as:

- Denial of Service (DOS) attacks
    - Attacks preventing the full use of a system in any way, typically through flooding bandwidth
- Sensor Spoofing/Altercation
    - Attacks involving transmitting false data to sensors
- GPS Attack Spoofing
    - Attacks like sensor spoofing attacks in general but requiring the specific format for GPS systems rather than just raw transmissions.
- Timing attacks
    - Attacks intentionally causing system delays.
- Home Attacks/Attempts at outsider control
    - Attacks where an attacker attempts to gain remote control of the system

5.4 Software Quality Attributes

- Software shall support updates and maintainability through following coding practices such as naming schemes, clear methods, and comments throughout.
- With abundant data down the car development process, the HIDS will improve and eventually achieve a robust security network.

Glossary

HIDS – Host-based Intrusion Detection System
ROS – Robot Operating System
DOS – Denial of Service Attack
CAN – Controller Area Network
GPS – Global Positioning System
V2V – Vehicle to Vehicle communication
V2X – Vehicle to Object communication