SYSTEMS ENGINEERING

COLORADO STATE UNIVERSITY

Student CyberTruck Experience Manual

Jeremy Daily

David Nnaji

Ben Ettlinger

Subhojeet Mukhergee

May 25, 2020

A special thank you goes out to Urban Jonson at the National Motor Freight Traffic Association, Inc. (NMFTA) for being a champion of the program. His passion for cybersecurity and education was insiprational in creating the content and program called the Student CyberTruck Experience. The Executive Director the NMFTA, Paul Levine, deserves special thanks as he provided resources and guidance on bringing the Student CyberTruck Experience to life. Our initial sponsors at the University of Tulsa include the NMFTA, Geotab, and PeopleNet. The continued support from Geotab through Ryan Brander and Glenn Atkinson are gratefully appreciated.

The students who significantly contributed to this manual, both in writing and participating in the Student CyberTruck Experience include: Hayden Allen, Duy Van, John Maag, Ryan Corley.

Contents

1 Introduction to the Student CyberTruck Experience

1.1 Trucking as a Critical Infrastructure

Transporting material by truck is a critical aspect of modern life. The Transportation Systems Sector has been declared a critical infrastructure sector by the Department of Homeland Security (DHS) [1]. Therefore, protecting trucks, trailers, vans, buses and the logistic engines supporting these vehicles is of the utmost importance for our way of life.

In the mid 1990s, the technology used on heavy vehicle transitioned from mechanical controls to digital controls. For many years, these digital controls and embedded systems were isolated from the rest of the world unless accessed by a technician. This is known as an air gap. However, the trucking industry realized process improvements through remote monitoring of the vehicles and their status. Technologies like over-the-air updates have become part of the truck and its supporting systems. This broke down the air gap defense and trucks may be connected to the Internet by way of a cellular modem. With the truck network connected to an outside network, there is a possibility for a vulnerability to be exploited and an attacker to remotely execute code on the truck. This is problematic, because the truck systems are trusting of the internal network and will react without regard the authenticity of a message. This means a well crafted message on the internal network can wreak havoc on the truck and potentially shut it down... or worse. Keep in mind, a truck traveling at 70 miles per hour carrying 80,000 lb has an large amount of energy. Here is a quick example of the kinetic energy calculation for moving truck:

$$KE = \frac{1}{2}mv^2$$

$$KE = \frac{1}{2}\frac{W}{g}(1.466S)^2$$

$$KE = \frac{80,000}{2(32.2)}[1.466(70)]^2$$

$$KE = 13,081,819 \text{ ft-lb}$$

For comparison, a .30-06 rifle can propel a 0.308 inch diameter bullet at 3000 ft/second, which is around 3000 ft-lbs of energy. In other words, a truck has as much energy as 4360 rifles being fired at once!

The ability for a truck to cause serious damage in a crash is significant. Crashes are perhaps the most

spectacular outcome of a cyber attack, but other subversive results can be achieved, like theft, logistics disruption, a traffic congestion. One particularly scary scenario is if a cyber attack affects many trucks at the same time. The tow and recovery services can easily respond to a couple broken-down trucks, but if thousands of truck came to a halt at the same time, the highway system would be broken. This means medical supplies, food, and fuel are no longer distributed. Its only a few days before serious civil unrest would start if trucking gets shut down. Brainstorming terrible scenarios is somewhat trivial, but our goal is to develop solutions. A compelling industry report from the NMFTA provides much of the rational to address the problem of heavy vehicle cybersecurity [2]. The reason for this book and the accompanying Student CyberTruck Experience is to teach the engineers and scientists the necessary elements needed to improve the cybersecurity posture of the transportation industry.

How the Student CyberTruck Experience Started

In 2016, Urban Jonson of the National Motor Freight Traffic Association, Inc. (NMFTA) reached out to Dr. Jeremy Daily while he was teaching mechanical engineering at the University of Tulsa (TU) to discuss some research findings related to heavy vehicle cybersecurity. In this conversation, Dr. Daily confirmed many of the hypotheses in the NMFTA whitepaper on the state of heavy vehicle cybersecurity. This conversation led to an invitation for Dr. Daily to attend the first NMFTA Heavy Vehicle Cyber Security (HVCS) meeting in Virginia. One of the results of the meeting was a recognition for the need to build the human talent needed to address the challenges associated with cybersecurity of heavy vehicles. This initiative is how the Student CyberTruck Experience came into being. A detailed account of the lessons learned in developing heavy vehicle cybersecurity talent was presented at the ESCAR USA 2017 conference [3].

1.2 Program Mission

The mission of the Student CyberTruck Experience is to *develop the talent necessary to improve the cybersecurity posture of heavy vehicles*.

This mission is intended to be high level and focused on the big picture. However, it is too general and hard to measure the actual achievement of the mission. There are a few indicators that the program is succeeding in its mission:

1. Graduates of the program have been hired into the industry in a cybersecurity capacity.
2. Graduates have gone onto graduate school and created scholarly works contributing to the body of science around heavy vehicle cybersecurity.
3. Student participants have attended and participated in the CyberTruck Challenge, thus building the overall awareness around heavy vehicle cybersecurity.
4. External sponsors continue to sponsor the research project, which enables undergraduates to focus summer research efforts towards heavy vehicle cybersecurity.

5. The ideas and projects from the Student CyberTruck Experience have been used as ideas for other funded research projects at the federal level.
6. Students consistently receive positive feedback at the HVCS bi-annual meetings.

While this is great feedback as to the success of the program, we need a more measured approach to help students and researchers progress through the program and maximize their potential. There is a famous saying:

> You can only improve what you measure.

This calls for some measurable program learning objectives.

1.3 Program Learning Objectives

As suggested by Sara Rathburn from Colorado State University's The Institute for Teaching and Learning, learning objectives are written statements organizing and defining the specific knowledge, skill-sets, and/or abilities students should acquire [4]. These learning objectives help students assess their skills and provide specifics for the learning journey to become a productive cybersecurity practitioner. An effective learning objective has three parts:

1. A description of what is expected.
2. The conditions necessary to demonstrate proficiency.
3. The criteria for evaluating performance.

For example, a Program Learning Example could be

- Create a program to reassemble messages that arrive out of order from a J1939 Transport Protocol - Data Transfer to properly receive the data in the same order in which it was sent.

Looking at this objective, there is a clear output of what output is expected. Specifically, the output is the computer program. However, the language writing the code is not specified, so any language would be acceptable. The conditions for the exercise is an out of order message stream from J1939 network. Success will be achieved when the original byte stream matches the reassembled message. The primary action verb in this objective is the word create. Different action verbs suggest different levels of learning, which has been popularized as Bloom's Taxonomy.

1.3.1 The Taxonomy of Educational Objectives (Bloom's Taxonomy)

When developing new course for university, we have to generate a set of course objectives. A well reasoned approach is to create objectives that start with the phrase "By the end of this course, students should be able to..." and the rest of the sentence is the learning objective. To create the learning objective, an action verb is used followed by the object of knowledge they are expected to acquire or construct. There are multiple

dimensions of knowledge as well as different levels of thinking about things. A succinct info-graphic was produced at Iowa State University under a Creative Commons License and displayed in Figure 1.3.1 on the following page. The examples and breakdown of the learning objective space shown in the figure will be used to span the basis for talent generation for the Student CyberTruck Experience.

The learning objectives should focus on accomplishing the program mission. Since the topic of cybersecurity is rapidly changing in the modern era, the learning objectives may need to change to accommodate advances in technology, knowledge, and application. For example, the advent of quantum computing promises to create a paradigm shift in modern cryptography as many "hard" problems that are used as the basis for modern cryptography can be calculated quickly, thus nullifying the underlying algorithms. As of this writing, there have been no practical exploits based on quantum computing.

Often Bloom's Taxonomy is constructed as a pyramid or cake as shown in Figure 1.3.2 on page 11. This is a simpler graphic to digest and use as a model for your own level of knowledge. Often the base layers are necessary to accomplish a mastery of a higher layer. For example, to apply a cryptographic algorithm on a heavy truck network, you would need to understand the limitations of J1939 and have knowledge of the truck system. The larger foundations are needed to achieve the higher order thinking. This means learning objectives that are focused on the top of the pyramid, like evaluate and create, require a mastery of the layers below. Thus students should strive to achieve the highest level of contemplation in an effort to have the broadest experience.

However, when a new student is introduced to a field, like heavy vehicle cybersecurity, they cannot immediately go to the top layer. Moreover, the process of building the knowledge foundation is critical to success. As such, the learning objectives in the program will vary across the cognitive process dimension. This enables a student to enter the program at their level and still have a objectives to challenge them. For example, a computer science student may have training and experience using data structures in C and an engineering student may have practical experience fixing air brake systems. These students would each bring their own level of expertise to the program and enter the quest for knowledge at different levels. The CS student may be able to quickly declare a structure for a CAN data frame in a C header file, but the ME student may need to learn about C data structures and data types, then understand how to create a struct. In other words, the level of effort in achieving mastery of the learning objectives may be different for each student.

1.3.2 Program Learning Objectives

To achieve the program mission, students who go through the Student CyberTruck Experience (CyTeX) should be able to...

1. List the major Original Equipment Manufacturers (OEM) and Tier 1 Suppliers for on-highway heavy vehicles.
2. Recognize a unified diagnostic service (UDS) message in SAE J1939 communication.
3. Recall the different layers of the Open Systems Interconnection (OSI) networking model.

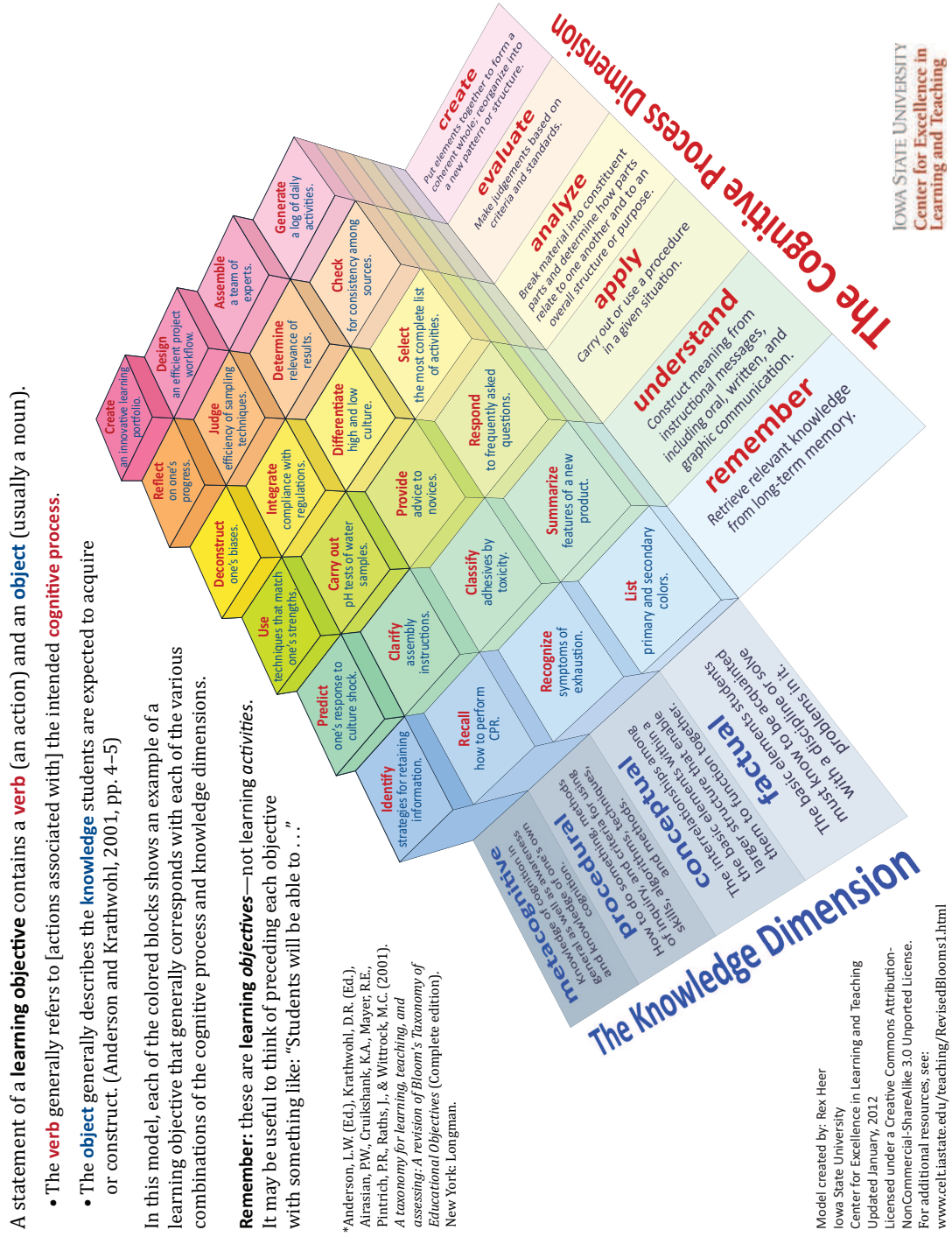


Figure 1.3.1: Revised Bloom's Taxonomy used in creating learning objectives [5].

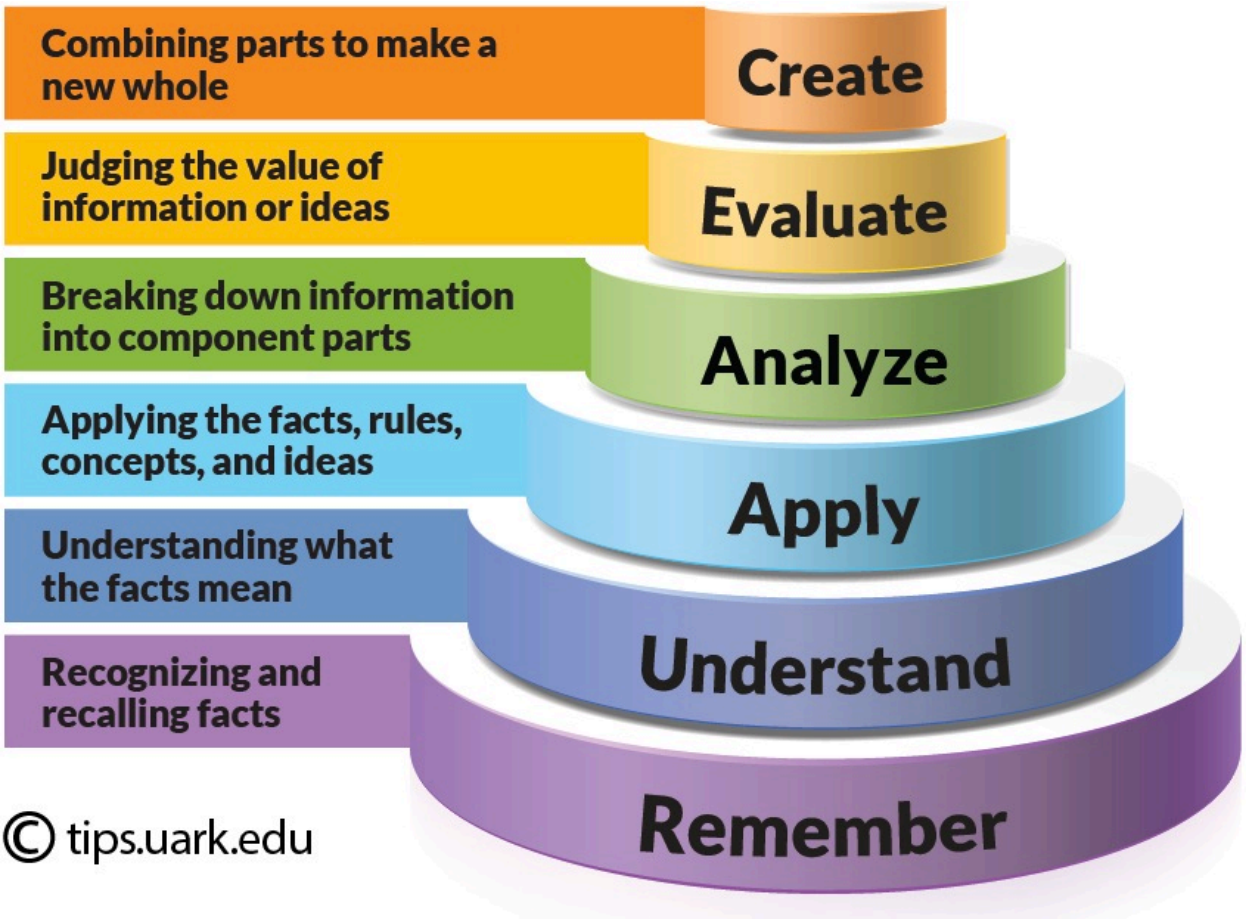


Figure 1.3.2: Bloom's Taxonomy from Jessica Shabatura.

4. Identify the different electronic control units (ECUs) used on heavy vehicles.
5. Summarize the strengths and weaknesses of the controller area network.
6. Classify trucks and trailers by weight class and use.
7. Clarify the way CAN bus data is used for the freight carriers, OEMs, tier 1 suppliers and telematics providers.
8. Predict potential cybersecurity attacks through an electronic logging device (ELD).
9. Respond to J1939 request messages for VIN, Component Identification, and other J1939 on request messages.
10. Provide a secure API for a CAN to Ethernet electronic device to network log data on a computer.
11. Carry out cryptographic message authentication for intra-vehicle communications.
12. Use certificates and public key infrastructure (PKI) to securely update a vehicle connected device.
13. Select the fastest cryptographic approach to secure diagnostic communication between a gateway and a diagnostic application.
14. Differentiate between best practices for IT security and heavy vehicle network security.
15. Integrate X.509 Certificates and PKI into embedded hardware security modules connected to vehicles.
16. Deconstruct heavy vehicle event data recorder (HVEDR) data sets from raw binary to time history graphs.
17. Check and verify digitally signed data from a CAN Logger
18. Determine the contents of a CAN frame based on an oscilloscope trace of the signals on the CAN bus wires.
19. Judge the real-time performance of HMAC and CMAC approaches to verifying the integrity and authenticity of CAN messages.
20. Reflect on the challenges for implementing secure solutions to legacy vehicles already on the road.
21. Generate a solution to stream maximum bit rate CAN data over an IP network.
22. Assemble a system to detect and defend against intrusions on the CAN bus.
23. Design a printed circuit board to bridge multiple networks while securely storing cryptographic key material.
24. Create an original research poster advancing the cybersecurity posture of heavy vehicles.

Notice these objective make use of every verb in the graphic shown in Figure 1.3.1 on page 10. Additional objectives will be enumerated within each chapter of the manual as the specifics of the content and exercises are matured.

1.4 Exercises

While the learning objectives are fantastic for what we want to achieve, the specific activities to accomplish those goals need to be defined. Through the manual, we will present exercises that focus on helping students to achieve a learning objective. These exercises are created at different levels of the taxonomy, which enables a learner to start at a level appropriate to their skills. Successful completion of all the exercises should lead to the accomplishment of all learning objectives. The idea of successful completion means we have to have a measurement for success.

Exercise 1.1. [Evaluate] Reflect on on the program learning objectives in Section 1.3.2 and determine what is missing. Write a new learning objective using a verb that is not already in the list. Justify the reason for the new objective and submit it to your cohort for discussion.

For each exercise, the

1.5 Assessments

In school, formal assessments typically come in the form of quizzes, exams, and essays. These tools are part of a traditional pedagogy where students learn to achieve the grade and the rewards associated with the good grades. In this model, the teacher is primarily responsible for the assessment tools. However, the teacher feedback and assessment needs to transition to self or peer assessment when schooling ends and graduates enter the work force. Since the Student CyberTruck Experience mission is workforce centric, these self assessment skills are necessary to develop.

The exercises used throughout the program should have measurable outputs. Many times the outputs are binary (e.g. the approach worked or it didn't), but some outputs are qualitative or quantitative. For quantitative outputs, there are typically thresholds of performance to measure success. For example, if an exercise is to build a CAN to Ethernet translator device, the quantitative measure of success could be the message rate through puts of the device. The exercise should include the testing functions required to perform the assessments; and the outputs of the tests should be compared to a threshold or standard of performance. For qualitative results, the arguments and analysis need to be presented clearly to peers, mentors, and instructors to ensure the concept is effectively communicated. Feedback is the primary form of assessing qualitative outputs.

1.5.1 Formative Assessments

Formative assessments are frequent small tests of ideas and concepts with low-stakes. They are meant to have little to no pressure and provide quick feedback on the progress towards achieving mastery of the learning objectives. Some types of formative assessments include:

- Conversations to determine levels of knowledge, like trivia questions

- What-if scenarios
- Bouncing ideas off peers
- Confirming concepts with mentors
- Sketching diagrams and flow charts

Formative assessments are frequently used in writing code and are synonymous with debug statements. Simply put, a debug statement is a small test to ensure the code is working according to your idea and design.

Students should develop their own set of self-critiquing skills and frequently perform their own formative assessments. The more of these frequent assessments that take place, the more likely for success. Furthermore, the documentation and communication of these formative assessment results are important to "managing your boss," which is an important workplace behavior where a subordinate will frequently update the manager on progress and challenges. Articulating what you've done, how you tested it, and explaining the results are welcome communications for any good manager.

An example of the formative assessment for Exercise 1.1 is the discussion with your peers and instructor. There are no grades associated with the exercise and the questions and discussion around your proposed objective will give you feedback on if it is clear and measurable.

1.5.2 Summative Assessments

In school, summative assessments are less frequent, high stakes exercises like examinations and term papers. Since the focus of the Student CyberTruck Experience is project based, the summative assessment is the successful completion of the project. This begs the question of what success means. In an engineering design, a customer will voice a need or desire. One of the first jobs of the design engineer is to restate the problem into a set of measurable requirements. Since a good requirement is measurable, the project essentially assesses itself. The project is completed once all the requirements are demonstrated to be satisfied. This means it is possible to perform a summative assessment completely by yourself. However, an external review of the project is highly recommended.

For the CyTeX program, the realization of the summative assessment is the research poster presented at the NMFTA's fall HVCS meeting. These posters are viewed by all the attendees, which range from motor carriers, to government researchers, military scientists, and cybersecurity professionals. Students are expected to explain their poster contents and address questions from the attendees.

An example of the summative assessment for Exercise 1.1 is the acceptance of the proposed learning objective into the program. Perhaps it will be included in the next version of this manual!

1.6 Research

Every year we continue to research topics that can be relevant and helpful to improving the cybersecurity posture of the trucking industry. These projects are curated from the needs of the trucking companies, current technologies, topics of interest, and achievability. The list of projects is curated in the fall of each year and proposed to the sponsors. Once agreed upon, the projects are presented to the students. Student select the projects based on interest, but often they don't have enough background to know where to go. In those cases, the exercises herein can lay the foundation of knowledge and experience to take steps towards accomplishing the research.

Research can lead new discoveries and novel methods, but the likely output of the research for this program is a well thought out engineering design project. The engineering design process is as follows:

Ask Understand the needs of the customer, which means you have to know who the customer and their use cases. Restate and define the problem with a series of measurable engineering specifications, criteria, and constraints.

Imagine Brainstorm and explore different possible solutions to the problem.

Plan Select the most promising solution and draw/sketch out a solution. Determine the resources needed

Create Implement a prototype the solution. Acquire the parts and materials needed and build the solution

Test Evaluate the prototype to determine if it works and satisfies the design constraints

Iterate the process and improve the implementation and refine the requirements.

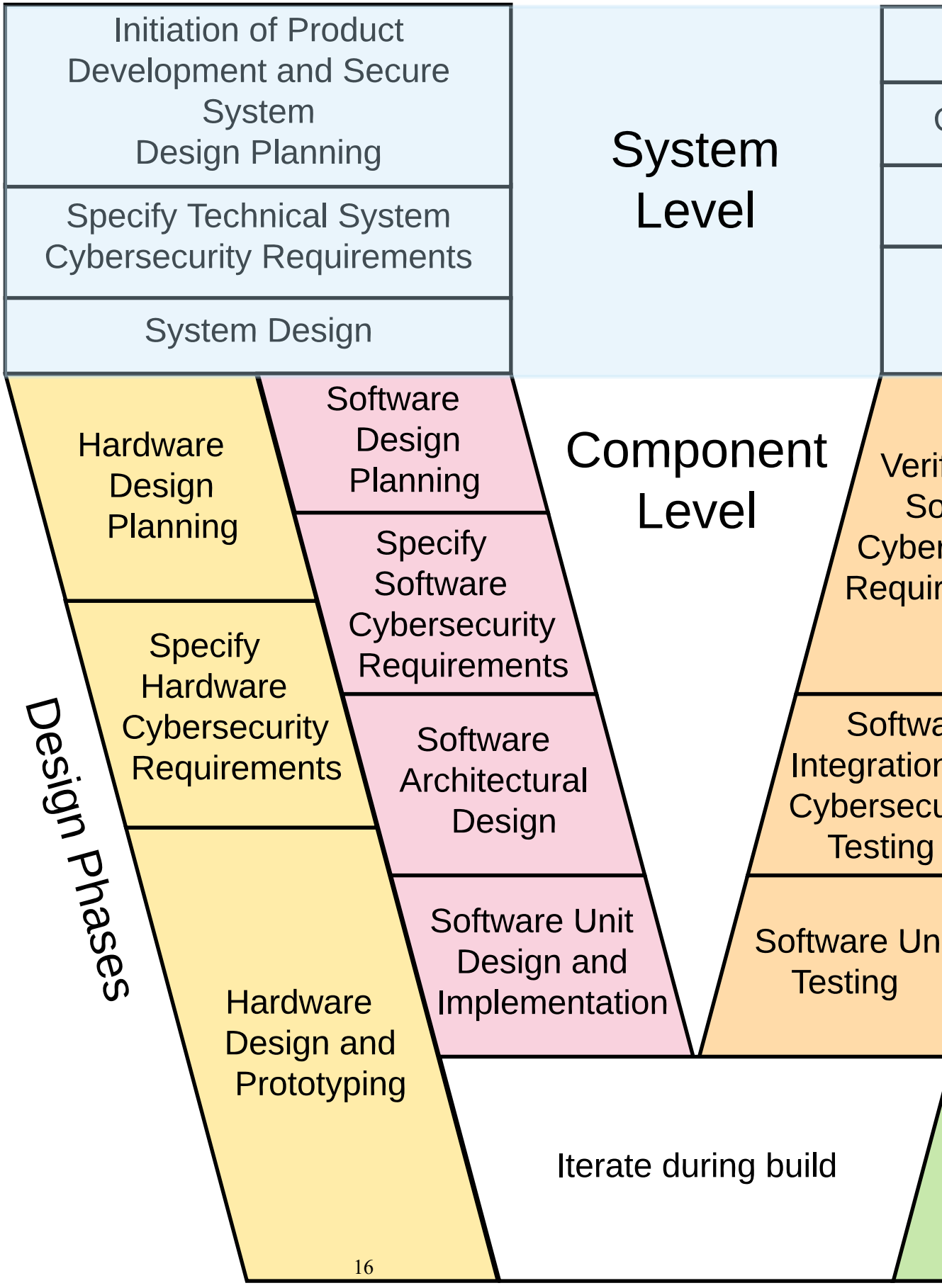
Share Communicate your design and solution to the appropriate audience.

There are many info-graphics on the Internet to visualize this process. The final poster presentation is the final step in sharing your results. The poster should capture the engineering design process and the things you've learned along the way, even if you learned that something doesn't work.

1.7 Systems Engineering Approach

The engineering design process is often visualized in an iterative loop. Another visualization, popular among systems engineers, is the systems V ("vee") diagram. The V-diagram is used by in SAE J3061: Cybersecurity Guidebook for Cyber-Physical Vehicle Systems to describe the process taken to design and develop more secure cyber-physical systems [6]. The graphic for this process is shown in Figure 1.7.1.

The V diagram is setup so the design phases are on the left and the testing phases are on the right. The top of the diagram contains the systems level approach, while the bottom V is separated into hardware and software designs. Many processes are iterative and follow the engineering design process within their own blocks. Often the hardware and software teams need to work closely together. For example, a systems level requirement may be to securely store cryptographic keys. Often this is done with a dedicated hardware


Initiation of Product Development and Secure System Design Planning
Specify Technical System Cybersecurity Requirements
System Design
System Level
Hardware Design Planning
Specify Hardware Cybersecurity Requirements
Hardware Design and Prototyping
Software Design Planning
Specify Software Cybersecurity Requirements
Software Architectural Design
Software Unit Design and Implementation
Component Level
Design Phases
Iterate during build

security module (HSM). The hardware team needs to build the circuit to support the HSM and the software team needs to use the key material stored on the HSM. The software team may have to use evaluation hardware prototypes to implement the communication protocols and product algorithms.

During your work in the Student CyberTruck Experience, take some time to reflect where you are working in the diagram of Figure 1.7.1. This reflection should help you understand the different levels of effort and teamwork needed to bring secure solutions into fruition.

2 Introduction to Trucking

2.1 Objectives

What is the trucking industry?

What is a truck?

Who makes trucks?

Cybersecurity for Trucking

2.2 Trucking Resources

The National Motor Freight Traffic Association, Inc. (NMFTA) has produced a great whitepaper [2]

3 Overview of Heavy Vehicle Systems

Weight Classes

4 Prototyping Electronics

The purpose of this chapter is to give students skills necessary to work with modern electronics and build their own hardware tools.

4.1 Intro to Microcontrollers

A **microcontroller (MCU)** is a tiny computer that can run one program at a time, over and over again. More often than not, they are connected to sensors and actuators allowing them to listen and interact with the physical world.

Sensors convert physical phenomena such as wheel speed, fuel level, and oil temperature into electrical signals. **Actuators** convert electrical energy back into physical energy such as linear motion, light, heat, and motor torque.

MCUs listen to sensors and talk to actuators. They decide what to do based on the instructions you have written and stored to its memory. MCUs and the devices connected to them form the basis of the **Electronic Control Unit (ECU)**.

4.2 Arduino and Teensy

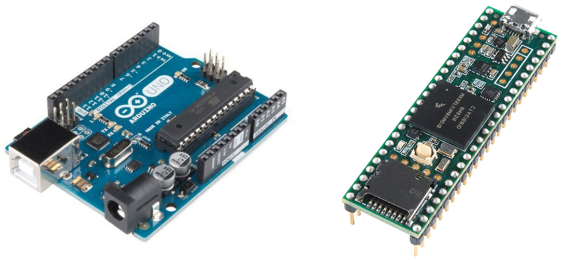


Figure 4.2.1: The Arduino Uno (left) and Teensy 3.6 (right)

Arduino offers some of the most recognizable open-source microcontroller boards today. This includes the Arduino Uno. When it comes to prototyping, these boards have many advantages including: widespread availability, low cost, thorough documentation, and a global community of users and forums. The Arduino

Uno uses the ATMEGA328P, has 32 KB of flash memory and runs at 16 MHz which are acceptable hardware specifications for simple projects.

Programs are written and compiled in the Arduino IDE program, a special text-editor, and uploaded via USB. When the Arduino is connected to an external power source it will run the instructions you uploaded.

The Teensy is another USB-based microcontroller development system. Although it is slightly more expensive than the Arduino, the 3.2 and 3.6 versions are preferred within CyTeX for their additional capabilities and features. Most notably, they support more communication protocols such as CAN, I2C, SPI, and Ethernet. The Teensy 3.6 footprint is roughly a third the size of the UNO, the MK66 Processor, has 1024 kB of flash memory, and runs at 180 MHz.

All programming is done via the USB port and the Teensy Loader Application. It is run automatically when using Verify or Upload within the Arduino software. Teensy Loader is available for Windows, Mac, and Linux.

Full documentation for the Teensyand Arduino UNOcan be found on their respective websites.

4.3 Intro to Microcomputers

A single-board computer (SBC) is a complete computer built on a single circuit board, with microprocessor(s), memory, input/output (I/O) and other features required of a functional computer. You *could* call it a "micro-computer." SBCs have the ability to run multiple programs, run an operating system, and can often support a simplified version of a typical desktop GUI.

The Raspberry Pi by the Raspberry Pi Foundation in the UK is by far the most common prototyping SCB. For automotive applications, the Beaglebone Black by Texas Instruments and the NXP S32K have been used within CyTeX. Both of these devices are capable of more advanced functionality.

4.4 Recommended Hardware Kit

Each student should have access to the parts in the basic kit. These items are minimum to successfully accomplish the programming and learning exercises.

Qty	Label	Description	Supplier	Supplier Part Number
1	A	Teensy 4.0 Development Board	PJRC	TEENSY40_PINS[1]
2	B	MPC2562 CAN Transceiver	Digi-Key	MCP2562FD-E/P-ND[2]
1	C	Solderless Breadboard	Digi-Key	BKGS-830-ND[3]
5	D	120 Ohm Axial Resistors	Digi-Key	S120CACT-ND[4]
1	E	Wiz850IO Ethernet Expansion Board	Digi-Key	1278-1043-ND[5]
1	F	30 Pack of Male-Male Breadboard Wires	Sparkfun	PRT-14284[6]
1	G	20 Pack of Male-Female Breadboard Wires	Sparkfun	PRT-12794[7]
1	H	Ethernet Cat6 Cable, 3 ft.	Sparkfun	CAB-08915[8]
1	J	USB micro Cable, 6 inch	Sparkfun	CAB-13244[9]
1	K	SOIC8 to DIP Converter Board	Sparkfun	BOB-13655[10]
1	L	ATECC608A Crypto Authentication Module	Digi-Key	ATECC608A-SSHDA-TCT-ND[11]
1	M	ATECC608 Crypto Co-Processor Breakout	Sparkfun	SPX-15838[12]
1	N	Trimpot 10K Ohm with Knob	Sparkfun	COM-09806[13]
1	P	Break Away Headers - Straight	Sparkfun	PRT-00116[14]
1	Q	Ambient Temperature Sensor	Sparkfun	SEN-14049[15]
1	R	Conductive 18 Compartment Organizer	Flambeau	C618[16]

Table 4.4.1: Basic Hardware Kit (Either use M or a the combination of K and L.)

Figure 4.4.1: Photograph Example Parts Kit

4.4.1 Basic Kit

Newer laptop computers may not have a physical Ethernet port, so a USB to Ethernet adapter may be necessary.

Exercise 4.1. [Knowledge] Download the Datasheet for the Teensy 4.0 processor. Answer the following questions:

1. What is the name of the processor?
2. What technology is the processor core built on?

[1] https://www.pjrc.com/store/teensy40_pins.html
[2] https://www.digikey.com/product-detail/en/microchip-technology/MCP2562FD-E-P/MCP2562FD-E-P-ND/4842807
[3] https://www.digikey.com/products/en?keywords=BKGS-830-ND
[4] https://www.digikey.com/product-detail/en/stackpole-electronics-inc/RNMF14FTC120R/S120CACT-ND/2617441
[5] https://www.digikey.com/products/en?keywords=Wiz%20850
[6] https://www.sparkfun.com/products/14284
[7] https://www.sparkfun.com/products/12795
[8] https://www.sparkfun.com/products/8915
[9] https://www.sparkfun.com/products/13244
[10] https://www.sparkfun.com/products/13655
[11] https://www.digikey.com/products/en?keywords=ATECC608A-SSHDA-TCT-ND
[12] https://www.sparkfun.com/products/15838
[13] https://www.sparkfun.com/products/9806
[14] https://www.sparkfun.com/products/116
[15] https://www.sparkfun.com/products/14049
[16] https://www.flambeaucases.com/18-compartment-box-1095.aspx

Qty	Description	Supplier	Supplier Part Number	URL
1	Teensy 4.1	PJRC	TEENSY41	[17]
1	Beagle Bone Black			
1	Quadrature Encoder with breakout board. Bournes PEC09			

Table 4.4.2: Advanced Hardware Kit

Qty	Description	Supplier	Supplier Part Number	URL
1	Smart Sensor Simulator 2	DG Technologies	TEENSY41	[18]
1	Beagle Bone Black			
1	Quadrature Encoder with breakout board. Bournes PEC09			

Table 4.4.3: Vehicle ECU Testing Kit

3. How many bits does the processor use when executing instructions?
4. How fast does the processor run?
5. How many CAN channels does the micoprocessor have?
6. What are the register number bases for the CAN Channels?

Exercise 4.2. [Comprehension] After reviewing the datasheet for the MCP2562, address the following questions:

1. Why do we need to connect the

4.4.2 Basic Tools

Multimeter Any modern digital multimeter is acceptable to get started. There are many on Amazon that are suitable.

Soldering Iron A

Tweezers

Wire Cutters

Wire Strippers

Carrying Tote

30 gauge Wire

4.4.3 Advanced Hardware Kits

4.5 Electronic Hardware Exercises

These exercises shouls serve to get you "up to speed" on Arduino hardware. Little to no computer programming experience is needed for this section, that comes later. However, you should know how to upload software in Arduino and view the console.

Exercise 4.3. Write a program to send CAN messages from one node to another.

Hints:

1. Check the continuity of each hookup wire before using it. They can be fragile and break easily.

4.5.1 Getting Started on Teensy Arduino (Stock Code that "should work")

4.5.2 Intro to CAN with Arduino (Move after CAN Section?)

4.5.3 Build a Breadboad Arduino (Move Challenge to Challenges Section)

Challenge: using the pieces below, send a CAN message.

Materials:

Breadboard Prototyping Wire ATmega328P MCP2551/MCP2562/MCP2558 MCP2515/MCP25625 LED 120ohm (2) or 60 ohm (1) resistor a 16MHz crystal 6 pin straight header a 10k resistor 18-22 pF capacitor (2)

Resources:

Use Datasheets to determine how to connect everything To find datasheets use google, digikey, and mouser Learning how to push code to the arduino: https://www.arduino.cc/en/Tutorial/ArduinoISP For help with SPI communication to MCP2515: http://avrbeginners.net/architecture/spi/spi.html

Challenge:

Send a CAN Message (Verify with oscilloscope) Decode the message from the oscilloscope to ensure that it is the mesage you told your arduino to send This can be done with Python Create an Altium Schematic of the layout that you created

[17] https://www.pjrc.com/store/teensy41.html
[18] https://www.pjrc.com/store/teensy41.html

5 Truck Systems for Computer Scientists

6 Computer Programming Fundamentals (Computer Programming for Engineers)

Computer programming may seem like a daunting field for beginner programmers. It does not need to be. At its core, computer programming is providing a set of instructions to a computer. Each line of code contains one or more instructions. Sometimes instructions interfere with each other and cause errors. Imagine telling someone who has never seen a sandwich before how to make a peanut butter and jelly sandwich. The steps may be:

1. Get Bread, Peanut Butter, Jelly, Knife, and Plate.
2. Place plate.
3. Place two pieces of bread side by side on plate. Call them Slice A and Slice B.
4. Put peanut butter on knife.
5. Use knife to spread peanut butter on Slice A.
6. Clean knife.
7. Put jelly on knife.
8. Use knife to spread jelly on Slice B.
9. Combine Slice A with Slice B by placing Slice B on Slice A.
10. Eat Sandwich.
11. Clean Knife
12. Clean Plate

Provided no external changes, this set of instructions will always lead to a proper peanut butter and jelly sandwich. However, some people may argue about if you should use grape jelly or strawberry jelly, chunky or smooth peanut butter. Some people are adamant that you must put Jelly on Slice A, never peanut butter. Some argue that if one starts with a loaf of bread instead of slices then it produces a better sandwich. These options represent the different ways that people program. There are many different sets of instructions that produce peanut butter and jelly sandwiches. Programming is very smiliar. There are many different ways to code a program that will lead to the same result.

However, there are ways which are more efficient than others, and there are ways to make code easier to understand. When writing code, it is important to write with a consistent programming style that allows others to understand your code. Code must be maintained over time, and may transition between programmers.

This chapter should serve to introduce the basics of computer programming as a platform for the Cyber Truck Experience program.

6.1 Arduino Programming Language

6.2 Python 3 Programming Language

An important tool in any hackers toolkit is the ability to interact with collected data. In this section, we will be using a programming language called Python to help with this.

6.2.1 Background

Python is a general purpose, versatile, and popular programming language. It is great as a first language because it is concise and easy to read. It is also a good language to have in any programmer's stack as it can be used for everything from web development to software development and data science applications.

This is different than what we have done to this point in Arduino. We will generally use the Arduino devices to directly interface with systems. These devices will then send data back to a computer or other device running a python script.

6.2.2 Codecademy

We will begin by having you learn the basics of python. We will be using an external system to aid in this. Navigate over to hereand create an account using your email and fill out any information they require. Please ensure that you do not pay for this service at this time. We will only need the services that the free version offers.

Please complete this course. Take notes throughout the course over functions that may be useful (importing and exporting files, searching through lists, data structures, etc.) Periodically, there will be additional trial material to supplement your learning.

6.2.3 Editors and IDE

6.2.4 Threading

6.2.5 PySerial

6.2.6 matplotlib

6.2.7 Making GUIs with PyQt5

6.3 C++ Programming Language

6.3.1 Codecademy

6.4 Computer Programming Fundamentals Exercises

7 Serial Communication with SAE J1708 and J1587

Exercise 7.1. CAN Frame Decoding

Capture a CAN Frame using an oscilloscpope on a J1939 network and decode it according to SAE J1939.

Exercise 7.2. Read Live Engine RPM Challenge

Using a BeagleBone Black and a Truck Cape, connect to an engine contoller that is broadcasting non-zero engine RPM. Gather this data using candump. Interpret the raw CAN frames and extract information for Engine RPM, or J1939 SPN 190. Plot 20 seconds of changing RPM with matplotlib. Print the properly labeled plot to PDF and show it to your instructor. Objectives

Learn how to interface with Linux SocketCAN and can-utils Be able to look up a signal definition in the J1939 Digital Annex (spreadsheet) Use grep to search for specific strings from a candump Have a reliable CAN datalogger for use in future projects Plot data using matplotlib in Python.

Suggested Materials

This exercise can be run with any Linux device with CAN hardware. An example of a commercial product with these features is the DG Technologies' Beacon device. An example of a hand built project is the BeagleBone Black with a TU TruckCape. Resources

J1939DA Internet Access (you may want to share your PC's connection sharing)

Exercise 7.3. Man in the Middle

Build a man-in-the middle board and box that takes CAN signals into one can channel and sends them out on another. Start a diagnostics session with a PC and RP1210 device to perform maintenance. Create a forwarding system that inspects and forwards network traffic in both directions. Attempt to hijack a a diagnostic session and affect a parameter change started with the PC diagnostics software.

Using DDEC Reports, try to prevent resetting the CPC clock during a data extraction on a CPC.

8 CAN Communication with SAE J1939

Exercise 8.1. CAN Frame Decoding

Capture a CAN Frame using an oscilloscpope on a J1939 network and decode it according to SAE J1939.

Exercise 8.2. Read Live Engine RPM Challenge

Using a BeagleBone Black and a Truck Cape, connect to an engine contoller that is broadcasting non-zero engine RPM. Gather this data using candump. Interpret the raw CAN frames and extract information for Engine RPM, or J1939 SPN 190. Plot 20 seconds of changing RPM with matplotlib. Print the properly labeled plot to PDF and show it to your instructor. Objectives

Learn how to interface with Linux SocketCAN and can-utils Be able to look up a signal definition in the J1939 Digital Annex (spreadsheet) Use grep to search for specific strings from a candump Have a reliable CAN datalogger for use in future projects Plot data using matplotlib in Python.

Suggested Materials

This exercise can be run with any Linux device with CAN hardware. An example of a commercial product with these features is the DG Technologies' Beacon device. An example of a hand built project is the BeagleBone Black with a TU TruckCape. Resources

J1939DA Internet Access (you may want to share your PC's connection sharing)

Exercise 8.3. Man in the Middle

Build a man-in-the middle board and box that takes CAN signals into one can channel and sends them out on another. Start a diagnostics session with a PC and RP1210 device to perform maintenance. Create a forwarding system that inspects and forwards network traffic in both directions. Attempt to hijack a a diagnostic session and affect a parameter change started with the PC diagnostics software.

Using DDEC Reports, try to prevent resetting the CPC clock during a data extraction on a CPC.

9 Princples of Cybersecurity

The Triads of Cybersecurity

Confidentiality, Integrity, and Availability

Or,

Authentication, Authorization, and Auditing

9.1 Introduction to Cryptography

9.2 Message Authentication

9.3 Encryption on CAN

10 Heavy Vehicle Digital Forensics

11 Challenge Problems

Problem 11.1. Transfer a JPEG image of the CSU logo over CAN and display it.

Nomenclature

air gap A physically disconnected network with no access to the outside world.

API Application Programming Interface

CMAC Cryptographic Message Authentication

CyTeX Student CyberTruck Experience

ECU Electronic Control Unit

ELD Electronic Logging Device

HMAC Hash based message authentication

HSM Hardware Security Module

HVCS Heavy Vehicle Cyber Security

HVEDR Heavy Vehicle Event Data Recorder

IT Information Technology

NMFTA National Motor Freight Traffic Association, Inc.

OEM Orignial Equipment Manufacturer

OSI Open Systems Interconnection

PKI Public Key Infrastructure

TU The University of Tulsa

UDS Unified Diagnostic Services

VIN Vehicle Identification Number

Bibliography

[1] Department of Homeland Security, *Transportation Systems Sector*, Original release date: July 06, 2009; Last revised: May 09, 2019; Accessed: May 24, 2020. "https://www.cisa.gov/transportation-systems-sector. 6

[2] National Motor Freight Traffic Association, Inc., *A Survey of Heavy Vehicle Cyber Security*, September 2015. http://www.nmfta.org/documents/hvcs/nmfta%20heavy%20duty%20vehicle%20cyber%20security%20whitepaper%20v1.0.3.6.pdf. 7, 18

[3] J. Daily, U. Jonson, and R. Gamble, "Talent generation for vehicle cybersecurity," in *Embedded Security for Cars (ESCAR) USA*, 2017. http://www.nmfta.org/documents/hvcs/vehiclecybersecurityeducationalinitiatives.pdf. 7

[4] S. Rathburn, *Developing Course Learning Objectives*, Accessed: May 24, 2020. https://tilt.colostate.edu/TipsAndGuides/Tip/92. 8

[5] Center for Excellence in Learning and Teaching, Iowa State University, *Revised Bloom's Taxonomy*, Accessed: May 24, 2020. https://www.celt.iastate.edu/teaching/effective-teaching-practices/revised-blooms-taxonomy/. 10

[6] S. International, *Cybersecurity Guidebook for Cyber-Physical Vehicle Systems*, Jan 2016. https://www.sae.org/standards/content/j3061_201601/. 15