Risk Assessment for Cooperative Automated Driving

Derrick Dominic
University of Michigan
ddominic@umich.edu

Sumeet Chhawri
University of Michigan
Transportation Research
Institute (UMTRI)
schhawri@mtu.edu

Ryan M. Eustice
University of Michigan
eustice@umich.edu

Di Ma
University of
Michigan-Dearborn
dmadma@umich.edu

André Weimerskirch
University of Michigan
andrewmk@umich.edu

ABSTRACT

Global investment and recent advancements in vehicle automation are making autonomous and cooperative automated driving (AD) a reality. Not only will automated vehicles incorporate more electronics and connectivity than ever before, but also, notably, they will transfer control and responsibility of monitoring the environment from a human driver to a robotic system. While prior work has assessed and provided security solutions for non-automated vehicles, there is much to understand regarding the security implications of AD. In this work, we begin to address this gap in understanding. This paper reports on a risk assessment framework for autonomous and cooperative AD. We aggregate the state of the art in AD research to define a reference architecture for automated vehicles, describing the new attack surfaces and data flow. Employing existing automotive threat models, we propose a novel application-based threat enumeration and analysis approach that is able to address different AD applications across all levels of automation. We demonstrate this framework with an example application assessment and summarize the results and security insights from analyses of other applications. The results of our risk assessment and future assessments with this framework will inform on the design of security solutions and secure architectures for production AD systems.

Keywords

Automated driving; autonomous vehicles; connected vehicles; cooperative driving; cybersecurity; risk assessment; threat modeling.

1. INTRODUCTION

Modern automobiles incorporate more electronics and connectivity than ever before in the form of driver assistance systems, personal electronics, and infotainment systems. Due to the increased complexity and remote accessibility of in-vehicle systems, modern automobiles are vulnerable to cyber-attacks against safety and privacy. Security research has uncovered a broad range of physical and remote attack surfaces that malicious agents can use to take advantage of a vehicle [17, 6, 10]. In response, solutions have been developed to secure in-vehicle networks (e.g. secure CAN) and detect potential security threats (e.g. intrusion detection systems).

More recently, the automotive world has seen an escalation in research toward partial and fully automated driving (AD). Since the DARPA Grand and Urban Challenges [35, 36, 27, 3, 19], there has been a significant investment by both industry [13, 39, 5, 9] and academia [21, 22, 20, 37, 18] to make AD a reality. While full automation remains in research and development, applications enabling partial automation such as self-parking, Adaptive Cruise Control (ACC), and Lane Keeping Assistance (LKA) are already being offered to consumers [34]. Unlike other automotive applications, AD transfers control and the responsibility for monitoring the environment from the human to the vehicle. From a security perspective, this transfer of control and the increased complexity of AD systems exposes more entry points for malicious and unintended cyber-attacks. And, depending on the situation and level of automation, neither a human driver/passenger nor the vehicle may be ready or able to respond.

The safety and privacy of future passengers will depend on the security solutions deployed to secure automated vehicles. As of now, an unsatisfied prerequisite to developing and appropriating AD security solutions is understanding the security implications of AD. In this paper, we work toward this goal by proposing and demonstrating a risk assessment framework for autonomous and cooperative AD.

Specifically, our contributions are:

1. Developing a reference architecture from the state of the art in AD to model AD applications.

2. Formulating a customizable threat model based on existing automotive threat modeling approaches to enumerate and assess potential threats to AD applications.

3. Demonstrating our risk assessment framework with an example AD application and summarizing the results and security insights from assessments of other applications.

CPS-SPC'16, October 28 2016, Vienna, Austria
© 2016 ACM. ISBN 978-1-4503-4568-2/16/10...$15.00
DOI: http://dx.doi.org/10.1145/2994487.2994499

This paper is organized as follows. Section 2 reviews automotive cybersecurity and surveys the related work in AD security. Section 3 clarifies the challenges of our work and motivates our risk assessment approach. Section 4 describes our AD reference architecture using the state of the art in autonomous vehicle research. Section 5 details our threat identification and modeling approach for AD applications. Section 6 demonstrates the use of this framework with an assessment of an AD application. And, sections 7 and 8 conclude with a summary of the results and security insights from all of our application assessments.

2. BACKGROUND

2.1 Automotive Cybersecurity

We first review the architecture of the modern automobile. Most vehicle functionality is implemented in 50-70 independent computers known as Electronic Control Units (ECUs) [17]. Because some automotive features require interaction across units, ECUs can share data through peer-to-peer connections or over several standard data buses using a number of protocols including CAN, LIN, FlexRay, and MOST [23]. While there are generally multiple buses for different component groups, they may not necessarily be physically isolated [17]. Modern vehicle attack surfaces include but are not limited to the physically accessible OBD-II diagnostic ports, passenger infotainment systems, wireless personal devices, the wireless Tire Pressure Monitoring System (TPMS), and key fobs for keyless entry [6].

Modern automotive systems have already been shown to have several physical and remote vulnerabilities. [17] demonstrated that attackers with physical access to modern vehicles can exploit several internal vulnerabilities. They showed, by experiment, that modern vehicles have few safeguards against attacks to ECUs or the internal vehicle network, and that infiltrating almost any ECU can allow an attacker to affect any other. [6] extended this work by demonstrating experimentally how malicious agents can take advantage of remote attack surfaces in addition to physical ones.

2.2 Automotive Threat Models

Interest in evaluating the security of current and future automobiles prompted the development of automotive threat models. A threat model defines the parameters of a potential threat and how those parameters can be used to evaluate the risk of a threat to a target. In our study, we explored the automotive threat models used by the National Highway Traffic Safety Administration (NHTSA) and the E-safety Vehicle Intrusion Protected Applications (EVITA) project.

The NHTSA threat modeling approach [23] is a composite model derived from STRIDE, Trike and Microsoft ASF [26, 29, 25]. Their approach first identifies automotive applications and then decomposes them into interconnection diagrams. Threats are then identified and analyzed by filling in the parameters of a threat matrix. Importantly, the NHTSA approach is limited in that it does not capture *Threat Agents*, has few factors influencing *Motivation*, and fails to consider the risk of being caught as a negative factor in motivation. Considering threat agents provides a better understanding of capabilities and motivations, leading to a more informed estimation of the parameters in the threat matrix.

The EVITA project developed a similar comprehensive threat analysis methodology [12] which they used to develop automotive security microcontrollers [2]. Their approach begins with the development of a reference architecture for a general understanding of the system in question. This is subsequently used to enumerate threats. Like the NHTSA approach, each threat is consolidated in a threat matrix with factors considering its severity and likelihood of success.

2.3 Related Work

Security literature has only recently considered the case of automated driving. [31] offered one of the earliest analyses of security in automated and connected vehicles in their identification of threats in high and full AD. The authors assessed potential attacks to AD sensors and infrastructure, using a threat matrix to categorize and prioritize risks by likelihood and impact. [38] also broadly covered security concerns and possible mitigation strategies for not only automated cars, but also aircraft, trains, and ships. And, [11] looked at security requirements and potential attacks to Intelligent Transportation System (ITS) applications in general, including automated driving; like [38], their list of cyber-threats was largely inspired by the well-studied attacks on wired and wireless communications.

Some studies have focused on specific vulnerabilities to cooperative and automated vehicles. [30] experimentally demonstrated that cameras and LIDAR sensors (critical sensors for most AD platforms) can be remotely attacked by blinding and spoofing. [7] highlighted how a malicious agent could easily disrupt GPS signals. [32] explored attacks that exploit vehicle connectivity; the authors describe attacks on location privacy leveraging location information broadcast on Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I) communication networks. [16] shows how a similar attack on privacy can be performed on location information that is broadcast from navigation software on personal devices carried by passengers. And, with the accelerated use of machine learning for automated vehicle perception, [24] explores the possibility of adversarial inputs to a machine learned model.

3. APPROACH

Our goal in this work is to gain an understanding of cybersecurity risks to autonomous and cooperative AD for the purpose of designing or appropriating prioritized security solutions and secure architectures. However, security analysis of AD poses important challenges that need to be addressed by our approach.

- *AD is still in development.* High and fully automated driving systems are still in research. Academic and industrial development teams focusing on the research problems of automated driving tend to design unique systems with minimal regard to future standardization. As a result, there is no reference architecture for AD systems.

- *AD components can realize different applications at different levels of automation.* A security assessment of AD must consider applications at different levels of automation (discussed further in section 4). As described in section 2.3, prior work in AD cybersecurity primarily addresses attack surfaces and approaches. We would refer to this as a zero-th order analysis of an interconnected system. In the literature, there has been little to no consideration of how components work together to realize applications. We do not that an approach that disregards the way components realize an application has the advantage of being generalizable to many applications. However, because different components are used in different ways to realize AD applications, taking into account the way components are used will help better predict the effect of a potential attack.

- *It is impossible to assess every application and enumerate every threat.* AD implementations can differ by manufacturer and new applications may be developed in the future. As technology improves and expertise becomes accessible, more threats become viable and an initial threat identification becomes obsolete.

We address these challenges by developing a *framework* for risk assessment of AD. And, in particular, we propose an *application-based* approach as opposed to a component-based one. In our proposed approach, a reference architecture based on the state of the art in AD is used to model the targeted AD application and to identify threats. Threats are characterized using a threat model combining the strengths of the NHTSA and EVITA automotive threat modeling approaches. As AD systems go into production and new applications are developed, our framework would need to be re-applied to incorporate new information.

For the purposes of the risk assessment in this work (refer to section 6), we limit our focus to attacks which exploit AD and, in particular, the targeted application. We do not assume that any security solutions have been appropriated, although it may be possible that resiliency may come directly from the system architecture (e.g. through redundancy in sensors). We will look primarily at input data vulnerabilities (e.g. forged sensor data) as opposed to software vulnerabilities (which we touch on in section 7).

4. AUTOMATED DRIVING

4.1 Background

Given an AD application, we need to understand its implementation to determine potential attack surfaces and begin to enumerate threats. In this section, we describe a reference architecture based on the state of the art in AD research.

We begin by clarifying two terms. First, when we refer to an "AD application", we mean an automotive function which employs automated driving in specific scenarios. These include, but are not limited to, Cruise Control, Adaptive Cruise Control, Lane Keeping, Automatic Emergency Braking, etc... An exhaustive list of AD applications (which we refer to for high level application descriptions) can be found in [4]. Second, we categorize AD applications by their "level of automation" using the SAE definitions [33]. They define automation as spanning from level 0 (no automation, or full control and responsibility maintained by the human driver) to level 5 (full automation, or full control and responsibility given to the vehicle).

4.2 Autonomous Automated Driving

As most AD systems (particularly for high levels of automation) are still in development, there exists no standard architecture for a commercial AD solution. To that end, we aim to distill a generalized architecture for an AD system that captures the state of the art and allows us to deconstruct applications at various levels of automation.

For our AD architecture, we look to the systems of Defense Advanced Research Projects Agency (DARPA) Grand and Urban Challenge finalists [35, 36, 27, 3, 19] and other academic and industrial AD research platforms [20, 37, 8, 39]. The interconnection diagram for this architecture can be seen in Fig. 1. Our diagram consists of functional blocks at two levels of abstraction: the higher level blocks (e.g. "maps" and "sensors") are more generalizable to different applications while lower level blocks (e.g. "road network map" and "GPS") are more specific to the application. Our architecture depicts an autonomous automated vehicle augmented with communications modules for cooperative driving. This reflects the fact that academic research on automated driving tends to focus on level 4 and 5 (highly and fully autonomous) driving. However, we can still use this high automation level definition to assess applications at lower automation levels.

This interconnection diagram describes a vehicle equipped with odometric sensors for inertial navigation, a GPS receiver for global navigation, and range sensors for perceiving the environment. We do not specify the sensor suite for the sake of generalization. For example, while several platforms employ laser range finding [35, 36, 27, 3, 19, 37], some projects [8, 39] rely on less expensive sensors like cameras for perceiving the environment.

Additionally, our system employs maps which encode prior (but not necessarily up-to-date) knowledge of the environment for localization and path planning. Maps allow for a drastic simplification of the autonomous driving problem. Instead of dealing with the difficulty of perception, vehicles employing maps need only localize onto the map (using, for instance, range sensors to detect features in the environment) to get pre-annotated information about lanes, traffic light locations, traffic signs, static objects, etc... Maps also simplify the global path planning problem (i.e. finding a route to the desired destination) because the possible routes that could be taken are known ahead of time.

We distinguish 2 kinds of maps: an environment map which captures dense information like the environment's 3D structure or appearance [21, 22, 20] and a road network map which captures sparse information like an abstract representation of the road network and lane markings [8]. We also consider in our architecture the case of maps being shared and updated over-the-air [8].

We organize our system's software architecture into three modules: *localization*, *object detection*, and *path planning.* Sensors and maps feed into these modules.

- *Localization*: This module is responsible for localizing the vehicle in its environment. From the vehicle's last estimated pose (position and orientation), we estimate a new pose using the odometric sensors and GPS receiver. This estimate is further refined using range sensors to localize the vehicle onto a dense environment map.

  We note that several localization pipelines use GPS only as initialization for a coordinate frame tracked solely by odometry [36, 27]. Although GPS has the advantage of bounding localization error, the discontinuities and jumps of its location estimate make it a poor localization solution for tasks like object tracking.

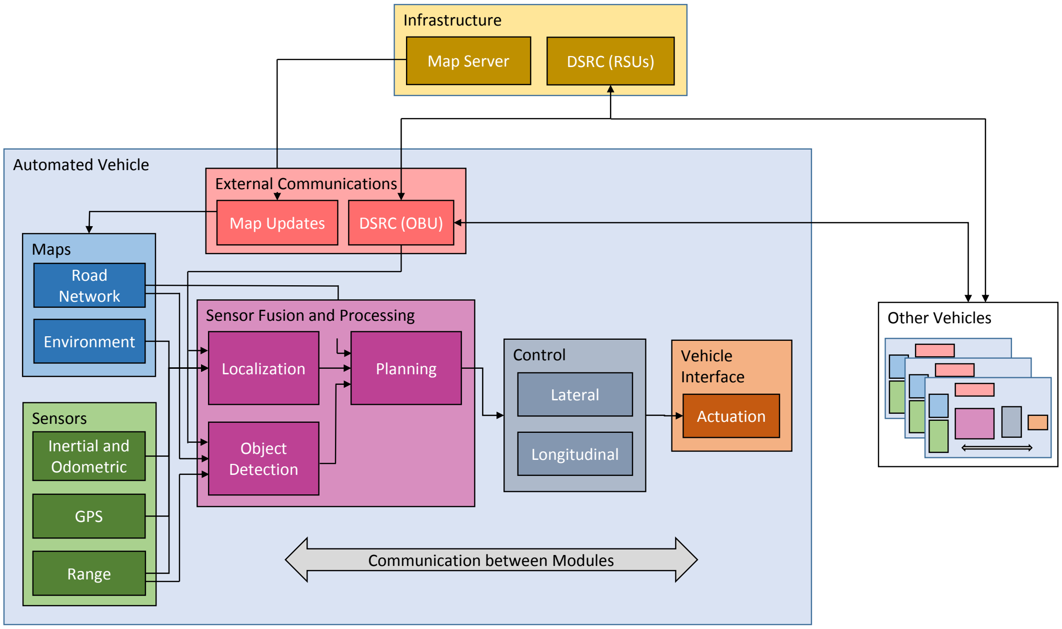

Figure 1: An interconnection diagram of our AD reference architecture. Rectangles represent functional blocks which take in, process, and output data. Data flow is represented using directional arrows. Best viewed in color.

- *Object Detection*: This module detects, classifies, and tracks objects in the vehicle's environment. Range sensors are used to identify objects near the vehicle which can be classified as either static (e.g. buildings, signs) or dynamic (e.g. vehicles, pedestrians, cyclists). In addition to locating static objects for avoidance, this module also processes static road infrastructure such as traffic lights and signs. Dynamic objects are tracked over time to facilitate prediction of their future paths.

- *Path Planning*: This module plans the vehicle's trajectory using the output of localization and object detection. We can further subdivide the path planner into 3 sub-modules: a route planner, a behavioral planner, and a motion planner [36, 27, 3, 8]. The route planner generates a path using the abstract road network map given a high level goal. The behavioral planner refines the path from the route planner by observing driving rules and accounting for obstacles. The motion planner computes inputs to the vehicle interface to track the refined path from the behavioral planner.

The final output of the path planning module is fed to the vehicle interface for control of the vehicle.

4.3 Cooperative Automated Driving

Our system can interact with other vehicles and infrastructure through a Dedicated Short Range Communication (DSRC) on-board unit (OBU). The DSRC module allows for both V2V and V2I (e.g. with static DSRC road-side units (RSUs)) and may be able to provide real time information about other vehicles and objects not necessarily in line of sight, and more precise localization than GPS.

Although automated driving research in academia tends to focus on autonomous driving, cooperative driving is another enabler of automated driving. V2V and V2I communication of vehicle location can minimally augment the range sensors and maps that feed into an autonomous vehicle's environment model [28]. But, in addition, V2V and V2I allow for communication at longer ranges and past line of sight. Just as with maps, using V2V and V2I information could allow an automated vehicle to sidestep perceptual challenges such as locating and tracking vehicles and detecting the state of the traffic light.

4.4 Modeling an AD Application

As described in section 3, looking at cybersecurity risks for automated driving from an application perspective allows for a more informed characterization of the impact of a threat to the application. This reference architecture can be used to model a potential AD application at an implementation-agnostic level. The resulting model can subsequently be used in risk assessment.

An application can be modeled with this architecture by first defining its capabilities. As a simple example, the "Lane Keeping Assist" application minimally automates lateral control to keep the vehicle inside of a marked lane. Next, the necessary functional blocks are selected from Fig. 1. In our example, we would need "Sensors" (specifically "Range Sensors") to capture the road markings, "Sensor Fusion and Processing" (specifically "Localization") to perceive and identify the lanes, and "(Lateral) Control" with "Vehicle Interface (Actuation)" to compute and execute lateral control trajectories. Note that the "Localization" functional block here only performs a subset of the full localization task described in section 4.2, namely just estimating the vehicle's lateral offset from the center of the current lane.

When modeling an application, not all functional blocks need be used and the low level jobs of those functional blocks may be subsets of their fully autonomous counterpart. Thus, the fully autonomous architecture introduced in section 4.2 can generalize to lower levels of automation and to arbitrary AD applications.

5. RISK ASSESSMENT

Using the reference architecture laid out in section 4, we can model an arbitrary AD application with an understanding of both the components involved and how they interact to realize the application. As detailed in section 3, this process of defining the application under attack constitutes the first step in our proposed risk assessment framework for AD. From there, threats to the application are enumerated and characterized using the STRIDE [26] classification (section 5.1.1). For each threat, threat model parameters are determined (section 5.1.2) and a *result vector* consisting of attack potential, motivation, and impact is computed, characterizing the risk of the threat (section 5.1.3). The result vectors of identified threats quantify the recommended prioritization of security solutions to protect the application.

5.1 Threat Model

We derive our proposed threat model by combining the strengths of the NHTSA [23] and EVITA [12] automotive threat models. While NHTSA and EVITA have contributed significantly to automotive threat modeling, this work serves as a further iteration, considering new variables and introducing a new visual depiction of the threat matrix.

5.1.1 Threat Identification

Given a defined application under attack, threats to the application are identified. Each threat consists of a threat agent (the entity performing the attack) [15], one or more attack surfaces, and one or more attack methods. Note that threat identification is not just a one time process [23], and new threats may be identified in later passes.

Threat Agent.

Potential threat agents targeting AD applications are listed in Table 1 with their primary motivations and capabilities. We note that the enumeration of attackers described here is not complete, and the motivations and capabilities assigned to each attacker may be changed under different assumptions. The capabilities associated with the agent of a particular threat will be used as parameters in the threat model to help determine the motivation and attack potential (see section 5.1.2). Unlike NHTSA and EVITA, we include a threat agent in our threat model to capture the different motivations and capabilities of potential attackers. The type of attacker will inform on the likelihood of an attack.

Attack Surface.

Several attack surfaces of AD systems are listed in Table 2 with characteristics about their resiliency and potential for attack. These will be used with the attack method and modeled AD application to estimate the resources required for the threat agent to successfully execute the attack. As with threat agents, these required resources will be used as parameters in the threat model.

The characterization of attack surfaces in Table 2 is determined from the reference architecture in section 4. "Remote Access" describes whether the attack surface can be reached remotely. "Expertise Required" qualifies the minimum level of expertise necessary to launch an attack on the attack surface. "Redundancy" lists other sources of information to corroborate or check information from the attack surface. And, "Relevant Attack Methods" lists some STRIDE classified attacks to the attack surface.

Attack Method.

Although a threat agent may pursue many different attack scenarios, we follow the convention of the NHTSA threat model [23] and categorize the attack method(s) using the STRIDE classification [26]: ***S**poofing Identity, **T**ampering with Data, **R**epudiation, **I**nformation Disclosure, **D**enial of Service,* and ***E**levation of Privilege.*

5.1.2 Threat Matrix

The result vector associated with a threat (which characterizes its risk) is modeled as a function of threat model parameters in a threat matrix. Each of the components of the result vector (attack potential, motivation, and impact) are broken down into parameters which can be determined either individually or jointly by the threat agent, attack surface, attack method, and targeted AD application.

Following [12] and [14], we establish numerical scales for each of our parameters and represent each component of the result vector as a weighted linear combination of its constituents. This enables simpler comparison of result vectors and provides a way to visualize relative levels of risk (see section 5.1.3). While the weights and numerical scales are arbitrary, we present our choices as used in the example risk assessment in section 6.

Attack Potential.

The attack potential P captures the difference between the threat agent's ability to execute a successful attack and the system's ability to withstand the attack. Each constituent of attack potential has one term for "Attacker Potential" and another for "System Withstand Potential". Many of the "Attacker Potential" parameters can be determined directly from the choice of threat agent in Table 1. "System Withstand Potential" parameters need to be determined using a combination of the attack surface characteristics in Table 2, the attack method, and the model of the AD application under attack. As in [14], the numerical scale for each constituent is an integer from 0 to 3 corresponding to the quantization levels (as in [12]) given below. The weights w are all set to 1.

- *Time Elapsed*: For the attacker, time required to identify a vulnerability and mount a successful attack [12] ($p_{a,t}$); for the system, time required to understand the system ($p_{s,t}$). Quantized as Minutes, Hours, Days, or Months.

- *Finances*: For the attacker, availability of finances with the attacker ($p_{a,f}$); for the system, minimum finances required to launch a successful attack ($p_{s,f}$). Quantized as None, Low, Medium, or High.

- *Expertise*: For the attacker, their skill level ($p_{a,ex}$); for the system, the required level of skill ($p_{s,ex}$). Quantized as Layman, Proficient, Expert, or Multiple Experts.

- *Knowledge of the System*: For the attacker, the level of knowledge of the system available ($p_{a,k}$); for the system, the level of knowledge necessary ($p_{s,k}$). Quantized as Public, Restricted, Sensitive, or Critical.

- *Window of Opportunity*: For the attacker, the maximum time available to attack ($p_{a,w}$); for the system,

Table 1: Threat agents. Refer to section 5.1.2 for details on the parameters used to characterize threat agents.

Threat Agent	**Motivations**	**Finances**	**Expertise**	**Knowledge of System**	**Equipment**
Thief	Financial (e.g. car or identity theft)	Low	Layman	Public	Standard
Owner (unlimited access to vehicle)	Financial (e.g. by performance tuning), Passion	Low	Layman	Public	Standard
Organized Crime	Financial	High	Proficient	Restricted	Bespoke
Mechanic	Financial (e.g. force vehicle into more maintenance than necessary)	Low	Expert	Critical	Specialized
Hacktivist	Ideology, Passion	Low	Multiple Experts	Sensitive	Multiple Bespoke
Terrorist	Ideology	Low	Layman	Public	Standard
Foreign Government	Financial, Ideology	High	Multiple Experts	Restricted	Multiple Bespoke

Table 2: Attack surfaces specific to autonomous and cooperative AD.

Attack Surface	**Remote Access**	**Expertise Required**	**Redundancy**	**Relevant Attack Methods**
Inertial / Odometric Sensors	No (internal)	Proficient (understanding of inertial sensor, ability to infiltrate vehicle sensor data channels)	Other inertial / odometric Sensors; range sensors localizing in map	Spoofing, Tampering (provide false sensor data); Denial of Service (jam sensor data channel)
Range Sensors	Partial (when in range and field of view)	Proficient (understanding of range sensor)	Inertial / odometric sensors; other range sensors; V2V/V2I; map	(in addition to those of inertial / odometric Sensors); Denial of Service (blind or jam from a distance)
GPS	Yes (within GPS range)	Layman (understanding of GPS, aided by commercially available jamming tools [7])	Inertial / odometric sensors; Range sensors localizing in map	Denial of Service (jamming); Spoofing
Map Update (*over-the-air*)	Yes (within wireless range)	Expert (understanding of map localization and encoding, ability to craft and transmit adversarial map updates)	Range sensors for environment perception	Spoofing, Elevation of Privilege (posing as map server); Tampering (modifying update messages); Denial of Service (jamming update channel)
V2V/V2I (*e.g. surrounding vehicle locations, traffic light state, ...*)	Yes (within DSRC range)	Proficient (ability to sniff, transmit, or modify DSRC packets)	Range sensors when in line of sight; None otherwise	Spoofing, Tampering, Information Disclosure, Denial of Service

the minimum access time necessary ($p_{s,w}$). Quantized as Short, Medium, Long, or Unlimited.

- *Equipment*: For the attacker, the equipment available ($p_{a,eq}$); for the system, the equipment required ($p_{s,eq}$). Quantized as Standard, Specialized, Bespoke, or Multiple Bespoke.

The attack potential term of the result vector is:

$$P = \sum_{j \in \{t,f,ex,k,w,eq\}} w_{p,j} g\left(p_{a,j} - p_{s,j}\right)$$

where $g(x)$ is the unit step function: $g(x) = 1 \ \forall x \geq 0$ and $g(x) = 0$ otherwise. $g(x)$ ensures that the attack potential correctly prevents an attacker with a single very high-valued "Attacker Potential" parameter from balancing out the detrimental effects of the other low-valued parameters.

Motivation.

Motivation M captures both the motivations and deterrents for the threat agent to execute the attack. As with attack potential, the numerical scale for each constituent is an integer from 0 to 3 corresponding to the quantization levels given below. The weights w are all set to 1.

- *Financial Gain* (m_f): Motivation of a financial reward. Quantized as None, Low, Moderate, or High.
- *Ideology* (m_i): Motivation driving hacktivists and terrorists. Quantized as None, Individual, Businesses, or Public.
- *Passion* (m_p): Motivation driving owners, mechanics, and some hacktivists. Quantized as None, Without Harm, Safety Implications, Criminal Intent.
- *Risk* (m_r): Deterrent of being caught. Quantized as None, Low, Moderate, High.

The motivation term of the result vector is:

$$M = \left(\sum_{j \in \{f,i,p\}} w_{m,j} m_j \right) - w_{m,r} m_r$$

Impact.

Impact I captures the loss to the stakeholders. As with attack potential, the numerical scale for each constituent is an integer from 0 to 3 corresponding to the quantization levels. All impact constituents are quantized as None, Low, Medium, High.

- Financial Loss (i_f)
- Privacy (i_p)
- Safety (i_s)

As in [14], the weight for privacy is kept at 1 while the weights for safety and financial loss are set to 2 to highlight the severe consequences of high impact in those areas. The impact term of the result vector is:

$$I = \sum_{j \in \{f,p,s\}} w_{ij} i_j$$

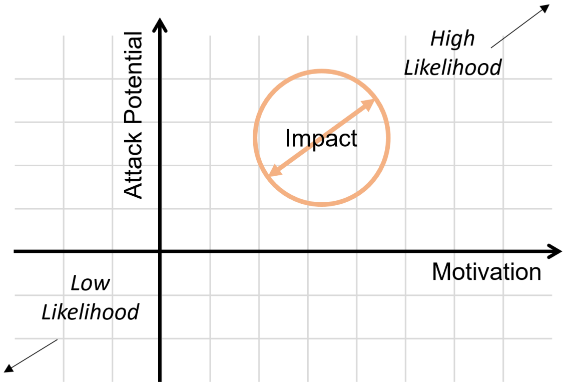

Figure 2: Threat matrix visualization. An example threat is shown as an orange circle: its center captures the Motivation and Attack Potential while its size captures the Impact.

5.1.3 Result Vector

As described above, weighted linear combinations of the threat matrix parameters output the result vector: *Attack Potential* (P), *Motivation* (M), and *Impact* (I). We can visualize this 3 dimensional characterization of risk in a plot as shown in Fig. 2.

Although risk is often defined by the two-dimensional likelihood and impact, we implicitly split up likelihood into motivation and attack potential. This lets us better understand the components of likelihood given that we know the typical motivations and abilities of the threat agent. In addition, the split gives us more insight into "medium likelihood" threats: while low motivation / high attack potential and high motivation / low attack potential threats seem equally "likely", high motivation / low attack potential threats are unique in that potential threat agents would be interested in developing a successful attack later on. In the future, as technology improves and such attacks become accessible, they may be more likely.

5.1.4 Customization

Many components of this threat model are designed to be customizable to different assumptions. For instance, while we base our characterizations of AD attack surfaces on our study of the state of the art, the specific values assigned to each parameter are not set in stone. This also goes for our list and characterization of threat agents and choices for the hyperparameters in our threat matrix (weights and choices of scale). We maintain these choices for the example risk assessment in this work (section 6), but we note that these choices are subject to assumptions and may be customized for different needs.

6. APPLICATION ASSESSMENT

Our AD reference architecture and threat model together comprise a risk assessment framework for assessing AD applications. We demonstrate its use by examining an example application, driverless valet parking. Following the proposed framework (outlined in section 5), we will define the application using our AD reference architecture, enumerate several representative threats to the application, and characterize the threats in a threat matrix and visualization that captures the relative risk to the application.

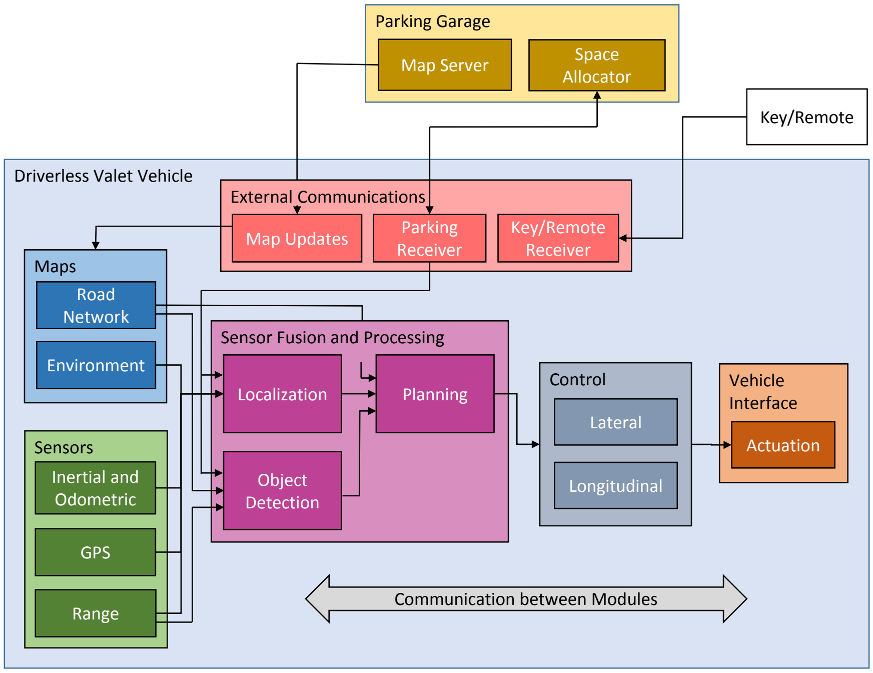

Figure 3: Architecture diagram for driverless valet parking. Best viewed in color.

6.1 Description and Model

Driverless valet parking, as defined in [4], is a level 4 AD application in which a human driver, having arrived at a parking garage, can exit the vehicle, initiate parking with a remote, and have the vehicle park in (and later retrieve itself from) the garage. Infrastructure in the garage for managing space allocation and serving maps supports this application [8]. When the maneuver is initiated by the driver, the vehicle communicates with the parking garage infrastructure to receive maps of the area and a high level goal (at which space to park). The vehicle then autonomously navigates to the parking space. Once the driver is ready to retrieve the vehicle, the driver initiates retrieval again with a remote, prompting the vehicle to autonomously return to the pick up location.

Fig. 3 shows the architecture diagram for a driverless valet parking vehicle. The interconnection diagram is very similar to the highly autonomous vehicle from Fig. 1. As a highly automated application, this vehicle performs many of the functions described in section 4 including localizing onto a map of its surroundings (receiving map updates over-the-air), building a model of its environment using its sensor payload, and autonomously planning trajectories. A few key changes are noticeable: first, the vehicle communicates with a central parking server in the garage which manages parking spots and serves maps; second, a key/remote (belonging to the owner) is introduced as a way for the owner to initiate automated parking and retrieval.

6.1.1 *Risk Assessment*

Given our understanding of the targeted application, we can now identify threats for analysis. Table 3 displays the threat matrix entries for threats we identified for this application. Recall that a threat is defined by a threat agent, attack surface(s), and attack method(s). For each threat, we picked a threat agent that we believe would be an exemplar. We then used Table 1 to fill in values of attack potential (attacker capability) and motivation (of attacker) in the threat matrix. Similarly, we used the characteristics of the attack surface as in Table 2 to help fill in attack potential (system withstand) and impact (to stakeholders). Here, we will discuss the threats in more detail and use the data from these tables in addition to the model of the application to justify our understanding of the attack potential, motivation, and impact of each threat.

- *Spoof GPS*: A thief seeking to steal an autonomously parking car spoofs the GPS signal. GPS jamming and spoofing are well known attacks with equipment for this purpose available commercially [7] and used militarily [38]. Expertise and knowledge would not be as necessary, and the equipment is not particularly specialized. The attacker would be motivated by the prospect of financial gain from a car theft. However, we know from the reference architecture that GPS is one of many sources of information about the vehicle's global location (the other key one being range sensors localizing in an environment map). As discussed in section 4.2, GPS is often only used for initialization and not relied on for precise localization. Thus, a spoofing GPS attack would not have much effect.

- *Modify Map via Update*: A hactivist group aims to steal an autonomously parking car by updating its map with false information, forcing it to plan a route toward an arbitrary destination. Due to the increased complexity of map based information and map updates, this attack requires far more resources to complete. But the impact is substantial—highly autonomous vehicles rely heavily on maps to sidestep perceptual challenges.

Table 3: Threat matrix for driverless valet parking. In parentheses next to each parameter is the corresponding numerical value as specified in section 5.1.2.

Attack Scenario					
Attack Name	*Spoof GPS*	*Modify Map via Update*	*Replay Retrieval*	*Blind Range Sensor*	*DoS Parking Space Allocator*
Threat Agents	Thief	Hacktivist	Thief	Terrorist	Hacktivist
Attack Surface	GPS	Map (over-the-air update)	Key/Remote	Range Sensor	Infrastructure (Parking Server)
Attack Method	Spoofing	Tampering	Spoofing	Denial of Service	Denial of Service, Elevation of Privilege
Description	Spoof GPS to lead parking vehicle to arbitrary location for car theft.	Modify map through over-the-air update by adding phantom obstacles to cause a failure of the application.	Replay recorded retrieval signal from target's key/remote to initiate automated valet of someone else's vehicle.	Blind range sensor (e.g. camera, LIDAR) to induce a crash.	Denial of service to parking space allocation server to induce a freeze of the automated valet service.
Attack Potential (System Withstand)					
Time Elapsed	Minutes (0)	Months (3)	Days (2)	Minutes (0)	Days (2)
Finances	Low (1)	High (3)	Low (1)	None (0)	Medium (2)
Expertise	Layman (0)	Expert (3)	Proficient (1)	Layman (0)	Expert (2)
Knowledge of System	Public (0)	Critical (3)	Restricted (1)	Public (0)	Sensitive (2)
Window of Opportunity	Short (0)	Medium (1)	Long (2)	Short (0)	Medium (1)
Equipment	Standard (0)	Multi Bespoke (3)	Specialized (1)	Standard (0)	Bespoke (2)
Attack Potential (Attacker Capability)					
Time Elapsed	Days (2)	Months (3)	Days (2)	Days (2)	Months (3)
Finances	Low (1)	Low (1)	Low (1)	Low (1)	Low (1)
Expertise	Layman (0)	Multi Experts (3)	Layman (0)	Layman (0)	Multi Experts (3)
Knowledge of System	Public (0)	Sensitive (2)	Public (0)	Public (0)	Sensitive (2)
Window of Opportunity	Short (0)	Medium (1)	Short (0)	Short (0)	Long (2)
Equipment	Standard (0)	Multi Bespoke (3)	Standard (0)	Standard (0)	Multi Bespoke (3)
Motivation (of Attacker)					
Financial Gain	High (3)	None (0)	High (3)	None (0)	None (0)
Ideology	None (0)	Businesses (2)	None (0)	Individual (1)	Public (3)
Passion	None (0)	Safety Implications (2)	None (0)	Safety Implications (2)	None (0)
Risk	Moderate (2)	Low (1)	Moderate (2)	Moderate (2)	Low (1)
Impact (to Stakeholders)					
Financial	None (0)	High (3)	High (3)	None (0)	Low (1)
Privacy	None (0)	Low (1)	Low (1)	None (0)	None (0)
Safety Violation	None (0)	High (3)	None (0)	None (0)	None (0)
Result Vector					
Attack Potential	6	4	2	6	5
Motivation	1	3	1	1	2
Impact	1	14	8	1	3

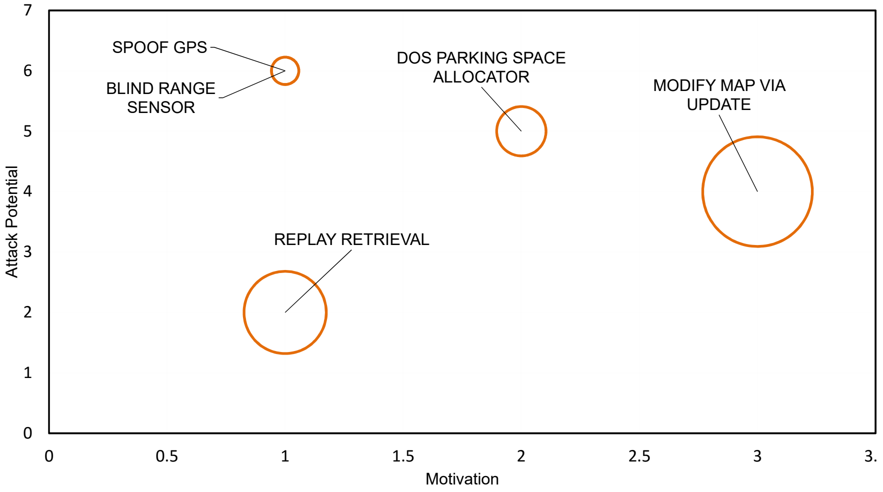

Figure 4: Threat matrix visualization for driverless valet parking.

- *Replay Retrieval*: A thief replays a recorded retrieval signal from the owner's key/remote to initiate the automated retrieval process. Although the "key/remote" is not an attack surface specific to automated driving, we can treat it as any other infrastructure that communicates wirelessly with the vehicle's external communications module. Replaying or spoofing the retrieval signal to steal the car bypasses many of the challenges of dealing with the AD system–the only system that this attack deals with directly is the key interface.

- *Blind Range Sensor*: A terrorist seeking to induce a crash could blind a range sensor on the vehicle. Assuming redundancy with other range sensors and the entire sensor payload, a single sensor failure is not damaging. In addition to not having the desired impact, such an attack requires physical access, limiting the window of opportunity and increasing the risk of being caught.

- *DoS Parking Space Allocator*: A hacktivist group seeking to induce a traffic jam or to freeze the driverless valet service can launch a denial of service attack on the parking space allocation server. The hacktivist group (based on the assumptions of Table 1) would have the resources to mount such a large scale attack on the service.

Fig. 4 visualizes the result vectors of the threats in question. From this plot, we see that our model characterizes the attacks in our list with the highest attack potential ("Spoof GPS" and "Blind Range Sensor") as having low impact. The attacks that demand prioritization are "DoS Parking Space Allocator" and "Modify Map via Update". Because the driverless valet application relies heavily on maps and the parking garage server, these attack surfaces become impactful targets. By incorporating an understanding of how these components realize the application in question, we have a more informed characterization of the risks of attacks exploiting them.

7. DISCUSSION

As stated in section 3, our goal wish risk assessment of AD is to determine how to prioritize the research and appropriation of security solutions to protect future automated vehicles. We note three insights toward this goal that were realized during the development of our reference architecture and our assessment of other AD applications. In addition to driverless valet parking, we looked at automated parking at levels 1 and 2, platooning (level 3), and the urban robot taxi (level 5) to sample a broad range of potential AD applications.

First is the importance of redundancy and not establishing too much trust in any one subsystem. In the case of platooning, for example, vehicles maintain close headway by relying on V2V for communication past line of sight. Without line of sight, local range sensors cannot check this data. This gives rise to threats to the V2V channel that have a direct impact on safety (i.e. a man in the middle attack on the V2V channel can remotely induce a platoon collision [1]). In an AD application, whether it is on-board perception, map-based localization (as in driverless valet parking), or V2V/V2I based sensing (as in platooning), reliance on any one subsystem can be exploited.

Second is the need for secure external communication. This includes, but is not limited to, over-the-air map updates, over-the-air firmware updates, V2V/V2I, key/remote signals, and other infrastructure communication (e.g. the parking garage server in driverless valet parking). While this consideration is not unique to automated vehicles and has been considered heavily in the case of purely connected and cooperative vehicles, the potential impact of external communication attacks on automated vehicles is especially high. Authentication and encryption are paramount considerations for the communications of any production AD system and infrastructure.

Third is the need for separation between safety-critical and non-safety-critical subsystems. Although we did not focus on software vulnerabilities in this work, consider the case of a malicious agent embedding malware into an over-the-air map update. If the system that received this update was connected without firewall or segregation to the rest of the automated driving software, the malicious agent immediately has access to this full stack and can potentially make the vehicle do anything. In defense, critical subsystems should be distributed from the rest with proper access control between any necessary bridges. And, as noted above, proper authentication and encryption protocols should be employed for any data that could potentially reach critical subsystems (i.e. map and firmware updates).

8. CONCLUSION

Automated driving is a new and promising research direction for the field of security. As AD starts to become more ubiquitous, it becomes increasingly important to address the gap we have in understanding its security concerns. In this work, we aim to make progress toward this goal by detailing and demonstrating a risk assessment framework for AD applications consisting of an AD reference architecture and threat model. In future work, we would apply this framework over more threat agents, attack scenarios, and applications. We would look into improving our approach by reducing the number of hyperparameters (e.g. weights in the threat matrix), making the model less reliant on subjective assumptions and non-specific terminology, and packaging the model into a completely automated threat enumeration and assessment tool. We would also explore a better characterization of impact, particularly as relates to privacy loss (which, unlike safety and financial impact, is not easily measured). We believe that the results of this and future AD application assessments will guide the design of the secure automated driving architectures that will inevitably and quickly become necessary.

9. REFERENCES

[1] M. Amoozadeh, A. Raghuramu, C.-n. Chuah, D. Ghosal, H. M. Zhang, J. Rowe, and K. Levitt. Security vulnerabilities of connected vehicle streams and their impact on cooperative driving. *IEEE Communications Magazine*, 53(6):126–132, June 2015.

[2] L. Apvrille, R. El Khayari, O. Henniger, Y. Roudier, H. Schweppe, H. Seudié, B. Weyl, and M. Wolf. Secure automotive on-board electronics network architecture. In *FISITA 2010 world automotive congress, Budapest, Hungary*, volume 8, 2010.

[3] A. Bacha, C. Bauman, R. Faruque, M. Fleming, C. Terwelp, C. Reinholtz, D. Hong, A. Wicks, T. Alberi, D. Anderson, S. Cacciola, P. Currier, A. Dalton, J. Farmer, J. Hurdus, S. Kimmel, P. King, A. Taylor, D. V. Covern, and M. Webster. Odin: Team VictorTango's entry in the DARPA Urban Challenge. *Journal of Field Robotics*, 25(8):467–492, Aug. 2008.

[4] A. Bartels, U. Eberle, and A. Knapp. AdaptIVe Delivrable D2.1: System Classification and Glossary. Technical report, Automated Driving Applications and Technologies for Intelligent Vehicles (AdaptIVe), 2015.

[5] M. d. Cava. Visiting the future in Mercedes' F 015 autonomous car. http://www.usatoday.com/story/tech/2015/03/18/mercedes-benz-f015-autonomous-car-first-ride/24964341/, 2015. [Online; accessed 2015].

[6] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, and T. Kohno. Comprehensive Experimental Analyses of Automotive Attack Surfaces. In *Proceedings of the 20th USENIX Conference on Security*, pages 6–6, San Francisco, CA, 2011. USENIX Association.

[7] J. Coffed, J. Rolli, and C. Slutsky. Detecting and Locating GPS Jamming. In *Proceedings of the ION 2015 Pacific PNT Meeting*, pages 484–492, Honolulu, Hawaii, 2015.

[8] P. Furgale, U. Schwesinger, M. Rufli, W. Derendarz, H. Grimmett, P. Muhlfellner, S. Wonneberger, J. Timpner, S. Rottmann, B. Li, B. Schmidt, T. N. Nguyen, E. Cardarelli, S. Cattani, S. Bruning, S. Horstmann, M. Stellmacher, H. Mielenz, K. Koser, M. Beermann, C. Hane, L. Heng, G. H. Lee, F. Fraundorfer, R. Iser, R. Triebel, I. Posner, P. Newman, L. Wolf, M. Pollefeys, S. Brosig, J. Effertz, C. Pradalier, and R. Siegwart. Toward automated driving in cities using close-to-market sensors: An overview of the V-Charge Project. In *2013 IEEE Intelligent Vehicles Symposium (IV)*, pages 809–816. IEEE, June 2013.

[9] A. Gibbs. Volvo to test autonomous cars in Sweden. http://www.cnbc.com/2015/03/03/volvo-to-test-autonomous-cars-in-sweden.html, 2015. [Online; accessed 2015].

[10] A. Greenberg. Hackers Remotely Kill a Jeep on the Highway-With Me in It. https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/, 2015. [Online; accessed 2015].

[11] E. Hamida, H. Noura, and W. Znaidi. Security of Cooperative Intelligent Transport Systems: Standards, Threats Analysis and Cryptographic Countermeasures. *Electronics*, 4(3):380–423, July 2015.

[12] O. Henniger, L. Apvrille, A. Fuchs, Y. Roudier, A. Ruddle, and B. Weyl. Security requirements for automotive on-board networks. In *Proc. Intelligent Transport Systems Telecommunications,(ITST),2009 9th Int. Conf*, pages 641–646, Oct. 2009.

[13] INRIA. D15.6 – Selection of offers for the city demonstrations. Technical report, Cities Demonstrating Automated Road Passenger Transport (CityMobil2), 2014. - 4 LIDARs for localization and obs detection - 4 other LIDARs for security detection? - Cameras for obs detection - odometric and inertial navigation - GPS Localization - GPS with inertial and odometric navigation - SLAM using LIDAR Comms - 2 CAN bus and Ethernet - reports location information to control center - V2V and V2I infrastructure.

[14] M. M. Islam, A. Lautenbach, C. Sandberg, and T. Olovsson. A risk assessment framework for automotive embedded systems. In *Proceedings of the 2Nd ACM International Workshop on Cyber-Physical System Security*, CPSS '16, pages 3–14, New York, NY, USA, 2016. ACM.

[15] ISO/IEC. Information technology – security techniques – evaluation criteria for it security – part 1: Introduction and general model. Technical Report ISO/IEC 15408-1: 2009, ISO, 2009.

[16] T. Jeske. Floating Car Data from Smartphones: What Google and Waze Know About You and How Hackers Can Control Traffic. In *Proceedings of Black Hat Europe*, pages 1–12, 2013.

[17] K. Koscher, A. Czeskis, F. Roesner, S. Patel, T. Kohno, S. Checkoway, D. McCoy, B. Kantor,

D. Anderson, H. Shacham, and S. Savage. Experimental Security Analysis of a Modern Automobile. In *2010 IEEE Symposium on Security and Privacy*, pages 447–462. IEEE, 2010.

[18] F. Kunz, D. Nuss, J. Wiest, H. Deusch, S. Reuter, F. Gritschneder, A. Scheel, M. Stübler, M. Bach, P. Hatzelmann, C. Wild, and K. Dietmayer. Autonomous driving at ulm university: A modular, robust, and sensor-independent fusion approach. In *Proc. IEEE Intelligent Vehicles Symp. (IV)*, pages 666–673, June 2015.

[19] J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher, E. Frazzoli, A. Huang, S. Karaman, O. Koch, Y. Kuwata, D. Moore, E. Olson, S. Peters, J. Teo, R. Truax, M. Walter, D. Barrett, A. Epstein, K. Maheloni, K. Moyer, T. Jones, R. Buckley, M. Antone, R. Galejs, S. Krishnamurthy, and J. Williams. A perception-driven autonomous urban vehicle. *Journal of Field Robotics*, 25(10):727–774, Oct. 2008.

[20] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt, M. Sokolsky, G. Stanek, D. Stavens, A. Teichman, M. Werling, and S. Thrun. Towards fully autonomous driving: Systems and algorithms. In *2011 IEEE Intelligent Vehicles Symposium (IV)*, pages 163–168, Baden-Baden, Germany, June 2011. IEEE.

[21] J. Levinson, M. Montemerlo, and S. Thrun. Map-Based Precision Vehicle Localization in Urban Environments. In *Proceedings of Robotics: Science and Systems*, pages 121–128, Atlanta, GA, 2007.

[22] J. Levinson and S. Thrun. Robust vehicle localization in urban environments using probabilistic maps. In *Proc. IEEE Int Robotics and Automation (ICRA) Conf*, pages 4372–4378, May 2010.

[23] C. McCarthy, K. Harnett, and A. Carter. Characterization of Potential Security Threats in Modern Automobiles: A Composite Modeling Approach ((Report No. DOT HS 812 074). Technical Report DOT HS 812 074, National Highway Traffic Safety Administration, Washington, DC, Oct. 2014.

[24] P. McDaniel, N. Papernot, and Z. B. Celik. Machine learning in adversarial settings. *IEEE Security & Privacy*, 14(3):68–72, May 2016.

[25] J. D. Meier, A. Mackman, and B. Wastell. Cheat Sheet: Web Application Security Frame. http://msdn.microsoft.com/en-us/library/ff649461.aspx, 2005. [Online; accessed 2015].

[26] Microsoft Developer Networks. The Stride Threat Model, 2005.

[27] M. Montemerlo, J. Becker, S. Bhat, H. Dahlkamp, D. Dolgov, S. Ettinger, D. Haehnel, T. Hilden, G. Hoffmann, B. Huhnke, D. Johnston, S. Klumpp, D. Langer, A. Levandowski, J. Levinson, J. Marcil, D. Orenstein, J. Paefgen, I. Penny, A. Petrovskaya, M. Pflueger, G. Stanek, D. Stavens, A. Vogt, and S. Thrun. Junior: The Stanford entry in the Urban Challenge. *Journal of Field Robotics*, 25(9):569–597, Sept. 2008.

[28] NHTSA. V2V communications fact sheet. Technical Report 11078-101414-v2a, National Highway Traffic Safety Administration (NHTSA), 2014.

[29] Open Web Application Security Project. Threat Risk Modeling - OWASP. https://www.owasp.org/index.php/Threat{_}Risk{_}Modeling, 2010. [Online; accessed 2015].

[30] J. Petit, D. Broekhuis, and M. Feiri. Connected Vehicles: Surveillance Threat and Mitigation. In *Black Hat Europe 2015*, Amsterdam, Netherlands, 2015.

[31] J. Petit and S. E. Shladover. Potential Cyberattacks on Automated Vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):1–11, 2014.

[32] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl. Remote Attacks on Automated Vehicles Sensors: Experiments on Camera and LiDAR. In *Black Hat Europe 2015*, Amsterdam, Netherlands, 2015.

[33] SAE. Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems. Technical report, SAE International, 2014.

[34] Tesla. Model S Software Version 7.0. https://www.teslamotors.com/presskit/autopilot, 2016. [Online; accessed 2016].

[35] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, K. Lau, C. Oakley, M. Palatucci, V. Pratt, P. Stang, S. Strohband, C. Dupont, L.-E. Jendrossek, C. Koelen, C. Markey, C. Rummel, J. van Niekerk, E. Jensen, P. Alessandrini, G. Bradski, B. Davies, S. Ettinger, A. Kaehler, A. Nefian, and P. Mahoney. Stanley: The robot that won the DARPA grand challenge. *Journal of Field Robotics*, 23(9):661–692, 2006.

[36] C. Urmson, J. Anhalt, D. Bagnell, C. Baker, R. Bittner, M. N. Clark, J. Dolan, D. Duggins, T. Galatali, C. Geyer, M. Gittleman, S. Harbaugh, M. Hebert, T. M. Howard, S. Kolski, A. Kelly, M. Likhachev, M. McNaughton, N. Miller, K. Peterson, B. Pilnick, R. Rajkumar, P. Rybski, B. Salesky, Y.-W. Seo, S. Singh, J. Snider, A. Stentz, W. Whittaker, Z. Wolkowicki, J. Ziglar, H. Bae, T. Brown, D. Demitrish, B. Litkouhi, J. Nickolaou, V. Sadekar, W. Zhang, J. Struble, M. Taylor, M. Darms, and D. Ferguson. Autonomous driving in urban environments: Boss and the Urban Challenge. *Journal of Field Robotics*, 25(8):425–466, Aug. 2008.

[37] J. Wei, J. M. Snider, J. Kim, J. M. Dolan, R. Rajkumar, and B. Litkouhi. Towards a viable autonomous driving research platform. In *Proc. IEEE Intelligent Vehicles Symp. (IV)*, pages 763–770, June 2013.

[38] E. Yagdereli, C. Gemci, and A. Z. Akta. A study on cyber-security of autonomous and unmanned vehicles. *The Journal of Defense Modeling and Simulation: Applications, Methodology, Technology*, Mar. 2015.

[39] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller, E. Kaus, R. G. Herrtwich, C. Rabe, D. Pfeiffer, F. Lindner, F. Stein, F. Erbs, M. Enzweiler, C. Knoppel, J. Hipp, M. Haueis, M. Trepte, C. Brenk, A. Tamke, M. Ghanaat, M. Braun, A. Joos, H. Fritz, H. Mock, M. Hein, and E. Zeeb. Making bertha drive —an autonomous journey on a historic route. *IEEE Intelligent Transportation Systems Magazine*, 6(2):8–20, Apr. 2014.