Merging safety and cybersecurity analysis in product design

The MIT Faculty has made this article openly available. Please share how this access benefits you. Your story matters.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>As Published</td>
<td><a href="http://dx.doi.org/10.1049/iet-its.2018.5323">http://dx.doi.org/10.1049/iet-its.2018.5323</a></td>
</tr>
<tr>
<td>Publisher</td>
<td>Institution of Electrical Engineers (IEE)</td>
</tr>
<tr>
<td>Version</td>
<td>Author’s final manuscript</td>
</tr>
<tr>
<td>Accessed</td>
<td>Mon Apr 15 11:02:18 EDT 2019</td>
</tr>
<tr>
<td>Citable Link</td>
<td><a href="http://hdl.handle.net/1721.1/119161">http://hdl.handle.net/1721.1/119161</a></td>
</tr>
<tr>
<td>Terms of Use</td>
<td>Creative Commons Attribution-Noncommercial-Share Alike</td>
</tr>
<tr>
<td>Detailed Terms</td>
<td><a href="http://creativecommons.org/licenses/by-nc-sa/4.0/">http://creativecommons.org/licenses/by-nc-sa/4.0/</a></td>
</tr>
</tbody>
</table>
Abstract: When developing cyber-physical systems such as automated vehicles, safety and cybersecurity analyses are often conducted separately. However, unlike in the IT world, safety hazards and cybersecurity threats converge in cyber-physical systems; a malicious party can exploit cyber-threats to create extremely hazardous situations, whether in autonomous vehicles or nuclear plants. We propose a framework for integrated system-level analyses for functional safety and cybersecurity. We present a generic model named Threat Identification and Refinement for Cyber-Physical Systems (TIRCPS) extending Microsoft’s six classes of threat modelling including Spoofing, Tampering, Repudiation, Information Disclosure, Denial-of-Service and Elevation Privilege (STRIDE). TIRCPS introduces three benefits for developing complex systems: first, it allows the refinement of abstract threats into specific ones as physical design information becomes available; Second, the approach provides support for constructing attack trees with traceability from high-level goals and hazardous events to threats. Third, TIRCPS formalizes the definition of threats such that intelligent tools can be built to automatically detect most of a system’s vulnerable components requiring protection. We present a case study on an automated-driving system to illustrate the proposed approach. The analysis results of a hierarchical attack tree with cyber threats traceable to high-level hazardous events are used to design mitigation solutions.

1. Introduction

The analyses of safety hazards (which we will sometimes refer to as “hazards”) and cybersecurity threats (which we will refer to as “threats”) are usually conducted separately when developing cyber-physical systems such as autonomous vehicles (AVs). The urgent need for a combined safety and cybersecurity analyses and processes in the product life cycle is recognized not only in the automobile industry, but also in other safety-critical areas such as railway, airline, nuclear and industrial control systems, etc. [30][31][35] In the automotive domain, although original equipment manufacturers (OEMs) have been trying to improve the coordination between functional safety and cyber-security activities [1] due to the impact of increases in the level of connectivity and automation, simultaneously ensuring system safety and security is challenging because of the large number of causal relations between hazards and threats. As Parkinson et al. [2] suggest, for complex systems such as AVs, “the full extent of which sensors might be compromised and their effect on a vehicle’s function is not known.” This complicates the design of safety-critical systems because the impact of threats on system functionality must be characterized to design effective mitigation strategies. Sun et al. [32] suggest that safety and cybersecurity teams are often isolated when developing complex systems. They also propose a formalism to identify contradictory requirements between safety and cybersecurity. However, this formalism does not include a structured process, which is necessary for maintaining traceability among safety and cybersecurity requirements and mitigation solutions for design changes.

In addition to the gap between functional safety and cybersecurity activities, threat identification is often conducted in an ad hoc way with little guidance [2][3]. Although multiple approaches for threat modelling have been adapted from IT systems, such as STRIDE [5], OCTAVE [6], Attack Trees [7], etc., they provide little support on finding threats. For example, although tree structures are widely used in decomposing complex system and there have been efforts of combining fault trees used in safety and attack trees in cybersecurity analyses [29][33][34], identifying off-nominal attack scenarios where one hazard can result from more than one threat is still an open question. The generic model based on STRIDE and control structure of cyber-physical systems described in this paper bridge this gap.

We propose a structured framework that combines safety and cyber security analyses. In this framework, we present a generic model named TIRCPS, which extends the STRIDE category of threat modelling for cyber-physical systems. We claim TIRCPS presents three benefits for product development: first, it provides guidance for engineers to refine abstract threats identified during concept stages into specific ones depending on actual designs of the systems; second, the approach enables safety and security engineers to construct attack trees that provide traceability among system-level goals, hazardous events (HES) and threats; and third, TIRCPS formalizes threat definitions such that automated tools can be built to detect most of a system’s vulnerably components in need of protection. We present a case study applying TIRCPS to an automated-driving system to illustrate the proposed approach.

2. Background

2.1. Motivation and Previous Work

For integrated hazard and threat analyses, there is no industry standard that accounts for safety and cyber-security simultaneously throughout product life cycle. Although it has been suggested that safety processes can be tailored to
support cyber-security analyses in the automotive cyber-security guidance—J3061 [1], the how and the when of communication between the safety and security processes has not been broadly investigated.

Parkinson et al. [2] summarize the knowledge gap between existing approaches for ensuring cyber security and potential threats for AVs. The authors suggest that the causal relations between hazards and threats, and the severity of threats in terms of injury or losses due to incidents, are unknown. For example, the effect of applying the emergency brake unexpectedly when an AV is moving on a highway -- because of, say, spoofed on-board vehicle sensors -- differs from the effect of the same action when the AV is approaching an intersection in city streets. Additionally, the physical architecture must adapt to design changes. For example, adding vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) capabilities to a car introduces new attack methods such as the Sybil attack [8], in which an adversary spoofs multiple vehicle identities to fool nearby vehicles and roadside units [4]. This threat cannot be mitigated by sensor fusion defences. For threat identification of AVs, Petit and Shladover [3] provide a summary of perception and localization threats for AVs and infrastructure. Yan et al. [9] demonstrate wireless attacks on perception modules in the Tesla S model, and S. Petit et al. [10] successfully spoof camera and Light Detection and Ranging (LIDAR) modules used by AVs to object detection and classification. Miller and Valasek [11] investigate the security of the CAN (controller area network) and control units in existing vehicle module in the market. For general threat identification, Humayed et al. [12] provide a survey of cyber-physical approaches. Based on a control-based hazard analysis technique System-Theoretic Process Analysis (STPA), Young [13] describes a method for identifying system vulnerabilities to cyber attacks [14] named STPA-Sec. However, his method focuses on concept development and does not provide guidance on threats identification of physical components. Siegel proposes a model on generic cyber-physical security [26].

2.2. Our Contributions

Although there are previous work considering the interdependency between safety and cybersecurity and graphical methods such as fault trees and attack trees are used to combine hazards and threats in a single model, to the best of our knowledge, little work has been done to support safety and cybersecurity co-engineering to meet two critical requirements, i.e., providing guidance of deriving threat scenarios for constructing attack trees and maintaining traceability between safety and cybersecurity in the product life cycle. Specifically, this paper contributes in the following aspects.

The STRIDE-based model (Fig. 2) and template (Table 1) provide guidance for constructing and refine threats and attack scenarios for constructing attack trees. Although the solution in [29] suggests the use of STRIDE for identifying threats, it does not provide any structured processes to guide engineers. In addition, our method is focused on cyber-physical systems, as illustrated in the control structure in the Fig. 2. In addition, the derived attack tree maintains traceability between safety and cybersecurity. In section 3, we define HEs and threats formally to achieve this goal. Maintaining traceability is critical to dealing with new threats and designing mitigation solutions to resolve conflicts among hazards and threats. We provide an example of mitigating digital scenery attacks in the case study (section 4) to illustrate how an intelligent tool built on our formal framework can support engineers in these tasks.

3. A framework for threat identification and secure architecture design

We present the TIRCPS approach for threat identification of cyber-physical systems. TIRCPS builds on Microsoft’s STRIDE model and is based on a control-based structure used in safety [14] and human performance analyses [15]. Similar to the STPA-SEC and STPA hazard and security analyses techniques for concept development, TIRCPS threat identification starts from the concept phase. However, the focus of TIRCPS is system-level design, in which detailed information about physical components is available. Figure 1 summarizes this approach.

The proposed framework (see Fig. 1) starts by initiating safety and security life cycles from features defined by system goals and stakeholder needs. For example, a high-level safety goal can be that the vehicle must avoid collisions by automatically applying longitudinal and lateral control, while a goal for the normal operation of transportation systems is that autonomous manoeuvring should not hinder the traffic flow in any type of roads unless there is an emergency situation.

Based on system functionalities supporting these tasks, engineers can perform hazard analyses to identify HEs with ASIL ratings [16] and derive corresponding functional threats. For example, for a hazard that involves unexpected emergency braking when an AV is moving on the highway, one of the corresponding threats in the functional level can be that the range of obstacles or vehicles in front was manipulated without authorization.

During system-level design, the high-level threats identified during concept development can be refined to finer granularity accounting for decisions on actual design choices (e.g., choosing LIDAR or a camera for object detection and tracking). We present a STRIDE-based generic model (see Fig. 2, Table 1) that provides guidance for identifying and refining threats to create risk scenarios, presenting details in the next section. In this section we define HEs and threats formally such that the analysis results are compatible with Attack Trees, a threat model approach proposed by Schneier [7] and included in industrial guidance for automotive cybersecurity [1]. This allows intelligent algorithms and tools to be built to automatically detect the component that is most vulnerable to cyber-attacks.

An attack tree is a collection of nodes in which a root node has a child node and each node has a reference to child nodes. There are three types of child nodes: an ‘and’ node indicating that a HE or an abstract threat becomes true only if all of its child nodes are true; an ‘or’ node indicating that a hazardous event or an abstract threat can result from any of its child node; and a ‘threat’ node which represents behaviours of an adversary and its risk level. Since we adopt
Microsoft’s STRIDE model to classify threats into six categories, a tree’s threat node can have six types.

Definition 1. A Hazardous Event (HE) is a four-tuple \(< F, F^T, Co, AI >\) where \(F\) is system functionality; \(F^T\) is a four-tuple \(< PR, NPR, EL, NT >\) where \(PR, NPR, EL,\) and \(NT\) represent ways that a function can be provided undesirably including provided, not provided, too early or too late or too much or not enough, guidewords that often used in hazard analysis [14][16]; \(Co\) corresponds to the context under which a hazard occurs; \(AI\) is a three-tuple \(< Se, C, E >\) representing severity, controllability and exposure of the hazardous event respectively. Each hazardous event identified at this step is related to high-level goals. For example, the hazardous event wherein the AV provides a deceleration command unexpectedly when moving on the highway with high speed can not only cause collisions but also hinder the traffic flow, both of which violate the AV’s high-level goals.

Definition 2. A Threat (T) is a five-tuple \(< Tc, Fb, Fa, P, S >\) where \(Tc\) is the threat category corresponding to six categories in the STRIDE model; \(Fb\) represents feedback information that a given threat refers to; \(Fa\) is the parent node that can be a hazardous event or a threat; \(P\) is the attack potential; \(S\) is a ChildNode that is a three-tuple \(< T, A, O >\) where \(T\) is a Threat, \(A\) is an AND node and \(O\) is an OR node [7]. Hazardous events identified in functional safety analysis become the starting point of threat identification during concept development. Consider threats that result in HEs related to a deceleration command unexpectedly. The same attack can cause moderate injury to people and moderate damage to vehicles or persons and is difficult but possible to control in the low-speed context, while potentially fatal and very difficult to control in the latter. The determination of severity and controllability of a given threat is based on risk assessment of corresponding HEs in functional safety process.

3.1. A generic model of threat identification for cyber-physical systems

A cyber-physical system [17] often consists of computers and networks (i.e., controllers) that monitor and control physical processes through actuators based on feedback from sensors’ signals. Threat modelling for cyber-physical systems differs IT-based solutions in that threats influence each element in the control loop in different ways and thus need to be treated independently. In other words, the threat model must be customized based on unique features of and the influence on cyber physical systems, as suggested by Myagmar et al. [18]. Figure 2 depicts a generic model that extends the STRIDE model to adapt it to cyber-physical systems. Table 1 is the corresponding template that provides guidance for identifying threats on physical components such as LIDAR, cameras, etc.

Consider AV threat identification based on STRIDE. A vehicle takes inputs from environment (e.g., scenery) or feedback from on-board sensors (e.g. LIDAR, camera, inertial measurement units, etc.) and stores information in memory chips for perception and localization, etc. Then, the automated controller can make safety-critical decisions based on contextual information including route information, vehicle’s current location and detected objects at surrounding environments. It also plans trajectories and generates low-level control commands for propulsion,
steering and braking systems, etc. Each element in the control loop above is subject to different threats in STRIDE.

**Spoofing:** an adversary can spoof the environment or sensory inputs by creating fake objects that are recognized as real ones by AV’s perception systems. For example, he or she can replay laser pulses from a transmitter to jam a LIDAR or falsify the presence of an object [19]. He or she can also spoof vehicle identities or those of high-importance controllers (e.g. traffic lights or service or dispatch centers) in V2X communication, which can be disastrous for decision-making. For example, an adversary who spoofs vehicle identity in an intersection can sending false position data, causing other vehicles to execute emergency stop manoeuvres.

**Tampering:** similar to spoofing, an adversary can take advantage of vulnerabilities in the perception systems and tamper with environmental or sensory inputs to AV’s. Consider the computer-vision systems that take image data from on-board cameras to detect and classify objects. Researchers have shown that for perception systems built on deep learning techniques such as convolution neural network (CNN), even small perturbations on 2-D image [20] or 3-D printed objects [21] can cause misclassification.

**Repudiation:** repudiation involves “users who deny performing an action without other parties having any way.

---

**Fig. 2. A generic model for threat identification of cyber-physical systems based on STRIDE**

<table>
<thead>
<tr>
<th>STRIDE Category</th>
<th>Threat ID</th>
<th>HEs</th>
<th>Information acquisition</th>
<th>Information Storage &amp; Analysis</th>
<th>Decision - Making</th>
</tr>
</thead>
<tbody>
<tr>
<td>Spoofing</td>
<td></td>
<td></td>
<td>Fake objects/scenery</td>
<td>Relay/replay signals</td>
<td>Misclassification, lack of context</td>
</tr>
<tr>
<td>Tampering</td>
<td></td>
<td></td>
<td>Modified objects</td>
<td>Man-in-the-middle modification</td>
<td>Misclassification, lack of context</td>
</tr>
<tr>
<td>Repudiation</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Any spoofing, tampering and DoS cannot be traced back to the adversary</td>
</tr>
<tr>
<td>Information Disclosure</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Disclosure of credentials for V2X application</td>
</tr>
<tr>
<td>Denial of Service</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>• Disclosure of passenger data e.g., routing information</td>
</tr>
<tr>
<td>Elevation privilege</td>
<td></td>
<td></td>
<td></td>
<td>Jamming or corrupting</td>
<td>Sensor flooding</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Access to critical data or network</td>
<td></td>
</tr>
</tbody>
</table>
to prove otherwise” [5]. This category can help engineers decide what information or behaviours need to be documented and projected for future reference. For example, any signals that trigger a minimum risk condition such as emergency stopping should be securely recorded in a log that can be traced back afterwards if any incident occurs. In the event that an adversary spoofs multiple identities to send “fake” vehicle position to other vehicles through V2X channels, “watch-dog” modules (i.e., cameras) can be installed on the traffic lights at intersections or within maintenance factory to monitor adversarial behaviours.

**Information Disclosure**: the impact of information disclosure threats on cyber-physical systems goes beyond privacy issues (e.g., leakage of customers’ confidential information). For example, design information may be disclosed. Therefore, vehicle credentials for V2X applications or authentication key exchange systems must be protected throughout the product life cycle including design, production, deployment and maintenance. An adversary who accesses valid credentials can not only send false messages to other vehicles to cause chaos [25].

**Denial-of-Service**: This type of attack may focus on the sensory inputs, receivers, or networks. For example, an adversary can jam signals from/to the infrared encoder installed on wheels such that wheel speed or rotation is unavailable [22]. The charge-coupled devices (CCD) within a digital camera can be damaged by high-energy neutrons created by laser systems that are 3-metre from the target camera [19]. If this happens, computer vision systems will be (partially) disabled. In addition to damaging, an adversary can also explore physical properties of on-board sensors to disrupt their normal operation. For example, the camera might experience overexposure when facing strong intensity LED light [10] or fail to detect objects under bright background. Moreover, an adversary with valid credential for V2X communication can fake multiple identities to overload vehicular networks and reduce availability of network infrastructure.

**Elevation privilege**: This type of threats refers to the privileged access gained by unauthorized users or malwares. An adversary who gets passengers’ routing information can track their vehicle and thus more easily initiate cyber attacks, while a malware installed on on-board units (OBUs) may access to critical messages sent over the CAN bus [11].

### 3.2 Change Management

Changes occurring from concept development and system-level design to post launch have significant implication on hazard and threat analyses. Three types of changes deserve attention.

#### Changes in system components or communication channels

adding components into existing system to support new functionalities can introduce new attack surfaces utilized by adversaries. For example, adding communication modules to support V2V and V2I application can lead to “Sybil” attacks [8] where an adversary “sends false safety messages using valid security credentials” such as position and speed of other vehicles on the road.

**Context change**: contextual information is critical for hazard and threat analyses. As mentioned before, an attack on sensors that detect obstacles and other vehicles can change environmental perception and has higher level of severity and less controllability for AVs moving on a highway than those moving through a city at low speed.

**Technology evolution**: as technologies evolve, original assumptions made during threat modelling may become obsolete. For example, LIDAR advances can make

---

1 Minimum Risks condition refers to A low-risk motor vehicle operating condition to which an automated driving system automatically resorts upon either a system failure or a failure of the human driver to respond appropriately to a request to take over the dynamic driving task [23].

---

BFS(currentNode)

if currentNode is a leaf node
    risk_level = risk_level(currentNode)
    // Lookup table based on controllability, severity and probability
    Add(currentNode, Threat Priority List)
else if currentNode is an AND node
    node_AND_Combined = getCombinedNode(currentNode)
    // Engineers should evaluate the potential that all child threats of
    // an AND node occur simultaneously
    Add(node_AND_Combined, Threat Priority List)
else if currentNode is an OR node
    node_OR_MaxRisk = findMaxRiskNode(currentNode)
    // It is assumed that an adversary will choose attack with max risk level
    Add(nodeMaxRisk, Threat Priority List)
else
    childNode_list -> getchildnode(currentNode)
    for each childNode in childNode_list
        childNode.controllability = currentNode.controllability
        childNode.severity = currentNode.severity
        BFS(childNode)

---

**Fig. 3. Breath-First Search (BFS) for creating priority list of cyber threats**
it more spoof-proof [19] and thus decrease the attack potential of related threats.

Since system changes can introduce new threats or change risk levels of existing threats (controllability, severity, etc.), existing mitigation solutions should be re-evaluated to account for newly identified threats with higher risk level. Coming up with mitigation solutions are out of the scope of our discussion. This paper only gives an algorithm based on breadth-first search strategy for iterating threat nodes in the attack tree to create a priority list based on the risk level of each node (Figure 3). A visualization tool is being developed at MIT Auto-ID lab that automatically constructs and searches attack trees. This tool can also provide information about vulnerable components — components whose related threats have the highest system-level risks.

4. A Case Study on An Automated Driving System

The system used in the case study is a fully AV that implements SAE Level-4 automation [23] and provides support for V2X safety applications such as collision avoidance at intersections [24] or platooning by automatic vehicle following in a fleet [25]. Multiple goals are defined to support driving tasks under different environmental conditions, including the two appearing below:

- Goal-1: the vehicle should avoid collisions by automatically applying longitudinal and lateral control.
- Goal-2: the manoeuvring of AVs should not hinder the traffic flow in all types of roads unless there is an emergency situation.

AV functionality supporting driving tasks includes deceleration, acceleration, driver warning, and steering, etc., Fig. 4. Functional architecture compared with (zoom-in) physical architecture

Fig. 4. Functional architecture compared with (zoom-in) physical architecture

Fig. 5. Attack Tree constructed using TIRCPs approach
as shown at the top-left of Figure 4. As can be seen in the control diagram, the automated systems estimate vehicle dynamics and perceive objects in the surrounding environment from feedback information given by on-board sensors. Based on estimated vehicle states, perceived objects and environmental conditions, the automated controller plans motion and sends reference signals to low-level platforms that interact with physical modules such as the propulsion, braking subsystems.

HEs involving vehicle’s deceleration function can then be derived, as shown in Figure 5. While HE-1, and 2 can lead to collisions thus violating Goal-1, HE-3 and 4 are related to both goals and can also hinder traffic flow if any happens.

- **HE-1.** Automated driving system not providing deceleration command, providing it too late or insufficiently can cause rear-end collisions when AV is approaching an intersection.
- **HE-2.** Automated driving system not providing deceleration command, providing it too late or insufficiently can cause rear-end collisions when AV is moving in a platoon on a highway and leading vehicles apply emergency braking.
- **HE-3.** Providing deceleration command unexpectedly when the vehicle is moving in a city street (< TBD mph) and approaching an intersection.
- **HE-4.** Providing deceleration command unexpectedly when the vehicle is moving in a platoon on a highway with an average speed of (>) TBD mph.

Based on the feedback information that the automated driving controller uses for decision-making, threats can be derived in the concept level. For example, since a given vehicle decelerates based on range or position of the vehicle in front and speed of itself, four related threats at different severity levels include:

- **Threat-1.** \{Spoofing/Tampering: control signals such as traffic lights at the intersection are manipulated without authorization\}, \{Traffic light, stop sign\}, HE-1, depend on specific designs).
- **Threat-2.** \{Spoofing/Tampering: The range of obstacles or vehicles in front was manipulated without authorization\}, \{Range, Position\}, HE-2, depend on specific designs).
- **Threat-3.** \{Spoofing/Tampering: The range of obstacles or vehicles in front was manipulated without authorization\}, \{Range, Position\}, HE-3, Moderate Attack Potential).
- **Threat-4.** \{Spoofing/Tampering: The range of obstacles or vehicles in front was manipulated without authorization\}, \{Range, Position\}, HE-4, Enhanced-Basic Attack Potential).

Note that at early stage of concept development, physical designs often have not become available. Therefore, each threat node above will have no child nodes until system-level design begins. In addition, since physical sensors or communication modules used to detect objects and decide their position are not known, engineers can only estimate the attack potential of each threat based on assumed specific contexts. For example, Threat-3 is assigned with a moderate potential [1] because it refers to the context where vehicles moving in a city street, while Threat-4 has a potential of “enhanced-basic” (< moderate) as the vehicle is moving on a highway and requires more efforts from an adversary although the latter is more severe and uncontrollable than the former.

It should be noted that other types of causal factors such as design flaws could also lead to the same HEs although the process for deriving other types of causal factors is not covered in this paper. For instance, a design flaw that causes HE-3 could be that the stopping distance for collision avoidance is miscalculated due to mismatch between true and simulated vehicle dynamics.

Detailed threats in system-level are then refined from abstract ones with the guidance by generic threat model (Figure 2, Table 1), as shown in Table 2. Consider Threat-4 that causes unexpected deceleration of an AV when moving on a single lane highway or road. As suggested by the generic STRIDE-based model in Fig. 2, in addition to spoofing/tempering attacks on DSRC, on-board camera, or signals from nearby vehicles, roadside units or traffic lights at intersections, this high-level threat can result from spoofing digital objects or scenery to fool the perception system of the target vehicle. Fig. 6 gives an example in which a “video wall” consisting of monitors mounts at the back of a truck to display the scenery of oncoming traffic for safe overtaking [27]. However, since the monitors are connected to the wireless camera attached to the front of the truck, an adversary will be capable of injecting fake images if he or she can jam the connection between the camera and the monitor or even gets full control over the digital monitors as an insider. If the target vehicle recognizes the faked 2-D vehicle image as a real one, it may issue an unexpected deceleration command during overtaking. A refined threat correspond to this scenario is described in Threat-4.1 in Table 2.

All refined threats are shown in Figure 5 that provides an example of attack trees for threats on deceleration function provided by the automated controller. The tree structure maintains traceable links among HEs, high-level threats identified during concept development and leaf nodes corresponding to refined threats at a more detailed level. Each leaf node corresponds to a sensor or communication channel used for providing position and speed of other vehicles on the road or vehicle dynamics in the zoom-in view of Figure 4 (left), including LIDAR, wheel encoders, cameras, inertia measurement unit (IMU) and DSRC module.

The attack tree with threats linking to hazards can provide decision support when designing mitigation solutions. Although the implementation of this function in software tools is left as future work, the tool can leverage the priority list based on the algorithm in Fig. 3 to help engineers make architectural trade-offs. For example, to mitigate threats for unintended deceleration shown in Fig. 6, engineers need to make trade-offs between redundancies of information source used for deciding vehicle identity and position and the overall cost. Threat-4.1 can be mitigated by conducting a cross-checking of digital objects with messages of oncoming vehicles’ position received from DSRC module (mitigation solution-1 in Fig. 6). However, this solution has dependencies with Threat-3.5 in that both of them can lead to the same high-level Threat-4, as indicated by the red dotted line in Fig. 5. Such dependencies raise two questions. First, engineers need to consider the priority of mitigation solution for each threat, i.e., which threat has higher risk-level? Specifically, if an adversary is
Table 2 Example refined threats derived from STRIDE-based generic model

<table>
<thead>
<tr>
<th>ID</th>
<th>Threat</th>
<th>HE</th>
<th>STRIDE Category</th>
<th>Stages</th>
<th>Description &amp; Mitigations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Threat-1.1</td>
<td>Threat-1</td>
<td>HE-1</td>
<td>Elevation Privilege Tampering</td>
<td>Comm. with infrastructure</td>
<td>An adversary pre-installs malicious malware on roadside units modified to send false traffic light signals through DSRC such that it indicates green light when the real light is red</td>
</tr>
<tr>
<td>Threat-1.2</td>
<td>Threat-1</td>
<td>HE-1</td>
<td>Denial of Service</td>
<td>Sensory inputs</td>
<td>AV may lose computer vision under saturation attack on camera. Redundancy is necessary for object detection. Moreover, AV should move to a safe location for maintenance.</td>
</tr>
<tr>
<td>Threat-2.X</td>
<td>Threat-2</td>
<td>HE-2</td>
<td>......</td>
<td>......</td>
<td>......</td>
</tr>
<tr>
<td>Threat-3.1</td>
<td>Threat-3</td>
<td>HE-3</td>
<td>Spoofing Tampering</td>
<td>Environment</td>
<td>An adversary spoofed traffic signs received by camera</td>
</tr>
<tr>
<td>Threat-3.2</td>
<td>Threat-3</td>
<td>HE-3</td>
<td>Spoofing Tampering</td>
<td>Sensory inputs</td>
<td>Replay original pulses to generate counterfeit signals to LIDAR for creating fake objects or masking real ones</td>
</tr>
<tr>
<td>Threat-3.3</td>
<td>Threat-3</td>
<td>HE-3</td>
<td>Spoofing</td>
<td>Comm. with other vehicles</td>
<td>An adversary spoofed DSRC to create a fake vehicle in-front by replaying recorded signals</td>
</tr>
<tr>
<td>Threat-3.4</td>
<td>Threat-3</td>
<td>HE-3</td>
<td>Tampering Info. Disc. Elev. P.</td>
<td>Comm. with other vehicles</td>
<td>An adversary manipulates inputs about vehicle dynamics into CAN before transmitting through DSRC to create fake vehicles in-front or to provide false vehicle position, speed, or deceleration</td>
</tr>
<tr>
<td>Threat-3.5</td>
<td>Threat-3</td>
<td>HE-3</td>
<td>Info. Disclosure</td>
<td>Comm. with other vehicles</td>
<td>An adversary spoofs vehicle identifies to create a fake vehicle in-front because he or she get the credentials by eavesdropping DSRC communication between a particular vehicle and others or stealing keys</td>
</tr>
<tr>
<td>Threat-3.6</td>
<td>Threat-3</td>
<td>HE-3</td>
<td>Elevation Privilege Tampering</td>
<td>Comm. with infrastructure</td>
<td>An adversary pre-install malicious malware on roadside units modified to send false traffic light signals through DSRC such that it indicates red light when the real light is green</td>
</tr>
<tr>
<td>Threat-4.1</td>
<td>Threat-4</td>
<td>HE-4</td>
<td>Spoofing Tampering</td>
<td>Environment Sensory inputs</td>
<td>Digital monitors attached to the back of truck in front display fake scenery or vehicle</td>
</tr>
</tbody>
</table>

An insider and has full control over the truck, he or she may dissuade the oncoming vehicle from broadcasting positioning information to the target vehicle because it is in a relatively “better” position [28]. Therefore, the mitigation solution for Threat-3.5 needs to be decided before considering Threat-4.1. Second, which information or feedback channel should have higher confidence level or is more trustable? Another modified solution (i.e., mitigation solution-2) can be built by augmenting the road with magnetic sensors that detect vehicle occupancy and put roadside units to broadcast vehicle positions to nearby vehicles. However, this solution only becomes economically viable if it can bring other benefits such as supporting vehicle platooning.

5. Conclusion
This paper proposes a structured framework that combines safety and cyber security analyses. The framework includes a generic model, named TIRCPS that is developed based on the STRIDE model. It provides guidance of identifying and deriving detailed threats of cyber-physical systems. The formal definitions of HEs and threats in TIRCPS based on formal logic provides support for constructing an attack tree that maintains traceability among high-level hazards, low-level threats and mitigation solutions. A BFS-based algorithm is also developed to construct a priority list from the attack trees. This priority list can help engineers make trade-offs when designing mitigation solutions.

6. Reference