Threat Modeling and Attack Simulations of Connected Vehicles: A Research Outlook

Wenjun Xiong, Fredrik Krantz and Robert Lagerström
School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden

Keywords: Security Architecture, Design Analysis, Threat Awareness, Vulnerability Analysis.

Abstract: Modern vehicles are dependent on software, and are often connected to the Internet or other external services, which makes them vulnerable to various attacks. To improve security for Internet facing systems, holistic threat modeling is becoming a common way to proactively make decisions and design for security. One approach that has not been commonly implemented is to enhance the threat models with probabilistic attack simulations. That is, incorporating security intelligence, attack types, vulnerabilities, and countermeasures to get objective security metrics and risk assessments. This combination has been shown efficient in other disciplines, e.g. energy and banking. However, it has so far been fairly unexplored in the vehicle domain. This position paper reviews previous research in the field, and implements a vehicle threat model using a tool called securiCAD, based on which future research requirements for connected vehicle attack simulations are also derived. The main findings are: 1) not much work has been done in the combined area of connected vehicles and threat modeling with attack simulations, 2) initial tests show that the approach is useful, 3) more research in vehicle specific attacks and countermeasures is needed in order to provide more accurate simulation results, and 4) a more tailored metamodel is needed for the vehicle domain.

1 INTRODUCTION

Modern vehicles are often coupled with cellular connections to the Internet, and they contain more than a hundred Electronic Control Units (ECUs) that control brakes, airbags, parts of the engine, and so forth. This combination of ECUs, sensors, and network buses creates a computerized system. The most commonly used network in a vehicle is called Controller Area Network (CAN), and there have been several known ways to breach into this network (Currie, 2017). Vehicles seem to be vulnerable to exploits in several ways (just as other systems are), but a malicious actor getting access to vital ECUs can have dire safety consequences. Vulnerabilities have been reported numerous times, and one famous example is when Miller and Valasek acquired remote control of a 2014 Jeep Cherokee (Miller and Valasek, 2015).

One way to improve security in these Internet connected systems is to use advanced tools to model and analyze them, we are then able to know what parts of the network are the most vulnerable ones, and how they can be secured. Holistic threat modeling has become a very common way to work with proactive cyber security and security by design, e.g. taking into account software, data, infrastructure, processes, and the most recent trend in threat modeling is to couple it with attack simulations, to provide probabilistic measures to security, e.g. Time-To-Compromise (TTC) (Johnson et al., 2016c; Johnson et al., 2018). This fairly new approach has been used successfully in other domains like energy (Vernotte et al., 2018; Korman et al., 2017). However, as far as we know, it has not been developed or tested for connected vehicles. Thus, we aim to explore and answer the research questions: 1) Can holistic threat modeling and attack simulations be used for connected vehicles? 2) What future research needs to be done in order for it to be efficient and successful?

A software called securiCAD is used in this work. It is a threat modeling and risk management tool in which the user is able to model e.g. a home Local Area Network (LAN) or a large corporate network. Then security measures are assigned to different objects, and the built in simulation engine is used to show the probability of different attacks succeeding. Some attack types that can be simulated are Denial of Service (DoS), device compromise, and replay attacks (Ekstedt et al., 2015; Korman et al., 2016).

Our literature review and practical tests using securiCAD show that threat modeling and attack simulations for vehicles is promising, while some aspects need to be further considered in future research in order for it be efficient and successful.

Xiong, W., Krantz, F. and Lagerström, R.
Threat Modeling and Attack Simulations of Connected Vehicles: A Research Outlook.
DOI: 10.5220/0007412104790486
In *Proceedings of the 5th International Conference on Information Systems Security and Privacy (ICISSP 2019)*, pages 479-486
ISBN: 978-989-758-359-9

2 RELATED WORK

This section is divided into four parts, first we describe the internal network of connected vehicles, then the core work in vehicle security, after which recent trends in threat modeling and attack simulations are described, and finally the intersection between threat modeling and vehicles.

2.1 In-vehicle Network

In 2014, Miller and Valasek did a survey of attack surfaces on several automotive models (Miller and Valasek, 2014), and the most famous one is the 2014 Jeep Cherokee. We will continue to use this vehicle model as a running example. Besides, a typical in-vehicle network is shown in (Miller and Valasek, 2014).

The internal network of a 2014 Jeep Cherokee consists of two CAN buses (CAN-C, CAN IHS) and one LIN (Local Interconnect Network) bus. The CAN protocol applied by this vehicle is called CAN-FD [1], which is an extension of the original CAN protocol, and allows for larger payloads and decreased latency. Moreover, it has a larger packet size and allows for some security implementations e.g. message authentication (Islinger et al., 2017). LIN is designed to complement CAN. Apart from these, MOST (Media Oriented Systems Transport) and FlexRay are also commonly used by vehicle network protocols, however, they are losing support. Overall, these network technologies create a data communication channel between different ECUs in a vehicle.

The software used on these ECUs is either made entirely by the Original Equipment Manufacturers (OEMs), or applies existing architecture standards, e.g. AUTomotive Open System ARchitecture (AUTOSAR[2]). AUTOSAR is a standardized software framework for vehicles and it offers a multi-level security architecture among others.

2.2 Vehicle Security

Previously, vehicle OEMs did not consider cyber attacks that much, since an attack was only possible if an attacker had physical access to the vehicle. However, as modern vehicles have multiple wireless connections to outside networks and devices (e.g. bluetooth, Internet), attacks are dramatically increasing.

Possible security mechanisms to secure vehicles internal communications were addressed by (HoliSec, 2017), which include message authentication codes (MAC) for traffic integrity, firewalls both for external traffic and for internal traffic implemented in gateway ECUs, use of Intrusion Detection Systems (IDSs) to detect unusual activities on the networks, and certificates for identification of various devices. Security mechanisms were also addressed by (Buttigieg et al., 2017) to mitigate the threats on assets, which include access control, packet filter firewall, message authentication, etc.

2.3 Threat Modeling and Attack Simulations

The work by (Shostack, 2014) and the Microsoft Threat Modeling tool [3] are commonly used in this area. However, they are mainly used for designing one secure software application, and not considering the system holistically, e.g. taking into account software, data, infrastructure, processes, etc. In (Williams and Yuan, 2015), the authors studied the usefulness of the Microsoft Threat Modeling tool and could show that the participants "improved their work on threat modeling with the tool compared with not using the tool".

Another way of working with threat modeling is to use attack (and defense) trees or attack graphs (Salter et al., 1998; Saini et al., 2008; Kordy et al., 2010). Although attack graphs are widely accepted and used, there are plenty of known problems. For instance, as stated in (Ou et al., 2006), "previous work on attack graphs has not provided an account of the scalability of the graph generating process, and there is often a lack of logical formalism in the representation of attack graphs, which results in the attack graph being difficult to use and understand by human beings".

As a response to the known problems in holistic threat modeling and using attack graphs for quantitative simulation, some approaches have been proposed, for example, pwnPr3d (Johnson et al., 2016d; Vernotte et al., 2017) and MAL (the Meta Attack Language) (Johnson et al., 2018) both focused on probabilistic measures.

2.4 Vehicle Threat Modeling

Threat modeling is a process that can be used to analyze potential attacks and threats. The work by (Karahasanovic et al., 2017) adapted two threat modeling methods from the computer industry, TARA and STRIDE, to fit the needs of the automotive industry.

[1] http://www.bosch-semiconductors.com/ip-modules/can-ip-modules/can-fd/

[2] https://www.autosar.org/

[3] https://www.microsoft.com/en-us/download/details.aspx?id=49168

Also, the work by (Ma and Schmittner, 2016) proposed a ”practical and efficient” approach to threat modeling to better fit the automotive systems. However, they have so far done a proof-of-concept implementation of their approach without further validation.

The process for automotive threat modeling proposed by (Park et al., 2018) starts with first defining automotive security use cases, then identifying assets and threats by using the STRIDE method, and finally rating risks and evaluating the threat level and impact level against the found threats. Besides, for assessing exploitability risks of vehicular on-board networks, the work by (Salfer and Eckert, 2018) automatically generated and analyzed attack graphs, which could aid vehicle development by automatically re-checking the architecture for attack combinations.

Furthermore, some research concentrated on designing security architectures related to vehicle security, e.g. Sancus (Noorman et al., 2013; Noorman et al., 2017), Vulcan (Van Bulck et al., 2017), SeP-CAR (Symeonidis et al., 2017) and ITU-T X.1373 [4].

3 VEHICLE MODELING AND ANALYSIS

When it comes to vehicle modeling and analysis, the first thing is to understand the internal network of a vehicle, and the main assets in it. As current threat modeling tools are more focused on architecture, and lacking of attack analysis, this section will be done with securiCAD, for both a generalized model and the 2014 Jeep Cherokee model. It can automatically generate probabilistic attack graphs from a given system specification (e.g. connected vehicles), which serves as inference engine that produces predictive security analysis results from the vehicle model.

3.1 Creating the Threat Model

The threat model of a connected vehicle can be built by using drag-and-drop functionality on pre-defined objects, and these objects will be assigned certain properties and pre-defined attacks. For example, for `Network`, there are DoS attack, ARP cache poisoning attack, compromise attack, etc. Moreover, each object has a selection of security implementations, which can be set as enabled, disabled or probability-based.

[4]https://www.itu.int/rec/T-REC-X.1373/en

Generalized Model. The generalized model of a vehicle's internal network is shown in Figure 1, which contains two CAN networks (Schweppe, 2012). Note that there are small markers on each object, which indicate other objects are connected to it.

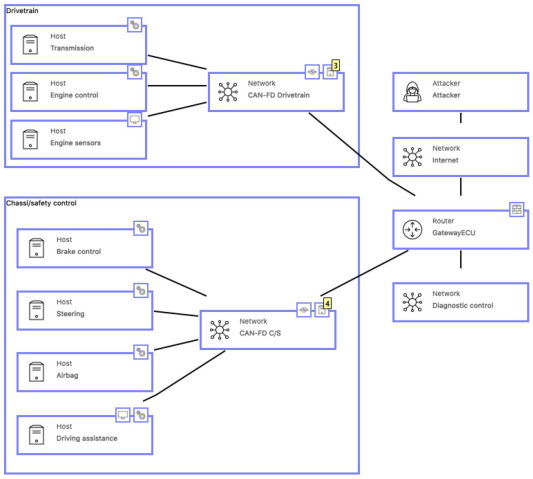


Figure 1: A generalized model of a vehicle's internal network by securiCAD.

In securiCAD, a `Host` is described as a kernel of an operating system, and is used to represent PCs or servers (Foreseeti, 2018), thus, it can be used to represent `ECU`. Besides, a `SoftwareProduct` is connected to each `Host` (not shown in Figure 1, but can be seen when double-click `ECU`), and here it represents AUTOSAR, as AUTOSAR is open-source and becoming a world standard for automotive embedded software.

In Figure 1, all `ECUs` have their specific names according to a real car. The two CAN-FD `Network` also have specific names, which are Drivetrain and Chassi/safety control. Besides, a `Router` named `GatewayECU` is connected to a `Firewall` and four `Networks` [5]. Among the four networks, there are two CAN-FD networks, an administrative network (required by securiCAD), and Internet (is available through the use of a Connectivity Control Unit attached to the gateway, which is not represented by an object in the model).

A `Firewall` has two security measurements, `Enabled` and `KnownRuleSet` (if the firewall ruleset is known to the modeler and configured properly). The default settings are enabled and `KnownRuleSet` is set as probability = 0.5, as no public information is available about how manufacturers configure their firewalls on Gateway ECUs.

[5]https://www.bosch-mobility-solutions.com/en/products-and-services/passenger-cars-and-light-commercial-vehicles/connectivity-solutions/central-gateway-cgw/

In order to model the broadcast behavior of the CAN network, ECUs are connected to a CAN-FD Network. Besides, Network has security measurements including DNSSec, PortSecurity and StaticARPTables that are TCP/IP related, but CAN-FD itself has no security measurements enabled.

In this generalized model, Service and Client can be connected to ECU, but an ECU do not require both of them. To be more specific, an ECU will be connected to Client only when it is required to send data to other ECUs. For example, the Driving Assistance ECU is without control over any electrical devices, and aims to calculate for the driving assistance functions based on input and send output to ECUs that handles driving functions. Both Service and Client have a security measurement called Patched and is enabled. Also, SoftwareProduct is connected to them, which means all Service and Client applied AUTOSAR standard.

Moreover, Dataflow is connected to CAN-FD Network, and is also connected to Service and Client, which represents the communication between Service and Client. The communication denotes how much access that Service and Client have to commands and function calls in the operating system and kernel. The setting applied the most secure option, as no information was found about how much access does a service on an ECU in AUTOSAR has.

Similarly, a Protocol is connected to Dataflow, which gives options to choose different security implementations to apply on the communication over the CAN-FD networks, and the security measurements available on Protocol are Authenticated, Encrypted and Nonce. Here, Authenticated is supported by CAN-FD Network and enabled, while the other two are disabled.

Furthermore, an Attacker is added to the threat model to make it complete. The Attacker is connected to the object where an attack start at. In this case we consider Internet as unsafe, therefore the Attacker is connected to the Internet Network with the connection type Compromise.

Overall, the security settings related to the model in Figure 1 are as follows:

- The security settings of Host:
  - ASLR: Address Space Layout Randomization (ASLR) fortifies against buffer overflow attacks; is disabled; because it is not implemented in AUTOSAR classic, but is available on the adaptive platform.
  - AntiMalware: detects, removes and deters malware attacks; is disabled; because it is not implemented.
  - DEP: Data Excecution Prevention (DEP) defends against buffer overflow, by making memory areas non-excecutionable; is disabled; because it is not implemented in AUTOSAR classic, but is available on the adaptive platform.
  - Hardened: represents the procedures where unused services, ports and hardware outlets are disabled; is unset; because no information is available.
  - HostFirewall: a firewall controls whether dataflow is blocked or allowed between hosts; is unset; because no information is available.
  - Patched: it means the host has the latest security updates; is enabled; because Internet connection gives improved software support and patch availability.
  - StaticARPTables: means mapping IP address to MAC address to avoid spoofing; is disabled; because this measurement is with Ethernet network, not a CAN network.
- The security settings of SoftwareProduct:
  - HasVendorSupport: means whether the software product is supported and has access to patches; is enabled; because the model has an Internet connection and is assumed to be supported.
  - NoPatchableVulnerability: means whether the software product has no patchable vulnerabilities; is unset; because no information is available.
  - NoUnPatchableVulnerability: means whether the software product has no unpatchable vulnerabilities; is unset; because no information is available.
  - SafeLanguages: means the software product is developed in languages that perform checking to reduce the risk of buffer overflow; is unset; because no information is available.
  - Scrutinized: whether the software has been thoroughly tested and checked for vulnerabilities; is unset; because no information is available.
  - SecretBinary: whether there is an access to the binary by an attacker who can then detect vulnerabilities; is unset; because no information is available.
  - SecretSource: whether the source code is a secret source; is disabled; because AUTOSAR is an open-source.
  - StaticCodeAnalysis: whether there is a code analysis tool to find vulnerabilities and bugs; is unset; because no information is available.
- Firewall connected to GatewayECU has KnownRuleSet enabled, and its probability is set to 0.5.
- The security settings of Network are disabled.

- `Service` and `Client` connected to `ECU` have `Patched` enabled.
- `Dataflow` connected to `Network` has `Authenticated` enabled.

2014 Jeep Cherokee Model. According to the topology by (Miller and Valasek, 2014), Figure 2 is created and shows the threat model of 2014 Jeep Cherokees's internal network, with some changes on the generalized model (Figure 1). Here, CAN networks are used instead of CAN-FD networks (Miller and Valasek, 2015), so `Authenticated` is disabled from the security settings of its network `Protocol`. The `Firewall` connected to Radio `ECU` is enabled, even though only one open port is accessible. But the open ports found by Miller and Valasek can be represented by disabling `Hardened` setting of `ECU`.

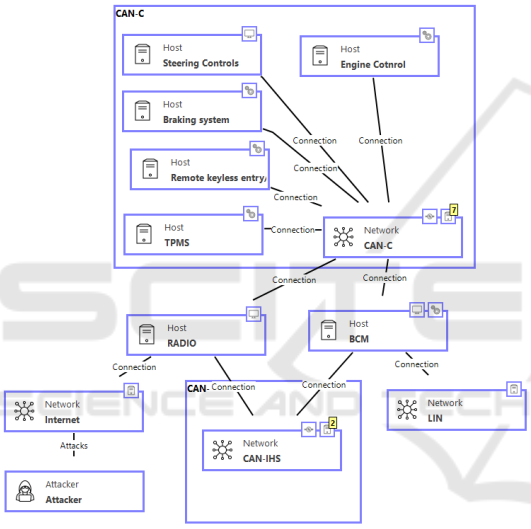
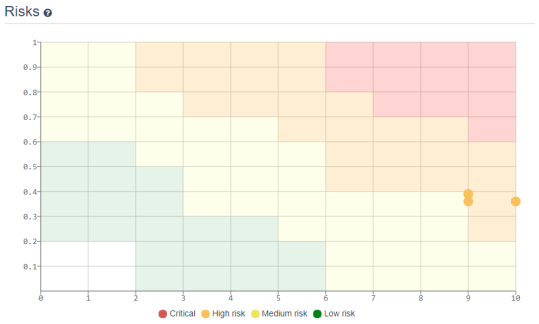


Figure 2: Internal network model of 2014 Jeep Cherokee by securiCAD.

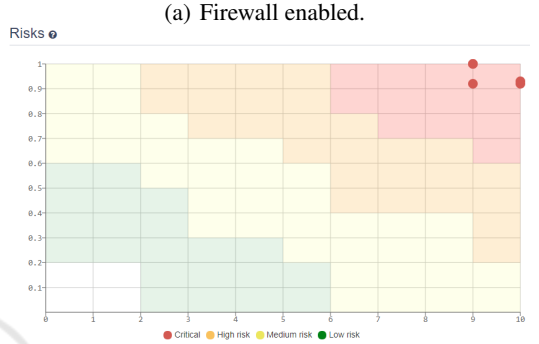
3.2 Running the Attack Simulations

Generalized Model. To conduct attack simulations, the attack consequence (from 0 to 10, while 10 indicates the most severity) is also required to set for each object. For Engine control, Transmission and Brake control `ECU`, the consequence of a compromise attack is set to 10, because a compromise and an access to these ECUs and services could lead to fatal road accidents. For CAN-FD `Network`, the consequence of a DoS attack is set to 9, because a DOS attack can shut down the access to ECUs of the network, and it will not lead to fatal road accidents compared to the formal one.

By running securiCAD attack simulations, the risk assessment is shown in Figure 3(a), and all attacks are considered to be of High risk. To show the consequence if the `Firewall` connected to `GatewayECU` is disabled, that all attacks are considered in the Critical zone (shown in Figure 3(b)). The result shows that `Firewall` is the most important object to secure the network.



(a) Firewall enabled.



(b) Firewall disabled.

Figure 3: Risk matrix from simulations performed on the generalized model.

The simulation results also show the attack path of an attack, which is aggregated by attack graphs to model the composition of vulnerabilities found in a system. For example, the attack steps for a DoS attack on the Drivetrain network is shown in Figure 4, with the `Firewall` enabled and `FirewallKnownRuleSet` set as probability=0.5.

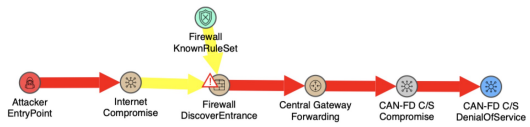


Figure 4: Attack path of a DoS attack on a CAN-FD network.

In Figure 4, the measurements that can be made to further improve security is shown by a green circle. In this case, it is related to `FirewallKnownRuleSet`. If it is set as 1.0, there would be 0 risk for all attacks.

2014 Jeep Cherokee Model. In order to simulate the attack consequences of the Jeep model, the settings are:

- Consequence of a compromise attack on Radio `ECU` is set to 3, which is used as a reference to see how much probability lowers after the initial entry of the network.
- Consequence of a compromise attack on Braking system `ECU` is set to 10.
- Consequence of a DoS attack on CAN-C `Network` is set to 10.
- Consequence of a replay attack on CAN-C `Network` is set to 10, which represents the actual attack made by Miller and Valasek (Miller and Valasek, 2015) where they to send commands over the network unhindered.

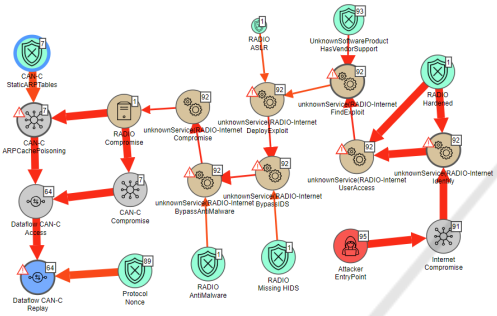


Figure 5: Attack path of the Jeep replay attack.

Figure 5 indicates the attack path of a replay attack on CAN-C `Network`, and the unknown service indicates the D-bus service accessed in an actual attack by (Miller and Valasek, 2015), they looked at a service (reflected by UnknownService in Figure 5) connected to D-bus, and discovered that D-bus was running as root, which helps them got access rights to connected systems and hacked the vehicle remotely. Just before the replay attack step, the attack paths are divided into Compromise path and ARPCachePoisoning path, while the compromise path is more likely to happen as ARP (Address Resolution Protocol) is not applied in CAN network.

Besides, several security measurements that could be implemented are shown by the green circles, and most of them are related to Radio `ECU`. Moreover, `Hardened` setting of Radio `ECU` is the most important, as it allows the attack to happen in the first place.

Furthermore, TTC is a measure of the effort expended by an attacker for a successful attack assuming effort is expended uniformly. The attacker will then take the shortest path, i.e. the least time consuming way to the end node. The TTC of the replay attack can be seen in Figure 6, which indicates how many days it takes to reach a certain risk probability. In this case, a replay attack takes 21 days and has a 50% probability to compromise the vehicle.

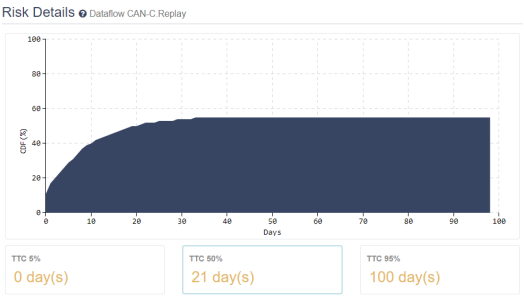


Figure 6: TTC of the Jeep replay attack.

The overall attack simulation results show that the modeled vehicles are not fully secure. According to the risk matrix, a firewall is the most important object to secure in this network. Also, the attack path shows what other security measurements that can be improved.

4 RESEARCH OUTLOOK

Holistic threat modeling and quantitative attack simulations of Internet connected vehicles seems promising, which allows a holistic identification and ranking of all the security related threats that are most likely to affect the systems. However, future work requires to be done in order for it to become efficient and useful.

A metamodel describes the fundamental assets and their associations of systems (e.g. connected vehicles). Thus, the threat modeling metamodel needs to be tailored (Lagerström et al., 2009) to fit the internal architecture of them, as most threat modeling metamodels today are created for office IT or similar systems, and they only reflect parts of a vehicle system.

The set of attack types and associated countermeasures (defenses) related to each asset in a vehicle needs to be further explored and validated. Some attacks are known for web applications or Windows-based systems, but they might not be relevant for vehicles (Checkoway et al., 2011; Välja et al., 2017). Also, there might be certain attacks only possible on vehicle systems. When it comes to countermeasures, a vehicle has certain limitations on its performance, cost and functionality that do not appear in other larger systems.

Quantitative measures of security (e.g. TTC or Time between vulnerability disclosure (TBVD) (Johnson et al., 2016a)) require quantitative inputs in order to provide reasonable and useful output. Although it has been done for other system types, vehicle specific statistical studies relating attacks and defenses quantitatively (and probabilistic) are still need

to be done. This can be realized through hacking exercises or experts.

Another important step is to validate and test the approach with case studies by modeling vehicles and iteratively enhancing the approach, a similar work has been done in the energy domain (Blom et al., 2016).

Currently, the proposed approaches have focused on automating the attack graph generation and thus reducing modeling efforts significantly. However, modeling holistic systems with many components can still be time-consuming and error-prone, thus, the approach needs to get aid in decreasing the model instantiation effort (Närman et al., 2009). This can be achieved by automatic modeling (Holm et al., 2014; Välja et al., 2015) and using reference architecture models (Korman et al., 2016; Vernotte et al., 2018), however, neither of these has been tested within the vehicle domain. To enhance the precision, it could also be useful to couple this approach with databases containing known vulnerabilities (Johnson et al., 2016b; Lagerström et al., 2017a).

Once the modeling is done with a higher degree of automation, and the security analysis is done automatically using the attack simulations, the final step of designing a more secure architecture remains to be done (Lagerström et al., 2017b).

5 CONCLUSION

This position paper first addresses the security issues on Internet connected vehicles. Then threat modeling and attack simulations are conducted through an advanced tool named securiCAD, for both a generalized vehicle model and a 2014 Jeep Cherokee. The simulation results show that this tool is useful in modeling a vehicle's internal network.

This work also has its limitations, for example, the Vehicle-to-Everything (V2X) communication is out of the scope, and some security mechanisms have not been implemented in the threat modeling metamodel, which influences the scope of the attack simulations.

Future work includes providing more accurate simulation results for vehicle-specific attacks and countermeasures. Moreover, a more tailored metamodel for vehicle is needed, which include more security mechanisms like access control, data privacy models.

REFERENCES

Blom, R., Korman, M., Lagerström, R., and Ekstedt, M. (2016). Analyzing attack resilience of an advanced meter infrastructure reference model. In *Cyber-Physical Security and Resilience in Smart Grids (CPSR-SG), Joint Workshop on*, pages 1–6. IEEE.

Buttigieg, R., Farrugia, M., and Meli, C. (2017). Security issues in controller area networks in automobiles. In *18th international conference on Sciences and Techniques of Automatic control and computer engineering*, pages 1–6.

Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., Kohno, T., et al. (2011). Comprehensive experimental analyses of automotive attack surfaces. In *USENIX Security Symposium*, pages 77–92. San Francisco.

Currie, R. (2017). Hacking the can bus: Basic manipulation of a modern automobile through can bus reverse engineering. The SANS Institute, InfoSec Reading Room report series.

Ekstedt, M., Johnson, P., Lagerstrom, R., Gorton, D., Nydrén, J., and Shahzad, K. (2015). Securi cad by foreseeti: A cad tool for enterprise cyber security management. In *2015 IEEE 19th International Enterprise Distributed Object Computing Workshop (EDOCW)*, pages 152–155. IEEE.

Foreseeti (2018). *SecuriCAD 1.4 User Manual*. Foreseeti AB, Sveavgen 166, 3TR, 113 46 Stockholm, 5,3 edition.

HoliSec (2017). A state-of-the-art report on vehicular security.

Holm, H., Buschle, M., Lagerström, R., and Ekstedt, M. (2014). Automatic data collection for enterprise architecture models. *Software & Systems Modeling*, 13(2):825–841.

Islinger, T., Mori, Y., Neumller, J., Prisching, M., and Schmidt, R. (2017). Autosar secoc for can fd. *CAN Newsletter*.

Johnson, P., Gorton, D., Lagerström, R., and Ekstedt, M. (2016a). Time between vulnerability disclosures: A measure of software product vulnerability. *Computers & Security*, 62:278–295.

Johnson, P., Lagerström, R., and Ekstedt, M. (2018). A meta language for threat modeling and attack simulations. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, page 38. ACM.

Johnson, P., Lagerström, R., Ekstedt, M., and Franke, U. (2016b). Can the common vulnerability scoring system be trusted? a bayesian analysis. *IEEE Transactions on Dependable and Secure Computing*, (1):1–1.

Johnson, P., Vernotte, A., Ekstedt, M., and Lagerström, R. (2016c). pwnpr3d: an attack-graph-driven probabilistic threat-modeling approach. In *Availability, Reliability and Security (ARES), 2016 11th International Conference on*, pages 278–283. IEEE.

Johnson, P., Vernotte, A., Gorton, D., Ekstedt, M., and Lagerström, R. (2016d). Quantitative information security risk estimation using probabilistic attack graphs. In *International Workshop on Risk Assessment and Risk-driven Testing*, pages 37–52. Springer.

Karahasanovic, A., Kleberger, P., and Almgren, M. (2017). Adapting threat modeling methods for the automotive industry. In *Proceedings of the 15th ESCAR Conference*, pages 1–10. Chalmers Publication Library.

Kordy, B., Mauw, S., Radomirović, S., and Schweitzer, P. (2010). Foundations of attack–defense trees. In *International Workshop on Formal Aspects in Security and Trust*, pages 80–95. Springer.

Korman, M., Lagerström, R., Välja, M., Ekstedt, M., and Blom, R. (2016). Technology management through architecture reference models: A smart metering case. In *Management of Engineering and Technology (PICMET), 2016 Portland International Conference on*, pages 2338–2350. IEEE.

Korman, M., Välja, M., Björkman, G., Ekstedt, M., Vernotte, A., and Lagerström, R. (2017). Analyzing the effectiveness of attack countermeasures in a scada system. In *Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, pages 73–78. ACM.

Lagerström, R., Baldwin, C., MacCormack, A., Sturtevant, D., and Doolan, L. (2017a). Exploring the relationship between architecture coupling and software vulnerabilities. In *International Symposium on Engineering Secure Software and Systems*, pages 53–69. Springer.

Lagerström, R., Franke, U., Johnson, P., and Ullberg, J. (2009). A method for creating entreprise architecture metamodels: applied to systems modifiability. *International Journal of Computer Science and Applications*, 6(5):89–120.

Lagerström, R., Johnson, P., and Ekstedt, M. (2017b). Automatic design of secure enterprise architecture: Work in progress paper. In *Enterprise Distributed Object Computing Workshop (EDOCW), 2017 IEEE 21st International*, pages 65–70. IEEE.

Ma, Z. and Schmittner, C. (2016). Threat modeling for automotive security analysis. *Advanced Science and Technology Letters*, 139:333–339.

Miller, C. and Valasek, C. (2014). A survey of remote automotive attack surfaces. *BlackHat USA*.

Miller, C. and Valasek, C. (2015). Remote exploitation of an unaltered passenger vehicle. *BlackHat USA*.

Närman, P., Johnson, P., Lagerström, R., Franke, U., and Ekstedt, M. (2009). Data collection prioritization for system quality analysis. *Electronic Notes in Theoretical Computer Science*, 233:29–42.

Noorman, J., Agten, P., Daniels, W., Strackx, R., Herrewege, A. V., Huygens, C., Preneel, B., Verbauwhede, I., and Piessens, F. (2013). Sancus: Low-cost trustworthy extensible networked devices with a zero-software trusted computing base. pages 479–498. USENIX.

Noorman, J., Van Bulck, J., Mühlberg, J. T., Piessens, F., Maene, P., Preneel, B., Verbauwhede, I., Götzfried, J., Müller, T., and Freiling, F. (2017). Sancus 2.0: A low-cost security architecture for iot devices. *ACM Transactions on Privacy and Security (TOPS)*, 20(3):1–33.

Ou, X., Boyer, W. F., and McQueen, M. A. (2006). A scalable approach to attack graph generation. In *Proceedings of the 13th ACM conference on Computer and communications security*, pages 336–345. ACM.

Park, J. S., Kim, D., Hong, S., Lee, H., and Myeong, E. (2018). Case study for defining security goals and requirements for automotive security parts using threat modeling. In *SAE Technical Paper*. SAE International.

Saini, V., Duan, Q., and Paruchuri, V. (2008). Threat modeling using attack trees. *Journal of Computing Sciences in Colleges*, 23(4):124–131.

Salfer, M. and Eckert, C. (2018). Attack graph-based assessment of exploitability risks in automotive on-board networks. In *Proceedings of the 13th International Conference on Availability, Reliability and Security*, pages 1–10. ACM.

Salter, C., Saydjari, O. S. S., Schneier, B., and Wallner, J. (1998). Toward a secure system engineering methodology. In *Proceedings of the 1998 workshop on New security paradigms*, pages 2–10. ACM.

Schweppe, H. (2012). *Security and privacy in automotive on-board networks*. PhD thesis, Télécom ParisTech.

Shostack, A. (2014). *Threat modeling: Designing for security*. John Wiley & Sons.

Symeonidis, I., Abdelrahaman, A., Mustafa Asan, M., Mennink, B., Dhooghe, S., and Preneel, B. (2017). Sepcar: A secure and privacy-enhancing protocol for car access provision. In *ESORICS 2017: Computer Security - ESORICS 2017*, pages 475–493. Springer.

Välja, M., Korman, M., and Lagerström, R. (2017). A study on software vulnerabilities and weaknesses of embedded systems in power networks. In *Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids*, pages 47–52. ACM.

Välja, M., Lagerström, R., Ekstedt, M., and Korman, M. (2015). A requirements based approach for automating enterprise it architecture modeling using multiple data sources. In *Enterprise Distributed Object Computing Workshop (EDOCW), 2015 IEEE 19th International*, pages 79–87. IEEE.

Van Bulck, J., Mühlberg, T., and Piessens, F. (2017). Vulcan: Efficient component authentication and software isolation for automotive control networks. pages 225–237. ACM International Conference Proceeding Series.

Vernotte, A., Johnson, P., Ekstedt, M., and Lagerström, R. (2017). In-depth modeling of the unix operating system for architectural cyber security analysis. In *Enterprise Distributed Object Computing Workshop (EDOCW), 2017 IEEE 21st International*, pages 127–136. IEEE.

Vernotte, A., Välja, M., Korman, M., Björkman, G., Ekstedt, M., and Lagerström, R. (2018). Load balancing of renewable energy: a cyber security analysis. *Energy Informatics*, 1(1):5.

Williams, I. and Yuan, X. (2015). Evaluating the effectiveness of microsoft threat modeling tool. In *Proceedings of the 2015 Information Security Curriculum Development Conference*, page 9. ACM.