Ontology Based Anomaly Detection System for Cellular Vehicular Communications

Quentin Ricard[1,2], Philippe Owezarski[1]
Laboratoire d'Analyse et d'Architecture des Systèmes (LAAS-CNRS)[1]
Continental Digital Services France[2]
Toulouse, France
firstname.lastname@laas.fr

***Abstract*—Intelligent Transportation Systems are being deployed all over the world, providing new applications and services that could prevent accident, help regulate traffic and the automotive industry in designing energy efficient vehicles. However, enabling vehicles to communicate with the rest of the world ultimately leads to new security challenges with connected vehicles becoming new interesting targets for malicious actors. Thus, safeguards need to be put in place to detect malicious and anomalous activities in vehicular communications. This paper presents an anomaly detection method based on an ontological representation of cellular vehicular communication.**

***Index Terms*—intelligent transportation systems, anomaly detection, C-V2X**

I. Introduction

In recent years, the automotive industry and the research community have engaged considerable resources on Intelligent Transportation Systems (ITS) technologies. In fact, vehicles nowadays embed interconnected networks that are responsible of complex features in a vehicle such as cruise-control, lane-tracking and assistance braking. These system make decisions based on different kind of sensors i.e., radars, cameras, brakes, thermometer. Enabling vehicles to share pieces of information from these sensors with the rest of the digital world facilities would allow the emergence of new services dedicated to improving the efficiency of transport in terms of safety, user experience and fleet monitoring. Two types of communication channels were envisioned for this task namely vehicle ad-hoc networks (VANET or V2V) and cellular vehicular networks (CVN or C-V2X).

However, introducing new communications channels to vehicles also creates new opportunities for malicious actors to disturb these new networks. In fact, there has been an ever increasing list of attacks that were successfully conducted against vehicles [1]–[5], some of them allowed remote exploitation by attackers. Specifically, the Jeep Cherokee hack [6] by Charlie Miller and Chris Valasek had a massive impact in the media and forced automotive companies to take into account cyber-security when designing their vehicles.

Moreover, malicious activities are not the only threat to connected vehicles. For example, in the case of automated driving or collision avoidance systems based on communications: vehicles and drivers will have to rely on knowledge gathered from the network in order to take complex decisions. Therefore, the integrity of data sent and received by vehicles must be verified and safeguards need to be put in place to prevent anomalies from disrupting these types of critical services.

Detecting these anomalies or intrusions in network communications has been an extensive topic of research in the past decades. The first methods were mostly based on signature [7] and deployed in traditional information and communication technology (ICT) as well as industrial control systems (ICS) networks. While providing good results on well-known attacks, these types of intrusion detection schemes are not able to detect new or variant of known attacks and rely on expert knowledg to build signatures. Therefore, statistical and machine-learning based [8] techniques were envisioned to cope with this limitation. However, these methods suffer from a higher rate of false positive compared to signature based approach, e.g. the classification of benign events as anomalies. Furthermore, these models rely on algorithms often specialized to specific types of anomalies that require expensive computation capabilities [9]. Most importantly, the lack of explainable results and accurate training dataset discourage their use in the industry [9].

However, when considering cellular vehicular networks we argue that such kind of systems could be successfully embedded in tomorrow's car. In fact, the nature of the communications occurring between vehicles and the rest of the world differs from those of traditional ICT networks. We believe that CVN inherit from both mobile and ICS networks as two types of services namely vehicle-related and user-related, are operating in the same communication channel. Vehicle-related messages are carrying high semantic meaning as they are dependent on the observation of physical events on sensors of the vehicle. On the other hand, user-related messages will mostly consist of infotainment applications communications such as music streaming, e-mails or map apps, which are closely related to current smartphone traffic. Thus, the content and frequency of vehicle-related communications are more predictable than user-related ones. Therefore, we believe that building an anomaly detector based on a model describing these the communication of the vehicle would be beneficial in terms of detection capabilities, adaptation to evolution and explainable results.

In this paper we present an anomaly detection method based on ontology representations of vehicular communications. In

section II relevant work on ontology for information security and anomaly detection in vehicular networks are presented. Then, section III introduces our anomaly detection method. Finally, section IV presents the different attack scenarios that we used to build a dataset to test our system. Conclusions and future work are presented in section V

II. Related Work

Ontologies are explicit formal specifications of terms and relations in a particular domain [10]. They have been greatly used in the World Wide Web in order to ease the search for information by automated processes (web crawlers) thanks to the use of expressive languages (RDFS, DAML, OWL ...). Such languages enable domain-specific information sharing by experts. Ontologies were used in previous work in the field of information security. In [11] the authors present the use of semantics for the detection of targeted attacks. In [12] the authors present an ontology for the detection of application level intrusion for hyper text transfer protocol (HTTP).

Authors in [13] modelled intrusions in terms of attacks directed at a particular system component caused by a defined input. Such input, has consequences on the system that are divided into two sub-classes, i.e. input validation error and exploit. The attack results are divided in a class of consequences, i.e denial of service, remote to local, user to root or probing. In their case they use the reasoner rules to detect intrusions while in our case we rely on the ontology to divide the network traffic to search for specific classes of anomalies in specific sub-set of the whole communication. Thus, we only use the reasoner to build a representation of the detected anomaly. In [14], the authors proposed a method based on an ontology and user-defined rules to represent network security situation in heterogeneous Internet of things (IOT) networks. Their goal is to provide security situation awareness based on multi-source information aggregation. In our case we rely solely on network traffic to detect anomalies in cellular vehicular communications.

III. Ontological Representation of Communications and Anomaly Detection

In this section, we introduce our anomaly detection method based on an ontological representation of the communications occurring between vehicles and the rest of the world.

A. Modelling the communications

As stated in our introduction, we believe that it is possible to formally describe the communications of a vehicle with the rest of the world. Such a representation would allow a better classification of the traffic in order to detect multiple types of anomalies. In fact, since the traffic carries high semantic meaning, it is possible to sort vehicle-related communications from user-related exchanges.

Therefore, we represent the communications at different scales, i.e. from flow to packet, as well as the entities that take part in the exchanges, i.e. network nodes. Finally, we also model the anomalies in order to ease the post analysis work and the sharing of anomalies alerts.

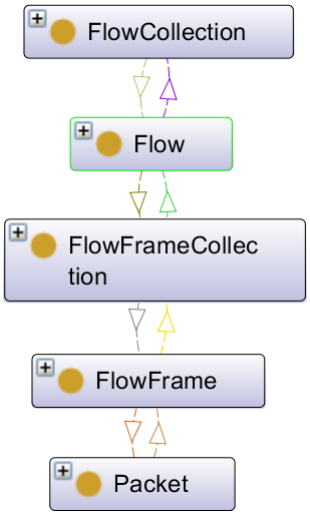

Fig. 1: Graphical Representation of the flows of packets in Protege

a) Representing flows of packets: The representation of packets and flows is depicted by Figure 1. We describe the communications as a *collection* of *flows* of *packets* occurring between specific *entities*, typically a vehicle and a server. A *flow* is composed of a collection of *frames* that are themselves composed of sequences of *packets*. This discrimination from *flows* to *packets* allows us to analyse the traffic on different scales and therefore detecting anomalies at different scale.

The main features that we create for the flows and frames consist mostly of key statistical aspects of the flow, number and size of packets in a frame, ratio of received packet over time of a frame etc. Based on our knowledge of the underlying processes that triggers the communications we will assign a type to the flow e.g. vehicle or user-related flow.

b) Representing the packets: Network packets carry a lot of information often represented in the Open Systems Interconnection model [15]. Each layer in the model corresponding to a specific function in the communication. In our work we only consider three layers of the OSI model, i.e. network, transport and application. We regroup session, presentation and application in a single layer for convenience. From each of these layers we extract interesting key informations in order to determine the semantic that the packet carries and assign it to a specific *packet class* in the ontology. For example, depending on a TCP flag, a packet could either represent a connexion establishment, closure of connexion or simply application data. When dealing with encrypted traffic we assign each packet a specific attribute in a similar fashion.

During the anomaly detection, independent processes analyse the packets and the flows. The packet analysis is based on the *packet class* that we extract from the traffic while the flow analysis is based on the features that we create based on our observations of the flows (size, number of packets etc..). Such multi-scale approach allows us to minimize the number of features that each anomaly detection process has to handle.

For instance, a volume anomaly, e.g. a denial-of-service, would be detected by comparing different features of the *frames* of a *flow* without having to consider every classes of *packet*. A sequence anomaly however, e.g. a syn-scan, would be detected by analysing usual sequences of *packets classes* inside a flow.

c) Representing the Anomalies: An important element of our model is to represent anomalies based on results from different algorithms in a way that allows sharing and understanding generated alerts.

In order to understand the anomalies we retrieve the context surrounding a detected anomaly. For example, if a packet triggered an alert it would be beneficial to an operator to get all the other packets of the same flow. In order to do so we need to propagate the anomaly associated to a *packet* instance to its corresponding *flow* instances. We use a simple inference rule on the composition relationship materialized in our ontology by the *partOf* axiom. Said axiom binds *packets* to *frames*, *flows* and *entities*:

- partOf(x, y) $\wedge$ isAnomalous(x) $\rightarrow$ isAnomalous(y)

Thus, if a *packet* is deemed anomalous, the *frame* as well as the *flow* is also categorized as anomalous. Finally, the entity that took part in the communication with the vehicle is also categorized as anomalous. Another similar rule is defined to contaminate the other flows that the anomalous *entity* might have established with the vehicle.

However, such rule could also have a detrimental effect on the understanding of the anomaly if it returns too much information, especially because of the fact that categorizing the *flow* as anomalous will retroactively contaminate every *packet* inside it. Therefore, we define a degree of abnormality that decrease depending on the distance between a *frame* where a packet was deemed as anomalous and other *frames*. Therefore, attention is drawn to the closest contextual knowledge of the anomaly.

IV. Training Dataset

In this section, we introduce our communication dataset generation procedure based on an emulation environment.

A. Emulation environment

Anomaly detection applied to network communications is well known to encounter issues when considering training datasets. Quite often, communication datasets used by researchers cannot be published due to privacy concerns or by law restrictions. Therefore, results on new methods for anomaly detection lack repeatability. Furthermore, recent datasets such as the one introduced in [16] are mostly dedicated to classical ICT networks. Thus, following the guidelines introduced by Shivari et al. in [17] we created a dataset dedicated to cellular vehicular network using an emulation environment named **Autobot**.

The emulation runs in a completely isolated environment using docker containers. Each container act a vehicle connected to a docker network as depicted in Figure 2. In order to respect cellular network behaviour `netem` [18] is used to shape the latency and bandwidth of every virtual interface for every container. **Autobot** allows us to emulate the networking behaviour of up to 300 vehicles on an server running Ubuntu 16.04 with 32 processors (2.6 GHz) and 64 gigabytes of RAM.

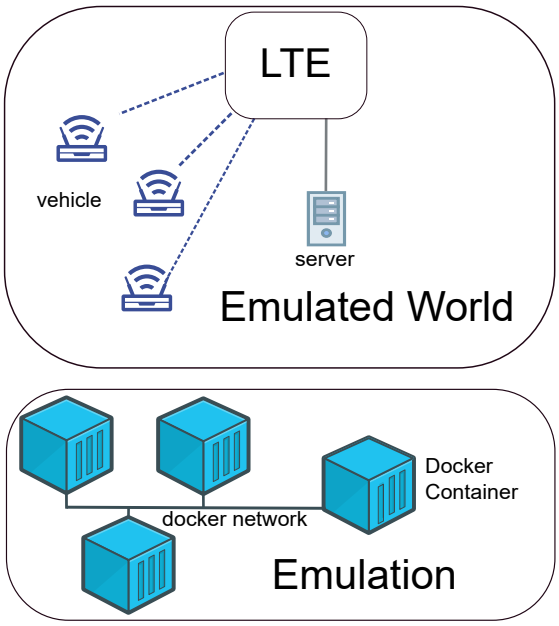

Fig. 2: Representation of the emulation environment Autobot.

We created three types of application that generates realistic traffic that could appear in connected vehicles in the future.

a) Vehicle Telemetry: In order to enable the new ITS services presented in the introduction vehicles will have to share information they gather from their sensors with remote servers. To this end, an exchange format was created by a consortium of automotive industrial. This message format, Sensoris, is meant to enable different parties to share knowledge between each other regardless of the constructor of the vehicle. We shipped inside every vehicle container a Message Queueing Telemetry Transport (MQTT) client that sends Sensoris messages to another container that act as a server. The data that is sent in this messages was extracted from real-life vehicles that are used on the E-horizon project at Continental Digital Services France for research and development purposes. The data contains information ranging from accelerometer to temperature and GPS coordinates.

b) Infotainment applications: In order to improve the experience of users when driving, vehicles will likely embed applications like music streaming. We plan on creating a containerized music streaming application based on the Spotify API in order to emulate the traffic from these applications. We will also embed map navigation applications in order to improve the realism of the communication of the emulated vehicles.

c) Updates over-the-air: Finally, in order to manage vehicles fleets, updates will need to be sent over the air to vehicles. Such application was used for users in Hurricane

Florence's path [1] by the Tesla company. It allowed them to increase the range of the vehicles that were located in that area thus facilitating their travels.

B. Emulated Anomalies

We generate our anomalies based on known attacks that were perpetrated against connected vehicles. Alongside these attacks, we also created several scenarios based on a survey introduced in [1]. Most attacks presented here require that vehicles be provided public IP address, or alternatively that mobile network operator allow device to device communications. However, we consider this pre-requisite to be very plausible as the upcoming 5G[2] planned device-to-device communications.

a) Network Scans: An attacker tries to find listening ports by sending specially crafted packets to the vehicles.

b) Remote Exploitation: Based on a discovered vulnerability an attacker is able to remotely access to the vehicle infotainment system and perform malicious activity such as shutting down the system or extracting private information about the users of the vehicle.

c) (Distributed) Denial-of-Service: An attacker tries to prevent the services of a vehicle from functioning properly by flooding its network interface.

d) Malware and Ransomware-based Attacks: Users of a vehicle inadvertently install a malicious infotainment application that act as a malware. The malware sends and receive control-and-command instruction from a remote server controlled by an attacker.

e) Anomalies in Vehicle Telemetry: We generate anomalies in the vehicle telemetry messages generated by the vehicles. For example, we prevent the vehicle from sending information regarding particular sensors (accelerometer for example), during a time interval.

Every scenario is performed independently and anomaly detection is performed using a capture of the network traffic on the interface of the attacked vehicle during the emulation.

V. Conclusion

In this paper we introduced an anomaly detection method based on an ontological representation of cellular vehicular communications. We showed how using a semantic approach to packet analysis could reduce the number of features that are involved in the detection process. In order to evaluate our approach we presented our dataset creation method based on a emulation environment dedicated to cellular vehicular networks.

The full version of this paper will contain a detailed representation of our ontology as well as the algorithms that were used for the detection of different classes of anomalies. Moreover, we will present how our model is integrated to an anomaly detection system based on IBM's Mape-K loop [19], [20]. Finally, we will present a performance comparison against other designs and algorithms used for network anomaly detection using our dataset.

[1]https://bit.ly/2Wy9PeO

[2]http://5gaa.org/

Future work on our method will involve the alert sharing process that could be put in place in order to help other vehicles detect more efficiently anomalies that were already experienced by other vehicles.

Acknowledgments

The authors wish to thank Continental Digital Services France for funding this work.

References

[1] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces."

[2] I. D. Foster, A. Prudhomme, K. Koscher, and S. Savage, "Fast and vulnerable: A story of telematic failures."

[3] S. Jafarnejad, L. Codeca, W. Bronzi, R. Frank, and T. Engel, "A car hacking experiment: When connectivity meets vulnerability," in *Globecom Workshops (GC Wkshps), 2015 IEEE*. IEEE, 2015, pp. 1–6.

[4] S. Mazloom, M. Rezaeirad, A. Hunter, and D. McCoy, "A security analysis of an in-vehicle infotainment and app platform."

[5] S. Bayer, T. Enderle, D.-K. Oka, and M. Wolf, "Security crash test-practical security evaluations of automotive onboard it components," *Automotive-Safety & Security 2014*, 2015.

[6] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," 2015.

[7] V. Paxson, "Bro: a system for detecting network intruders in real-time," *Computer networks*, vol. 31, no. 23-24, pp. 2435–2463, 1999.

[8] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1-2, pp. 18–28, 2009.

[9] R. Sommer and V. Paxson, "Outside the closed world: On using machine learning for network intrusion detection," in *Security and Privacy (SP), 2010 IEEE Symposium on*. IEEE, 2010, pp. 305–316.

[10] T. R. Gruber, "A translation approach to portable ontology specifications," *Knowledge acquisition*, vol. 5, no. 2, pp. 199–220, 1993.

[11] R. Luh, S. Marschalek, M. Kaiser, H. Janicke, and S. Schrittwieser, "Semantics-aware detection of targeted attacks: a survey," *Journal of Computer Virology and Hacking Techniques*, vol. 13, no. 1, pp. 47–85, 2017.

[12] A. Razzaq, H. F. Ahmed, A. Hur, and N. Haider, "Ontology based application level intrusion detection system by using bayesian filter," in *Computer, Control and Communication, 2009. IC4 2009. 2nd International Conference on*. IEEE, 2009, pp. 1–6.

[13] J. Undercoffer, A. Joshi, and J. Pinkston, "Modeling computer attacks: An ontology for intrusion detection," in *International Workshop on Recent Advances in Intrusion Detection*. Springer, 2003, pp. 113–135.

[14] G. Xu, Y. Cao, Y. Ren, X. Li, and Z. Feng, "Network security situation awareness based on semantic ontology and user-defined rules for internet of things," *IEEE Access*, vol. 5, pp. 21 046–21 056, 2017.

[15] H. Zimmermann, "Osi reference model-the iso model of architecture for open systems interconnection," *IEEE Transactions on communications*, vol. 28, no. 4, pp. 425–432, 1980.

[16] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization." in *ICISSP*, 2018, pp. 108–116.

[17] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *computers & security*, vol. 31, no. 3, pp. 357–374, 2012.

[18] S. Hemminger *et al.*, "Network emulation with netem," in *Linux conf au*, 2005, pp. 18–23.

[19] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *Computer*, no. 1, pp. 41–50, 2003.

[20] A. Computing *et al.*, "An architectural blueprint for autonomic computing," 2006.