Emulation of Vehicular Networks For Anomaly Detection Purposes

Quentin Ricard, Philippe Owezarski

Emulation of Vehicular Networks For Anomaly Detection Purposes

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*—This paper introduces an ontology-based communication model for the detection of anomalies occurring in cellular vehicular communications. We describe network exchange on multiple scales in order to allow the detection of different classes of anomalies. In order to test our model we built a test-bed emulation environment named AutoBot to generate realistic network traces of vehicular communications.**

***Index Terms*—network emulation, docker, anomaly detection**

I. Introduction

A. Introduction

In recent years, car manufacturers and the research community showed a lot of interest in connecting vehicles to the digital world facilities. In fact, today's vehicles embed complex networks. They consist of numerous computer components called Electronic and Telematic Control Unit (ECU, TCU). Each of these components are responsible of specific features in a vehicle (braking, steering wheel, Assistance Braking System) and operate by communicating with others using a CAN (Control Area Network). It allows vehicles to take complicated decisions based on multiple sources of information. The introduction of a new communication channel between vehicles and the rest of the internet allows the extraction of interesting pieces of information from a vehicle enabling new ways to improve traffic efficiency, safety, and energy consumption.

To this end, two types of networks were envisioned namely, Vehicular Ad-hoc Networks (VANET) and Cellular Vehicular Networks (CVN). The former was designed to provide safety applications such as collision avoidance that require low latency between vehicles. The latter, introduces two types of services. The first type is driver assistance services, e.g. by providing relevant traffic information (congestions, accidents) or monitoring a fleet of vehicle that requires vehicles to send telemetry information to remote servers. The second type of service is related to Infotainment applications, e.g. video streaming, mail services, application download. These new types of cyber-physical systems are called Intelligent Transportation Systems (ITS). In our work, we focus only on CVN.

Introducing cellular communication capabilities to vehicles ultimately creates new attack vectors against such vehicles [1]. In recent years, an ever increasing list of attacks were conducted against vehicles, some of them allowed remote exploitation by attackers. The Jeep Cherokee hack [2] by Charlie Miller and Chris Valasek is the most famous one. Thus, safeguards need to be built around the communication channels of vehicles in order to prevent such attacks to be perpetrated against them. Besides attacks, anomalies data sent from vehicles or directed at them, especially for safety applications, could also have detrimental consequences for users of the road. Thus, the integrity of the data sent and received by vehicles also need to be verified.

Detecting anomalies or intrusions in traditional Information and communications technology (ICT) networks has been a prolific realm of research during the last 20 years. It started with signature based models that are still in use today [3]. 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 knowledge to build signatures. Thus, anomaly based [4] approach that are able to detect unknown attacks and anomalies are under constant development. However, these methods suffer from a higher rate of false positive detection compared to signature based approach, e.g. the classification of benign events as anomalies. Furthermore, these models rely on machine learning algorithms often specialized to specific types of anomalies that require expensive computation capabilities [5]. Most importantly, the lack of explainable results and accurate training dataset discourage their use in the industry [6].

B. Related Work

In their paper [7] Shiravi and al. introduced the notion of profiles that contain an abstract representation of events and patterns seen on the network. They enable researchers to generate the network behaviour of numerous hosts, protocols and malicious or abnormal activity. In our work, we applied this approach to CVN.

However, reproducing such communications is challenging. Over years, the research community studied vehicle networking using several network simulators, mainly `ns3` and `omnetpp`. However, results from these experiments do not reflect realistic communication behaviour due to the small amount of devices used, combined with a small number of interactions between them. Yet, network simulation and

This work has been funded by Continental Digital Services France (CDSF)

emulation provide cost effective ways to validate new designs and application before large scale deployment. Nonetheless, most of the attention was directed towards vehicle-to-vehicle (V2V) simulation and few studies were dedicated to cellular networks only [8], [9].

Because of the rise of connected vehicles nowadays and in a very short future, new threats in ITS applications need to be studied. in the remainder of this paper we introduce an ontology based anomaly detection method for CVN in section II. Then, we introduce an environment that allows researchers to generate networking datasets dedicated to vehicular communications in section III and conclude this paper in section IV.

II. Intrusion and Anomaly Detection

A. Motivation

Based on the limitations of anomaly detection methods argued in our introduction, we designed an anomaly detection method following these requirements :

- The system must be able to detect different types of anomalies, e.g. volume, ratio, contextual and sequential anomalies [5].
- The system must explain the root cause that triggered an anomaly alert, allowing the operator to understand the anomaly.
- The system must be autonomous, i.e. adapt to the evolution of the network traffic (caused by the installation of new application by users of the car for example).
- The system must operate in real-time.

Due to the nature of the traffic to analyse, most anomaly detection system rely on identifying deviation from a standard profile. Only few studies [10], [11] tried to represent such profile in a normalized way and were mainly focused on industrial control systems (ICS) or supervisory control and data acquisition (SCADA) systems. In fact, these types of networks differs from traditional ICT networks as they are more deterministic since they operate repetitive industrial processes. We believe that CVN inherit from both ICT and ICS networks since two types of services, namely vehicle-related and user-related, are operating in the same communication channel. The main feature of vehicle-related messages is that they are carrying high semantic meaning: their content and frequency are more predictable than user-related messages. Therefore, we believe that building an anomaly detector based on a model describing these two types of communication would be beneficial in terms of detection capabilities, adaptation to evolution and explainable results. We designed an ontology to describe this communication model.

Ontologies are explicit formal specifications of terms and relations in a particular domain [12]. 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 [13].

The integration of ontologies in our work is twofold. First, it allows us to pre-classify the communication into different categories to ease the anomaly detection process in terms of computation requirement and detection abilities, e.g. the detection of the different types of anomalies requires different algorithms. Secondly, the formalism induced by the ontology supports the representation of detected anomalies in a graphical and easy to read manner. Therefore, operators will easily understand why an anomaly alert was triggered and what caused it.

B. Overview of the model

The basic representation of a communication between two nodes is a collection of *Flows* of *Packets* occurring between specific *Entities*, e.g. a vehicle and a server. A *Flow* is composed of a collection of *frames* that are themselves composed of *SequenceOfPackets*. This discrimination from *Flows* to *Packets* allows us to analyse the traffic on different scales in order to detect a wide variety of anomalies. 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 *Packet* and their payload. A sequence anomaly however, e.g. a syn-scan, would be detected by analysing usual sequences of TCP flags inside a *Flow* of *Packets*.

The ontology classes are provided with attributes corresponding to the feature of interests of the traffic. These are directly extracted from the capture of the communications. For example, a flow is either vehicle or user-related, A frame is provided with numerous statistics on the size of packets e.g. minimal, maximal, average and standard deviation, or the time between two received packet inside a frame. These features are used by algorithms in charge of the detection of specific classes of anomalies. This minimizes the number of features that each algorithm has to handle.

In order to ease the analysis process, we need to extract 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. To do so we need to propagate the anomaly associated to a *packet* instance to its corresponding *flow* instance. We use a simple inference rule on the composition relationship materialized in our ontology by the *partOf* axiom. Said axiom binds *packets* to *frames* and *flows* :

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

Thus, if a *packet* is deemed anomalous, the *PacketSequence* is also categorized as anomalous, as well as the *Frame* and the *Flow*. Other rules could be used to gather additional knowledge of a situation for the operator.

III. Autobot : Emulation Environment For Anomaly Detection Purposes

In order to test our model, we built an environment emulating communications between vehicles and a remote server. It combines several existing tools such as docker and traffic-control that respectively emulate real application behaviours,

and simulate LTE connectivity behaviour by shaping the traffic.

We tried to reproduce the experiments done in [8]. However, the NS3 simulator [14] was not able to emulate enough nodes for our needs. Thus, in order to emulate network communication in the most realistic way, we used `docker` in combination with `traffic-control`. `Docker` is responsible for the simulation of a Long-Term-Evolution network and the communications between vehicles and the remote server. In order to emulate realistic network characteristics we use `traffic-control` to shape the communications in terms of delay, bandwidth, corruption, reordering and packet loss.

Containers are operating system-level (OS-level) virtualization. They run and isolate processes, packaging their own libraries, dependencies and configurations. Since they avoid shipping an entire OS for each container, they provide a lightweight solution compared to virtual machines. In our environment, containers are connected to each other over a virtual network (docker-network).

The use of containers has several advantages. First, we are able to reproduce the networking behaviour of real applications that could be embedded in vehicles in the near future. Secondly, it reduces the amount of work that is needed to create new applications and allows researchers to develop their own containers and connect them to vehicles and server nodes. Lastly, we are also able to connect containers to the internet in order to emulate realistic use of infotainment applications, i.e. video streaming, mail services, application download, while maintaining a strong isolation of the emulated network.

A. Proof of Concept

Our simulation environment runs on a Ubuntu 16.04 with 32 processors (2.6 GHz) and 64 gigabytes of RAM. The docker containers are built and connected to each other over a virtual network. Then, we tune the connectivity parameters of each interface in order to reproduce realistic LTE connectivity behavior by calling `traffic-control` scripts written in python.

We built two types of containers. The first one is designed to act as a cloud service provider. It starts an MQTT server that waits for messages from the fleet of vehicles. The second type of container represents the network behaviour of the vehicles. At the time of writing this paper, our simulation is able to emulate the communications of 20 vehicles. The vehicles run an MQTT client that sends telemetry informations to the same MQTT server at a regular pace. These messages contains speed, and gps coordinates.

We gather all traffic from vehicles and the server using tcpdump as packet capture (pcap) files. They contain all the communication that occurred during the simulation, packets are captured once they reach the interface of the container. We are also working on embedding real-life infotainment systems inside the containers of the vehicles in order to emulate user-related communication. Moreover, we are also building containers that will act as attackers against our network based on scenarios such as scanning, malware contamination, denial-of-service etc...

IV. Conclusion and Future work

In this paper, we introduced a model for anomaly detection based on an ontology. In order to test this model we built Autobot, an emulation environment for the creation of a vehicular communication dataset.

In future work, we aim at introducing realistic connectivity changes for vehicles containers, based on a recent dataset [15]. Furthermore, adding functionalities to the containers such as more telemetry informations inside MQTT messages (fuel consumption, wheel orientation, etc...) as well as updates over the air of the infotainment system. Finally, once our dataset is accurate enough we will perform anomaly detection using our ontology to validate our approach using different algorithms.

Acknowledgment

The authors wish to thank Continental Digital Services France for funding this work. Also, we thank Clement Cassé for his technical support on docker and network namespaces.

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

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

[4] 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.

[5] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[6] 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.

[7] 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.

[8] T. Molloy, Z. Yuan, and G.-M. Muntean, "Real time emulation of an lte network using ns-3," 2014.

[9] A. Fouda, A. N. Ragab, A. Esswie, M. Marzban, A. Naser, M. Rehan, and A. S. Ibrahim, "Real-time video streaming over ns3-based emulated lte networks," *Int. J. Electr. Commun. Comput. Technol.(IJECCT)*, vol. 4, no. 3, 2014.

[10] I. Garitano, R. Uribeetxeberria, and U. Zurutuza, "A review of scada anomaly detection systems," in *Soft Computing Models in Industrial and Environmental Applications, 6th International Conference SOCO 2011*. Springer, 2011, pp. 357–366.

[11] N. Erez and A. Wool, "Control variable classification, modeling and anomaly detection in modbus/tcp scada systems," *International Journal of Critical Infrastructure Protection*, vol. 10, pp. 59–70, 2015.

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

[13] 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.

[14] G. F. Riley and T. R. Henderson, "The ns-3 network simulator," in *Modeling and tools for network simulation*. Springer, 2010, pp. 15–34.

[15] D. Raca, J. J. Quinlan, A. H. Zahran, and C. J. Sreenan, "Beyond throughput: a 4g lte dataset with channel and context metrics," in *Proceedings of the 9th ACM Multimedia Systems Conference*. ACM, 2018, pp. 460–465.