Evaluating Misbehavior Detection for Vehicular Networks

Rens W. van der Heijden, Frank Kargl
Institute of Distributed Systems
Ulm University
Ulm, Germany
Email: rens.vanderheijden@uni-ulm.de

***Abstract*—The literature proposes a large variety of different misbehavior detection mechanisms for vehicular networks, which are designed to separate attacks and faulty data from legitimate data. Most of these mechanisms are evaluated using techniques adopted from the field of intrusion detection. However, because misbehavior detection is content-oriented, and includes the detection of faulty data, it is possible that such data may be indistinguishable from an attack. This paper discusses and critically reflects upon different evaluation strategies used in the literature, and provides some recommendations for authors.**

I. Introduction

Vehicular networks are one area where misbehavior detection provides a significant benefit, because integrity is more important than confidentiality, and the data correctness plays a much more significant role than for regular networks. The literature has already developed many schemes, which are evaluated in widely different ways and with very different tools. Some are analytical, some are simulative and sometimes there are limited experimental cases.

Similarly, the data that is verified is normally application-specific, and not independent of surrounding vehicles (i.e., a ground truth cannot be established for just the particular message content that is being evaluated), which makes analytical approaches very difficult. Additionally, analytical approaches often require simplifications that could again be exploited by an attacker. Experimental approaches have different issues; experiments are expensive to perform, and cannot necessarily be applied to all settings, especially when the consequences are dangerous (e.g., attempting to trigger a crash on a highway for experimentation is not ethically or financially feasible). Nevertheless, experimental approaches can still provide some benefit in estimating the behavior of a system in an attacker-free environment, and the resulting data set can be artificially extended with attacks [1].

Work on Intrusion Detection System (IDS) validation has already seen that it is difficult to compare results between IDSs, but this is made harder by the fact that Misbehavior Detection Systems (MDSs) are much more varied, and often designed to work with data. Many other questions play a role: should detection happen on a per-packet basis or a per-node basis? Is the goal to eventually detect all attackers, or to detect the most attackers with the lowest latency? What kind of attacks are considered? Is there application logic that tries to be fault tolerant?

The evaluation of misbehavior detection mechanisms is often strongly influenced by what kind of attacks the authors were considering. This makes comparisons across published results very difficult. There seem to be no satisfactory evaluation metrics in the literature that both catch the subtleties of misbehavior detection, while still remaining general enough to be applicable to more than just a specific mechanism.

In this paper, we aim to give an overview of the most common evaluation techniques and metrics, and discuss their suitability for evaluating misbehavior detection. The contribution is two-fold; the paper can be used as reference material, but we also aim to provide a common ground that authors can use to compare their work to that of other authors.

II. Evaluation Strategy

In this paper, we limit ourselves to simulated evaluation strategies. These strategies enable reliable and reproducible experiments in a variety of settings, and enable analysis of detector performance on a large scale. Although there has been work on empirical analysis of attacks [1], such studies are extremely difficult to generalize and verify without a significant deployment of vehicular communication systems, which are still in their deployment phases at the time of writing. There is a wide variety of network and vehicle behavior simulation tools, with different scenarios; however, these are not the primary topic of this paper. For more information on this topic, refer to studies that compare simulation environments [2], [3].

However, what is important to consider for this paper is the significant variables that could have an impact on overall detection performance, and which should be analyzed individually to make general statements about the suitability of a detection mechanism. The most significant parameters here include the amount of attackers, the amount of legitimate vehicles and the driving scenario. By studying these parameters, authors can gain an insight into whether their detector has weaknesses in specific scenarios, and how it performs when significant message loss is encountered, and these are often also used in the evaluation of vehicular applications. In all cases, repetitions under different simulation seeds are used whenever probabilistic models of, e.g., the communication channel are considered. The amount of attackers is a first step to determine whether the detector is resilient against attacks.

Resilience against attack is a metric that is difficult to quantify, however; it is particularly challenging to consider sufficiently distinct attacks, and to implement these attacks correctly. Because detectors are often designed to detect specific attacks, it is tempting to implement exactly these attacks and then show that the detector performs well. Therefore, it is important to clearly specify the attacker model, including how exactly this attack works, and whether or not attackers can cooperate (collaborative attackers) and whether or not attackers can create a limited amount of additional identities (Sybil attackers). In the latter case, the attacker uses different pseudonymous identities that are possibly linkable by an authority; in the former case, multiple independent vehicles are attacker-controlled (e.g., this could be due to malware).

Having considered the evaluation strategies, authors should next consider what an appropriate metric for validation is. Most mechanisms will perform well in some metrics, and worse in others; it is therefore important to clearly state what the evaluation metric is. For comparative studies, this means that different metrics need to be considered, requiring an evaluation strategy that is as independent from the evaluated mechanisms as possible.

III. Evaluation Metrics

Almost all papers discussing detectors evaluate these through some measure of detection quality. However, there are many different strategies and nuances to this process, e.g., how it is computed and how it is aggregated over different messages and vehicles. In addition to detection quality, some alternative metrics exist, e.g., how long it takes for an attacker to be detected, or the message overhead (if any); we discuss these metrics separately.

A. Confusion Matrix

The most obvious approach to evaluate the quality of a detector's results is to use the confusion matrix, which is also used in other fields (e.g., medicine and machine learning). This matrix describes various combinations of false positives, false negatives, true positives and true negatives and their rates with respect to the entire population. This includes metrics like accuracy, which is the amount of correctly classified events divided by the total population, and false omission rate, which is the amount of false negatives in the set of all negatives. Most papers in the area of misbehavior detection that use this type of metric use the false negative rate to quantify the risk of missing detections, and the false positive rate as the risk of an incorrect detection event. Opinions differ widely on what acceptable values are (e.g., for intrusion detection in networks with a lot of traffic, a false positive rate over 0.001 is considered very bad), and it is difficult to make general statements about this, because the impact of a false detection event is significant.

However, it is actually not trivial to classify a detectors' results into these categories when evaluating the detector in a simulation. Notably, in distributed detection scenarios such as vehicular networks, proximity to the attacker plays a significant role in how likely a vehicle is able to detect an attack. Not all nodes in the network will hear all messages, so if the data is aggregated across different detectors, one must take care to normalize these results: instances of a detector do not necessarily produce the same output for a given message, and not every message is seen equally often. Simple normalization may not be sufficient here: if a specific subset of receivers produces very high false negatives, while the aggregate of all receivers on average performs very well, this could still mean the detector is bad. In other words, aggregating and normalizing over an entire simulation may hide the fact that there is a weakness in a specific scenario.

Having established a way to normalize and aggregate the detection results still leaves other questions open. The definition of the input of the detector is one of these factors: does the detector take a single message and output a classification, or does it take a stream of messages and output an eventual classification? In the latter setting, one needs additional metrics to establish the timeliness of the system (e.g., detection latency: how long does it take for a detector to correctly classify an attack, after this attack starts?).

For a data-centric setting, using the confusion matrix is particularly difficult, because of the fact that many attacks are impossible to distinguish from legitimate messages. This can happen in two cases: either the message was sent by an attacker, but follows the expected behavior of vehicles (i.e., it is not a malicious message), or the attack is so marginal that it has no impact (and is thus indistinguishable from expected behavior or sensor noise). For example, an attacker might transmit a beacon with a position a few centimeters from its' actual position. This led some authors to conclude that application-oriented evaluations may be more suitable: if the attacker cannot achieve a goal, because the impact of false data is too small, then clearly the detection mechanism is effective. The disadvantage of this strategy is that the evaluation depends not only on the simulation aspects and the attackers' implementation, but also on the application implementation.

For data-centric detection mechanisms that are based on consistency, i.e., they consider multiple data sources and detect inconsistencies, it is often implicitly assumed that previously received data is true, and only the incoming packet is classified as legitimate or malicious. However, this means that message order is particularly significant: whenever a malicious message arrives first, it may trigger an additional false positive, because the next legitimate message differs from the malicious message.

Many papers implicitly discuss that detectors should also perform revocation or response – they exclude specific messages from those that are received, in order to prevent errors in the application. Similarly, some detection algorithms are inherently incompatible with the idea of evaluating individual message on a sequential basis, because they perform some batch processing (e.g., classifying vehicles instead of messages [4]).

Some authors use a pre-classified set of messages, which is not necessarily data-centric (e.g., Grover et al. [4] use a set of 3101 legitimate and 1427 malicious samples, with several types of attacks). This is common in the field of machine learning, where classifiers are often tested using this type of approach. However, since most detection algorithms in VANETs have different inputs, it is difficult to find a conclusive set that considers these various inputs, as well as contain the necessarily distinct types of attacks (especially when reputation is considered, which can be built over time). In other words, this approach is only valid if each sample is well-defined, which is not the case in our more general setting.

B. Alternative Accuracy Metrics

This led authors to use application-specific or detector-specific evaluation strategies, in order to demonstrate specific strengths of individual detectors in comparison to others from the literature.

a) Application behavior metrics: in many VANET scenarios, the specific values transmitted by an attacker are not necessarily relevant, but what is of interest is whether a receiving vehicle makes incorrect decisions about the state of the world. Application metrics aim to capture this subtle concept into a concrete metric. These metrics are useful, because they can also consider errors from other (i.e., non-malicious) sources, and are independent of a deep understanding of the detection mechanism. However, they require an application implementation, which is bug-prone and makes attacker implementation more complex.

One specific category application metrics is that related to schemes that detect routing misbehavior. Because routing misbehavior is historically closely linked to evaluation of routing schemes, some authors use routing performance metrics (such as arrival rates, consumed bandwidth and similar metrics) and changes to vehicle mobility [5], [6].

Another class of application metrics is much closer to the data that many data-centric detection mechanisms analyze; for example, this includes collision avoidance applications [7], [8] and in-network aggregation [9]. A disadvantage of these strategies is that it is hard to use them as a baseline for other studies, because often they are very specific.

b) Detector specific metrics: Some detectors have known sources of potential errors, often inherent to their design; a common strategy to deal with this is to approach and analyze these issues specifically. This is particularly useful when it is very clear what kind of error sources are to be considered. For example, Bimeyer et al. [10] used this type of metric to also include GPS error as a potential source of additional false positives for their scheme to analyze these effects in detail.

Another example of such an approach would be an evaluation where the impact of the attacker on the analyzed variable is. This only works for continuous variables, such as position information, where a simple distance metric is available to estimate the error between real positions and falsified positions. Rather than looking at how well attacks are detected only, this strategy would measure the distance between accepted attacker-generated data that is accepted as valid, and use this as a metric for detection quality. This approach is also viable if error sources are considered for legitimate vehicles. A disadvantage of this approach is that it may only be suitable for specific classes of detectors (or at least, it may put other detectors at a disadvantage).

C. Other Metrics

There are many other types of metrics available to authors seeking to evaluate their detection mechanisms. Some notable examples include:

a) Detection Latency: the time required for an attack to be detected. The exact definition of this metric differs type of detector, but typically it is the time between the first malicious packet received and the first detected malicious packet. This is particularly significant to measure the impact of reputation abuse in trust-based node-centric detection mechanisms: if trustworthy vehicles transmit malicious information, the potential impact of an attack is large.

b) Computational Cost: although relatively uncommon for vehicular networks, more traditional benchmarks such as computational cost can also be used. Most authors only check whether reasonable estimates of locally received messages can be analyzed in reasonable time (which, using 100 vehicles in range transmitting at 10Hz, is at most a millisecond per message), but particularly with proposals that include multiple detectors, scalability could be an issue.

c) Financial Cost: traditional IDSs are commonly evaluated by examining the cost associated with response and the corresponding variants in the confusion matrix [11]. In the case of vehicular networks, this is a bit more complex: the cost of a false negative is difficult to estimate, and strongly dependent on the scenario (i.e., high speed collisions have much higher cost, even though the classification of the event is exactly the same). Estimating this cost is also ethically complicated, because human lives are included in this process, which suggests that such an evaluation requires more extensive knowledge on how to deal with this suitable (as done, e.g., in invasive medicine).

d) Stability & Usability: one factor that is often forgotten or stated as future work only is that the detection mechanism should be sufficiently stable, such that the users' experience with the system is good enough. This is important, because users' trust in the system is strongly dependent on their perception of the systems' reliability. If the system continually warns or makes changes in response to possible attacks, but has an overall higher performance than other systems, then users may still perceive that system as very unreliable. This is one point where node trustworthiness over time will perform significantly better.

IV. Recommendations

We recommend authors to follow the developments in the simulation community, and take note of extensive standardized scenarios, such as the LuST scenario [12] for traffic simulation. This also includes pro-actively porting source code that is under development to more recent versions of the simulation environment wherever possible: ideally code should always be based on the most recent stable version of the simulator available while the paper is under review. This allows authors to benefit from quality of life improvements in the simulation environment, but allows them to consider, e.g., newer channel models.

For reasons outlined above, we also recommend using a variety of different attack strategies, which ideally have very different goals. This ensures that the authors can describe the qualities, as well as the limitations of their detector, which in turn can help future authors decide whether new attacks against this detector may be possible. It is also important to define a baseline in addition to these attacks, which can be undertaken in various ways (and this depends strongly on what the authors actually designed), and if possible a simulation of potential non-malicious sources of error. The simplest example is GPS error: adding an error to a GPS coordinate is relatively simple, but it makes the results much more representative than an idealized perfect positioning system for each vehicle, where it just "knows" its' position.

The authors should at least consider how they aggregate the detection results across messages, vehicles and simulation repetitions, as discussed above, and clearly specify their approach. If possible, we recommend developing the simulation in such a way that multiple aggregation approaches can be used; there may not be a one size fits all solution.

Finally, we recommend publishing source code, or at least making it available to other researchers on request. This enables reproducibility, one of the core principles of scientific research, which allows research in this area to make faster and more meaningful progress. It would also enable studies that analyze the behavior of a variety of algorithms in different settings a much more efficient and less error-prone process. Finally, the authors themselves benefit from this process, because it is easier to compare to the literature.

V. Conclusion

In this paper, we have discussed several evaluation approaches for misbehavior detection. After describing several challenges and briefly surveying existing solutions, we gave some concrete recommendations that should be useful for authors that are studying misbehavior detection mechanisms. We plan to use these recommendation in combination with a framework that we are developing, named Maat, to evaluate potential detection mechanisms. As part of this project, we are also looking to decouple the execution of the simulation and the detection mechanisms, which would enable us to provide a data set to the community in addition to publishing our results.

Acknowledgment

This work was supported in part by the Baden-Württemberg Stiftung gGmbH Stuttgart as part of the project IKT-05 AutoDetect of its IT security research programme.

VI. References

References

[1] N. Bissmeyer, K. H. Schroder, J. Petit, S. Mauthofer, and K. M. Bayarou, "Short paper: Experimental analysis of misbehavior detection and prevention in vanets," in *Vehicular Networking Conference (VNC), 2013 IEEE*. IEEE, 2013, pp. 198–201.

[2] R. Stanica, E. Chaput, and A.-L. Beylot, "Simulation of vehicular ad-hoc networks: Challenges, review of tools and recommendations," *Computer Networks*, vol. 55, no. 14, pp. 3179 – 3188, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128611001629

[3] C. Sommer, J. Hrri, F. Hrizi, B. Schnemann, and F. Dressler, "Simulation Tools and Techniques for Vehicular Communications and Applications," in *Vehicular ad hoc Networks*, C. Campolo, A. Molinaro, and R. Scopigno, Eds. Springer International Publishing, 2015, pp. 365–392, dOI: 10.1007/978-3-319-15497-8_13. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-15497-8_13

[4] J. Grover, N. K. Prajapati, V. Laxmi, and M. S. Gaur, "Machine learning approach for multiple misbehavior detection in vanet," in *Advances in Computing and Communications*, ser. Communications in Computer and Information Science, A. Abraham, J. L. Mauri, J. F. Buford, J. Suzuki, and S. M. Thampi, Eds. Springer Berlin Heidelberg, 2011, vol. 192, pp. 644–653.

[5] J. Hortelano, J. C. Ruiz, and P. Manzoni, "Evaluating the usefulness of watchdogs for intrusion detection in VANETs," in *IEEE International Conference on Communications Workshops (ICC)*. IEEE, May 2010, pp. 1–5.

[6] T. Leinmüller, E. Schoch, F. Kargl, and C. Maihöfer, "Decentralized position verification in geographic ad hoc routing," *Security and Communication Networks*, vol. 3, no. 4, pp. 289–302, Jul. 2010. [Online]. Available: http://doi.wiley.com/10.1002/sec.56

[7] J. Petit, M. Feiri, and F. Kargl, "Spoofed data detection in vanets using dynamic thresholds," in *Vehicular Networking Conference (VNC)*. IEEE, nov. 2011, pp. 25 –32.

[8] T. H.-J. Kim, A. Studer, R. Dubey, X. Zhang, A. Perrig, F. Bai, B. Bellur, and A. Iyer, "VANET alert endorsement using multi-source filters," in *Proceedings of the seventh ACM international workshop on VehicularAr InterNETworking (VANET)*. New York, NY, USA: ACM Press, 2010, p. 51. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1860058.1860067

[9] R. W. van der Heijden, S. Dietzel, and F. Kargl, "Sedya: secure dynamic aggregation in vanets," in *Proceedings of the sixth ACM conference on Security and privacy in wireless and mobile networks*. ACM, 2013, pp. 131–142.

[10] N. Bißmeyer, C. Stresing, and K. M. Bayarou, "Intrusion Detection in VANETs Through Verification of Vehicle Movement Data," in *Vehicular Networking Conference (VNC)*. IEEE, 2010, pp. 166–173.

[11] A. A. Cárdenas, J. S. Baras, and K. Seamon, "A framework for the evaluation of intrusion detection systems," in *IEEE Symposium on Security and Privacy*. IEEE, 2006, pp. 15–pp.

[12] L. Codeca, R. Frank, and T. Engel, "Luxembourg sumo traffic (lust) scenario: 24 hours of mobility for vehicular networking research," in *Vehicular Networking Conference (VNC), 2015 IEEE*. IEEE, 2015, pp. 1–8.