proceedings

MDPI

Proceedings

Time-Aware Detection Systems †

Manuel López-Vizcaíno *, Laura Vigoya, Fidel Cacheda and Francisco J. Novoa

CITIC, Universidade da Coruña, A Coruña 15071, Spain
* Correspondence: manuel.fernandezl@udc.es
† Presented at the 2nd XoveTIC Conference, A Coruna, Spain, 5–6 September 2019.

Published: 5 August 2019

Abstract: Communication network data has been growing in the last decades and with the generalisation of the Internet of Things (IoT) its growth has increased. The number of attacks to this kind of infrastructures have also increased due to the relevance they are gaining. As a result, it is vital to guarantee an adequate level of security and to detect threats as soon as possible. Classical methods emphasise in detection but not taking into account the number of records needed to successfully identify an attack. To achieve this, time-aware techniques both for detection and measure may be used. In this work, well-known machine learning methods will be explored to detect attacks based on public datasets. In order to obtain the performance, classic metrics will be used but also the number of elements processed will be taken into account in order to determine a time-aware performance of the method.

Keywords: IDS; early-detection; communication networks; time-aware metrics

1. Introduction

The systems dedicated to detect intrusions in communication networks are called Network Intrusion Detection Systems (NIDS) and have attracted a lot of attention due to the growth of networks and the importance of their correct behaviour to ensure business continuation [1]. As it was defined by Lockheed & Martin in 2011 [2] the time elapsed since the begin of an attack will affect directly to the possible damage caused. To avoid further risks, intruders and attackers should be detected as soon as possible in order to minimise the damage.

As part of this systems there are multiple works that explore the use of machine learning in order to detect anomalies in communication networks as it can be seen in [3–5]. This techniques are usually evaluated through the use of classical metrics as Precision, Recall [6] or F1 as a combination of both [7] which take into account the number of elements correctly and incorrectly classified.

In this article, results from the measurements with classical metrics and number of packets used to take the decision will be presented. Kitsune IoT dataset for OS Scan attack [8] will be used to perform experiments with several machine learning methods [5].

2. Methods

To perform this analysis, OS Scan from Kitsune dataset is used [8]. As the objective is to determine if a sequence of elements belongs to one class and to measure how the system performs, individual packets have been grouped into flows. Using the definition of flow [9] which are a set of packets with same source IP, destination IP, source port, destination port and protocol in a period of time, bidirectional flows have been created [10].

The dataset is divided randomly into 75% and 25% sets for training and testing. Then, each one has been splitted into 10 chunks containing 10% of the packets belonging to the flows. This is done in order to study the performance of the methods in different time points.

To conclude, several machine learning methods are then applied to all the chunks obtaining the predicted value for the classification or a delay if no decision is taken. This could happen if there are no packets in the flow yet or if there is not a majority in the flow, as individual packets are evaluated.

3. Results

Results are shown in Table 1 where chunks 1, 2 and chunks from 5 to 9 have been grouped together because there are no variation in the metric values. This, alongside with the 0.0 values for 1 and 2 chunks can be explained due to the dataset characteristics. As it represents an OS Scan, there are a high number of two packet size, scan and reset, flows which will not affect the results in any chunk but on chunks 4 and 10.

An increase in F1 values can be seen for the presented methods, as the number of packets evaluated increase. This rise is shown by the mean and the maximum number of packets. Also an increase in standard deviation can be seen as there is a big difference between two packet sized flows and the rest of the normal traffic.

Table 1. Performance for state-of-the-art machine learning models.

	Metrics	**Chunks**				
		1–2	**3**	**4**	**5–9**	**10**
RF	Precision	0.0	1.0	0.875	0.8452	0.8454
	Recall	0.0	0.0001	0.0004	0.8505	0.8519
	F1	0.0	0.0002	0.0009	0.8478	0.8487
J48	Precision	0.0	1.0	0.875	0.8452	0.8454
	Recall	0.0	0.0001	0.0004	0.8505	0.8519
	F1	0.0	0.0002	0.0009	0.8478	0.8487
JRip	Precision	0.0	1.0	0.8571	0.8451	0.8454
	Recall	0.0	0.0001	0.0004	0.8504	0.8518
	F1	0.0	0.0001	0.0007	0.8477	0.8486
Number of Packets	Max	45	68	91	206	229
	Avg	4.0203	6.0442	8.0848	19.1218	22.0911
	STD	11.4049	17.1423	22.9199	51.4238	57.0210

4. Conclusions

As it can be seen in Table 1 even if classical metrics show a good performance for the machine learning methods, it should be taken into account that more packets need to be processed. More packets imply longer times and an increase in the risk created by this particular threat. This is the reason why this metrics should be penalised depending on how much records have been processed to obtain this result.

Also, it must be said that even if an IoT environment could benefit from an early detection system, these techniques could also be applied to other fields where early detection is relevant to reach a good system performance.

Author Contributions: All authors have equally contributed to this article.

Funding: This research was supported by the Ministry of Economy and Competitiveness of Spain (Project TIN2015-70648-P) by the Xunta de Galicia (Centro singular de investigación de Galicia accreditation ED431G/01 2016-2019) and the European Union (European Regional Development Fund—ERDF).

Conflicts of Interest: The authors declare no conflict of interest. The founding sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Ashoor, A.S.; Gore, S. Importance of intrusion detection system (IDS). *Int. J. Sci. Eng. Res.* **2011**, *2*, 1–4.
2. Hutchins, E.M.; Cloppert, M.J.; Amin, R.M. Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. *Lead. Issues Inf. Warf. Secur. Res.* **2011**, *1*, 80.
3. Kaur, H.; Singh, G.; Minhas, J. A review of machine learning based anomaly detection techniques. *arXiv* **2013**, arXiv:1307.7286.
4. Buczak, A.L.; Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. Tutor.* **2015**, *18*, 1153–1176.
5. Chawathe, S.S. Monitoring IoT Networks for Botnet Activity. In Proceedings of the 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA), Cambridge, MA, USA, 1–3 November 2018; pp. 1–8.
6. Metz, C.E. Basic principles of ROC analysis. *Semin. Nucl. Med.* **1978**, *8*, 283–298, doi:10.1016/S0001-2998(78)80014-2.
7. Chinchor, N.; Nancy. MUC-4 evaluation metrics. In Proceedings of the 4th Conference on Message Understanding—MUC4 '92, McLean, VA, USA, 16–18 June 1992 ; Association for Computational Linguistics: Morristown, NJ, USA, 1992; p. 22, doi:10.3115/1072064.1072067.
8. Mirsky, Y.; Doitshman, T.; Elovici, Y.; Shabtai, A. Kitsune: An Ensemble of Autoencoders for Online Network Intrusion Detection *arXiv* **2018**, arXiv:1802.09089.
9. Aitken, P.; Claise, B.; Trammell, B. *Specification of the IP Flow Information Export (IPFIX) Protocol for the Exchange of Flow Information*; RFC 7011; Internet Engineering Task Force: Fremont, CA, USA, 2013, doi:10.17487/RFC7011.
10. Trammell, B.; Boschi, E. *Bidirectional Flow Export Using IP Flow Information Export (IPFIX)*; Technical Report; Internet Engineering Task Force: Fremont, CA, USA, 2008.