Bridging the gap to real-world for network intrusion detection systems with data-centric approach

Gustavo de C. Bertoli
PhD Candidate
Aeronautics Institute of Technology
gustavo.bertoli@ga.ita.br

Lourenço A. Pereira Jr., Filipe A. N. Verri
Department of Computer Science
Aeronautics Institute of Technology
ljr,verri@ita.br

Aldri Santos
Department of Computer Science
Federal University of Minas Gerais
aldri@dcc.ufmg.br

Osamu Saotome
Department of Electronics Engineering
Aeronautics Institute of Technology
osaotome@ita.br

Abstract

Most research using machine learning (ML) for network intrusion detection systems (NIDS) uses well-established datasets such as KDD-CUP99, NSL-KDD, UNSW-NB15, and CICIDS-2017. In this context, the possibilities of machine learning techniques are explored, aiming for metrics improvements compared to the published baselines (model-centric approach). However, those datasets present some limitations as aging that make it unfeasible to transpose those ML-based solutions to real-world applications. This paper presents a systematic data-centric approach to address the current limitations of NIDS research, specifically the datasets. This approach generates NIDS datasets composed of the most recent network traffic and attacks, with the labeling process integrated by design.

1 Introduction

There is an increase of connected devices nowadays as aggregated by the Internet of Things (IoT) paradigm and further with the rise of 5G [1, 2]. This scenario results in a greater heterogeneity of devices and network architectures to make the solutions available, such as autonomous driving, remote surgery, connected infrastructures, among others. In this context, cybersecurity is one of the properties that must be in place to make these solutions available with trust.

To deal with the pace of increasing network agents, architectural changes, diversity of attacks, and the increasing amount of network traffic, the use of machine learning (ML) based network intrusion detection systems (NIDS) raises as a technical approach to deal with the evolving context bringing trust to these applications [3]. In contrast with the NIDS research trend, ML-based NIDS are still not prevalent in real-world applications [4]. Instead, most research uses well-established datasets like KDD-CUP99 [5], NSL-KDD [6], UNSW-NB15 [7], and CICIDS-2017 [8], that from a machine learning perspective, are good as baseline comparisons on different ML techniques. However, to derive these findings to real-world application, we claim that a data-centric approach must be in place to continuously generate datasets and re-train models, address the following limitations: evolving network traffic, aging of attacks, and capability to generalize for different network architectures.

The remaining of the paper presents the related works about the challenges faced by NIDS datasets in Section 2. Section 3 and 4 details our approach to generate NIDS datasets from a data-centric perspective. Section 5 presents the conclusion, next steps of our research, and open questions.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

2 Challenge and Related Work

[4] reported the low performance of NIDS in the real-world environment as a research challenge. Some probable causes are using old datasets and not evaluating the proposed solutions considering a realistic environment. In addition, the authors report the lack of a systematic approach for dataset generation capable of being updated frequently. The challenges about datasets for NIDS research are also reported by [9]. It presents the inherent network traffic diversity characteristic and the difficulty for a dataset to cope with all possibilities. Furthermore, it is highlighted the difficulty of data availability (public datasets) and criticizes generalizing results obtained in small network architecture to larger networks. Regarding dataset aging, [9] reports that the most used datasets in 2010 were already one decade old (DARPA98, and KDD-CUP99). [10, 11] highlights the current trend for using a deep learning approach in the NIDS research but also points out the use of old datasets such as KDD-CUP99 and NSL-KDD. It also reports the lack of available datasets and low performance in a real-world environment. Next, [12] provides a survey about NIDS datasets and reports the outdated datasets in use by NIDS research as a challenge.

The survey by [13] reports the "perfect dataset" composed of up-to-date traffic, labeled, publicly available, with real network traffic and a multitude of attacks and normal traffic, spanning a long time frame. However, they are skeptical about the availability of datasets comprising all these properties, with the challenge for a labeled dataset containing long-time traffic. In this context of dataset aging for NIDS applications, it is important to clarify that the network traffic and attacks have a lifespan. In other words, the NIDS are susceptible to concept drift as reported by [14]. It reports a six-week of good performance of the trained models. This study indicates that a continuous update is required to address the evolving characteristics of the network traffic.

In summary, the related works present dataset aging and poor performance in a real-world environment as a research gap. Our proposition is a framework for dataset generation capable of generating NIDS datasets with the most recent network traffic and attack patterns shifting the current NIDS paradigm from model-centric to data-centric. The downside of our approach is the limited possibilities to benchmark ML solutions using the generated dataset with published baselines. Nevertheless, we envision this approach as a next step to the current NIDS solutions to bridge the gap to the real world. Thus, after performing the traditional NIDS research to obtain the best model, our approach must be part of an end-to-end pipeline to continuously re-train this model with a dataset composed of up-to-date traffic and network attacks. Another limitation is that we do not evaluate the use of commonality for NIDS features between our generated datasets and those publicly available [15].

3 Methodology

Our methodology starts with the generation of network traffic that represents the attack behavior. This step aims to overcome the current datasets' challenges: not containing the most recent attacks, not reproducible, and not properly labeled [16]. For the attack traffic, we take advantage of virtualization and cloud infrastructure to generate this traffic on-demand. We use a Docker container with the base image of Kali Linux distribution (a well-known distribution for cybersecurity tasks). The use of Kali Linux is a common approach to generate attack traffic [17]. This container is responsible for generating all attack traffic against the cloud infrastructure that we manage. This container is a customized step that can include different attacks and targets through configuration files. Knowing the attacker instance and the targets in advance is crucial because this information supports the automation of the labeling process. For example, it is possible to construct a tuple with the attacker and target IP addresses to determine the attack traffic.To compose the NIDS dataset, we use a method known as salting [18]. This method merges the legit traffic data with the attack traffic generated on-demand. We obtain this legit traffic after removing the malicious traffic from open-data traffic providers as MAWILab [19], from other public NIDS datasets, or from the deployment environment that increases the effectiveness of the solution (better data). Using traffic from the deployment environment requires authorization of the IT infrastructure in conjunction with the network administrators to label the legit traffic correctly. This labeling process can make use of firewall rules to determine the legit traffic. The generation of the dataset becomes part of the overall process of NIDS development (Figure 1). This process considers the generation of up-to-date datasets from both attack and benign traffic perspectives to bridge the gap between research and the real-world application of ML-based NIDS.

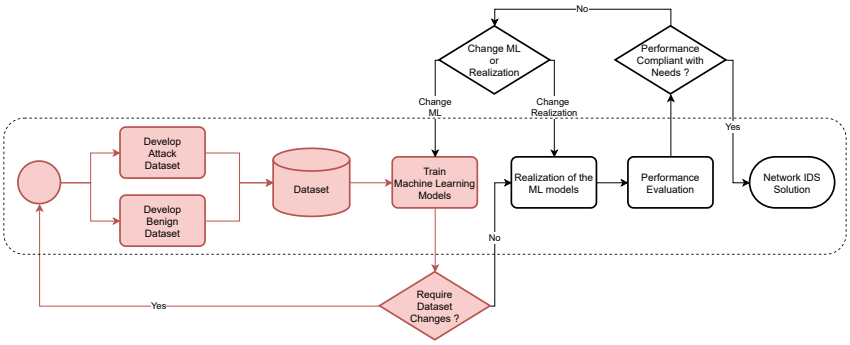

Figure 1: Workflow for the development of a ML-based NIDS solution. Dataset is part of the process.

We also ensure that the data generated is *tidy* [20], that is, has a rectangular structure composed of observations (rows) and variables (columns). Such an organization, popular among data scientists, enables the utilization of common tools for exploratory analysis, data manipulation, and model building [20]. Each packet in the dataset has its row (Table 1). Each column represents a packet feature that can be either context-aware ($X_1^C, \ldots, X_{n_c}^C$) that are common to multiple observations, such as source and destination IP addresses or context-free, that is, intrinsic of the packet ($X_1^P, \ldots, X_{n_p}^P$). One of the columns (Y) represents the variable that informs whether the packet is from benign traffic or an attack.

As a result, one can easily reshape the generated dataset to meet the requirements and aspects of the desired machine-learning task. For instance, in a stateless approach, one can discard all context-aware variables (Table 2a); while, in a stateful approach, one can use the consolidated summarizing operations in data science to aggregate rows (Table 2b). The common and consistent shape of the data also eases selection and filtering operations.

Table 1: Shape of the data generated in our framework. Context-aware and packet-intrinsic features.

X_1^C	...	$X_{n_c}^C$	X_1^P	...	$X_{n_p}^P$	Y
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...
...	...	...	...	...	...	...

Table 2: Different shapes for manipulated data.

(a) Stateless approach. The number of rows is kept, but the context-aware variables are discarded, reducing the number of columns.

X_1^P	...	$X_{n_p}^P$	Y
...	...	...	...
...	...	...	...
...	...	...	...
...	...	...	...

(b) Stateful approach. The number of rows is reduced since all observations that share the same context (X_1^C, ..., $X_{n_c}^C$) are combined. New variables ($\Gamma_1, \ldots, \Gamma_{n_\gamma}$) are the result of the aggregation operations.

X_1^C	...	$X_{n_c}^C$	Γ_1	...	Γ_{n_γ}	Y
...	...	...	...	...	...	...
...	...	...	...	...	...	...

4 Results and Discussion

We created an environment using a cloud service provider to create a geographically distributed testbed. For the attacker instance, the container's configuration file is available on our public repository. To support the labeling process, we set up a UDP daemon on each of the cloud instances that receive remote UDP commands from the attacker container to start and stop the network traffic capture using `Tcpdump`[1] on these cloud instances. The choice for the UDP protocol is because we focused on the TCP protocol, so there is no conflict between the command and control traffic and the one that makes

[1]Tcpdump: tcpdump.org

up our dataset. Figure 2 presents the overall picture of the process. From the UDP message starting the recording, the source IP address (i.e., attacker address) is used to compose the `Tcpdump` filter to capture only the traffic from the attacker to the target instance.

Hence, we generated a specific file containing the network traffic with a filename composed by the label name (e.g., attack type) and its start timestamp for each attack. Finally, all network traffic files generated during the attack are retrieved from multiple cloud instances, processed to retrieve the network traffic, and merged into a single dataset.

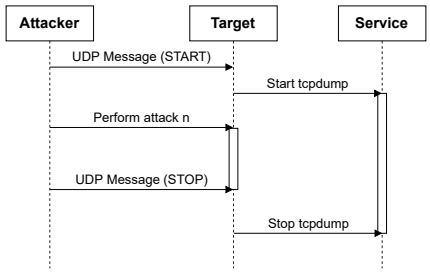

Figure 2: Attack dataset lifeline. Managing the service to capture packets through UDP messages between Attacker and Target.

For the benign dataset, we use the up-to-date MAWILab dataset, which is a daily 15-minutes network recording from trans-pacific backbone traffic between the USA and Japan. It has labels for the following classes: anomalous and suspicious. Our rationale to obtain benign traffic is to remove from the MAWILab dataset all the traffic inferred as anomalous and suspicious by the MAWILab classifiers. Normally, MAWILab traffic contains tens of millions of packets on each trace, so we introduced a step to random sample packets from these filtered traces to work with a not too much-imbalanced dataset. It is important to highlight that our current applications use a stateless approach (analysis of each packet without the context), so for a stateful analysis (based on flow), the sampling step must retrieve packets that are part of the same context. Figure 3 presents a summary of the process to obtain the benign dataset.

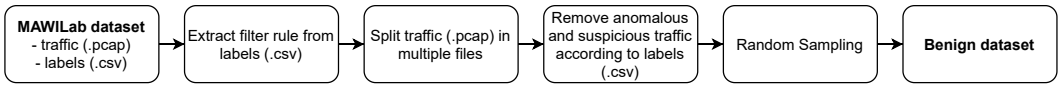

Figure 3: Benign dataset process. After obtaining the MAWILab traffic, and the labels for anomalous/suspicious traffic, the processing is performed to obtain a benign dataset.

Our research focuses on the problem of the detection of port scanning attacks. This Attack attempts to identify the system's available services or characteristics, which is normally the first step of a cyber attack. Thus, blocking port scanning stops an attack in its early stages, reducing the risks and the resources to secure a system. We effortlessly generated a dataset for the port scanning problem with this proposed framework for the Internet environment. The dataset comprises 22 classes (TCP port scanning attacks) targeting 4 cloud instances, resulting in $455{,}503$ correctly labeled attack samples. For the benign samples, we obtained from MAWILab traffic from November 10[2] and 29[3], 2020. After preprocessing the MAWILab traffic to remove the attack samples, we got a total of $380{,}438$ samples (packets) of benign traffic, resulting in a dataset with $835{,}941$ packets with the following distribution: $Benign = 46\%$ and $Attack = 54\%$ in a *tidy* dataset with 41 features (packet-intrinsic and context-aware). Furthermore, the NIDS is an inherently imbalanced problem (higher occurrences of benign traffic); indeed, our solution provides a controlled approach over it, either by managing the repetition of attacks, the number of target instances, or sampling benign traffic.

5 Conclusion

We presented the challenges for ML-based NIDS of aging datasets, the difficulty of coping with the constantly evolving characteristics of network traffic, and bad performance in the real-world environment. To address these gaps, we presented a systematic data-centric approach capable of generating up-to-date NIDS datasets. The approach exploits the public and up-to-date MAWILab dataset in conjunction with virtualization to generate the attack traffic. Then, it combines those two sources in a salting process to generate labeled NIDS datasets. The reproducible source-code to generate datasets as presented on this paper is available in our repository: `https://github.com/c2dc/AB-TRAP` (steps A and B from [21]).

[2] fukuda-lab.org/mawilab/v1.1/2020/11/10/20201110.html
[3] fukuda-lab.org/mawilab/v1.1/2020/11/29/20201129.html

References

[1] Shadi Al-Sarawi, Mohammed Anbar, Rosni Abdullah, and Ahmad B. Al Hawari. Internet of things market analysis forecasts, 2020–2030. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 449–453, 2020.

[2] Ericsson. Ericsson mobility report 2021. `https://www.ericsson.com/4a03c2/assets/local/mobility-report/documents/2021/june-2021-ericsson-mobility-report.pdf`, 2021.

[3] Ross Anderson. *Security engineering: a guide to building dependable distributed systems*, chapter 21.4.2, page 722. John Wiley & Sons, 3rd edition, 2020.

[4] Zeeshan Ahmad, Adnan Shahid Khan, Cheah Wai Shiang, Johari Abdullah, and Farhan Ahmad. Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1):e4150, 2021.

[5] S Hettich. Kdd cup 1999 data. *The UCI KDD Archive*, 1999.

[6] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali A. Ghorbani. A detailed analysis of the kdd cup 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6, 2009.

[7] Nour Moustafa and Jill Slay. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE, 2015.

[8] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A Ghorbani. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, 1:108–116, 2018.

[9] Robin Sommer and Vern Paxson. Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE symposium on security and privacy*, pages 305–316. IEEE, 2010.

[10] Hongyu Liu and Bo Lang. Machine learning and deep learning methods for intrusion detection systems: A survey. *Applied Sciences*, 9(20):4396, 2019.

[11] Yang Xin, Lingshuang Kong, Zhi Liu, Yuling Chen, Yanmiao Li, Hongliang Zhu, Mingcheng Gao, Haixia Hou, and Chunhua Wang. Machine learning and deep learning methods for cybersecurity. *IEEE Access*, 6:35365–35381, 2018.

[12] Hanan Hindy, David Brosset, Ethan Bayne, Amar Seeam, Christos Tachtatzis, Robert Atkinson, and Xavier Bellekens. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. *IEEE Access*, 2020.

[13] Markus Ring, Sarah Wunderlich, Deniz Scheuring, Dieter Landes, and Andreas Hotho. A survey of network-based intrusion detection data sets. *Computers & Security*, 86:147–167, 2019.

[14] E. Viegas, A. O. Santin, and V. Abreu Jr. Machine learning intrusion detection in big data era: A multi-objective approach for longer model lifespans. *IEEE Transactions on Network Science and Engineering*, 8(1):366–376, 2021.

[15] Mohanad Sarhan, Siamak Layeghy, Nour Moustafa, and Marius Portmann. Towards a standard feature set of nids datasets. *arXiv preprint arXiv:2101.11315*, 2021.

[16] Marek Małowidzki, P Berezinski, and Michał Mazur. Network intrusion detection: Half a kingdom for a good dataset. In *Proceedings of NATO STO SAS-139 Workshop, Portugal*, 2015.

[17] Abdullah Alsaedi, Nour Moustafa, Zahir Tari, Abdun Mahmood, and Adnan Anwar. Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. *IEEE Access*, 8:165130–165150, 2020.

[18] Z. Berkay Celik, Jayaram Raghuram, George Kesidis, and David J. Miller. Salting public traces with attack traffic to test flow classifiers. In *Proceedings of the 4th Conference on Cyber Security Experimentation and Test*, CSET'11, page 3, USA, 2011. USENIX Association.

[19] Romain Fontugne, Pierre Borgnat, Patrice Abry, and Kensuke Fukuda. Mawilab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In *ACM CoNEXT '10*, Co-NEXT '10, New York, NY, USA, 2010. Association for Computing Machinery.

[20] Hadley Wickham and Garrett Grolemund. *R for Data Science*. O'Reilly Media, 1st edition, 2017.

[21] Gustavo De Carvalho Bertoli, Lourenço Alves Pereira Júnior, Osamu Saotome, Aldri L. Dos Santos, Filipe Alves Neto Verri, Cesar Augusto Cavalheiro Marcondes, Sidnei Barbieri, Moises S. Rodrigues, and José M. Parente De Oliveira. An end-to-end framework for machine learning-based network intrusion detection system. *IEEE Access*, 9:106790–106805, 2021.