Carlos Garcia Cordero, Emmanouil Vasilomanolakis, Nikolay Milanov, Christian Koch, David Hausheer and Max Mühlhäuser, ID2T: A DIY dataset creation toolkit for Intrusion Detection Systems, in Proc. of IEEE Conference on Communications and Network Security, 2015, ISBN 978-1-4673-7876-5

# ID2T: a DIY Dataset Creation Toolkit for Intrusion Detection Systems

Carlos Garcia Cordero[b], Emmanouil Vasilomanolakis[a,b], Nikolay Milanov[c],
Christian Koch[c], David Hausheer[c], Max Mühlhäuser[b]
[a]AGT International, Germany
[b]Telecooperation Group, Technische Universität Darmstadt / CASED, Germany
{carlos.garcia, emmanouil.vasilomano, max}@tk.informatik.tu-darmstadt.de
[c]Peer-to-Peer Systems Engineering Lab, Technische Universität Darmstadt, Germany
{ckoch, hausheer, nikolay.milanov}@ps.tu-darmstadt.de

***Abstract*—Intrusion Detection Systems (IDSs) are an important defense tool against the sophisticated and ever-growing network attacks. These systems need to be evaluated against high quality datasets for correctly assessing their usefulness and comparing their performance. We present an Intrusion Detection Dataset Toolkit (ID2T) for the creation of labeled datasets containing user defined synthetic attacks. The architecture of the toolkit is provided for examination and the example of an injected attack, in real network traffic, is visualized and analyzed. We further discuss the ability of the toolkit of creating realistic synthetic attacks of high quality and low bias.**

## I. Introduction

Cyber-attacks are increasing in quantity and sophistication. To cope with this, network defense mechanisms such as IDSs are nowadays considered mandatory for critical networks. However, for the development of useful intrusion detection algorithms and IDSs, the evaluation of their performance, detection accuracy and false positive rate is essential. Labeled datasets are required for this.

Existing datasets in the area of intrusion detection exhibit certain disadvantages. First, many are outdated and, thus, lack new attack trends or contain obsolete network traffic patterns. Second, many datasets are not publicly available, due to copyright and privacy protection issues, and are not reproducible. Moreover, datasets are frequently tailored to specific scenarios (e.g., Distributed Denial of Service (DDoS) attacks in backbone networks) and often do not provide ground truth or attack labels.

We argue that IDSs must be evaluated against datasets that meet four main requirements. *Labeled attacks* are essential and, *high quality* normal and attack data must contain modern network traffic patterns and exclude data defects. A defect is any type of disturbance that hinders the credibility of the data (e.g., accidental anomalous patterns). These two requirements are crucial for comparing different IDSs and assessing their accuracy and false positive rate in different scenarios. Datasets need to be *publicly available* or, in case of synthetic datasets, *reproducible* to enable peer reviews and comparisons. Lastly, *flexibility* is important for testing different scenarios, e.g., attacks tailored to a specific network, intertwined with different attack types.

We present *Intrusion Detection Dataset Toolkit (ID2T)*, a toolkit for creating intrusion detection datasets that fulfill the aforementioned requirements. *ID2T* is able to create datasets from user supplied network packet captures by injecting attacks configured by the user. We argue that a dataset cannot fit all scenarios; rather, researchers should be able to create datasets on-the-fly corresponding to a specific scenario, in a reproducible manner. *ID2T* is open-source and highly modular; implementing custom attacks is straightforward. In addition, the toolkit supports the injection of up-to-date attacks such as DDoS attacks against IPv6 targets. Moreover, we envision support for processing arbitrarily large network capture files. Finally, the toolkit pays special attention to the creation of high quality datasets that do not include defects.

## II. Related Work

In order to evaluate and compare different IDSs, researchers rely on publicly available datasets, on simulating and recording network attacks, or on modifying well known datasets to contain synthetically created attacks. Most datasets that fall within one of these three categories do not conform to our requirements. Common publicly available datasets do not provide ground truth or are difficult to obtain in practice. The MAWI dataset is a modern and commonly used public dataset which lacks ground truth [1]. Other publicly cited datasets, such as the UNB ISCX [2], are not currently available. Simulated datasets often lack data quality and do not represent modern traffic patterns such as the popular 1999 DARPA dataset [3]. Given these problems, researchers often inject well known attacks into datasets but often do not make these resulting datasets public, as in the case of [4].

To the best of our knowledge, only one other publicly available tool exists capable of injecting attacks into user supplied datasets. The FLAME tool [5] takes a network flow trace as input and injects known attacks into it. This tool works exclusively with network flows. This is a limitation that hinders the types of attacks that may be injected. Only attacks that leave traces in flow records can be injected, e.g., DDoS or port scans. In contrast, ID2T injects attacks into network packet captures which enables a larger range of synthetically created attacks, e.g., network exploits.

## III. ID2T: an Intrusion Detection Dataset creation Toolkit

ID2T gives researchers the ability to inject labeled attacks in a scenario where they have full control. Using information and statistics extracted from network packet captures supplied as input, attacks are injected so as to mimic the input characteristics as close as possible. A DDoS attack, for instance, utilizes observed IP addresses as victims and TTL values of packets distributed normally using parameters learned from the input data, among other techniques. In ID2T, network packet captures (*pcap* files) are used as inputs. Moreover, ID2T creates attacks, labels them and finally creates new pcap files containing them. Labels are provided either externally as different files specifying time intervals where attacks are present or internally as user defined modifications to the attack packets, e.g., assigning all attack MAC addresses to a particular value.

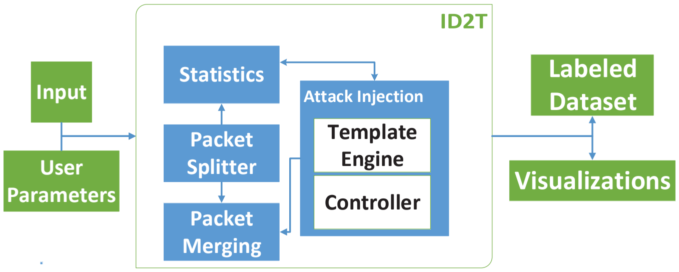

Fig. 1: ID2T architecture

ID2T achieves the aforementioned tasks by utilizing the architecture presented in Figure 1. There are four internal modules that react to user input to create labeled datasets. The *packet splitter module* is responsible for processing large pcap files by splitting them and applying map-reduce techniques to extract information. The *statistics module* is responsible for calculating statistics required by the attacks to replicate the quantitative and qualitative characteristics of the input. Attacks are modeled utilizing a framework provided by the *attack injection module*. Each attack can possess multiple *templates* that represent their basic structure. Templates are, essentially, network packets with replaceable placeholders and variables. The *controller* of an attack is responsible for filling out templates and for creating new network packets programmatically. Lastly, the *packet merging module* receives the network packets created by the attacks and merges them with the previously split input. A dataset with labels, accompanied with visualizations useful for understanding the injections, are the result of this process.

## IV. Visualizing Attacks Injected with ID2T

Figure 2 provides an example visualization of a DDoS attack injected with ID2T into real world data extracted from the MAWI dataset [1]. The period where the attack is injected does not contain known DDoS attacks. In time windows of 1 second the source and destination IP, and the source and destination ports are analyzed. The Shannon entropy of the counts of the four features is plotted in a time series. The colored area, between 60 and 70 seconds, shows how the entropy changes due to the injected attack. Before and after the attack it is evident that the entropy does not vary much. This implies that the counts of the observed features do not vary much. During the attack the *source IP entropy* and the *source port entropy* increases significantly. This is expected as a DDoS attack introduces many new source IPs and ports that increase the uncertainty of these two features [4]. On the other hand, almost no new destination IPs or ports are introduced and, in consequence, the entropy of these does not vary much.

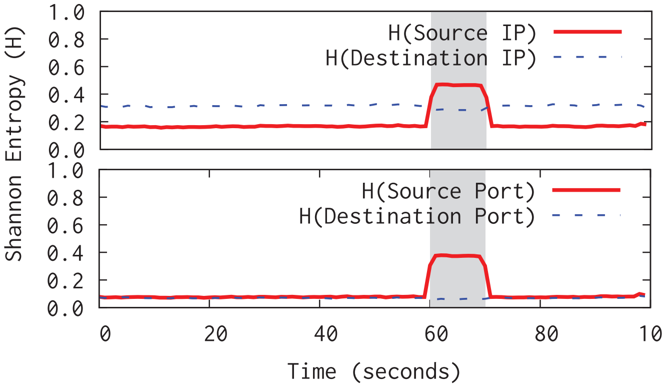

Fig. 2: DDoS attack injected with ID2T.

## V. Conclusion and Future Work

The intrusion detection community is in need of datasets that enable the evaluation and comparison of different IDSs. A single dataset cannot satisfy all use cases and scenarios; hence, a systematic and reliable mechanism to create datasets is preferred. We present ID2T, a toolkit for creating datasets that conform with the standards required to evaluate intrusion detection systems and algorithms. ID2T is able to inject realistic and modern attacks as shown in Section IV. We demonstrate an example DDoS attack being inserted into normal data and how the Shannon entropy of selected features is modified as expected.

The ID2T toolkit is still being developed. It is our intention to gather more requirements and use cases from the community to expand and improve the capabilities of the toolkit. To this extent, we are focusing on the creation of new evaluation techniques to assess the quality of the generated datasets. We are investigating graph analysis for representing normal and attack traffic, and the utilization of signature-based IDSs, such as Snort, to guarantee realistic looking synthetic attacks by detecting defects or biases present in them.

## References

[1] R. Fontugne, P. Borgnat, P. Abry, and K. Fukuda, "MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking," in *Conference on emerging Networking EXperiments and Technologies (CoNEXT)*. ACM, 2010, pp. 1–12.

[2] 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.

[3] R. Lippmann, J. W. Haines, D. J. Fried, J. Korba, and K. Das, "The 1999 DARPA off-line intrusion detection evaluation," *Computer Networks*, vol. 34, no. 4, pp. 579–595, 2000.

[4] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Conference on Applications, technologies, architectures, and protocols for computer communications (SIGCOMM)*. ACM Press, 2005, pp. 217–228.

[5] D. Brauckhoff, A. Wagner, and M. Martin, "FLAME: a Flow-Level Anomaly Modeling Engine," in *The conference on Cyber security (CSET)*, 2008.