INSecS-DCS: A Highly Customizable Network Intrusion Dataset Creation Framework

Nadun Rajasinghe*, Jagath Samarabandu†, Xianbin Wang‡
Department of Electrical and Computer Engineering
University of Western Ontario, London, Ontario
* nrajasin@uwo.ca , † jagath@uwo.ca , ‡ xianbin.wang@uwo.ca

***Abstract*—One critical challenge in design and operation of network intrusion detection systems (IDS) is the limited datasets used for IDS training and its impact on the system performance. If the training dataset is not updated or lacks necessary attributes, it will affect the performance of the IDS. To overcome this challenge, we propose a highly customizable software framework capable of generating labeled network intrusion datasets on demand. In addition to the capability to customize attributes, it accepts two modes of data input and output. One input method is to collect real-time data by running the software at a chosen network node and the other is to get Raw PCAP files from another data provider. The output can be either Raw PCAP with selected attributes per packet or a processed dataset with customized attributes related to both individual packet features and overall traffic behavior within a time window. The abilities of this software are compared with a product which has similar intentions and notable novelties and capabilities of the proposed system have been noted.**

I. Introduction

Network intrusion techniques are continuously evolving and intrusion detection algorithms must evolve alongside to keep pace. Network Intrusion Detection System (NIDS) community builds and tests security algorithms based on standard datasets. A common problem with standard datasets is that they are a snapshot(often simulated) of intrusion and normal traffic at the time of creation and quickly becomes outdated when newer types of network intrusions occur [1]. Another issue with datasets is the limitation of the attacks that are presented in them [1], [2].

Intrusion detection systems (IDS) encounter different types of traffic depending on the type of network and the nature of services offered. However, standard datasets are created on a specific network under a specific scenario. For example in the case of DARPA dataset, they created their own network and collected intrusion traffic for the attacks they simulated [3]. The other popular datasets that were derived from DARPA were KDD 99 and NSL-KDD which also suffer from the same drawback [4].

There have been attempts to create more recent Datasets by different researchers with improvements in the number and quantity of attacks, the quality of networks where data is captured from, attack simulation methods etc. [5]. Still, when a dataset user tries to train their IDS with these newer datasets, they will be training on a subset of scenarios the dataset creators faced even though the IDS user will not be facing the same threats locally [6]. It is also likely that the dataset user runs different versions of software and services than what was used by the dataset creators, causing the user to have a different set of vulnerabilities.

Another factor to consider is the security features that were present in the network that hosted the dataset creation platforms. Firewalls and encrypted communication channels are common security features in a network and traffic observed with and without these features are different to each other. This acts as further justification for the need of a custom dataset creation method, that would produce datasets specialized for local traffic.

There have been a few attempts to create on demand datasets and all of them have limitations in different aspects (see next sub section for details).

In this paper we propose an on-demand dataset creation software that can be run on a network of choice and gives the user the ability to fully customize the dataset according to their requirements. Details on the customizability and benefits of our implementation are given in section II and section III. The dataset generated by the proposed software framework includes not only just the raw PCAP files for each packet but also a processed dataset file in csv or JSON format, that can be fed into a machine learning tool easily. This software will also be made available under MIT license so that the research community is able to create much new and relevant data sets by customizing every aspect of the dataset.

A. Related work

Although tools that can create custom data sets have been proposed, they can only insert attack packets into the network traffic capture files and create a Raw dataset [1][2]. A key feature of the proposed data set creation software (INSecS-DCS) is the ability to process the data captured from the PCAP files in order to construct customized and meaningful attributes in a logical manner. The output is similar to the KDD 99 and NSL-KDD datasets in presentation style, where the dataset can be directly fed into a learning algorithm with no preprocessing because of its tabular form.

Shiravi *et al.* [7] has proposed a profile based custom dataset creation methodology where profiles are derived from observed traffic to represent certain features or events captured from network traffic. The profiles are based on HTTP, SMTP, SSH, IMAP, POP3, and FTP protocols and the output is a raw dataset. In contrast, INSecS-DCS allows the users to add any

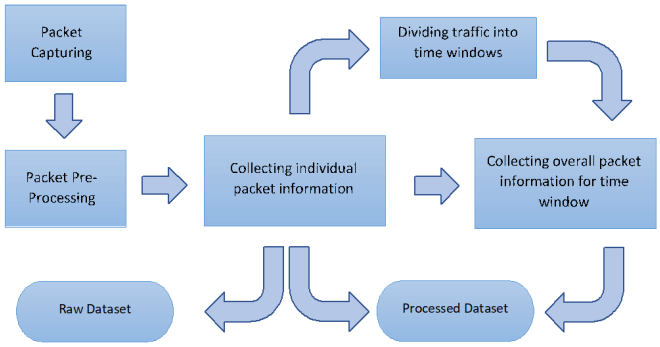

Fig. 1. The flow of dataset creation

protocol they wish to track and set any attribute they wish to see in their dataset.

The remainder of this paper is presented in the following manner. Section II explains the key factors to be considered in dataset creation together with comparisons on various strategies used in the standard datasets. Section III outlines our methodology and test cases, followed by an evaluation of the proposed software and future work.

II. Dataset Requirements

A dataset should possess certain qualities to be deemed fit to be used in the field of network intrusion detection and these have been identified and verified by researchers [8]. In this section we highlight several key requirements and their importance along with details on how we accommodate them in our software.

A. Realistic nature of Data

Simulated attacks, generated using intrusion tools, are used to create intrusion traffic in most standard datasets [3][9]. The exception is datasets created using honeypots where the traffic you get at the honeypot is always intrusion traffic [10]. Both approaches produces realistic intrusion data and thus we allow the user to select either option. This is achieved by operating the software package at any suitable network node and a honeypot, respectively.

B. Proper labeling of Dataset

Labeling an intrusion dataset properly is critical regardless of whether it is being used to train an IDS or used for research. Insufficient or inaccurate labeling requires users to utilize unsupervised learning methods to determine any attack classes or manually label the dataset based on anomaly identification [7] which takes a considerable amount of time and effort. Our software does automatic labeling of records provided that the attack IPs are known.

C. Accuracy of normal traffic

When collecting data to be included in the normal traffic portion of the dataset, there is a possibility that intrusion traffic may get included in it, unknown to the dataset creators. This is due to undetected attacks present in traffic that is assumed to be normal. This is avoided by generating normal traffic by simulation or getting a trustworthy group of volunteers to generate the normal traffic by interacting in the network habitually.

D. Customizability of Dataset

This is a novel and yet indispensable property of the output from INSecS-DCS, as compared to other dataset creation methods and tools discussed above. What we offer is the chance for the software user to be able to add more protocols to be tracked, filter out the undesired traffic, decide on where the data collection should happen inside the network, decide which attributes to have in the dataset, change the time window for average traffic analysis and the ability to run the software inside the user's network, server or personal computer to see what the traffic looks like and use that data to find what the possible threats are.

E. Payload availability

This is a debatable topic due to the loss of privacy of the original owners of the captured traffic [11]. There are anonymizing techniques to hide this information [12]. However, it is not a service we provide with the software. We provide the ability to get the payload in the raw dataset format but leave it upto the users to anonymize the data they capture when creating datasets inside networks.

F. Freedom to choose input

We offer two choices of data input to create datasets using our software. One is to run the software on a network node and generate a raw (records are in PCAP format) or processed (records are saved in csv file in a tabular form, similar to NSL-KDD dataset) dataset based on captured traffic (see figure 1). The other option is to use raw PCAPs provided by other dataset providers and feed it to the software and generate a raw dataset or processed dataset.

G. Freedom to structure of output Dataset

The dataset generated from our software can be in two forms. The raw form which only allows attributes related to individual packets be present. The alternative is the processed format which allows attributes to represent individual packets

TABLE I
OVERALL ATTRIBUTES FOR TIME WINDOW

Feature name	Feature description	Type
connection pairs	the number of different source and destination pairs.	Discrete Integer
IP addresses	the different IP addresses used in the connections	Discrete Integer
num ports	number of different port numbers used	Discrete Integer
num packets	total number of packets sent and received	Discrete Integer
src bytes	the total amount of source traffic	Discrete Integer
dst bytes	the total amount of destination traffic	Discrete Integer
tcp frame length	the total amount of frame bytes for TCP traffic	Discrete Integer
tcp ip length	the total amount of IP data for TCP traffic	Discrete Integer
tcp length	the total amount of TCP data	Discrete Integer
udp frame length	the total amount of frame bytes for UDP traffic	Discrete Integer
udp ip length	the total amount of IP data for UDP traffic	Discrete Integer
udp length	the total amount of UDP data	Discrete Integer
arp frame length	the total amount of frame bytes for ARP traffic	Discrete Integer
num tcp	total number of tcp packets sent and received	Discrete Integer
num udp	total number of udp packets sent and received	Discrete Integer
num arp	total number of arp packets sent and received	Discrete Integer
num ssl	total number of packets containing SSL traffic	Discrete Integer
num http	total number of http requests that were sent and received	Discrete Integer
num ftp	total number of ftp packets sent and received	Discrete Integer
num ssh	total number of packets containing SSH traffic	Discrete Integer
num smtp	total number packets containing SMTP traffic	Discrete Integer
num dhcp	total number packets containing DHCP traffic	Discrete Integer
num dns	total number packets using DNS traffic	Discrete Integer

as well as average properties of network traffic for a specific time window. The added advantage of this processed dataset is that it can be fed directly into ML algorithms without any preprocessing, which in turn saves a lot of time and effort.

III. METHODOLOGY

The workflow used in the dataset creation process is shown in figure 1. In this section, the procedure is described in detail. This software was developed in Python due to the availability of many libraries related to network packet processing. This also allowed this software to work easily with our future developments.

A. Dataset creation

The steps involved in the dataset creation is given in the following subsections of the paper. Each step has its own software component available for public use under an MIT license.

1) Packet capturing : First step in dataset creation is capturing network packets. After evaluating several packet capture tools, we selected *tshark* as the packet capturing tool as it provides a great GUI to view the captured packets with custom filters. No filtering is done at this stage and all the information available on each packet is captured.

2) Packet pre processing : Captured packets are in PCAP format and preprocessing involves converting these PCAP files to JSON format for easier processing. The rationale behind this that we plan to connect future projects with this software and it is important that we use the same data format. The preprocessor converts the data in a packet into key value pairs, where the key is the name given by *tshark* for the value, which are then stored in a python Dictionary.

3) Collecting individual packet features : During the next step, features for individual packets are collected by selecting key value pairs of interest. These features include the protocols used (such as TCP, UDP, IP, FTP, SMTP, SSH, SSL, ARP, DHCP,HTTP etc), the source and destination addresses (IP address) and port numbers used etc. The software comes with a list of preselected attributes but the user can customize this.

4) Dividing the individual traffic flow into time windows and collecting overall features : An important step in creating a processed dataset involves selecting a time window and analyzing the traffic flow during that time [13]. This will give information about the overall traffic behavior during given time interval and identify common trends in traffic; as opposed to considering information from individual packets only. For example if both TCP and UDP packets are used in an exploit, analyzing the TCP stream alone or the UDP stream alone would not suffice. With that taken into consideration, we collect certain parameters about total traffic flow, such as total length of TCP packets sent, total length of UDP packets sent etc. This is done for a time window, where the length is chosen by the user.

B. The processed dataset

The term Processed dataset in this paper is an intrusion dataset that is converted to a tabular format. The format is similar to that of the popular NSL-KDD dataset, which provides the dataset in a tabular manner. The processed dataset can be considered as the standard output of the INSecS-DCS. The attributes in the dataset contain information related to individual packets as well as information about overall traffic behavior during a specified time window. This is a much better dataset in terms of usability and usefulness as explained previously. Details of the attributes used are included later

TABLE II
CAPABILITIES OF DATASET CREATORS

Capability	INSecS-DCS	ID2T
Ability to Label dataset	yes	yes
Open Source	yes	yes
Raw PCAP dataset	yes	yes
Has a GUI	no	yes
Allows attack injection within the software	no	yes
Ability to divide traffic into time window and get overall traffic attributes	yes	no
Ability to select input method (packets captured on a network of choice or get a raw PCAP dataset from another source)	yes	no
Processed dataset that can fed into WEKA and other ML tools directly	yes	no
Attribute selection for processed dataset	yes	no

in this section. The steps in creating the processed dataset is depicted in figure 1.

1) The attributes in the processed dataset : The built-in attributes in the processed data set that are, related to overall traffic behavior within a time window, are given in the table I. A key factor to be considered here is the ability for the user to add more features or remove features in their dataset creation.

C. The raw dataset

Unprocessed or raw dataset is a dataset that presents data in a raw PCAP format. Each record contains attributes related to a single packet which has been preprocessed. The attributes include all the data related to the packet, including payload. Figure 1 shows the steps involved. This process takes less time than creating the processed dataset. This would provide researchers the ability to set their own attributes when they want to find the overall traffic information.

IV. EVALUATION

The tool or software that is closest to our software (INSecS-DCS) is the ID2T toolkit developed by Vasilomanolakis *et al.* [1] at the Telecooperation Lab, Technische Universit Darmstadt. In order to evaluate INSecS-DCS, compared to ID2T toolkit, key advantages of INSecS-DCS are highlighted in table II.

The attack generation to capture intrusion traffic is built into the software in ID2T implementation. This is done by attack scripts and PCAP modification. However, network attack generation softwares are already highly advanced and the hacker community keeps on developing better versions. Thus, we felt no need to include attack generation as an inbuilt software feature.

V. CONCLUSION AND FUTURE WORK

This paper introduced a software framework (INSecS-DCS) that is capable of creating a labeled network intrusion dataset, with the option to choose between a raw and processed dataset as the output. A real-time packet capture stream as well as imported PCAP files are accepted as inputs. INSecS-DCS is highly customizable and depending on the needs of the researcher or IDS user, they are able to recreate a dataset with the attributes they want, instead of the attributes selected by another researcher.

In future, we plan to utilize the dataset creation capability to provide a Real Time Network Intrusion Detection System (Real Time NIDS), with real time datasets. Customization options of the dataset will give the NIDS user, the ability to customize the performance of the NIDS to fit the local network. This would have a huge impact on commercial and private network security industry.

REFERENCES

[1] E. Vasilomanolakis, C. G. Cordero, N. Milanov, and M. Mühlhäuser, "Towards the creation of synthetic, yet realistic, intrusion detection datasets," *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, pp. 1209–1214, 2016.

[2] C. G. Cordero, E. Vasilomanolakis, N. Milanov, C. Koch, D. Hausheer, and M. Muhlhauser, "ID2T: A DIY dataset creation toolkit for Intrusion Detection Systems," *2015 IEEE Conference on Communications and NetworkSecurity, CNS 2015*, no. December, pp. 739–740, 2015.

[3] V. L. Cao, V. T. Hoang, and Q. U. Nguyen, "A scheme for building a dataset for intrusion detection systems," *2013 3rd World Congress on Information and Communication Technologies, WICT 2013*, pp. 280–284, 2014.

[4] G. Meena and R. R. Choudhary, "A review paper on ids classification using kdd 99 and nsl kdd dataset in weka," in *2017 International Conference on Computer, Communications and Electronics (Comptelix)*, July 2017, pp. 553–558.

[5] S. Bhattacharya and S. Selvakumar, "Ssenet-2014 dataset: A dataset for detection of multiconnection attacks," in *2014 3rd International Conference on Eco-friendly Computing and Communication Systems*, Dec 2014, pp. 121–126.

[6] A. Gharib, I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "An Evaluation Framework for Intrusion Detection Dataset," *ICISS 2016 - 2016 International Conference on Information Science and Security*, no. Cic, pp. 0–4, 2017.

[7] A. Shiravi, H. Shiravi, M. Tavallaee, and A. A. Ghorbani, "Toward developing a systematic approach to generate benchmark datasets for intrusion detection," *Computers and Security*, vol. 31, no. 3, pp. 357–374, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.cose.2011.12.012

[8] R. Zuech, T. M. Khoshgoftaar, N. Seliya, M. M. Najafabadi, and C. Kemp, "A New Intrusion Detection Benchmarking System," *Proceedings of the Twenty-Eighth International Florida Artificial Intelligence Research Society Conference*, no. McHugh, pp. 252–255, 2015.

[9] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, pp. 1–6, 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7348942/

[10] J. Song, H. Takakura, Y. Okabe, M. Eto, D. Inoue, and K. Nakao, "Statistical analysis of honeypot data and building of Kyoto 2006+ dataset for NIDS evaluation," *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security - BADGERS '11*, pp. 29–36, 2011. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1978672.1978676

[11] M. Maowidzki, P. Berezinski, and M. Mazur, "Network intrusion detection: Half a kingdom for a good dataset," 04 2015.

[12] S. E. Coull, F. Monrose, M. K. Reiter, and M. Bailey, "The challenges of effectively anonymizing network data," in *2009 Cybersecurity Applications Technology Conference for Homeland Security*, March 2009, pp. 230–236.

[13] P. Aggarwal and S. K. Sharma, "Analysis of KDD Dataset Attributes - Class wise for Intrusion Detection," *Procedia Computer Science*, vol. 57, pp. 842–851, 2015. [Online]. Available: http://dx.doi.org/10.1016/j.procs.2015.07.490