Traffic Generation using Containerization for Machine Learning

ABSTRACT

KEYWORDS

Network security, datasets, machine learning, intrusion detection

ACM Reference Format:
. 2019. Traffic Generation using Containerization for Machine Learning. In *DYNAMICS '19: DYnamic and Novel Advances in Machine Learning and Intelligent Cyber Security Workshop, December 09–10, 2019, San Juan, PR.* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

. . . . . .

. . . . . .

. . . . . .

This work provides the following contributions:

(1) We present a novel network traffic generation framework that is designed to improve several shortcomings of current datasets for NIDS evaluation. This framework is openly accessible for researchers and allows for straightforward customization.
(2) We define four new requirements a network intrusion dataset should fulfil in order to be suitable to train machine-learning based intrusion detection methods.
(3) We perform a number of experiments to demonstrate the suitability and utility of our framework.

1.1 Outline

The remainder of the paper is organized as follows. Section 2 discusses existing NIDS datasets and the problems that arise during their usage as well as background information about network traffic data formats and virtualization methods. The section concludes with a set of requirements we propose to improve the training and evaluation of machine-learning-based methods. Section 4 describes the general design of our framework, and how it improves on the discussed problems in existing datasets. We also discuss a specific example in detail. Section 5 discusses several experiments to validate the improvements and utility our framework provides. Section 7 concludes the results and discusses limitations of our work and directions for future work.

DYNAMICS '19, December 09–10, 2019, San Juan, PR

ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

2 BACKGROUND

2.1 Data formats

2.2 Related work and existing datasets

2.3 Problems in modern datasets

We can import here a lot from the existing paper, but add the following issues:

(1)
(2)
(3)

2.4 Containerization with Docker

2.5 MiniNet

3 DATASET REQUIREMENTS

We can refer to requirements by Cordero et al. (https://arxiv.org/pdf/1905.0 on requirements for generating synthetic datasets, and combine them with the existing set of requirements by us.

These include:

3.1 Benefits of our framework

This should potentially be merged with the dataset requirements, I am currently unsure where to put this.

1. Fidelity to real traffic 1. Real traffic, consistent (not invalid after Cordero et al.) 2. Structural richness on packet level (in contrast to) Induced due to the different levels at which traffic variation is introduced 3. Temporal activity levels? (actually not something we improve) We can look at test for realism of distributions (IP discovery, etc)

2. Ground truth labels through containerisation 1. Ground truth for attack behaviour, able to label 1002. Labels for different types of behaviour, reproducable useful for evaluation of model failures, what kind of behaviours cause failure applies to a large range of models also useful for evaluation of privacy infiltration methods, more niche 3. Ground truth for label matching between traffic and program logs/sys logs useful for models that try to correlate events for detection this is more niche, but potentially because of the lack of data

3. Extensive capture 1. Packet availability 2. Syslogs and for multiple scenarios program logs 3. Potentially host logs? Depends if we want to cater to cloud computing applicability

4. Better for ML-based methods 1. Flexibility "The models should allow researchers to generate different classes of data, such as augmenting the amount of data representing sparse events, or choose different topology" 2. Automisation of variable datasets through randomisation, automatically create structurally different datasets, but faithful to realism Especially novel in terms of network topologies, should emphasise this in use-cases 3. Structural richness allows for learning deeper and more generalisable knowledge in models,

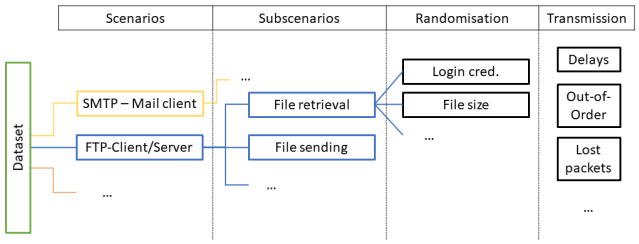

Figure 1: Visualization of the different levels at which traffic variation is introduced in DetGen.

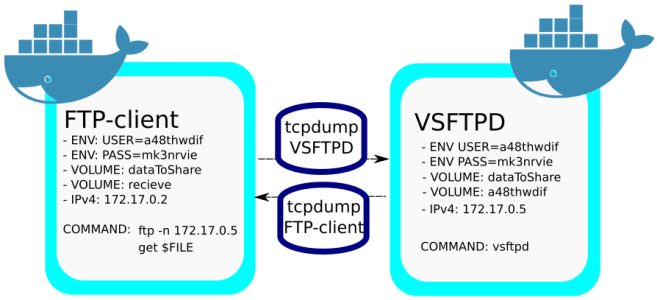

Figure 2: Diagram of FTP scenario

less prone to overfitting 2. Scalability "Train on as much data as necessary"

Variation.

Ground truth.

Modularity.

Scalability.

4 DESIGN

4.1 Modes of Operation

4.2 Scenarios and subscenarios

4.3 Randomization

4.4 Network transmission

4.5 Implementation Process

4.6 Implemented scenarios

4.7 Network-simulation mode

4.7.1 Dataset coalescence.

4.8 Network-simulation mode

5 FIDELITY CONFIRMATION EXPERIMENTS

This section is important to demonstrate that our data is valid and overcomes the difficulties entailed with synthetic data generation. Cordero et al. have proposed some more simple test that we can refer to first

Question to be answered: What requirements are there for the additional data, program logs and system logs, that we collect? Should we put less emphasise on these data sources in general if we are not able to perform these tests, and refer to them in future work? I am not aware of any papers that discuss these requirements in a similar way.

5.1 Data correctness tests

This section is concerned with dataset defects, artifacts, or invalid data (inconsistent MTU etc.). These are very straightforward to test and should not take up much space.

5.2 Diversity tests

These tests, also from Cordero et al. quantify diversity via the entropy of different quantities such as IP diversity, Time-to-Live, Maximum-segment-size, Window size, ToS. I think we should keep this relatively short and omit comparison to other datasets since this is already done by Cordero et al.

5.3 Structural richness and predictability

To quantify the

This experiment is more novel and should be a more significant contribution to the paper

Measure structural richness of Also measure divergence across same activities (same activity and same port) Demonstrate benefit of structural richness Closer to reality

5.4 Reproducible scenarios

5.5 Explorating Artificial Delays

This section is already existing, we could potentially expand this. I think it is sufficient and analysing it more does not add much to the paper as the performance of TC netem is relatively well accepted. I think we could even move this section to the appendix

Distribution	Mean	Jitter	RF Accuracy
No Delays (Baseline)	0	0ms	0.8176
Constant Delay	40ms	0ms	0.6730
Normal	60ms	5ms	0.6028
Pareto	60ms	10ms	0.5979
Paretonormal	50ms	10ms	0.6015
Weibull	60ms	10ms	0.5540

Table 1: Worst Random Forest accuracy rates for a given distribution

6 USE-CASES

6.1 Benefits of ground-truth labels/Benefits of Dynamic Dataset Generation

Extensive ground-truth labels for our activities are arguably the most important contribution of the DetGen framework, so we should highlight their benefit more. Since the benefit of ground-truth attack data is obvious, we should emphasise the benefit of having labels for different activities. In my eyes,

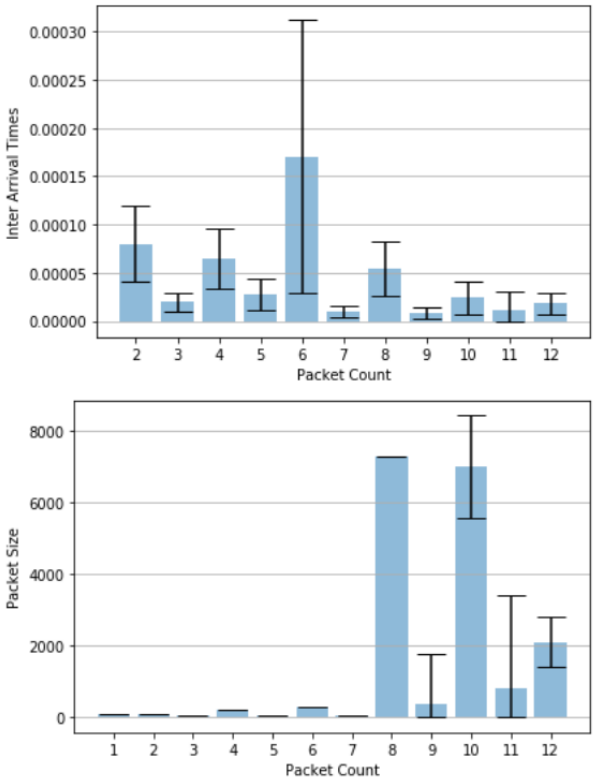

Figure 3: Means of IATs & packet sizes along with standard deviation bars for the first twelve packets in the Apache scenario.

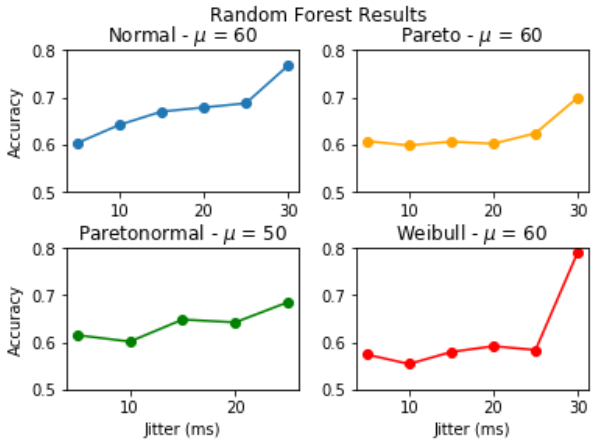

Figure 4: Results of Random Forest Classifier for a given distribution at the best performing delay mean μ. Note that a score of .5 indicates total indistinguishability.

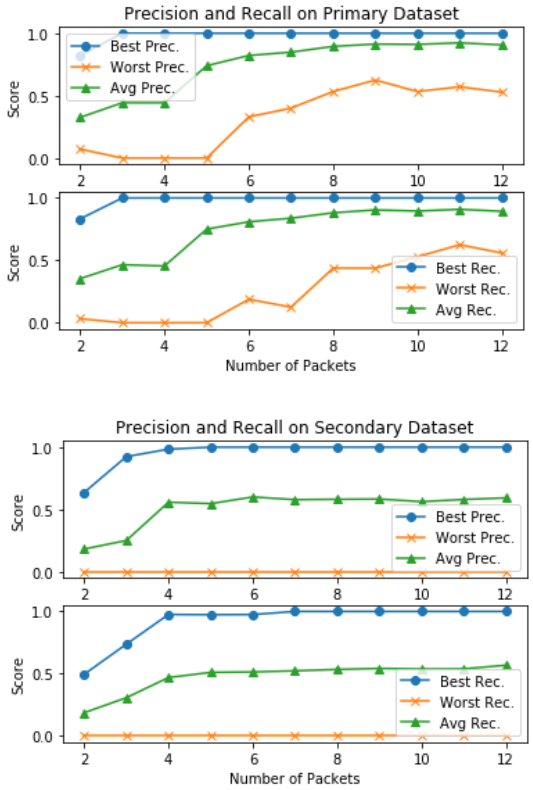

Figure 5: Results of Random Forest Classification on Primary dataset (Above) and Secondary dataset (Below)

the most striking benefit arises for false-positive analysis, which we could then combine with showcasing the benefit of being able to generate different amounts of traffic for different activities.

Idea 1: Implement the LSTM in the paper "An LSTM-Based Deep Learning Approach forClassifying Malicious Traffic at the Packet Level", train it on our data (both benign and attack traffic). Extract labels of traffic responsible for false-positives, show how much they are clustered around particular activities (potentially rare activities) compared to the overall traffic. Give potential reason for this. Generate a new dataset with increased amounts of the activities responsible for false positivies. Demonstrate that false-positives decrease.

Idea 2:

6.1.1 Show utility of flexible topology. o Find useful model, show that training on one topology leads to overfitting o Show that training on multiple datasets prevents overfitting Or that detection results can differ vastly • Measure effect of new traffic types on IDS performance -

6.1.2 Show utility of tuning amount of rare events.

6.2 Benefits of structural richness

- Learned model larger and detection problem is closer to reality - Demonstrate this by learning a model with less

7 CONCLUSIONS

7.1 Difficulties and limitations

7.2 Future work