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, 3 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 in the following aspects:
  (a) Ground truth labels documenting conducted activities
  (b) Increased richness of data
  (c) Tunable topology and data composition
  (d) Additional capture of program logs and system calls

  This framework is openly accessible for researchers and allows for straightforward customization.

(2) We perform a number of experiments to demonstrate the fidelity to realism of the generated data.

(3) We present a number of use-cases to demonstrate how the design of our framework can boost performance for ML-based network intrusion detection systems, in particular for false-positive analysis and effective model training.

1.1 Outline

……

……

……

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/1905.00304.pdf) 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
  - Real traffic, consistent (not invalid after Cordero et al.)
  - Structural richness on packet level (in contrast to) Induced due to the different levels at which traffic variation is introduced
  - 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
  - Ground truth for attack behaviour, able to label 100
  - 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
  - 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
  - Packet availability
  - Syslogs and for multiple scenarios program logs

DYNAMICS '19, December 09–10, 2019, San Juan, PR
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-9999-9/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

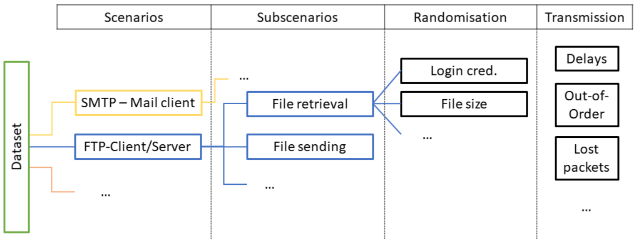

Figure 1: Visualization of the different levels at which traffic variation is introduced in DetGen.

- Potentially host logs? Depends if we want to cater to cloud computing applicability

(4) Better for ML-based methods
- 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"
- 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
- Structural richness allows for learning deeper and more generalisable knowledge in models, less prone to overfitting
- 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

......

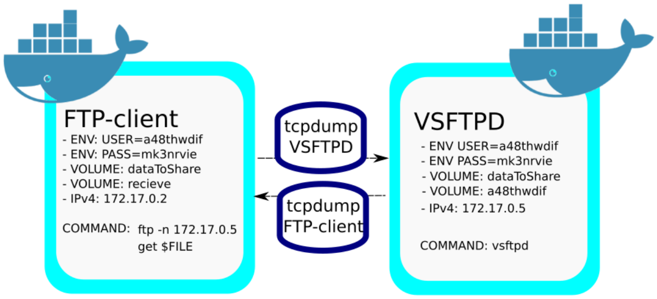

Figure 2: Diagram of FTP scenario

......

......

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 dataset dimensionality

Autoencoders are often used to compress non-deterministic, noisy data. Bahadur et al. have developed a procedure to estimate the „dimensionality" (to be understood as the complexity) of a dataset using variational autoencoders. I believe we can transfer this concept to sequence compression and estimate the overall complexity of connection sequences in our framework with both real traffic captures and existing network intrusion datasets. Showing that our data is closer to real-world data would be a good test for "artificially predictable patterns", as described by Cordero et al., and go hand-in-hand with demonstrating the benefits of our framework for the training of deep-learning models.

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

6 USE-CASES

6.1 Benefits of ground-truth labels/dynamic dataset generation

Possible title: **Dataset tuning to decrease false-positives**

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

Plan. Implement the LSTM-model 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