Progress Report: Henry Clausen

Henry Clausen

March 8, 2020

0.1 Problems with publishing in Network Anomaly Detection

The original scope of this PhD-project was to built contextual models that represent software behaviour primarily from network traffic and potentially other external data sources, for adaptive anomaly intrusion detection. Over the course of the last year, it became clear that the successful publication of anomaly-based methods network intrusion detection methods is difficult today. This

This ... was reflected during our unsuccessful attempts to publish ... at the ACSAC and CODASPY conferences. Despite improving detection rates and false positive rates significantly on U2R and R2L on contemporary datasets, the main criticisms were a lack of novelty in terms of intrusion detection model and a general scepticism of real-world applicability. This is in line with conversations I had with senior researchers at the ACSAC conferences, who highlighted that anomaly-based methods have been applied to network traffic for over 15 years without convincing results. General advice were to identify very specific applications that have the potential to yield good results. A similar advice comes from bring in Sommer and Paxson, who advise carfing out specific applications

0.2 Short-term contextual model of network flows using LSTM networks

In the context of the overall research goal of this project, some exploratory work on contextual anomaly detection for network events was conducted before my PhD started by Marc Sabate, Gudmund Grov, Wei Chen, and David Aspinall. This work uses a recurrent neural network to capture meaningful sequences of *NetFlows* and reflect reccurring patterns in a model. For that, recorded NetFlows are grouped according to the generating host. Furthermore, to filter out sequences of flows that are unrelated to each other, a squence of flows that are close in time is grouped into what is called a "session" as an approximation of the true relation. Each session then serves as a training or test sequence for a behavioural model. Learned semantic behaviour is reflected through the capability of the model to predict traffic protocols and network ports of flows in a session from a smaller subset of flows, with more accurate predictions being rewarded in the training process. Sessions which deviate from previously observed behaviour are then predicted poorly by the model and flagged as potentially malicious.

This work however was not yet complete for successful publication, which is why I took the responsibilty to bring it to a state of .... During my work, I made significant changes and additions to both improve the model performance as well as extend the evaluation to highlight the benefits and novelty of this work. In particular, I made the following major changes:

1. I specified the scope of the model to the detection of U2R and R2L attacks, to which it is more suited than high volume attacks. I also exchanged the evaluation datasets to mutiple ones that are more suitable for this task and more realistic in nature.

2. Originally, the model only incorporated the protocol and the port for each flow. I extended this to the direction (from or to host) as well as the size of each flow, which improved detection rates especially for brute-force attacks and sql injections. For this, I came up with an efficient way to process these additional inputs without significant increase of model size.

3. Similarly, the original model was a standard recurrent neural network with one layer. I extended this to a bidirectional LSTM network with multiple layers, which required careful parameter calibration and further boosted detection rates.

4. The detection method was changed from an overall host classification to a session classification using a scoring threshold, which is more realistic in a real-world deployment due to the high imbalance between benign and malicious traffic.

5. I carfed out the novelty and contribution of this work more, and implemented three comparision benchmark models to highlight the benefit of our model. I also highlighted the differences between our work and recent applications of LSTMs to intrusion detection.

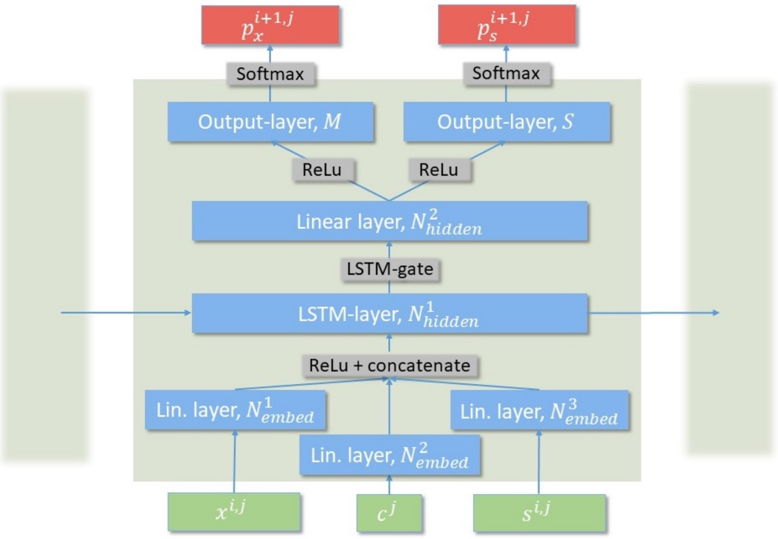

Figure 1: Bla

Since the paper is now in under my responsibility, and due to all the efforts I put into its improvement, we agreed that I would be the first author of this work in the event of publication.

difficulties [1]

0.3 Traffic Generation using Containerization for Machine Learning

Building contextual models of network traffic means to build an understanding how different network interactions can be distinguished via their traffic trace. However, available network traffic datasets do not contain ground truth labels about the nature of computer interactions and often suffer from a lack of realism. To improve this and ensure that our models extract meaningful sets of sequences that represent these different interactions, we started developing a containerised traffic generation framework to generate traffic with ground truth labels.

This framework generates controlled and isolated network traffic from a variety of applications and tasks. For this, a virtual network was created using the virtualisation program *Docker*. In this network, two or more parties can communicate through containers, which are sandboxes containing programs inside a minimal virtualised operating system. The benefit of this design is that individual containers can only communicate with each other via the virtualised network while the host is in complete control of the parallel execution of tasks in multiple containers. To capture the traffic, every container in the network was complemented with a *tcpdump*[1] container hooked onto the network interface. The captured traffic can then be labelled according to the particular scenario it was generated by. We implemented a variety of network service scenarios to capture a diverse set of network traffic.

We emphasised the following particular strengths in our framework, which makes the generated data particularly suitable for ML-applications:

[1] A common packet capture utility

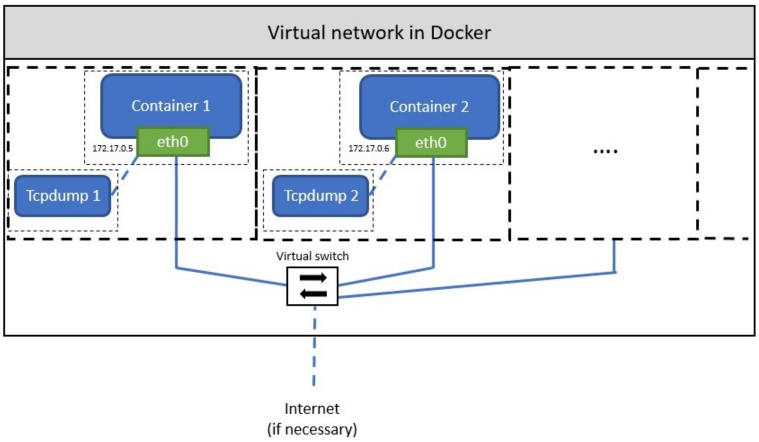

Figure 2: Visualisation of the virtual network in Docker

- traffic variation through a diversity of generation scenarios as well as transmission disturbances,
- ground truth labels through containerised separation of generation scenarios,
- scalability of the amount of generated traffic,
- and modularity of the framework to easily extend and update the set of scenarios.

This work was started in summer 2018 by me and Nikola Pavlov[2], and continued in summer 2019 by me and Robert Flood[3]. My responsibilities lied in overall design of the framework as well as the requirements on the generation features, testing and extending individual scenarios as well as implemention, and the design and conduction of validation experiments.

In autumn of 2019, Robert Flood and I described the framework in a paper of which I am first author, which was submitted and accepted at the ACSAC DYNAMICS workshop 2019, and will appear in the corresponding proceedings this March.

Due to very positive feedback at the workshop as well as encouragements and suggestions to extend the framework, we are working further on this project, which is described in Section .....

0.4 1. stint at BT - Stepping stone detection and connection correlation

As part of my CASE PhD scholarship, I spent six weeks with my industrial sponsor at BT Labs Adastral Park in August and September 2019. Before starting this stint, I met with my industrial supervisors to define a project to work on that is both related to my PhD and is a relevant problem for BT. During this meeting, we agreed that the problem of detecting stepping stones was a suitable topic.

[2] an LFCS summer intern

[3] an LFCS MSc student

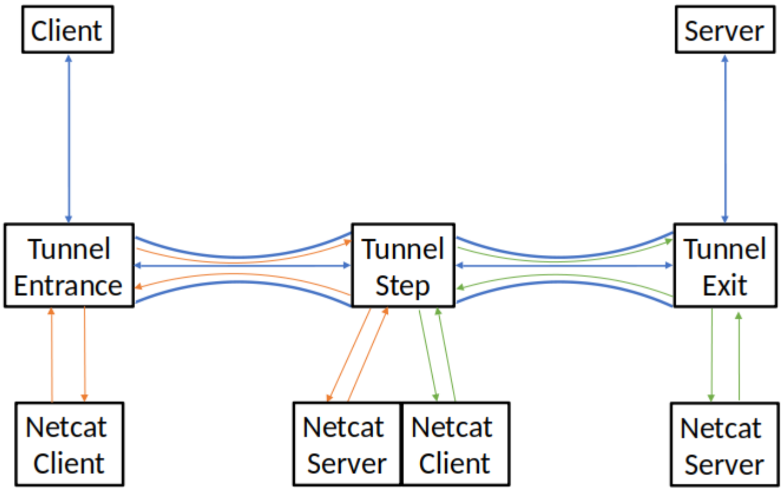

Figure 3: Implementation of ...

In a stepping stone scenario, an attacker launches an attack not from their own computer but from intermediary hosts within an enterprise network that were previously compromised, often using an interactive relay session. A common approach in the literature to detect stepping stones is to identify correlation between an two connections on a potential intermediary host citations. Attackers try to evade detection by inserting chaff packets and delays to make the connection appear uncorrelated.

The biggest challenge for this problem is that there are no available datasets available that describe stepping stone behaviour. Due to the success of the implemented traffic generation framework, I started to implement several scenarios of interactive traffic relays using ssh-tunnels and netcat/netem for chaff and delay insertion. With this, I was able to generate significant amounts of traffic with a controllable amount of noise and delay to train and assess correlation models. I furthermore implemented a state-of-the-art method for connection correlation insert DeepCorr citation that uses a deep convolutional neural network as a benchmark to compare a future model with.

After finishing the stint at BT Labs, I continued to work on the traffic generation for a bit further before we had a call with Jake Hill, a security expert at BT. This call was meant to provide field knowledge to confirm and/or improve the data generation set-up. However, Jake stated that the problem of attackers launching attacks from intermediary hosts is after all not of great relevance for BT's operations. Instead, his notion of stepping stones described simple proxies relaying services (more on this in Section reference. In addition, further investigation suggests that the consensus in the literature is that connection correlation produces too many false positives and is not applicable in real-world scenarios.

With this information, I decided to scale this project down and produce a simple evaluation of existing connection correlation methods. Since the major contribution of this work so far is the large amount of detailed and varied data I can produce, this allows for the first time an independent assessment of existing methods. So far I have implemented and evaluated four different methods, and am in the process of finishing this project. I am currently producing a small paper from it, which I intend to publish at a smaller workshop or conference.

0.5 Modelling of connection establishment via LSTM encoders

0.6 Future plans

Bibliography

[1] R. Sommer and V. Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316, May 2010.