# Semantic Modelling of Network Traffic for Anomaly Detection

PhD 2nd Year Progress Report

Henry Clausen

April 5, 2021



# 1 Research progress in the last year

## 1.1 DetGen framework

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.

After focusing on the dynamic and scalable capabilities of DetGen suitable to train Machine Learning models in the ACSAC DYNAMICS last year, which has now finally been submitted to be published by the DYNAMICS organisers, I tried to emphasise the original design idea of DetGen more: The controlled and reproducible generation of traffic traces with ground truth information to understand how ML-models process specific network activities, which we call *model probing*.

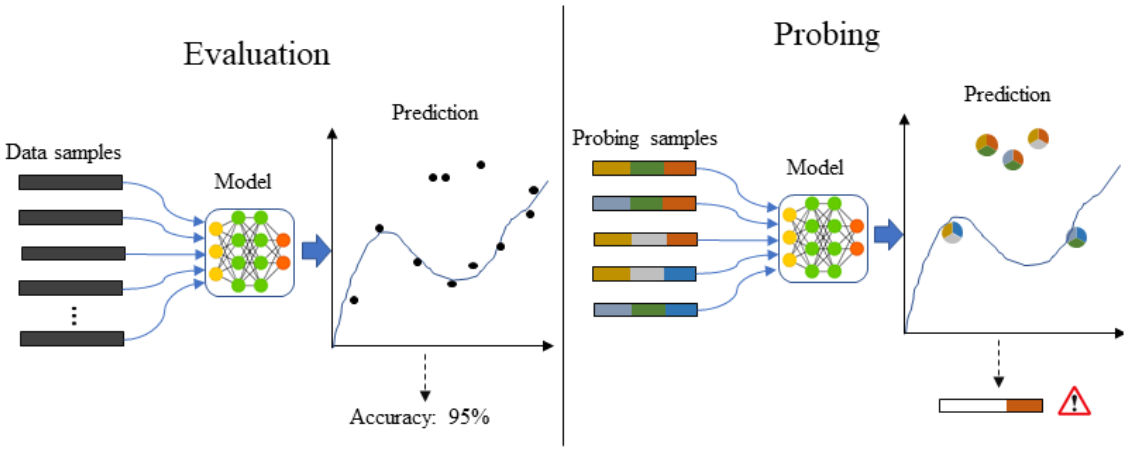


Figure 1: Model evaluation and model probing with controlled data characteristics.

For this, I examined how two state-of-the-art traffic classifiers could be probed, and performed several experiments to examine the control of DetGen over various traffic characteristics as well as the influence of these traffic characteristics on captured traffic traces with three novel traffic comparison metrics. The results from the probing of the classifiers were prepared into a paper and submitted and accepted at the *Workshop on Traffic Measurements for Cybersecurity*, which is hosted at the *S&P*-conference in May. The examination of traffic determinism and their impact on traffic traces along with a detailed description of DetGen were submitted as a paper to the 2021 *SecureComm* conference, for which we will receive acceptance notice on 17th May.

## 1.2 Short-term contextual model of network flows using LSTM networks

One of the main strains of work in year 1 and 2 of my PhD was focused on a building a LSTM-based neural network to capture meaningful sequences of *NetFlows* and reflect reccurring patterns in a model. Learned contextual 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.

Despite the promising performance of this model, several submissions a paper to well-known conferences were unsuccessful and the paper was rejected mostly for a lack of novelty and scepticism of the real-world applicability. Last year, this work was finally accepted at the *Machine Learning for Networking*-conference 2020. Some aspects that I improved for acceptance include:

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. I extended the model input features and model architecture to capture more complex sequences and decrease areas where the model was not performing well.

| Src | Dst | DPort | bytes | # packets |
|---|---|---|---|---|
| A | B | 80 | 247956 | 315 |
| A | B | 80 | 7544 | 13 |
| A | B | 80 | 328 | 6 |
| A | B | 80 | 2601 | 10 |
| A | B | 80 | 328 | 6 |
| A | B | 80 | 328 | 6 |
| A | B | 80 | 380 | 7 |
| A | B | 80 | 328 | 6 |
| | | | ⋮ | |

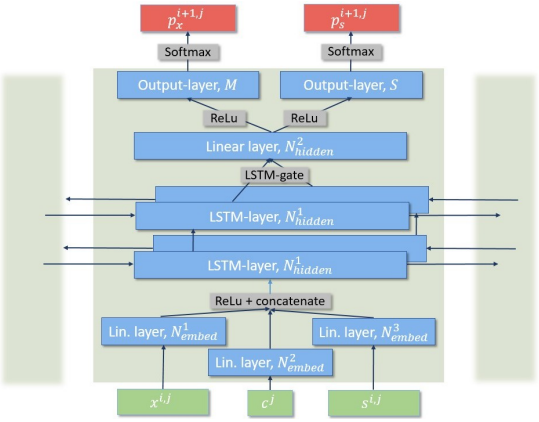


Figure 2: Architectures of the model and corresponding traffic sequence of XSS-attack.

3. I replaced the CTU-14-dataset with the more suitable datasets CICIDS-17 and UGR-16, and identified suitable data traces for modelling and analysis.

4. I focused more on the traffic structures the model is able to detect and explained the corresponding novelty of the model better.

5. I included several state-of-the-art models as benchmark and also compared the now more complex model to a more shallow version to highlight how the learning of specific traffic structures was improved.

## 1.3 Stepping stone detection

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 two connections on a potential intermediary host. 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 DetGen 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.

Since this problem in the described form turned out to not be relevant for BT anymore, I did not proceed to design my own detection method. I instead implemented 7 state-of-the-art methods for connection correlation and performed an extensive evaluation of their performance under various circumstances. The corresponding paper was submitted, accepted and published at the *Network and Systems Security* conference 2020.

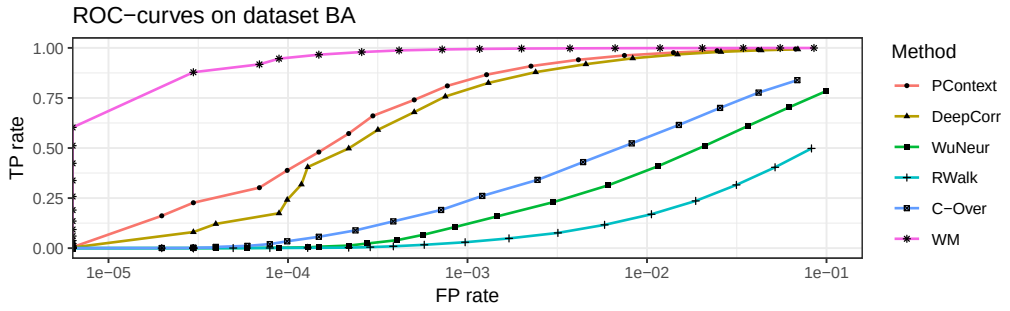


Figure 3: Evaluation results for methods on the stepping stone data.

## 1.4 BT project

I am currently involved in a project with BT on the detection of proxies relaying protected content from the perspective of an ISP. Specific characteristics of this problem are:

- Unlike the original stepping stone problem, these relay proxies operate in a simplistic fashion and do not use evasion tactics. However, content can be forwarded using a different application or protocol than in the first connection.
- Packets are sampled before observation, meaning that only every n-th packet for each connection is observed.
- Benign proxies exist that can increase false-positives.

Since the start of the project, I have been involved in designing a data generation setup to fit our needs and provide sufficiently challenging data comparable with real-world traffic. I have furthermore examined generated data traces for features and characteristics that we could use in the detection process, and considered multiple potentially suitable detection methods:

**Statistical statements about similarity between incoming and outgoing traffic** By comparing features for both incoming and outgoing traffic on a host during a time window, it might be possible to make statements about unlikely similarity between the two directions. This would again be based on distributions of features like mean observed packets sizes, the number of connections, average transferred bytes, etc. A test on all features could then determine likelihood the two channels being a relay.

**Classification-based approaches** Mining the above described features for a variety of hosts, proxy and non-proxy, and training a robust classifier (logitistic regression, decision tree) on it, such as done in “Detecting Scans at the ISP Level”.

**Direct application of Deep-Learning method** The packet stream on a model could be fed directly into an LSTM model to classify each node. Computationally expensive, but could be used as 2nd step once light indicators are triggered. However likely to overfit our data.

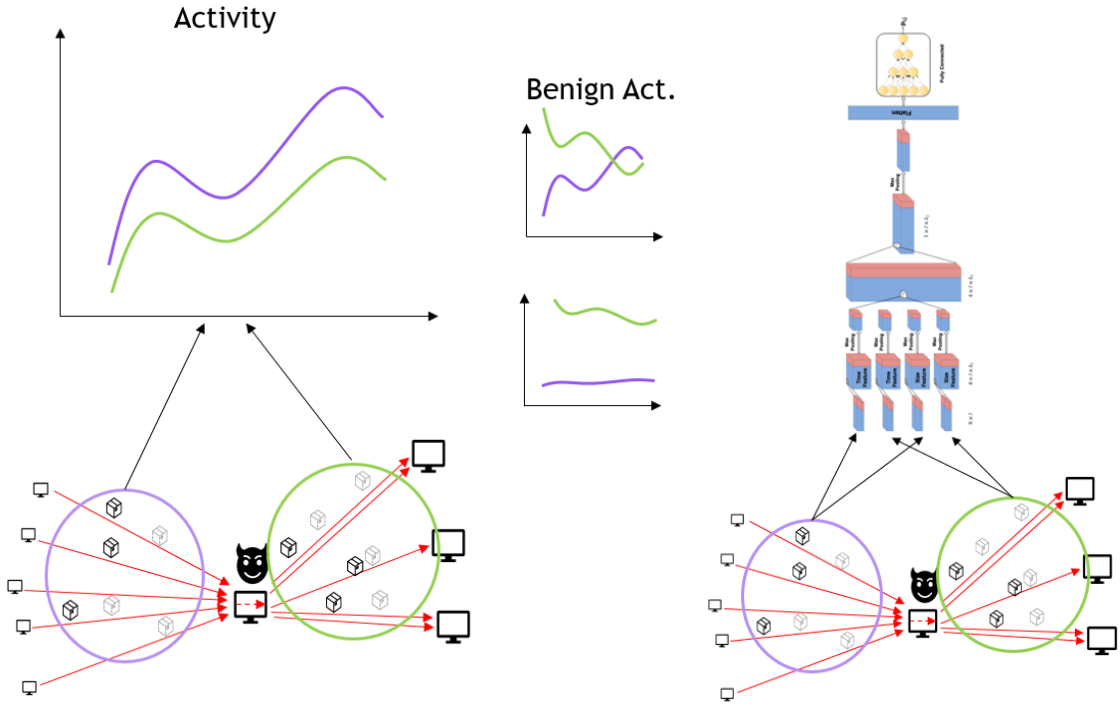


Figure 4: Potential methods to detect proxies.

At the moment, I am in the process of examining the latest data tranche and will test some of the methods on it. The project recently started moving more quickly, and we will hopefully have a good idea which methods might work in the summer (June/July).

# 2 Future plans

## 2.1 QUIC/HTTP examination

QUIC seeks to improve performance of connection-oriented web-applications using TCP. It is implemented on top of UDP and implements its own TCP-like packet control mechanisms. It allows for multiplexed HTTP-connections and resolves head-of-line blocking during packet loss as well as reducing latency by minimising round-trips during connection establishment. Furthermore, much of the QUIC implementation is moved from kernel to user space, which allows continuous updates of the protocol.

According to an NGINX-spokesperson, "since the protocol is new and things like stack optimization etc. still to catch up and since it's all user-space, there is a possibility to make changes to protocol very rapidly. This generates a higher attack surface than you would have over more layered approach with TCP and http on top." Initial investigations showed that there exists no research on mitigating intrusions over the QUIC protocol.

Due to the implementation in user-space, QUIC may introduce vulnerabilities that allow remote code execution on a host, such as in *CVE-2017-15407* and *CVE-2017-15398*. The detection of such code executions by building a combined model of QUIC packet exchanges and process starts/system calls therefore seems like a promising project that provides both novelty and relevance.

In particular, I believe the following conditions would allow for interesting results:

- Operation on a host (server or client) level
- Combination of unencrypted packet packet stream and system calls/process starts corresponding
- A sequential model that applies NLP-techniques to packet content and predicts probability of process start or particular system call (which could then correspond to code execution)
- Traffic and system call/process log collection using a containerized framework.

## 2.2 LSTM anomaly detection: Special issue

After the acceptance and presentation of our LSTM-based anomaly detection model at the MLN conference 2020, we were invited to submit a paper to the *Special Issue "Selected Papers from the 3rd International Conference on Machine Learning for Networking (MLN'2020)"*. Since over time, a significant amount of material has accumulated that was not included in the MLN-paper, this seems to be a good opportunity to get the remaining work published in this special issue. Specific parts of this material include:

- Additional details on model operation, likelihood functions, other methods of performing analysis
- More examination of UGR-content, combined with more analysis of long-term performance
- UGR-port-scan detection results
- More extensive comparison with smaller benchmark LSTM-model
- Results from the LANL dataset

## 2.3 BT Proxy detection project

As described above, I am currently investigating data from BT to design appropriate methods to detect proxies in the wild. The goal of this project is to have produced sufficient data, tested methods appropriately, and concluded their reliability and performance by June/July. Depending on the results, this project could produce a joint publication together with BT, who have ambitions to file a corresponding patent.

# 3 Publications

**Published:**

- *Better Anomaly Detection for Access Attacks Using Deep Bidirectional LSTMs*, H Clausen, G Grov, M Sabate, D Aspinall, Machine Learning for Networking: Third International Conference, MLN 2020, Paris, France, November 24–26, 2020,

- *Evading Stepping-Stone Detection with Enough Chaff*, Clausen, Henry, Michael S. Gibson, and David Aspinall, International Conference on Network and System Security, Melbourne, Australia, November 25-27, 2020

**Accepted and to be published**

- *Traffic generation using containerization for machine learning*, Henry Clausen, Robert Flood, David Aspinall, ACSAC DYNAMICS'19: DYnamic and Novel Advances in Machine Learning and Intelligent Cyber Security Workshop in December 09-10, 2019, San Juan, PR

- *Examining traffic microstructures to improve model development*, Henry Clausen and David Aspinall, Workshop on Traffic Measurements for Cybersecurity at 42th Symposium of Security and Privacy, May 2021, San Francisco

**Submitted**

- *Controlling network traffic microstructures for machine-learning model probing*, Henry Clausen, Robert Flood, David Aspinall, SecureComm September 6-9 2021, Canterbury UK

**Planned**

- *Additional Material on Anomaly Detection for Access Attacks Using Deep Bidirectional LSTMs*, Special Issue "Selected Papers from the 3rd International Conference on Machine Learning for Networking (MLN 2020)", Deadline: April 30th

- *Results from proxy detection project*, Appropriate venue not identified yet

# 4 Thesis structure plan

1. **Central research questions**

   **Research Question 1**
   *How well-structured is the space of contextual behaviours observed in the traffic of a machine or a network? How much does noise or input variation blur the observable contextual differences between clearly distinct actions?*

   **Research Question 2**
   *To what degree can contextual structure in network traffic be captured in a model from a training dataset, and how can we achieve this?*

   **Research Question 3**
   *What is a meaningful representation of traffic structures? What requirements must a labelled traffic generation framework fulfill to provide realistic data?*

   **Research Question 4**
   *What will a contextual model be able to prevent?*

2. **Background.** This section gives a general overview over

   - network intrusion detection
   - anomaly-detection
   - network traffic

3. **Design and requirements on semantic anomaly-detection models.** This section acts as a motivation on

   - what the design of a semantic anomaly-detection model can potentially identify, i.e. anomalous packet or flow sequences that are a results of specific exploits.
   - what model design could capture corresponding structures for the detection

   This section would furthermore discuss

   - why microstructures in the traffic are necessary to be leveraged by such a model
   - what influence different factors could have on these structures and corresponding models

4. **Flow-level semantic anomaly-model and structures it can learn/detect**. This section will contain the majority of the MLN LSTM-paper and the additional material on applications to the LANL data. It will highlight

   - What structures can be observed on a flow level
   - which attacks affect structures on a flow-level, and in what way
   - how our model is able learn flow-level structures and identify these deviations
   - what steps were necessary in terms of model expansion to learn complex structures

5. **Packet-level semantic encoder and how it internalises traffic structures**. I am not sure whether to include this section. It would contain the work I made to build an encoder-decoder model of packet sequences. I did not perform any anomaly-detection with it, but it could give very interesting insights into how a model internalises packet-sequence structures due to the latent-space projection inherent to this model.

6. **Quantifying effect of factors on microstructures.** This section would contain most of the work conducted in the SecureComm-paper, i.e.

   - the proposal and explanation of traffic similarity metrics
   - measurement and analysis of the influence of different factors (congestion, implementations, activities, etc.)

7. **Examining the effect of microstructure perturbations on model performance.** This section would contain most of the work conducted in the WTMC-paper, i.e.

- two examples on how models are affected by traffic influence factors

8. **Effect of adversarial microstructure perturbations on stepping-stone detection** This section would contain the results of the stepping-stone paper, but
   - focus on the perturbations
   - why specific models (DeepCorr, PContext) are not able to identify learned structures anymore when facing sufficient perturbations
9. **Controlling influence on traffic microstructures.** This section would contain the setup of DetGen and the corresponding experimental verifications of its determinism.

## 4.1 Thesis Completion plan

- April
  - Collate LSTM-materials into a paper and submit to special issue on 30th April deadline
  - Finish section 2 and 3 (background and detection requirements) of the thesis
  - Implement simple benchmark methods on proxy-data
  - Discuss final scope of QUIC/HTTP-project
  - Potential work on the QUIC/HTTP-project
- May
  - Extend and test proxy detection methods on larger background data
  - Finish section 4 and 5 (both on anomaly models) of the thesis
  - Potential work on QUIC/HTTP-project
- June
  - Finish section 6 and 7 (WTMC and SecureComm-papers) of the thesis
  - Present proxy detection results at the Stats and Cyber Security Session at JSM 2021 (were I was invited to speak)
  - Revise literature review
  - Potential work on QUIC/HTTP-project
- July
  - Finish proxy-detection work, hopefully with a well-performing model as the result
  - Discuss potential publication of the proxy-detection results
  - Finish section 8, 9, and 1 of the thesis
  - Potential work on the QUIC/HTTP-project

- August
  - Revise thesis and add potential additional examinations (models, traffic)
  - Potential work on the QUIC/HTTP-project
  - Potentially write paper on proxy-detection work (needs permission from BT-supervisors)
- September
  - Revise and submit thesis
  - Prepare for viva

# 5 Impact of COVID-19 on progression

The closure of the PhD-offices in the Informatics forum and the work-from-home directive lead to a less productive work environment at home with significantly more distractions. Roughly I am 1-2 hours less productive every day. I also had to invest a significant amount of time rearranging the work situation appropriately. Face-to-face meetings over Skype have been helpful and easy to set up, but it is more difficult to go through paper drafts or discuss work in detail. Overall, I had significantly less contact and conversations with people who could potentially have valuable input and stimulating thoughts, loss of opportunity to network at international conferences. Internet speed has had a minor impact on data availability.