# Semantic Modelling of Network Traffic for Anomaly Detection

PhD 2nd Year Progress Report

Henry Clausen

March 27, 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.

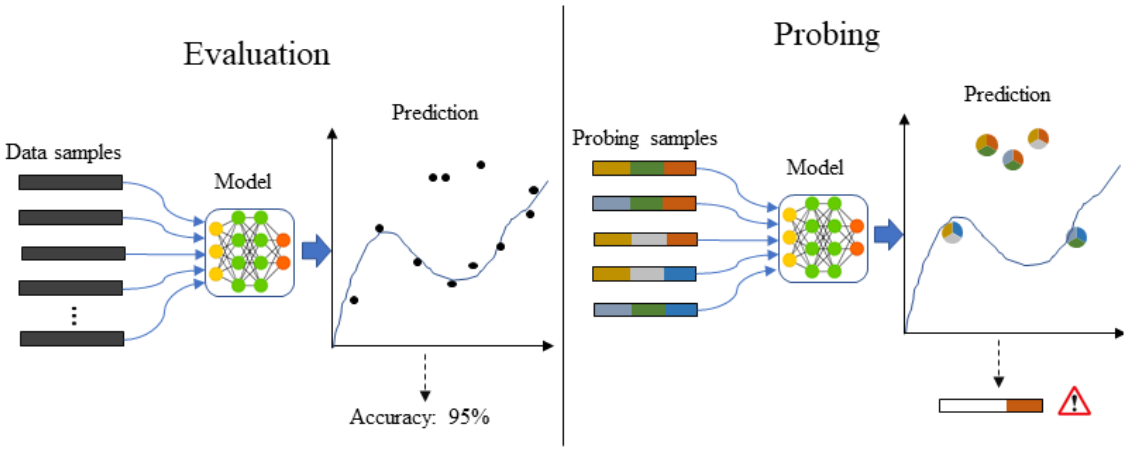


Figure 1: Model evaluation and model probing with controlled data characteristics.

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*.

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

| 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 |
| | | | ⋮ | |

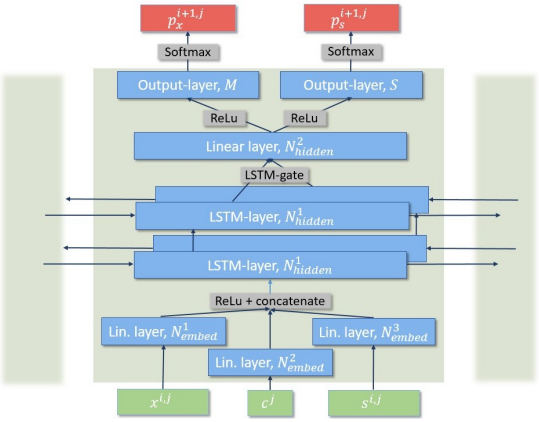


Figure 2: Architectures of the model and corresponding traffic sequence of XSS-attack.

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.

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

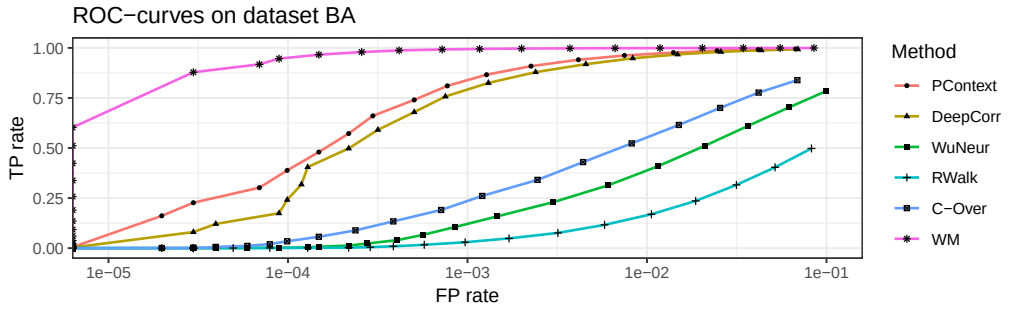


Figure 3: Evaluation results on the generated data.

extensive evaluation of their performance under various circumstances. The corresponding paper was submitted, accepted and published at the *Network and Systems Security* conference 2020.

# 2 Future plans

## 2.1 Challenges with publishing in Network Anomaly Detection and ACSAC conference

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 realisation was reflected during our unsuccessful attempts to publish the contextual network flow model described in Section 1.2 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. Prior attempts publication attempts by my supervisor and collaborating researchers yielded similar responses.

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. There seems to be a general scepticism of the detection capabilities and improvements that general-purpose anomaly-detection methods can provide in real-world scenarios due to high false-positive rates, in particular for network traffic. In the context of the proposed project, this raises doubt if research question 4 can be answered adequately for a PhD-project concerning general models of network behaviour. This also made us re-evaluate the scope of the project discussed in Section 1.4.

General advice were to identify very specific applications for which taylored machine learning methods have the potential to yield good results. This is in line with the findings of Sommer and Paxson [13] that anomaly detection methods should be "keeping the scope narrow" and providing an answer for "why the particular choice promises to perform well in the intended setting, considering domain-specific properties". "If one cannot make a solid argument for the relation of the features

to the attacks of interest, the resulting study risks foundering on serious flaws."

## 2.2 BT project

As part of my CASE PhD scholarship, I am spending 12 weeks in total with my industrial sponsor at BT Labs Adastral Park in the context of a student placement. While there, I am supposed to conduct research that is both relevant to my PhD-topic and to the operations of BT. Choosing a suitable topic is therefore not completely straightforward and could mean that the research conducted during the placement does not align completely with my PhD-topic.

In August and September 2019, I spent the first six weeks at Adastral Park. Before starting this stint, I met with my industrial supervisors to define a suitable project to work on. During this meeting, we agreed that the problem of detecting stepping stones was a suitable topic, as it was identified as a pressing issue for BT by their operational team, and could potentially benefit from from the developed DetGen framework described in Section 1.3 and the traffic sequence embedding techniques described in Section 1.4.

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 3.5). 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.

## 2.3 Submission of Stepping stone project and negotiation phase model

With work being almost finished for the stepping stone project and the negotiation phase model also being well on the way, I am planning to write each project up into a paper for publication. Since the scope for each project was scaled back a bit, I am intending to submit them to smaller venues with a higher acceptance rate, and do not intend to spent more than three months completing both. A suitable venue for both might be the 2020 Conference on Machine Learning for Cyber Security.

## 2.4 QUIC anomaly detection

An advice I received at the ACSAC conference from Guofei Gu, the Program Chair, was to identify very specific and novel applications of machine learning methods

to intrusion detection in order to produce publications that could be accepted at renowned venues. One such applications could be to the new transport layer QUIC. 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.

In total, I would assign about six months work in this project to conclude.

One difficulty in this project is that there do not exist many identified vulnerabilities in the QUIC protocol yet, of which none are known to have been used in a successful attack. Therefore, it might be difficult to get enough malicious data for result validation. At this stage of my PhD, beginning a project that is not certain to be successful bears significant risk. I am therefore currently spending a small amount of time reaching out to experts to verify that data for model validation is available and that the scope of this project is of relevance before fully committing to it.

## 2.5 Service relay detection

As mentioned in Section 1.3, disussion with a security expert at BT Labs indicated that the detection of simple proxy servers that relay protected services to third parties from the perspective of an ISP is a relevant and promising research project. 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.

Initial investigations that I conducted showed that there are no publications yet that address this problem in any way, which indicates the potential for novel contributions. Since there exists no description or analysis of such proxies at all, this however also means that we are reliant on BT for knowledge and guidance on the problem definition and proxy operation mode. Also since no public dataset exists, we would have to implement and generate our own dataset and feedback with BT to check if it resembles the behaviour of such proxies. At the moment it seems that a suitable start would be to focus on video relay. Necessary steps for this project are:

- Implement a docker traffic generation scenario to generate realistic data, and parse them in a sampled manner.

- start designing and implementing detection methods. As there are no evasion tactics involved, simple cumulative sum or moving window statistics with a p-value threshold seem like a good start.

- Test these methods on the data with a suitable background of independent connections as well as benign proxies. Obtaining simple background data is not challenging, however we will have to discuss how and to what extend benign proxies should be included in the background data.

- Evaluate and adjust/improve existing methods.

As we are reliant on the cooperation with BT Labs, we are not completely certain yet if this project will go ahead yet. Currently, they are retrieving data of service relay examples. We will have a further discussion with them later in April, which will bring more clarity.

## 2.6 Extension of Docker framework

Due to very positive feedback at the workshop as well as encouragements and suggestions to extend the framework, Robert Flood and I are working further to extend the current traffic generation framework. The overall goal is to provide a tool that enables researchers to generate large data quantities of specific services of interest in arbitrary network configurations, complemented by multi-step attack scenarios in differing executions. This would be of specific interests to researchers who apply machine-learning techniques to specific network applications for which there is not sufficient realistic public data available.

Particular goals are:

1. Extend the framework to create datasets that resemble full fledged computer network with variable topology. For this, several subtasks need to be completed:

(a) Embed all scenarios in a *Mininet* framework in order to allow the fast inclusion of switches, routers, etc.

(b) Build a mechanism that can generate variable or random network topologies.

(c) Create a launch mechanism that starts and executes different traffic scenarios for a given network topology at times and locations drawn from appropriate distributions that resemble empirical network behaviour.

2. Include the collection of application logs and system call logs for each container. These will then be matched with the corresponding traffic captures and receive the same ground truth labels.

3. Generate different multi-step attack executions on a set of network topologies to capture the similarities and disimilarities between different attack techniques and tactics.

An advantage of this project is the pre-existing DetGen body, that is being extended here. This means that the timescale of the project depends on how many extensions should be implemented. It should be relatively easy to finish the project quikcly at any given time after about two months of work while having sufficient results to publish.

# 3 Thesis plan

In the light of the problems described in Section 3.1 that we encountered in the field of anomaly-based intrusion detection, I believe that the scope of this PhD-project has to be altered slightly. As there will not be enough material directly concerning anomaly detection, the formulation of building anomaly-detection models describing software behaviour should be relaxed to general applications of ML (not just anomaly detection) in specific areas related to software defined behaviour. The overall unifying theme of this PhD so far has been centered around small-scale traffic structures generated by software-defined computer interactions and applications of ML language models to it, in other words **Modelling computer interactions as a language**. This could be based of the reoccurring use of traffic (and potentially system/program logs) generation for machine learning. The generated traffic could then be verified as valuable by the implemented intrusion detection applications.

All applications implemented so far except for the flow-level model are driven (in part) by our traffic generation:

- Flow-level LSTM model
- stepping stone detection
- traffic relay
- LSTM encoder
- QUIC anomaly detection

Overall, all these applications except for the traffic relay detection are concerned with software-defined fine-grained structures. Below, I outline the different chapters that I believe should be included in my thesis:

1. Introduction and related work

   This chapter could largely draw from the existing introduction in the research proposal as well as the extensive literature survey I conducted. Updates on the scope as well as recent developments in related work would have to be added.

2. Background: Anomaly detection and challenges in security

   This chapter outlines the motivation to use anomaly-based detection models instead of classification-based ones. It furthermore highlights why anomaly detection is more difficult in security than in other areas, and what currently prevents it from being deployed outside of academia.

3. Background: Language models and their application to security

   This chapter briefly outlines different techniques used in language modelling, and how they can be used to build data representation useful for anomaly-detection. It also discusses previous applications of language models in security.

4. Data sources, datasets, and data generation with realistic small scale structures

   This chapter will draw largely on the work conducted for the DetGen traffic generation framework.

5. Computer communication structures and the effect of malicious behaviour on them

   This chapter will describe the different structures that the different layers of software-defined communication can have on the collected data. These descriptions and their corroboration can be taken from work in each of the projects described above. Furthermore, this chapter will describe how attacks can alter these structures due to the distinct approach they are taking, and why it is worth building models that focus on small-scale traffic structures.

6. Traffic as a language and detection models

   This chapter will describe how network traffic can be described by a language model, and how to construct anomaly-based models from it. The chapter will then proceed to describe the different approaches I worked on and their respective results:

   (a) Flow-based modelling
   (b) Connection setup model
   (c) Traffic relating to process initialisations in the QUIC protocol
   (d) Detecting similarity in computer connections

7. Conclusions

## 3.1 Thesis completion plan

1. month
    - Finish stepping stone project
    - Inquire about the prospect of QUIC project
    - Discuss with BT the boundary conditions and data sources for a service relay problem
2. month
    - Work on connection setup project
    - Start producing paper from connection setup project
    - Finish inquiry about the prospect of QUIC project
3. month
    - Finish connection setup project
    - Start developing anomaly-detection model for QUIC project
4. month
    - Continue development of model for QUIC project
    - Work on dataset for QUIC project
5. month
    - Work on dataset for QUIC project
    - Gather data representing malicious QUIC traffic
6. month
    - Evaluate and tune results for QUIC project
    - Start producing paper for QUIC project
7. month
    - Finish paper for QUIC project
    - Work on data generation for service relay project
8. month
    - Begin 2. stint at BT
    - Work on model for service relay project
9. month
    - Finish 2. stint at BT
    - Evaluate model for service relay project
10. month

- Discuss further steps in service relay project, potentially produce paper
- Extend DetGen framework

11. month

    - Extend DetGen framework

12. month

    - Extend DetGen framework
    - Produce paper for DetGen framework

**Thesis write-up**

13. month

    - Revise literature review
    - Complete chapter 1 and 2
    - Compile a map of different language models for chapter 3

14. month

    - Complete chapter 3
    - Assess the requirements for chapter 4 and compare to results of DetGen project
    - Start working on chapter 4

15. month

    - Complete chapter 4
    - Compile a map of different traffic structures observed in each project
    - Compile a map of different attacks and their effect on respective structures
    - Start chapter 5

16. month

    - Complete chapter 5
    - Assess the results of each project, and decide order and weight of each project on chapter 6
    - Decide if anything is needed as input for chapter 6
    - Start chapter 6

17. month

    - Complete chapter 6

# References


[1] S. Badiger, S. Baheti, and Y. Simmhan. Violet: A large-scale virtual environment for internet of things. In *European Conference on Parallel Processing*, pages 309–324. Springer, 2018.

[2] L. Bernaille, R. Teixeira, I. Akodkenou, A. Soule, and K. Salamatian. Traffic classification on the fly. *ACM SIGCOMM Computer Communication Review*, 36(2):23–26, 2006.

[3] M. Crotti, M. Dusi, F. Gringoli, and L. Salgarelli. Traffic classification through simple statistical fingerprinting. *ACM SIGCOMM Computer Communication Review*, 37(1):5–16, 2007.

[4] J. Crussell, J. Erickson, D. Fritz, and J. Floren. minimega v. 3.0. Technical report, Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), 2015.

[5] M. Du, F. Li, G. Zheng, and V. Srikumar. Deeplog: Anomaly detection and diagnosis from system logs through deep learning. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1285–1298. ACM, 2017.

[6] R. Fujdiak, V. Uher, P. Mlynek, P. Blazek, J. Slacik, V. Sedlacek, J. Misurec, M. Volkova, and P. Chmelar. Ip traffic generator using container virtualization technology. In *2018 10th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pages 1–6. IEEE, 2018.

[7] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai. Kitsune: an ensemble of autoencoders for online network intrusion detection. *arXiv preprint arXiv:1802.09089*, 2018.

[8] M. Nasr, A. Bahramali, and A. Houmansadr. Deepcorr: Strong flow correlation attacks on tor using deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 1962–1976, 2018.

[9] B. J. Radford, L. M. Apolonio, A. J. Trias, and J. A. Simpson. Network traffic anomaly detection using recurrent neural networks. *arXiv preprint arXiv:1803.10769*, 2018.

[10] S. Roshan, Y. Miche, A. Akusok, and A. Lendasse. Adaptive and online network intrusion detection system using clustering and extreme learning machines. *Journal of the Franklin Institute*, 355(4):1752–1779, 2018.

[11] I. Sharafaldin, A. Gharib, A. H. Lashkari, and A. A. Ghorbani. Towards a reliable intrusion detection benchmark dataset. *Software Networking*, 2018(1):177–200, 2018.

[12] Y. Shen, E. Mariconti, P. A. Vervier, and G. Stringhini. Tiresias: Predicting security events through deep learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 592–605, 2018.

[13] 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.

[14] K. Wu, Z. Chen, and W. Li. A novel intrusion detection model for a massive network using convolutional neural networks. *IEEE Access*, 6:50850–50859, 2018.