Securing Future Networked Infrastructures through Dynamic Normal Behaviour Profiling

Kyle A. Simpson
2029567s@student.gla.ac.uk

January 23, 2017

Contents

1 Introduction

Regular attacks on online services are becoming an ever-present reality. Hackers seek to breach the security of organisations to compromise users and gain confidential information, or deny others access to internet shops and services to inflict monetary harm. System administrators and security researchers are engaged in an unending game of cat-and-mouse with these adversaries, and can often only react to attacks and security holes after the fact. More proactive, automated methods are needed to identify potentially harmful or disruptive traffic as it presents itself. Abnormal traffic may well have machine-detectable features which can be learned and leveraged to differentiate it from normal network activity. Denial-of-service attacks and application-level exploits designed to consume resources might have a consistent profile or content which correlates heavily with execution cost, covert channels used for data exfiltration (such as Loki[1]) may cause excess traffic over unusual protocols such as ICMP, and any shellcode payload bundled into an attack packet will appear distinctly unlike regular textual requests. Crucially, all of such features are deviations from what might be considered *normal* traffic behaviour.

Machine learning-based approaches such as neural networks, support vector machines [5], naïve Bayes classifiers and principal components analysis have been used within intrusion detection systems to attempt to identify anomalous traffic flows and outlying packets. However, these approaches favour offline training against established (potentially pre-labelled) data sets; in practice, features and characteristics of inbound network traffic will vary over time, mandating expensive re-training and management of training data. Real-time response and preemption of threats as they arrive and evolve demands investigation of more adaptive learning strategies, particularly those able to act on incomplete data sets collected over short time-scales.

The most common solution in use is to install expensive bespoke monitoring boxes at network ingress points, each capable of performing deep packet analysis and filtering of traffic intended to disrupt service. However, these demand that *all* incoming traffic pass through such middleware boxes, which can make it difficult to scale up or reconfigure a network. Recent advances such as Software-Defined Networks (SDN) [6] and Network Function Virtualisation (NFV) [7] enable dynamic configuration and routing of flows throughout a network without physical layout or wiring changes, and allow

[1] http://phrack.org/issues/49/6.html

installation of new monitoring and analysis tools throughout on standard x86 hardware. This new paradigm presents a potentially feasible platform for distributed machine analysis and handling of anomalous traffic without any of the standard reconfiguration times or costs, and should even allow dynamic relocation of monitoring components on-the-fly.

2 Background

The concept of *software defined networking* (SDN) was first introduced by McKeown et al. as *OpenFlow*, a programmable interface for switches and routers designed for researchers working with new protocols [6]. OpenFlow-enabled switches coordinate securely (i.e. over TLS) with one or more *controller* machines within a network to build flow tables and obtain routing rules for traffic—enabling packets to be dropped, redirected or handled by hardware routing rules where needed.

This paradigm goes hand in hand with *network function virtualisation* (NFV) [7]. NFV attempts to tackle the problems of space use, power consumption and cost associated with specialist hardware installation and upgrading within traditional networks. Rather than have functionality like industry-grade firewalls, network address translation or statistical analysis be performed by dedicated (and potentially proprietary) hardware, NFV focuses on providing these features dynamically through the use of lightweight virtual machines and containers on commodity computing hardware (i.e. x86 Unix servers).

Ali et al. discuss the merits of a combination of these advances for adapting to (and resisting) *distributed denial-of-service* (DDoS) attacks in cloud infrastructures, where a controller node remains uses measurement functions distributed across the network [1]. Their approach places functions for DDoS monitoring and remediation in different locations across the network, testing against modern attacks such as DDoS amplification attacks—they are able to show that a distributed approach can achieve better resilience than a middleware box at each exposed service.

Analysis of traffic flows and packet content will have to be performed very quickly if service is to be kept as close to line-rate as possible. While many complex machine learning approaches such as deep learning and convolutional neural networks are seeing applications in diverse areas from data analysis to image classification [4], many of these models require massive sets of (typically labelled) data and exhibit large training and execution times. Furthermore, these can often mandate the installation of powerful many-core graphics processing units to operate efficiently.

Clustering approaches like *k-means clustering* [3] can allow for relatively performant classification, typically by constructing "mixture-of-Gaussians" models to describe clusters of data. Unsupervised learning from data often requires an estimate of *how many* clusters are expected in the data to be effective. Techniques such as the Dirichlet process [2] extend the traditional notion of clustering by producing a probability distribution of probability distributions—this can allow for more flexible cluster detection in an unsupervised environment by allowing an estimation of a model's likelihood of accuracy.

3 Hypothesis and Objectives

Automated, unsupervised detection of abnormalities such as those described in Section 1 may well be an essential part of defence and threat mitigation for internet services in future. I hypothesise that cutting-edge machine learning techniques will be able to spot these deviations from the norm, and that monitoring locations distributed across the network will be able to communicate new findings and classifications with one another to throttle or block attack traffic. I aim to support and challenge this idea by investigating:

1. Classification and clustering of flows and packet content to detect abnormal behaviour—testing against the inclusion of arbitrary shellcode in packets and known denial-of-service attacks;
2. Coordination of knowledge and classification consensus between measurement functions at ingress points by various means, considering peer-to-peer models and centralised decision making at SDN control nodes;
3. Correlation of processing cost or resource usage with classification—this could be potentially useful when facing application-specific attacks such as XML-RPC abuse in the WordPress blogging software;

4. Traffic analysis at egress points—abnormal messages originating from *inside the network* corresponding to another incoming flow (a covert channel) might hint that an internal machine or service is compromised;
5. The performance impact of coordination across measurement points: how quickly can new routing rules be introduced to the network once an attack begins? What is the computational cost of the learning and classification process?

For this research to see real-world use, it is important that any functions introduced to the network do not overly impact the throughput or response times—even if such a system proves capable of attack detection and prevention, it will likely be unusable if regular service is overly affected. For these reasons it will be important to record and measure usage statistics, per-packet execution costs, service response times during and outside of attacks, and any other potentially useful metrics. It is hoped that any interim results such as system designs, quantitative performance, and optimisations relative to the state-of-the-art would be published in the relevant journals.

4 Work Plan

Funding for this PhD project is expected to last for 3½ years; this time must be allocated to account research, implementation, experimentation and production of a final thesis document. A rough breakdown is provided within Figure 1, and an elaboration on the main tasks is provided below.

Initial Research: The current background survey on both systems and any relevant machine learning approaches is incomplete—the start of the project will focus on further in-depth research to establish the current state-of-the-art.

Classifier Experimentation: This period of time will be spent applying the most relevant classification and learning techniques against existing datasets and new datasets constructed from known attacks. Results from this stage will provide a good basis for online classification, and will indicate which features are realistically detectable.

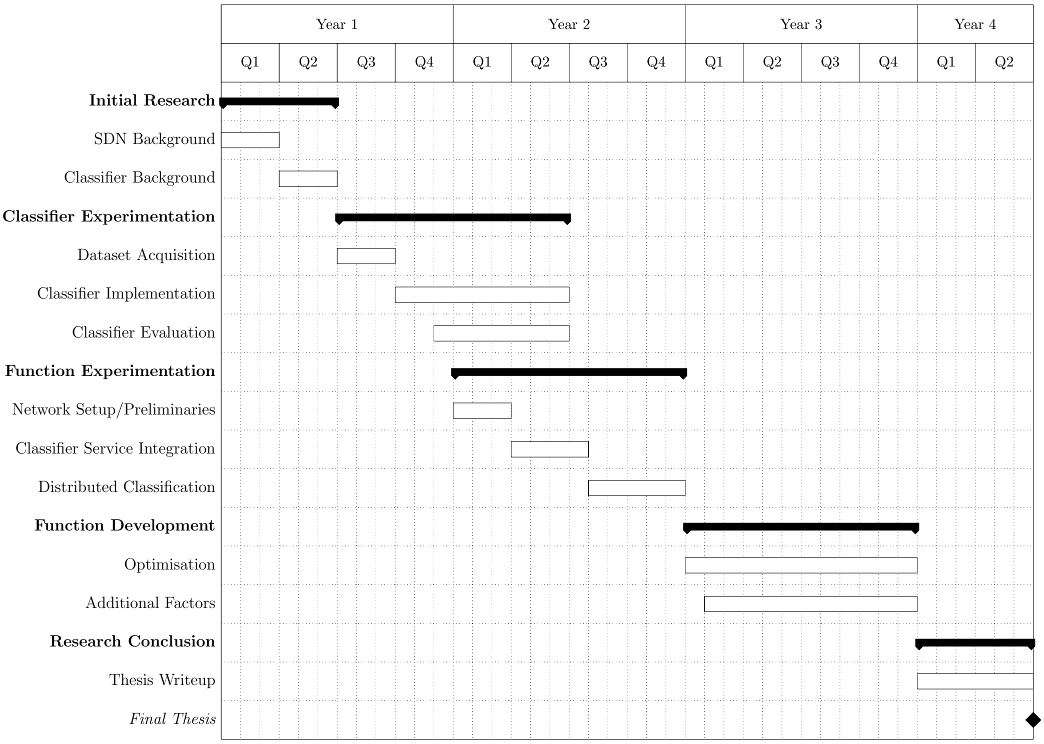

Figure 1: *Gantt chart of the proposed work plan*

Function Experimentation: Around this time I expect to be investigating initial implementations of any work on classification into test networked environments, working on using such services to filter and control anomalous traffic. I also aim to begin performance measurement and work on different communication strategies to aid classification between distributed monitoring points.

Function Development: By this point, it is hoped that I will be attempting to examine the effects of correlating egress traffic or execution and performance costs against inbound flows to determine if these can aid classification, among other potential factors.

Research Conclusion: This final period contains only the collation of all related work undertaken over the PhD's duration into a final thesis. While some work may occur throughout the prior 3 years, a large period of time is dedicated solely to the production of this document.

References

[1] Abeer Ali, Richard Cziva, Simon Jouët, and Dimitrios P. Pezaros. *SDNFV-based DDoS detection and remediation in multi-tenant, virtualized infrastructures.* Unpublished book chapter. 2017.

[2] Thomas S. Ferguson. "A Bayesian Analysis of Some Nonparametric Problems". In: *Ann. Statist.* 1.2 (Mar. 1973), pp. 209–230. DOI: 10.1214/aos/1176342360. URL: http://dx.doi.org/10.1214/aos/1176342360.

[3] Tapas Kanungo, David M. Mount, Nathan S. Netanyahu, Christine D. Piatko, Ruth Silverman, and Angela Y. Wu. "An Efficient k-Means Clustering Algorithm: Analysis and Implementation". In: *IEEE Trans. Pattern Anal. Mach. Intell.* 24.7 (2002), pp. 881–892. DOI: 10.1109/TPAMI.2002.1017616. URL: http://dx.doi.org/10.1109/TPAMI.2002.1017616.

[4] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. "ImageNet Classification with Deep Convolutional Neural Networks". In: *Advances in Neural Information Processing Systems 25: 26th Annual Conference on Neural Information Processing Systems 2012. Proceedings of a meeting held December 3-6, 2012, Lake Tahoe, Nevada, United States.* Ed. by Peter L. Bartlett, Fernando C. N. Pereira, Christopher J. C. Burges, Léon Bottou, and Kilian Q. Weinberger. 2012, pp. 1106–1114. URL: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.

[5] Aleksandar Lazarevic, Levent Ertöz, Vipin Kumar, Aysel Ozgur, and Jaideep Srivastava. "A Comparative Study of Anomaly Detection Schemes in Network Intrusion Detection". In: *Proceedings of the Third SIAM International Conference on Data Mining, San Francisco, CA, USA, May 1-3, 2003.* Ed. by Daniel Barbará and Chandrika Kamath. SIAM, 2003, pp. 25–36. ISBN: 978-0-89871-545-3. DOI: 10.1137/1.9781611972733.3. URL: http://dx.doi.org/10.1137/1.9781611972733.3.

[6] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru M. Parulkar, Larry L. Peterson, Jennifer Rexford, Scott Shenker, and Jonathan S. Turner. "OpenFlow: enabling innovation in campus networks". In: *Computer Communication Review* 38.2 (2008), pp. 69–74. DOI: 10.1145/1355734.1355746. URL: http://doi.acm.org/10.1145/1355734.1355746.

[7] *Network Functions Virtualisation*. 2012. URL: https://portal.etsi.org/NFV/NFV_White_Paper.pdf.