SDN-PANDA: **S**oftware-**D**efined **N**etwork **P**latform for **AN**omaly **D**etection **A**pplications

This paper appeared in the PhD Forum of ICNP 2015

Brian R. Granby
Department of Computer Science
Liverpool John Moores University
Liverpool, UK
Email: b.r.granby@2009.ljmu.ac.uk

Bob Askwith
Department of Computer Science
Liverpool John Moores University
Liverpool, UK
Email: r.j.askwith@ljmu.ac.uk

Angelos K. Marnerides
Department of Computer Science
Liverpool John Moores University
Liverpool, UK
Email: a.marnerides@ljmu.ac.uk

***Abstract*—The proliferation of cloud-enabled services has caused an exponential growth in the traffic volume of modern data centres (DCs). An important aspect for the optimal operation of DCs related to the real-time detection of anomalies within the measured traffic volume in order to identify possible threats or challenges that are caused by either malicious or legitimate intent. Therefore in this paper we present SDN-PANDA; a 'pluggable' software platform that aims to provide centralised administration and experimentation for anomaly detection techniques in Software Defined Data Centres (SDDCs). We present the overall design of the proposed scheme, and illustrate some initial results related to the performance of the current prototype with respect to scalability and basic traffic visualisation. We argue that the introduced platform may facilitate the underlying functional basis for a number of real-time anomaly detection applications and provide the necessary foundations for such algorithms to be easily deployed.**

Keywords*—*Software Defined Networking, Software Defined Data Centres, Network Function Virtualization, Anomaly Detection

I. Introduction

Advancements in Internet technologies have caused an exponential growth in the number of reported network attacks we face on a constant basis. Hence, *anomaly-based network intrusion detection* has evolved and constitutes a basic asset on addressing this issue. A number of existing approaches rely on either centralized or distributed measurement and monitoring sensors within a given backbone network in order to capture and further profile the traffic-wise behaviour within a given Anomaly Detection (AD) methodology [1].

Profiling incorporates a technique whereby a given system's operational behaviour is monitored over a period of time in order to define an accepted 'normal' operational profile. This profile is then used as a benchmark for defining reasonable measures of deviation for detecting anomalous events in network traffic. Recent research in *Software-Defined Networking* (SDN) [2], and its application towards Internet and data centre (DC) security has demonstrated exciting opportunities for exploring, and creating effective AD methods that can adequately confront the highly dynamic and cost-demanding network management requirements as derived by real-time security challenges [2].

In this paper we present our work to date in implementing SDN-PANDA, an SDN-enabled platform that will act as the underlying basis for configurable AD where related algorithms may be easily "plugged" and deployed. We describe the design and features our platform intends to provide, and present initial benchmarking results that mainly focus on the real-time operational overhead as well as on the traffic visualization enabled by SDN-PANDA. Similarly with the studies in [2] and [3] this paper describes an architectural SDN-enabled platform design whereby centralised monitoring is performed against aggregated flow statistics that make up the global network view. However, unlike [4] where the focus is directed towards small home and office networks, we focus our platforms application towards large scale virtualized and multi-tenant DC environments. Further, the SDN-PANDA platform differs from [3], as we place an emphasis on the advantages provided by coupling SDNs and Network Function Virtualization (NFV) [5] and their role on remediating anomalous events in our proposed platform.

Overall, we propose that by taking advantage of SDNs centralized control plane, and its global view of a given network, we are able to mitigate the challenges and limitations associated with current distributed monitoring models and techniques. Our proposed solution takes advantage of statistical data that is obtainable from OpenFlow-enabled switches [6], in the form of network flow table statistical traffic aggregate records. Our method implements Switch → Controller aggregate flow monitoring, and presents an interface whereby researchers of the AD network intrusion field are able to configure algorithm-specific preprocessing techniques for traffic feature selection. To prevent adaptability and scalability limitation we have intentionally avoided any dependence on controller libraries. We have developed our modules in python, and conduct experimentation with Floodlight [7] and POX [8] controllers. Hence, the herein presented research aims to contribute by providing a generic and module-independent experimental platform that can be adapted throughout the network security research community to experiment with, and further develop AD based security techniques for SDDCs and SDN-enabled networks.

The remainder of this paper is structured as follows: section II is dedicated at presenting the architectural design of the SDN-PANDA platform whereas section III presents some initial benchmarking results of the proposed platform that relate to traffic visualisation and real-time performance cost. Finally, section IV provides the future directions of this work and concludes this paper.

II. SDN-PANDA Design

SDN-PANDA consists of three controller-centric application modules -

1) Network Monitoring Module - (NMM)
2) Network Intrusion Detection Module - NID
3) Network Intrusion Response Module - (NIRM)

Figure 1 depicts a high level diagram of the proposed SDN-PANDA platform, and its placement within a generic data centre topology. Each module residing within the SDN-PANDA architecture allows for centralized administration via the controller management plane. Similar to [3], we design each of the modules as separate entities so that adaptable implementation of existing AD methods is possible whilst retaining configurable synergy to support each other. By aiming to provide this degree of flexibility, network operators are free to plug and further configure AD applications, network flow sampling requirements and rate intervals without deploying distributed monitoring agents.

A. Network Monitoring Module (NMM)

The NMM is responsible for data collection and preprocessing of switch aggregated flow statistics (e.g. packet match fields, packet counters, src/dst IP address, src/dst Ports, instructions and timeouts). The monitoring approach within the NMM differs to that in [3], as it does not rely on packet sampling techniques in order to construct statistical flow aggregates. Instead the NMM monitoring approach utilises the controller's REST API used for pushing switch commands, and also for delivering flow-table counter statistics to the controller.Thus we are able to obtain complete flow aggregate records without inflicting further processing demand on the controller. We achieve this by utilising the data used for monitoring the network state and provide network operators a centralised global view of the network.

B. Network Anomaly Detection Module (NID)

The NID is a pluggable interface for rapid deployment and configuration by network operators. Hence, it is intended to accommodate a broad variety of flow-level AD techniques available. By providing a high degree of customisation, operators will be able to deploy new state-of-the-art tools faster, and break away from the vendor lock in paradigm associated with traditional middle boxes as discussed in [5]. We propose a flexible interface in our NID so that others are free to further implement alternative intrusion detection methods such as packed based signature detection engines. By doing so, we intend to provide a foundational grounding for the rapid deployment of, and experimentation with both AD and packets based intrusion detection techniques in unity.

In order to address the potential for service degradation or controller resource exhaustion we align with the work in [9] whereby a flexible packet based sampling extension for OpenFlow is presented. We argue that this could prove beneficial when considering methods for obtaining packets for signature based analysis.

C. Network Intrusion Response Module (NIRM)

We design NIRM to function independently to NID so that response actions are defined based on standard policies, which are non-dependent on the AD method used. Anomalous instances detected by NID raise an alert message which is fed into the NIRM. Thus, the NIRM is responsible for autonomous execution of appropriate remediation policies in the event of a TRUE-POSITIVE intrusion alert.

Remediation policies must be accurately defined so that they only mitigate the identified attack or malicious intent in the DC. Ensuring any remediation action does not cause loss of service, or disruption to other users of the infrastructure. This makes applying the standard practice of dropping network packets, or isolating a physical host inapplicable in multi-tenant DCs.

By virtue of addressing the aforementioned issue, our work is inspired by the work in [10], whereby virtual machine instances are invoked on demand to provide scalable DC attack resilience. However, instead of invoking virtual machines (VMs) to saturate processing demand, the NIRM relies on NFV replication of legitimate tenant services into a different cluster of the DC infrastructure. Subsequently the NIRM aims to update all flow routes towards these replicated instances, leaving only the malicious or compromised tenants VM instance. By doing so it is therefore feasible to contain developing incidents and further provide vital input to the operators on investigating the source of the anomalous behaviour.

III. Results

In order to have a generic assessment of the performance of our proposed platform, we have implement our monitoring module on both the Floodlight and POX network controllers and are currently exploring avenues for inclusion of the OpenDaylight[12], and RYU[13] controllers.

To allow our platform to be utilised by future controllers we employed a standardised JSON interface and message response format compromising uniform value keys across all controllers. Our initial experimentation indicates that both the POX and FloodLight controller REST Response messages differ in formatting, as well as what can be utilised as attribute labels required by different AD applications. Furthermore, each of the controllers employ system-specific JSON request interfaces, which could cause configuration errors and limit the global intelligence available to SDN-PANDA.

When considering the application of our monitoring module towards multi-controller domains these variations cause scalability issues. We propose that by standardising the JSON aggregated flow record labelling we are able to adopt the included values as labelled attributes. Through this implementation, the SDN-PANDA is able to reduce the overall flow record pre-processing time. In standardising the above we are able to assume with confidence that flow records used as input data for training purposes are globally consistent, and globally accessible in scalable SDDCs.

In order to include traffic variation within our aggregate logs, we replay a network trace file containing 60,000 TCP/UDP flows into a simulated mininet network [14]. We configure a small SDDC tree topology containing 8 switches,

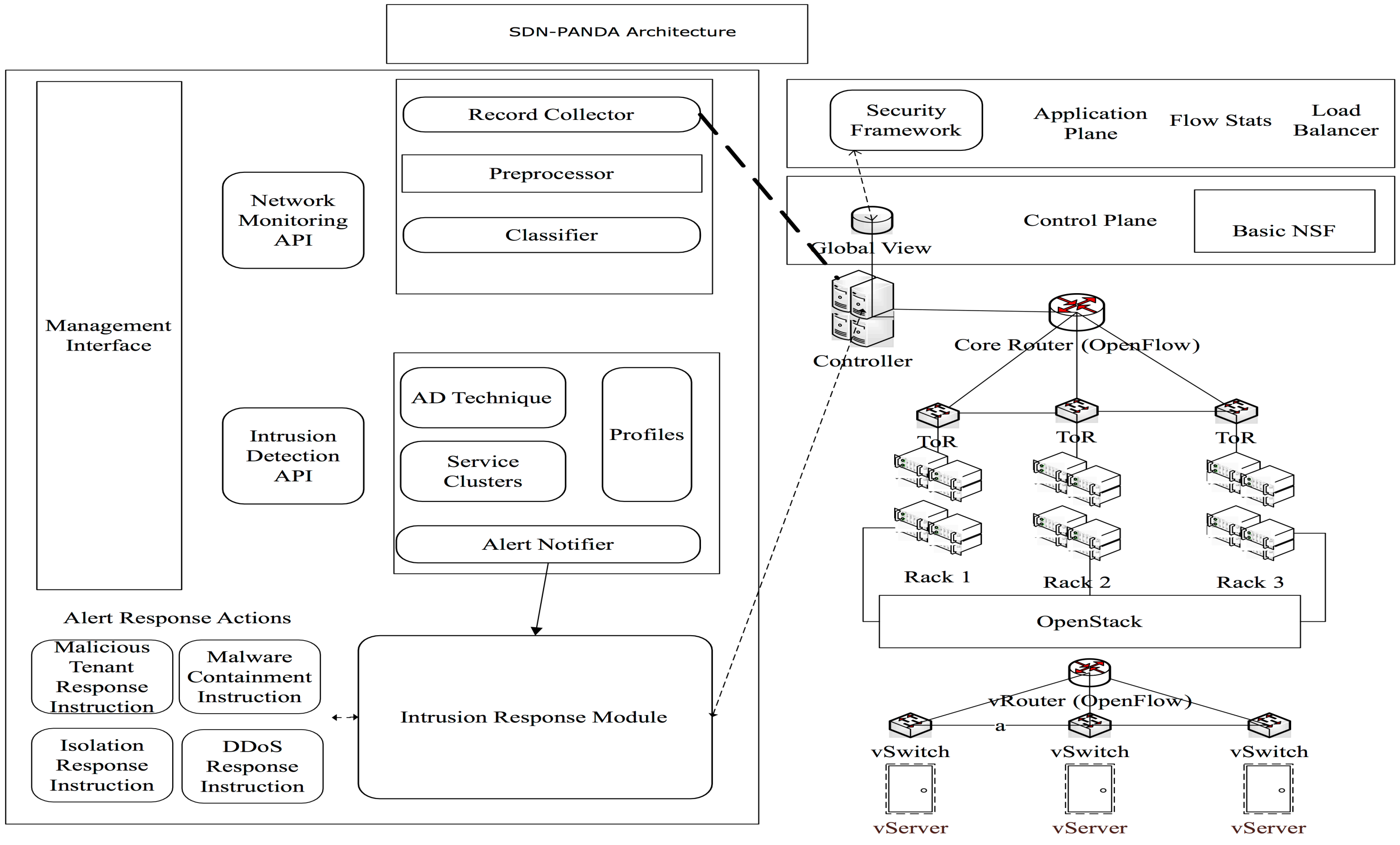

Fig. 1. SDN-PANDA Design Diagram

and one POX network controller where we deploy our platform modules. The resultant real-time monitoring aggregate produces $11,906$ records collected by our NMM throughout the duration of our experiment. Figure 2 provides the traffic visualisation enabled by SDN-PANDA and depicts the variation in network service traffic present within our experiment duration. It is evident, that the greatest amount of bytes was consumed by the SMB protocol followed by several FTP sessions.

For validation of the performance capabilities of our platform, we perform a benchmarking process by incrementing system load balancing. During the benchmarking analysis we execute a ping flood performance test against the SMB service depict by port 445 within figure 2. We initially conduct our experimental runs with one switch to test the real-time feasibility capabilities of our platform, and increment the number of switches as demand increases as presented in figure 3. Our results suggest that our application is capable of scaling to meet the demanding requirements of dynamic environments typically associated with DC networks. Our simulation attempts to replicate a scenario whereby an increasing service demand is met through the initiation of virtualised service instances and switches so that system performance remains unaffected. We demonstrate that utilising aggregated network flow statistics accessible via the JSON-REST API as input data for our platforms pluggable AD algorithm module we maintain scalable properties.

IV. Conclusions

The rapidly changing traffic demands on internet infrastructure require new network security tools and support. As

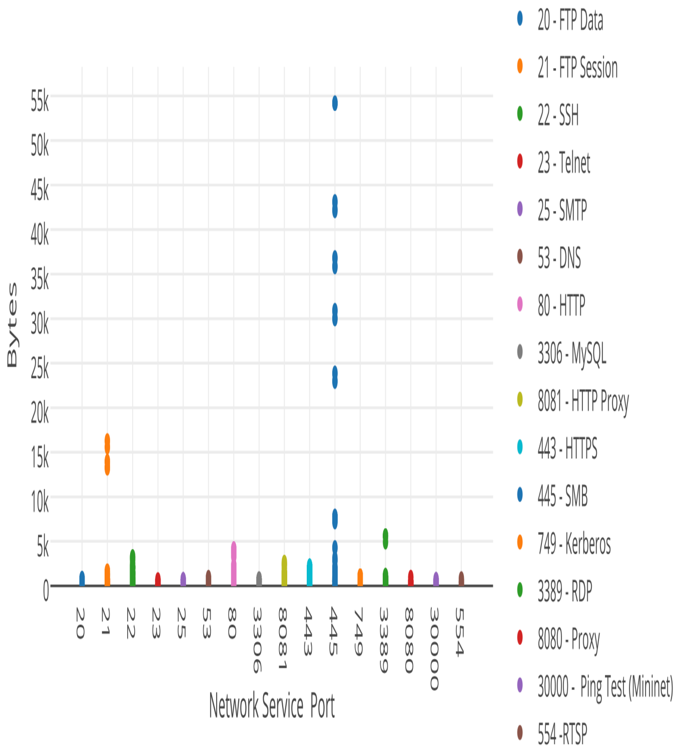

Fig. 2. Traffic volume distribution per protocol.

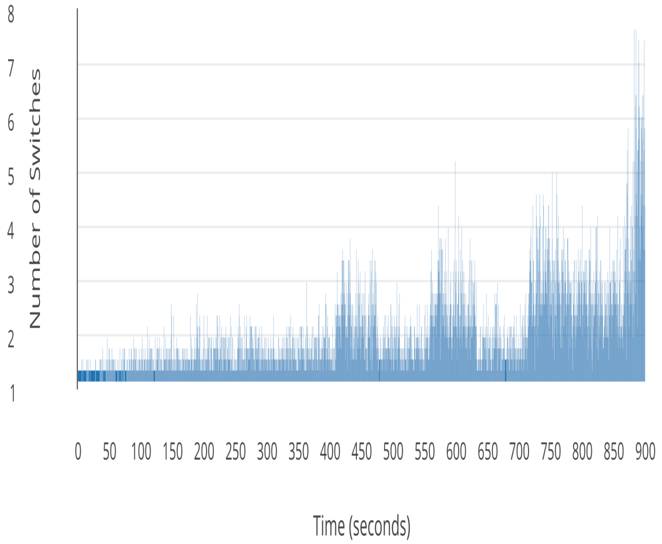

Fig. 3. Computational time cost vs. number of switches

more and more embrace the heterogeneous benefits of SDDC service hosting, traffic volume and targeted cyber attacks will increase in parallel. In this paper we have described SDN-PANDA – an SDN and NFV architectural platform for reaping the benefits of AD in a SDDC. The aim behind SDN-PANDA is to provide an experimental platform capable of being used as the underlying basis for the real-time deployment and operation of AD methods in SDDCs. Consequently, the real-time operation of AD methods via the SDN-PANDA platform would contribute towards the reactive identification of threats that could challenge the optimal operation of the SDDC. We describe that by utilising switch recorded flow statistics as input data towards our pluggable AD application module, network operators are able to specify appropriate remediation actions for detected malicious anomalies.

For future work we plan to compare SDN-PANDA against similar systems, e.g. - [3], [4],[10] and their suitability towards large-scale virtualized environments. Furthermore, we intend on providing an intuitive library of existing AD techniques for experimentation and implementation that meet scalability requirements of real world SDDCs.

References

[1] A., K., Marnerides, A., Schaeffer, and A., Mauthe, *Internet Anomaly Diagnosis: A Survey*, in Elsevier Computer Networks (COMNET), Vol. 73, ISSN: 1389-1286, pp. 224-243., Nov. 2014

[2] S. Scott-Hayward, G. O'Callaghan, and S. Sezer, SDN Security: A Survey, 2013 IEEE SDN Future. Networks Serv., pp. 1-7, Nov. 2013.

[3] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments, Comput. Networks, vol. 62, pp. 122-136, 2014.

[4] S. A. Mehdi, J. Khalid, and S. A. Khayam, Revisiting Traffic Anomaly Detection using Software Defined Networking, Entropy, vol. 6961, pp. 1-20, 2011.

[5] B. Han, V. Gopalakrishnan, L. Ji, and S. Lee, Network function virtualization: Challenges and opportunities for innovations, IEEE Commun. Mag., vol. 53, no. 2, pp. 90-97, 2015.

[6] A. Lara, A. Kolasani, and B. Ramamurthy, Network Innovation using OpenFlow: A Survey, IEEE Commun. Surv. TUTORIALS, vol. 16, no. 1, pp. 493-512, 2013.

[7] Floodlight OpenFlow Controller -Project Floodlight Available: http://www.projectfloodlight.org/floodlight/.

[8] POX. Available: http://www.noxrepo.org/pox/about-pox/.

[9] Shirali-Shahreza, S., & Ganjali, Y. (2014). Traffic statistics collection with FleXam. In Proceedings of the 2014 ACM conference on SIGCOMM - SIGCOMM 2014 (pp. 117-118). New York, 2014.

[10] R. Miao, M. Yu, and N. Jain, NIMBUS, ACM SIGCOMM Comput. Commun. Rev., vol. 44, pp. 121-122, 2014.

[11] H. Zhou, C. Wu, M. Jiang, B. Zhou, W. Gao, T. Pan, and M. Huang, Evolving defense mechanism for future network security, IEEE Commun. Mag., vol. 53, no. 4, pp. 45-51, 2015.

[12] OpenDaylight. Available: http://www.opendaylight.org/.

[13] Ryu SDN Framework. Available: https://osrg.github.io/ryu/.

[14] Mininet. Available: http://mininet.org/.

Brian R. Granby is a PhD student with the Department of Computer Science at Liverpool John Moores University, UK. He was awarded his B.Sc in Cyber Security from Liverpool John Moores University in 2014. Brian is currently in his first year of studies. Brianś research interests include software defined networking, autonomous network security, intrusion detection & prevention systems as well as emerging threats of ubiquitous network connectivity.

Bob Askwith is a Principal Lecturer in the Department of Computer Science at Liverpool John Moores University. He received a BSc in Software Engineering in 1996 and a PhD in Network Security in 2000, both from LJMU. He leads the development and delivery of Cyber Security programmes within the department. His research interests are focussed on the security of computer networks, especially mobile, wireless, and ad hoc. He has been involved in security projects funded by UK Government and EU.

Angelos K. Marnerides is Senior Lecturer (Assistant Professor) in the Department of Computer Science at Liverpool John Moores University, UK. Prior to that he was a Research Associate in the department of Computing & Communications at Lancaster University (2012-2014), a Postdoctoral Research Fellow in the Carnegie Mellon University - Portugal postdoctoral scheme at IT , University of Porto (2011-2012) and an Honorary Research Associate with the department of Electronic & Electrical Engineering at University College London (UCL) (2012-2013). He has been involved in a number of NSF/FCT, EU and UK EPSRC research projects and his research interests include network and cloud security, smart-grid resilience, anomaly detection and fault diagnosis. He obtained his M.Sc and PhD in Computer Science from Lancaster University in 2007 and 2011 respectively. He has been a member of the IEEE since 2007.