Testbed for Security Orchestration in a Network Function Virtualization Environment

Aapo Kalliola[*/***], Shankar Lal[*/***], Kimmo Ahola[**], Ian Oliver[*], Yoan Miche[*] and Silke Holtmanns[*]

[*]{aapo.kalliola, shankar.lal, ian.oliver, yoan.miche, silke.holtmanns}@nokia-bell-labs.com, Nokia Bell Labs
[**]kimmo.ahola@vtt.fi, VTT Technical Research Centre of Finland
[***]{aapo.kalliola, shankar.lal}@aalto.fi, Aalto University

***Abstract*—We present a testbed implementation for the development, evaluation and demonstration of security orchestration in a network function virtualization environment. As a specific scenario, we demonstrate how an intelligent response to DDoS and various other kinds of targeted attacks can be formulated such that these attacks and future variations can be mitigated. We utilise machine learning to characterise normal network traffic, attacks and responses, then utilise this information to orchestrate virtualized network functions around affected components to isolate these components and to capture, redirect and filter traffic (e.g. honeypotting) for additional analysis. This allows us to maintain a high level of network quality of service to given network functions and components despite adverse network conditions.**

I. Introduction

In recent years the provision of network services has swiftly moved from dedicated hardware installations to cloud platforms. While moving functionality formerly handled by dedicated hardware to the cloud has the obvious advantages of resource usage optimization as well as flexibility, it also widens the attack surface in many cases. Interfering with physical hardware used to often require physical access to the elements, while software versions of identical functionality can be more extensively accessed remotely, and often utilize heavily overlapping physical resources in e.g. processing and network bandwidth.

In order to enable this massive move of network functionality to the cloud while retaining sufficient security, several initiatives have been started. The European Telecommunication Standard Institute (ETSI) [1] has an operator-initiated Industry Specification Group (ISG) for Network Function Virtualization (ETSI NFV). NFV allows for the desired property of flexible software based connectivity. At the same time, the network functions being transformed to virtual network functions (VNF) increase the risks of them being compromised remotely either in targeted attacks or through resource exhaustion by e.g. distributed denial-of-service (DDoS) attacks.

In this paper, we address the need for a flexible testbed capable of running realistic attack and mitigation scenarios for evaluation of mitigation mechanisms in conjunction with security orchestration actions. More specifically, we present a DDoS attack scenario continued with targeted attacks, and show the actions taken by the mitigation/orchestration system to defend the system against these attacks.

In section II, we describe the nature and some details of the telco cloud. Sections III and IV describe the specific scenario we are deploying into the testbed with implementation specifics. Finally, we look at some of the limitations of the proposed solution and discuss possible future research directions along these lines in section VI and conclude in section VII.

II. Environment

The Telco Cloud is the telecommunication industry's take on cloud computing realised through the Network Function Virtualisation (NFV) concept [2]. The idea being that instead of hosting dedicated hardware for functions such as firewalls, base stations and the plethora of services that make up an operator's infrastructure (HLRs, VLRs, MMEs etc) are virtualized, bringing the benefits of cloud computing, e.g. in increased flexibility and efficiency.

Such components can be distributed as Virtual Network Functions (VNFs) which are then provisioned as one or more virtual machines to be deployed within the cloud. In practise a single VM may host multiple VNFs and similarly a single VNF may require a number of VMs to provide its functionality; a common simplification is to assume single VNF to single VM in most discussions.

In addition, the inclusion of network programmability through software defined networking (SDN) gives the ability to partition networks, services and dynamically route traffic based on certain conditions in real-time. Hence, NFV and SDN become the technologies which together constitute the Telco Cloud.

In terms of security (and privacy) provisioning, such a dynamic combination of service provisioning/chaining with a dynamic network topology means that reaction to attacks such as DDoS and others can be highly specific in themselves. In this paper we present one method of reacting to DDoS using these features, which can be further generalised to a number of other cases.

III. Attack scenario

Distributed denial-of-service (DDoS) attacks can take many forms. They may be volumetric in either packet count or bandwidth, or they may target vulnerable protocols or applications. In the scope of this paper we are considering the case of attacks

978-1-5386-3285-7/17/$31.00 ©2017 IEEE

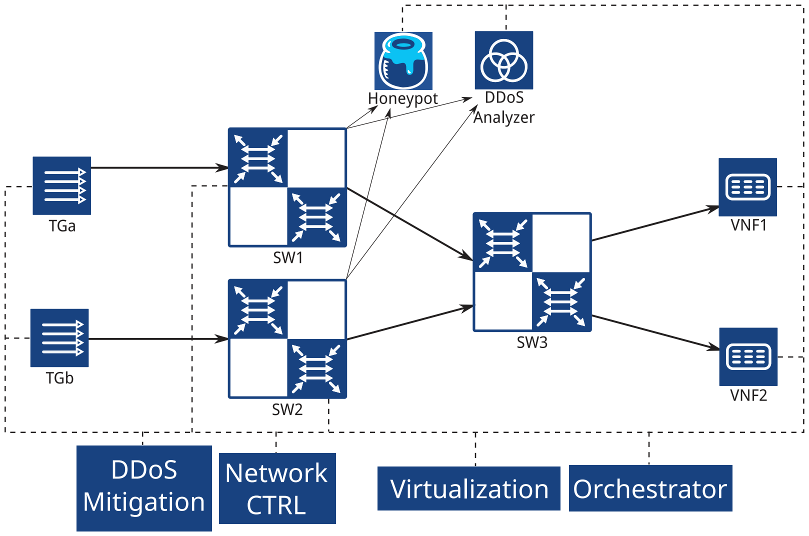

Fig. 1. Example network setup: Two traffic generators (TGa and TGb) send packets to the ingress switches SW1 and SW2. The traffic from both switches bottlenecks at switch SW3, which delivers the packets to both VNF1 and VNF2. Control plane information paths are abstracted as dashed lines, while solid lines denote data links.

impacting multiple systems in the Cloud environment due to shared bottleneck resources. Thus our focus is on volumetric attacks where communications link capacity is the bottleneck resource.

An interesting special case of volumetric DDoS is the case where the attack traffic origin is located in the same Cloud environment as the targeted host. This co-hosted DDoS attack case can potentially avoid some defences which are protecting the Cloud instances from external threats.

In real life DDoS attack traffic volumes greatly exceed the normal traffic. We recreate this in our experimental testbed by pre-recording a wide range of different extreme-bandwidth attack traffic traces and replaying these traces on top of normal traffic. The key factor in the attack traffic is the plausibility of the traffic from the view of the defence mechanism. In order to gain representative insight into the performance of the system we have experimented with attack traffic ranging from single-origin through DNS amplification and botnet-like sources to completely distributed spoofed traffic sources.

In addition to the volumetric attack our attack scenario also includes a targeted attack, i.e. a specific attack targeting a software vulnerability. While the range of different targeted attack varies widely from simple service vulnerability scanning to social engineering, our focus is strictly on network-based defence mechanisms. As such the targeted attack scenario which we aim to mitigate is the scanning and eventual exploitation attempt of vulnerable services in the Cloud.

As a tool for the targeted attack we utilize the Metasploit exploit development and penetration testing framework. The targeted attack is run against a service in the network during the DDoS attack, after a scan of the available vulnerable services.

Concerning the attack scenario in general it is important to note that for the deployment scenario we consider the Telco Cloud to be an important platform. One of main differences compared to generic Cloud environments is that the locations of VNFs in Telco Cloud may not be as flexible as in generic Cloud environments due to business and legal reasons. This mandates that the defence mechanism must be capable of protecting existing capacity instead of relying on extensive scale-out of the targeted services.

IV. Mitigation Testbed

Our mitigation architecture is designed for the Cloud environment, and uses the underlying SDN capable network elements for a view of traffic and for the attack mitigation. For a comprehensive view of the network traffic we utilize sFlow sampling on the relevant virtual network switches. The relevance of a switch is heavily network topology dependent, and thus sampling can be optimized with knowledge of the network graph. It is desirable to have an accurate view of network traffic while avoiding sampling same traffic flows multiple times in the network. The effect of attack mitigation is also achieved by using forwarding rules on the switches, which makes the mitigation nearly zero overhead from network element point of view.

The basic structure of the experimentation testbed is shown in Figure 1. This testbed is implemented in an OpenStack environment consisting of around 250 cores (Intel Xeon), 0.5Tb RAM over 4 dedicated compute nodes, 1 controller/compute node and 1 networking node running OpenStack Kilo on Ubuntu 14.04. Our switches are running Open vSwitch [3] and the Ryu [4] SDN controller is directed through a REST API by the mitigation and orchestration components.

Based on the information about historic traffic view an out-of-band machine learning component builds a model of the normal traffic profile. During an attack this model is compared to the current traffic profile and the traffic best fitting the

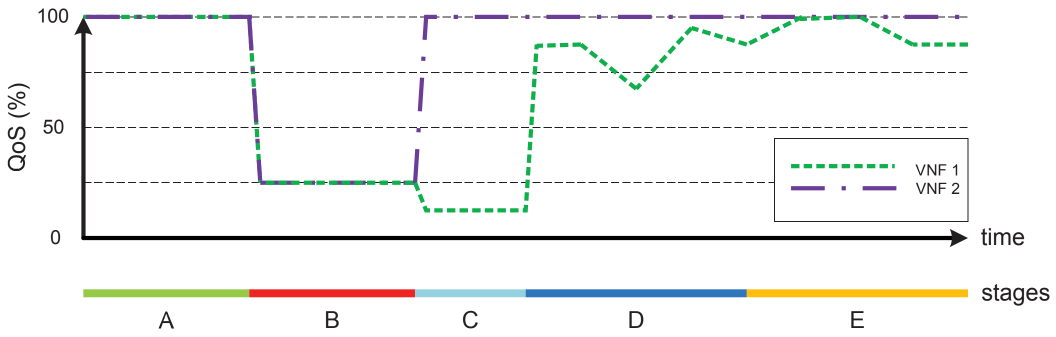

Fig. 2. Testbed example scenario timeline

normal traffic model is prioritized over traffic with worse fit with the model until available capacity is used up. In practise the machine learning component creates allow/deny flow rules, which are deployed to the switches through an application running on top of the SDN controller. These rules are updated dynamically based on current traffic in a control loop of a few seconds. The details of the machine learning and defence filter deployment have been covered previously by Kalliola *et. al.* [5].

Figure 2 shows an example of the stages of the defence mechanism. The length of the stages is scaled for clarity and does not proportionally reflect the delays related to activation of different stages.

In Stage A:

The system is in normal operation, i.e. both VNF 1 and VNF 2 are only receiving normal traffic and no part of the system is overloaded. During this time the *DDoS mitigation* mechanism receiving sFlow traffic samples from SW1 and SW2 is in learning mode and builds up a profile of normal traffic.

In Stage B:

The volumetric attack traffic targeting VNF 1 hits the system, heavily degrading the QoS to both VNFs. Both VNFs are impacted by the attack because they share a bottleneck in the network, namely SW3. This stage can also be considered to be the reaction delay of the defence system, and it should be as short as possible. In performance experimentation the reaction delay of our defence mechanism is in the order of a few seconds.

At the start of Stage C:

The *network isolation* defence mechanism is activated. This divides the network capacity leading to VNFs into N shares, with each share having a guaranteed amount of capacity available. This completely isolates VNF 2 from the detrimental effects of the DDoS attack targeting VNF 1. The QoS of VNF 1 is likely to suffer further due to the network isolation, since now the normal and attack traffic heading to VNF 1 is placed in a capacity share of less than full capacity.

At the start of Stage D:

The *DDoS mitigation* defence mechanism is activated. This happens somewhat after the activation of network isolation since the DDoS mitigation mechanism contains a more complex control loop and requires more extensive updates to the network flow rules. In practise the delay between network isolation and DDoS mitigation becoming active is again quite short, in the order of few seconds. The QoS of benign clients varies slightly over time during the attack mitigation depending on changes in the normal traffic and attack traffic patterns.

Finally, in stage E:

The system is hit with a targeted attack directed to VNF 2 in addition to the ongoing DDoS attack. In real-life scenarios this would indicate that the original attack may have been intended as a diversion. The targeted attack is mitigated in the system by having honeypot VNFs, in our case running Dionaea [6], laid over the real VNFs covering all services not explicitly provided for external use by the VNFs themselves. This is achieved in the packet forwarding by only forwarding packets destined to specific TCP/UDP ports to real services, and other packets to honeypots. External access to the services emulated by the honeypots results in an autonomous analysis of the access and alerts if the access is malicious.

Testing of the system indicates that the defence mechanism is capable of maintaining 70%-100% service quality in terms of packet drops for benign users even during an attack exceeding the bottleneck capacity by a factor of 20. While the defence mechanism is relatively insensitive to the attack traffic volume, the QoS does decrease somewhat when the attack traffic profile gets close to normal traffic profile. Nonetheless, the overall performance of the system even in the face of an overwhelming attack is good and the defence can perform attack mitigation autonomously.

V. Related Work

Software defined networking testbeds are commonly employed for testing with various levels of performance, ranging from fully emulated [7] to small-scale hardware/software hybrids [8] to fully fledged internationally distributed testbeds [9]. Within the scope of SDN, sFlow is recognized to be a useful sampling tool for gaining network traffic visibility [10].

A very large cloud-native testbed has also been explored by Mambretti et. al. [11]. In such a shared environment the resource control and distribution concerns are of paramount importance, and the federation of multiple large testbeds becomes a potential consideration.

The integration of SDN and NFV in Telco Cloud has previously been examined by Costa-Raquenna et. al. [12]. Their technology analysis and testbed work supports our view of the relevance of cloud environments for future telco networks.

In contrast to previous work, our paper presents a specific mid-sized cloud architecture and an example security evaluation scenario for flexible modeling and testing of in-cloud security functions.

VI. Discussion

The experimentation testbed outlined in this paper is a limited example of a possible network graph of an in-cloud VNF deployment. In our case the bottleneck could be avoided also e.g. by using a fat tree topology for interconnecting the switches to VNFs.

The fundamental issue of potential in-cloud network hotspots remains valid, nonetheless. Network graph optimization in an environment with heavily varying dynamic traffic loads, of which a DDoS attack is an extreme example of, is a very difficult problem. Ideally the very configuration of the network itself should be scalable and dynamic enough to prevent even attack-time hotspots from forming, but with current state of the art it is likely that infrastructure realities limit the flexibility of the dynamic network configuration. Thus, attack-time congestion of links is likely to occur also in real deployments.

In addition DDoS attacks are likely to affect networks across SDN domain boundaries, and therefore intelligent single-point link congestion control is a valuable addition even for very dynamic network configurations. Furthermore, separate DDoS mitigation mechanism can complement the dynamic network management if both are controlled by the same orchestration entity.

On a wider scope the analysis and detection of attacks is a complex closely related topic. While we have used reasonably simple analysis steps, it is possible to integrate extensive analysis solutions to our defensive approach. Outputs of such analysis can be further used for reporting and recording attacks for subsequent law enforcement response and use by other legal bodies. Specification of the related interfaces is an extensive topic on its own.

In general, the approach we have chosen combines centralized analysis and control with sampling and filtering functionalities distributed in the network. While this solution is directly applicable to current SDN architecture and devices, in the future an in-network intelligence and control may provide further scalability benefits through distributed analytics and centralized data aggregation.

VII. Conclusion

We have detailed and presented a flexible security orchestration testbed with an example scenario for DDoS and targeted attack mitigation. We have implemented the defence in OpenStack with SDN enabled network environment, and demonstrated its effectiveness with different types of attack traffic.

In addition to the direct technical contributions our approach indicates further possibilities for extending and improving the defensive capabilities of in-cloud networks. The operational insights gathered during the experimentation also indicate that visualization and output reporting interfaces are essential for effective information dissemination to the human operator, and this remains a topic for more extensive research.

Acknowledgment

This work was supported by the DIMECC Cyber Trust programme.

References

[1] "The european telecommunications standards institute, ETSI," http://www.etsi.org/.

[2] "Network function virtualisation: An introduction, benefits, enablers, challenges and call for action," ETSI Technical Report, October 2012.

[3] B. Pfaff, J. Pettit, T. Koponen, E. J. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar, K. Amidon, and M. Casado, "The design and implementation of open vswitch," in *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'15. Berkeley, CA, USA: USENIX Association, 2015, pp. 117–130. [Online]. Available: http://dl.acm.org/citation.cfm?id=2789770.2789779

[4] "Ryu SDN Framework," https://osrg.github.io/ryu/, accessed: 2017-06-30.

[5] A. Kalliola, K. Lee, H. Lee, and T. Aura, "Flooding DDoS mitigation and traffic management with software defined networking," in *Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on*, Oct 2015, pp. 248–254.

[6] "Dionaea honeypot," https://github.com/DinoTools/dionaea/, accessed: 2017-06-30.

[7] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, ser. Hotnets-IX. New York, NY, USA: ACM, 2010, pp. 19:1–19:6. [Online]. Available: http://doi.acm.org/10.1145/1868447.1868466

[8] H. Kim, J. Kim, and Y. B. Ko, "Developing a cost-effective openflow testbed for small-scale software defined networking," in *16th International Conference on Advanced Communication Technology*, Feb 2014, pp. 758–761.

[9] J. K. et. al., "OF@TEIN: An OpenFlow-enabled SDN Testbed over International SmartX Rack Sites," in *Proceedings of the Asia-Pacific Advanced Network*, 2013, pp. 17–22.

[10] S. U. Rehman, W. C. Song, and M. Kang, "Network-wide traffic visibility in OF@TEIN SDN testbed using sFlow," in *The 16th Asia-Pacific Network Operations and Management Symposium*, Sept 2014, pp. 1–6.

[11] J. Mambretti, J. Chen, and F. Yeh, "Next generation clouds, the chameleon cloud testbed, and software defined networking (sdn)," in *2015 International Conference on Cloud Computing Research and Innovation (ICCCRI)*, Oct 2015, pp. 73–79.

[12] J. Costa-Requena, J. L. Santos, V. F. Guasch, K. Ahokas, G. Premsankar, S. Luukkainen, O. L. Prez, M. U. Itzazelaia, I. Ahmad, M. Liyanage, M. Ylianttila, and E. M. de Oca, "Sdn and nfv integration in generalized mobile network architecture," in *2015 European Conference on Networks and Communications (EuCNC)*, June 2015, pp. 154–158.