Detection as a Service: An SDN Application

Mehrnoosh Monshizadeh*†, Vikramajeet Khatri*, Raimo Kantola†

*Nokia Bell Labs, Finland

† Department of Comnet, Aalto University, Espoo, Finland

mehrnoosh.monshizadeh@nokia-bell-labs.com*, mehrnoosh.monshizadeh@aalto.fi†, vikramajeet.khatri@nokia-bell-labs.com*, raimo.kantola@aalto.fi†

Abstract— **In a cloud computing environment, future networks will most probably utilize network functions virtualization (NFV) which is a network architecture concept that proposes virtualizing network node functions into "building blocks" or entities that may be operationally connected or linked together to provide services. However, applying these mechanisms brings security challenges. Due to the programmability of software defined networking (SDN), if attackers gain access to an SDN controller, then the whole network may be exploited by the attackers. The attackers may change forwarding paths and pass malicious traffic to infect the SDN enabled network. To detect the security attacks and malicious traffic early enough and to protect the network, centralized monitoring and intrusion detection system (IDS) monitoring may be used for enhancing SDN, NFV and OpenFlow security. If the network traffic is analysed and the anomalies are detected, the SDN controller may be used to block such traffic from passing through the network by flow control, i.e. forwarding paths in a switch. IDS and intrusion prevention system (IPS) may be deployed at the gateway node to detect a security intrusion. Thus, the data traffic originated from a subscriber passes through each network element until the traffic reaches the gateway node. Such traffic may attack the network elements and may also cause a denial of service (DoS) attack in the network. IDS devices are designed to handle network traffic in real time, yet the cost and high processing time is a challenge for handling the traffic load. Combining dynamicity and programmability of SDN together with traffic filtering of IDS, enables a scalable, redundant and reliable anomaly detection for mobile network operators. In this study, we propose an architecture that combines IDS with programmability features of SDN for detection and mitigation of malicious traffic. Mitigation will be performed by SDN controller using flow control techniques. The proposed architecture can be applied to an SDN enabled mobile network with two different approaches for improved performance in terms of computation power.**

Keywords— **Anomaly Detection, Cloud, Controller, DaaS, IDS, OpenFlow, SDN, Security**

I. INTRODUCTION

The mobile network is evolving and moving towards cloud with programmable and adaptive characteristics. In particular, it is assumed that 5G will follow the software defined networking (SDN) principles of separating the control and data planes as well as will use network function virtualization (NFV) for running network (control) functions on the cloud infrastructure. That brings additional security challenges and increases attack surface. Also, in the case of successful attacks on the centralized components, the potential impact is higher than in traditional networks and thus the security risks are heightened. In a cloud infrastructure, security mechanisms must address the new threats and ensure robustness of the centralized elements in the face of all security threats. This calls for urgent efforts to improve the individual security mechanisms, the processes how they are managed by the network administrators as well as for new approaches to network security. For example, we should study the ways of harnessing the power of the cloud to boost the network and end system security [1].

Due to programmability of SDN, if attackers gain access to SDN controller, the whole network can be exploited [2-3]. Attackers can change forwarding paths and pass malicious traffic to infect the SDN enabled network or compromise the confidentiality of communications. To detect these attacks and malicious traffic early enough, centralized monitoring and intrusion detection system (IDS) is recommended to be used for SDN and NFV [4].

IDS is still a standard defense solution for networks. IDS monitors network traffic in real time and compares the received packet patterns with known patterns to detect anomalies in network. Yet the cost and high processing time to handle traffic load is a challenge in IDS [5]. Therefore, we should study how by combining dynamicity and programmability of SDN with traffic filtering of IDS we can create a scalable, redundant and reliable anomaly detection system for mobile operators.

Traffic originated from mobile subscriber passes through all network elements till it reaches the gateway to the Internet. Traffic originated in the Internet traverses the mobile network elements in the opposite direction. Such traffic may attack network elements and can also cause a denial of service (DoS) attack on the network. Therefore, it is very important to stop the abnormal traffic to reach the network elements. To solve this problem, IDS is used to analyze the network traffic and detect the anomalies. The SDN controller can be used to isolate the infected users and elements. In this paper, we present a design and analysis of an IDS based mechanisms combined with SDN application to identify and prevent malicious traffic by blocking flows via programing SDN controlled switches. Moreover, we describe the actions that can be performed in the SDN application, if detection as a

service (DaaS) detects intrusions and conclude the implications on flows.

The rest of the paper is organized as follows. Section II briefly reviews related work for intrusion detection and prevention in SDN. In section III, we discuss SDN security challenges and the need for detection of threats and attacks and their mitigation in forthcoming 5G mobile networks. In section IV, we propose an architecture to detect and prevent attacks on SDN and NFV using flow control techniques. In our terminology flow refers to forwarding rule in OpenFlow enabled switches, so the analysis would be done on packet level. In addition, two approaches for the framework are discussed. Finally, the conclusion is presented in the last section.

II. Related Work

Han at el. [6] proposed an SDN based network security defense system for distributed DoS (DDoS). The scenario comprises of an SDN controller, an IDS decision making server and an IDS device. In the first phase IDS device samples and detects suspicious traffic. Upon detection of suspicious traffic, IDS device informs IDS server; and IDS server prepares action plan based on the type of anomaly. The action plan made by IDS server will be send to SDN controller for threat processing, which will also reduce the load on SDN controller for analysis. This study does not state that SDN controller will instruct the underlying switch to remove and drop malicious flows. The architecture also does not include any load balancer or clustering mechanism to reduce the load. The proposed system samples the packets for analysis, therefore it does not process each packet. Furthermore, this patent explains more about IDS defense system and considers different scenarios such as ICMP flooding, UDP flooding and explains working methodology in each scenario.

Luo [7] proposed a security platform on software level, which comprises of an application layer, a management layer and a data layer. The application layer comprises a network application, network services and a custom application interface. The management layer includes SDN controller, whereas data layer includes firewall, a web application protective system and IDS that detect unsafe traffic. SDN controller updates custom firewall security policy to block unauthorized access. However, the detection of malicious traffic in SDN enabled network and its mitigation i.e., removal of malicious flows has not been discussed in this study.

Jo et al. [8] proposed a network intrusion prevention system (IPS), comprising of an IDS module which analyses and detects threats in an SDN enabled network. Upon detection, IDS module updates IPS module with threats list; and IPS module takes necessary mitigation actions.

Puttaswamy et al. [9] discussed about flow deflection within SDN and data centers. The flow deflection is applied on computing resources including central processing unit (CPU) and ternary content addressable memory (TCAM). The resources are monitored and upon detection of overuse, flow deflection is enabled to direct flows from one network element to other network element and avoid failures in SDN. When resources are not overloading, the flows are deflected back to the first network element. Overall, the study aims to reduce load at SDN controller and computing resources to decide which network element (i.e., switch) flows should be forwarded. The proposed flow deflection mechanism does not perform any intrusion detection or clustering mechanisms but simply protects SDN from overloading by resource over-utilization attacks.

III. Literature Review

In large scale data center environment, hardware switches rarely handle end user traffic directly as the traffic is tunnelled via overlay tunnels (like VXLAN and MPLS over GRE/UDP) between virtual tunnel end-points (VTEPs). Hardware switches may have limited capabilities to inspect overlay payload and thus create the filtering rules on switches is not feasible. This is especially important in case the overlay carrier transports another overlay for example, GTP-U over VXLAN. Also, the scale of hardware switches doesn't support end user filtering since current switches support few tens of thousands of flow rules. In case the switch is processing traffic for SAEGW VNF, it would be subjected to millions of end user flows. The processing power needed for inspecting every single packet in software switches on compute hosts requires lots of CPU resources and amount of those resources may make the solution commercially too expensive. Therefore, improvements in hardware capabilities may change the commercial aspect on the long run. It should be noted that in some use cases cost of network security is not the primary concern, this would include networking between financial institutions or emergency services and thus these use cases deploy all techniques needed and bear the related costs.

In cloud computing technology, use of an SDN controller for the control of switch flow forwarding state is an unavoidable general feature of all SDN traffic control mechanisms. SDN has been designed to manage traffic in flexible manner within SDN network. Management in common language means forwarding traffic to proper endpoint to perform the load balancing and otherwise optimize use of available network resources on the way. By default, it is not used to mitigate malicious traffic and even if it would be, the SDN management system is missing the detection module that could identify malicious flows. While previous defense mechanisms are typically designed to stop malicious traffic at the border of the network like firewalls or border gateway protocols (BGPs). In our study, we tackle the forthcoming situation, where malicious traffic is already flowing inside the network, since the first suitable control point is in the middle of the network. Assuming our mechanism applies also to software based virtual switches residing in the hypervisors server and on compute hosts then solution is more feasible. A box called detection as a service (DaaS) detects the anomalies in the network traffic. Considering the amount of network traffic and the computation requirements, various DaaS units can be utilized for load balancing purposes. Our proposed mechanism detects anomalies in the traffic that passes through SDN network and

prevents malicious traffic from flowing into SDN network. Traditional IDS and IPS are often deployed at the gateway, and can detect intrusion. But still the traffic passes through the network if it is subscriber originated and is destined towards the Internet. With this mechanism, we aim to stop the malicious traffic from entering further and polluting the network elements.

Furthermore, with our approach, each packet would be analyzed rather than sampled traffic; which means traffic mirroring would be used and a copy of packet would be analyzed. Unless the traffic is detected as malicious by a DaaS node, the traffic will be flowing normally. As soon as it is detected as malicious, flow removal process will be initiated so such traffic will not be forwarded. The flow tables of switch combined with clustering will be used to identify which traffic belongs to which cluster and a decision will be made to forward certain categories of traffic to respective DaaS node for analysis. Any clustering algorithm can be used with unsupervised learning since we do not need to specify the number of clusters in advance. The DaaS node analyzes the traffic and concludes whether it is malicious or not. If it is malicious, then it will inform SDN application. SDN application acts as an orchestrator and gathers the results from multiple DaaS nodes, formulates and sends a flow to controller. The controller then instructs the switch to perform actions including removal, modification or installation of flow. Thus, such malicious traffic would be dropped and will not pass through the switches.

Our proposed architecture does not only protect against DoS or DDoS, but also against more targeted and less volumetric attacks such as exploit deployment and data in-or-ex-filtration. The fundamental difference between this architecture and similar earlier studies is that the prior mechanisms use only switch to or from controller interaction and only has access to the initiating packet of the flow, and use this as a basis for the block or pass decision. Such approach has several drawbacks, such as vulnerability to SYN flooding with fake source addresses, which forces the controller to process each incoming packet as a separate flow and results in controller DoS [10-11], and vulnerability to malicious traffic which is transmitted after a harmless-seeming flow initiation. In addition, previous studies don't focus on preventing malicious traffic from switch by flow control mechanisms (i.e., flow removal, modification and installation of flows to accommodate safe traffic). In contrast, our architecture does the load balancing of packets for scalability, and also the heavy processing of malicious traffic detection in an entity that is completely separated from the SDN controller. The proposed architecture is also capable of total analysis of the network traffic and not only the beginning of new traffic flow. In addition, our architecture is in no way tied to a specific clustering algorithm. The clustering algorithm and its parameters need to be selected in such a way that the memory and processing limitations of both the element performing the clustering and the element doing cluster-based traffic forwarding are not exceeded. This selection is an embodiment and implementation specific task. There is no dependency between clustering features and SDN controller in our architecture. In the beginning, we consider all the features that are related to both mobile network and OpenFlow so we don't miss any important feature for detection. After clustering and detection are done, we filter out the features that are applicable for OpenFlow in SDN environment. Such filtering of features is made by SDN application. SDN application after filtering features will formulate a flow that will be sent to the controller. Therefore, the features used by SDN application include source IPv4 & IPv6 address, destination IPv4 & IPv6 address, source MAC address, destination MAC address, source port number, destination port number, metadata of packet, VLAN id, IP explicit congestion notification (ECN), IP diffserv code point (DSCP), SCTP source and destination ports, ICMP type and code, ARP opcode, ARP source and target IPv4 address, ARP source and target MAC address, MPLS label, TCP flags and other fields as specified in OpenFlow Switch 1.5.1 specifications [12]. However, similar techniques exist for example for mitigation of DDoS attacks in form of BGP flow specification but this technique lacks the described detection mechanisms, it only addresses countermeasures. BGP flow specification as mentioned in RFC 5575 [13], RFC 7674 [14] can be seen as early flow-based SDN that targets a very specific use case [15].

Considering there is already means to deploy countermeasures, there probably exists means to detect malicious traffic directly on the mobile gateway and base stations as it would allow for pinpointing accuracy on the traffic of mobile subscriber. Applying detection mechanisms solely on switches means dealing more with aggregate flows, for example between gateway products and thus detection is either more demanding (looking up deeper in the packet required) or even impossible (traffic between gateway nodes may be encrypted like it is possible in case of SAEGW to and from eNB S1-U interface). Therefore, we couple our approach with the already mentioned BGP flow specification technique since we should not restrict ourselves to controlling only SDN aware switches but also edge routers. Especially in the case of DDoS, edge router play crucial role in mitigating attacks that originate from the Internet and possible towards user equipment (UE) in the mobile network. Malicious traffic detection can occur in the mobile network by SDN, but attack mitigation should take place as near the Internet edge as possible in order to avoid loading the core network with malicious traffic. Encryption of end user traffic can be used to circumvent detection. Compression of end user traffic may make the analysis too demanding [16] and thus may render the solution infeasible. Also, if end user uses the onion router (ToR) [17], the proposed architecture cannot analyze detailed flow information nor block individual flows as analyzer only sees single encrypted flow from the end user to the ToR relay. In general, type of encrypted or compressed flow is possible to deduce by analyzing beginning of the flow. Each network service tends to have a pattern specific to it that it follows when initializing the flow. These characteristic patterns are used, e.g., in dynamic experience management (DEM) [18]

functions to identify types of flows in order to be able to prioritize flows and to adjust the network parameters to support the best possible customer experience. In case of compressed flow contents, if content analysis is necessary, the system needs to be able to decompress flow contents starting from the beginning of the flow. Many of the most common compression methods introduce the used method in the beginning of the compressed data to facilitate decompression. Some of them also have similar characteristic patterns as mentioned above. This information can be used to perform decompression before actual analysis of the flow contents. It is possible to terminate the encryption of connections prior to the detection mechanism, allowing the mechanism to operate unhindered by the encryption. This is commonly done by proxies and traffic inspection systems by effectively performing a man-in-the-middle attack against the encrypted connection [19]. For example, in the context of secure sockets layer (SSL) connections by acquiring a trusted certificate for terminating the encryption for incoming traffic, decrypting and analyzing the traffic, and forming another encrypted connection with the destination through which the traffic is then sent [20-21]. Compression can be considered a subcase of encryption, where the similar approach of decryption and re-encryption can in theory be applied. In any case, the added complexity of dealing with encrypted communications is common to all network based traffic analysis [22-23].

IV. ARCHITECTURE

There are two approaches to implement our architecture; the differences between these approaches lie in the cluster determining method, which are explained in detail here.

In our architecture, the first (or any consequent) packet that arrives to switch would be mirrored into a clustering system and will be forwarded normally towards the destination (for the load balancing or other criteria). The mirrored packet will be forwarded for analysis to a corresponding DaaS node. The DaaS node will decide whether the traffic is malicious or normal. If DaaS node tags the traffic as malicious, the DaaS node will inform the SDN application running on top of SDN controller that the flow should be blocked. SDN application will direct SDN controller to perform the blocking. Then SDN controller will insert the blocking flow entry to switch and finally the switch will block the flow. The DaaS node will be provided with mirrored traffic. It can do any type of detection for that traffic that has been considered meaningful and are possible within given time and with available resources. Then as mentioned also earlier, the DaaS node will decides whether traffic is malicious or not and will inform appropriate SDN application to take actions. In a simple scenario, DaaS node could be even a rule based module without any capabilities for adaptation, further analysis or identification of unseen types of malicious traffic.

Analysis on traffic is not carried in line, but rather traffic mirroring is used. In approach 1, a single flow will forward a copy of incoming traffic to clustering unit. In approach 2, there will be a single flow for each cluster that will forward a copy of incoming traffic from subscriber to respective DaaS node. Unless the traffic will be detected as malicious by a DaaS node, the traffic will be flowing normally. As soon as it is detected as malicious, flow actions (i.e., removal, installation or modification) will be initiated and then such malicious traffic will not be forwarded.

A. Approach 1

The approach 1 and its flowchart can be seen in Figure 1 and Figure 2 respectively.

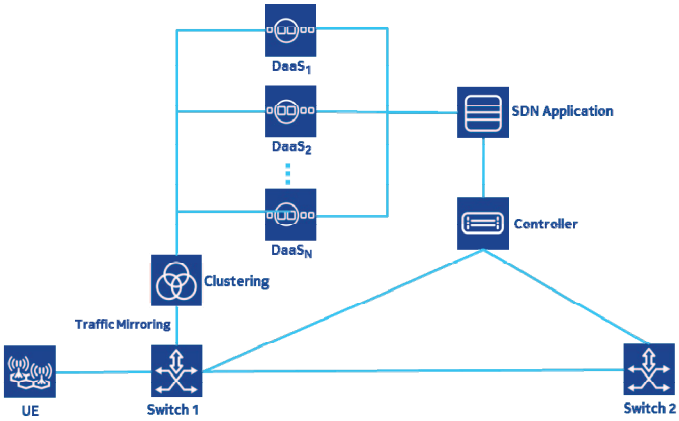

Figure 1. Approach 1 for DaaS in SDN

We consider a scenario where traffic originates from subscriber i.e., UE and is destined towards the Internet. There are $DaaS_N$ units that will analyze traffic and find anomalies and maliciousness in traffic. In this approach, a single flow will forward a copy of incoming traffic to clustering unit.

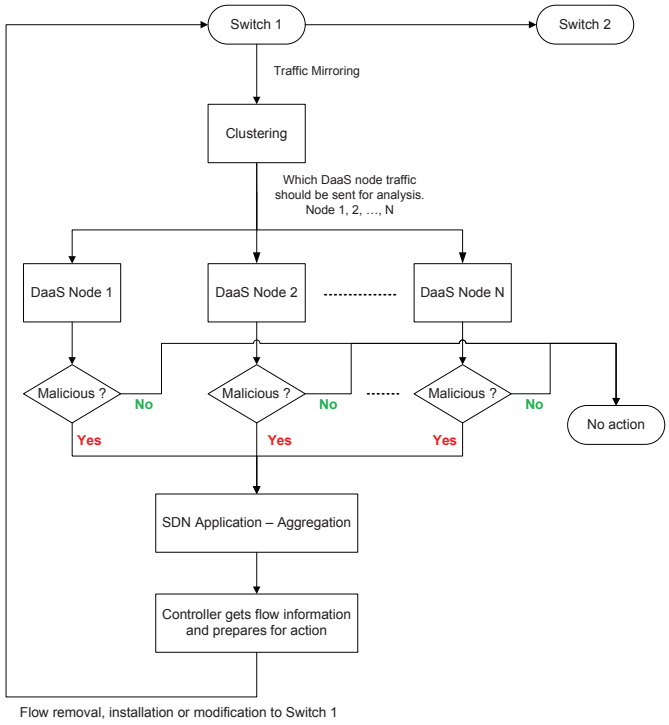

Figure 2. Flowchart for approach 1 for DaaS in SDN

The clustering algorithm will act as a load balancer or switching agent and decide which instance of DaaS the packet should be forwarded for inspection. It will read each packet and will place them into cluster x (x refers to any cluster that is initially programmed based on features of packet e.g., if packet comes from a specific IP address then it should be put into cluster 1 [24]). If the cluster x belongs to $DaaS_1$, then packet will be forwarded to $DaaS_1$ for inspection. Similarly, $DaaS_2$, $DaaS_3$ and so on $DaaS_N$. If malicious packets are detected, the DaaS instance will send the malicious packet information to SDN application that will correlate and remove any redundant or duplicate information; and send the instruction (flow information) to controller to take needed flow action. The controller instructs the switch to perform actions (flow removal, modification or installation) so that suspicious or potentially malicious traffic is not forwarded to the network elements.

B. Approach 2

We consider same scenario as considered in approach 1. The approach 2 and its flowchart can be seen in Figure 3 and Figure 4 respectively.

In this approach, there will be a single flow for each cluster that will forward a copy of incoming traffic from subscriber to respective DaaS node. Here, the behavior of the earlier mentioned approach 1 is distributed to multiple components for load balancing and latency minimization purposes. We call it as adaptive cluster creation which is discussed in next paragraph.

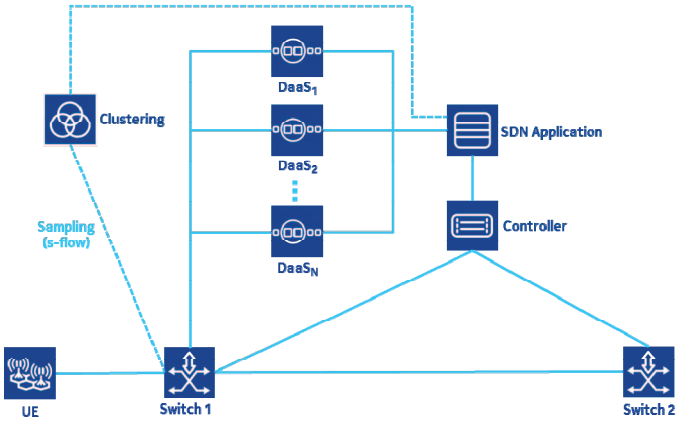

Figure 3. Approach 2 for DaaS in SDN

Initially clustering algorithm along with SDN application and controller takes care of forwarding incoming traffic to the respective DaaS node. A sample of packet (using s.flow) will be fed to clustering algorithm and the clustering algorithm will create even-sized clusters. Cluster information will be sent to SDN application. From cluster information, SDN application will formulate forwarding rules (flows) for different DaaS nodes and for load balancing. SDN application forwards flow(s) to controller and controller will deploy or install the respective flow(s) at the switch (switch 1 in Figure 3) so that all the traffic matching the flow will be forwarded to respective DaaS node. Eventually, traffic will be flowing to different DaaS nodes evenly. The traffic analysis and mitigation process will remain same as in approach 1 i.e., analyzing the traffic via DaaS node, sending its detection results to SDN application that will orchestrate and send action strategy to controller which eventually will stop malicious traffic from entering into network.

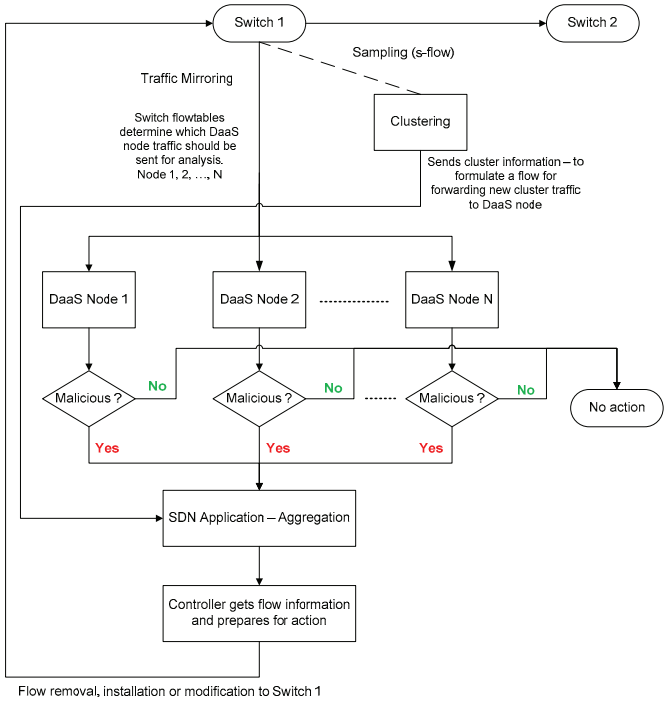

Figure 4. Flowchart for approach 2 for DaaS in SDN

V. CONCLUSION

Intrusion detection systems (IDSs) can be combined with programmability features of SDN in order to create a redundant, reliable and scalable anomaly detection system for mobile network operators.

Our network traffic security architecture has been suggested to be based on a software definition and refers to a security platform on software level. Our architecture comprises an SDN application, an SDN controller, a clustering algorithm, two switches and several DaaS nodes. The platform comprises of an application layer, a management layer and a data layer. The application layer comprises an SDN application and an application interface. The management layer includes the SDN controller and switches (control plane), whereas the data layer includes switches (user plane), a clustering algorithm that groups input data based on the traffic feature and several DaaS nodes to detect unsafe traffic. In proposed platform DaaS has been suggested to act similar to IDS which analyses and detects threats. Often conventional IDS are deployed at the gateway, and can detect intrusion; but still the traffic passes through the network if it came from subscriber and is destined towards internet. With our architecture, we aim to stop the malicious traffic from entering further and polluting the network elements.

The malicious traffic should be stopped as early as possible in a mobile network so it doesn't affect the network elements. We have proposed utilizing IDS (DaaS) in conjunction with SDN controller in an SDN enabled network and mitigating malicious traffic using flow control techniques. DaaS will be analysing the network traffic and detecting the anomalies in it. Upon detecting malicious packet(s), DaaS will inform SDN application about threats. SDN Application will act as an orchestrator and will be gathering results from multiple DaaS nodes and formulate a flow that will be sent to SDN controller. The SDN controller then instructs the switch to perform actions including removal, modification or installation of flow, and thus such anomaly traffic doesn't pass through switch and is dropped. Considering the huge amount of traffic, various instances of DaaS should be used and clustering mechanism would be distributing traffic evenly among DaaS instances for load balancing purposes. Two approaches have been considered for the proposed architecture. In the first approach, traffic clustering is carried out on each packet of the mirrored traffic. While in second approach clustering is done on sampled traffic instead of each packet. An implementation with different types of attacks for both approaches is already ongoing by authors to evaluate the performance.

The proposed architecture will protect SDN from being overloaded and from resource abuse attacks. In addition, applied load balancing mechanism together with clustering on the sampled traffic would reduce SDN controller load and computing resources and therefore computation cost and latency.

REFERENCES

[1] M. Monshizadeh, V. Khatri and A. Gurtov, "NFV Security Considerations for Cloud-Based Mobile Virtual Network Operators_v06", *The 24th International Conference on Software, Telecommunications and Computer Networks (SoftCOM 2016),* Split, 2016.

[2] Open Networking Foundation (ONF), "Threat Analysis for the SDN Architecture," Technical Report 530, v1.0, Jul. 2016.

[3] Anthony Lim, "Security Risks in SDN and Other New Software Issues," RSA Conference, Jul. 2015.

[4] M. Monshizadeh, Z. Yan, L. Hippeläinen and V. Khatri, "Cloudification and security implications of TaaS," *Computer Networks and Information Security (WSCNIS), 2015 World Symposium on*, Hammamet, 2015, pp. 1-8.

[5] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems," Technical Report NIST SP 800-94, National Institute of Standards and Technology, 2007.

[6] H. Han, L. Yan, Z. Li and J. Zhang, "Network security defense system based on software-defined network and working method of network security defense system," Patent CN104539625, Apr. 22, 2015.

[7] C. Luo, "Network security traffic platform based on software definition," Patent CN104753951, Jul. 1, 2015.

[8] J. Y. Jo, J. U. Kong and K. M. Lee, "System and method for preventing network intrusion," Patent KR101553264, Sep. 15, 2015.

[9] N. K. P. Puttaswamy, F. Hao and T. V. Lakshman, "Securing software defined networks via flow deflection," Patent CN104685850, Jun. 23, 2015.

[10] W. Eddy, "TCP SYN Flooding Attacks and Common Mitigations," RFC 4987, IETF, Aug 2007.

[11] M. Liyanage, A. B. Abro, M. Ylianttila and A. Gurtov, "Opportunities and Challenges of Software-Defined Mobile Networks in Network Security," in *IEEE Security & Privacy*, vol. 14, no. 4, pp. 34-44, July-Aug. 2016.

[12] Open Networking Foundation (ONF). *OpenFlow Switch Specification Version 1.5.1*, Mar 2015. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-switch-v1.5.1.pdf

[13] P. Marques, N. Sheth, R. Raszuk, B. Greene, J. Mauch and D. McPherson, "Dissemination of Flow Specification Rules," RFC 5575, IETF, Aug 2009.

[14] J. Haas, Ed., "Clarification of the Flowspec Redirect Extended Community," RFC 7674, IETF, Oct 2015.

[15] L. Serodio, "Traffic Diversion Techniques for DDoS Mitigation using BGP Flowspec," May 2013. https://www.nanog.org/sites/default/files/wed.general.trafficdiversion.serodio.10.pdf

[16] A. Bremler-Barr, S. T. David, D. Hay and Y. Koral, "Decompression-free inspection: DPI for shared dictionary compression over HTTP," *INFOCOM, 2012 Proceedings IEEE*, Orlando, FL, 2012, pp. 1987-1995.

[17] Tor Project: Overview https://www.torproject.org/about/overview.html.en

[18] Mark O. Riedl, Andrew Stern, Don Dini, and Jason Alderman, "Dynamic Experience Management in Virtual Worlds for Entertainment, Education, and Training," *International Transactions on Systems Science and Applications,* Special Issue on Agent Based Systems for Human Learning, vol. 3(1), 2008.

[19] John Mattsson and Mats Näslund, "Detection and Mitigation of HTTPS Man-in-the-Middles and Impersonators," *W3C Workshop on Authentication, Hardware Tokens and Beyond*, Sep. 2014.

[20] Jeff Jarmoc, "SSL Interception Proxies and Transitive Trust," Blackhat Europe, Mar. 2012.

[21] J. Michael Butler, "Finding Hidden Threats by Decrypting SSL," SANS Analyst Whitepaper, Nov 2013.

[22] A. Yamada, Y. Miyake, K. Takemori, A. Studer and A. Perrig, "Intrusion Detection for Encrypted Web Accesses," *Advanced Information Networking and Applications Workshops, 2007, AINAW '07. 21st International Conference on*, Niagara Falls, Ont., 2007, pp. 569-576.

[23] John W. Pirc, "SSL Performance Problems: Significant SSL Performance Loss Leaves Much Room for Improvement," Analyst Brief, NSS Labs, Jun 2013.

[24] M. Monshizadeh and Z. Yan, "Security Related Data Mining," *Computer and Information Technology (CIT), 2014 IEEE International Conference on*, Xi'an, 2014, pp. 775-782.

Mehrnoosh Monshizadeh is finalizing her Ph.D. at Electrical School of Aalto University, Finland. She is working at Nokia Bell Labs as security research specialist. Her research interests include cloud security, mobile network security, IoT security and data analytics.

Vikramajeet Khatri has M.Sc degree in information technology from Tampere University of Technology, Finland. He is working as research security specialist at Nokia Bell Labs. His research interests include intrusion detection, malware detection, IoT security and cloud security.

Raimo Kantola has a D.Tech degree in computer science from Helsinki University of Technology, Finland. He is a professor in networking technology at department of Comnet, Aalto University, Finland. His research interests include SDN, customer edge switching, trust in networks and cloud security.