DDoS Attack Mitigation in Internet of Things Using Software Defined Networking

M. Ejaz Ahmed and Hyoungshick Kim

***Abstract*—Securing Internet of Things (IoT) systems is a challenge because of its multiple points of vulnerability. A spate of recent hacks and security breaches has unveiled glaring vulnerabilities in the IoT. Due to the computational and memory requirement constraints associated with anomaly detection algorithms in core networks, commercial in-line (part of the direct line of communication) Anomaly Detection Systems (ADSs) rely on sampling-based anomaly detection approaches to achieve line rates and truly-inline anomaly detection accuracy in real-time. However, packet sampling is inherently a lossy process which might provide an incomplete and biased approximation of the underlying traffic patterns. Moreover, commercial routers uses proprietary software making them closed to be manipulated from the outside. As a result, detecting malicious packets on the given network path is one of the most challenging problems in the field of network security. We argue that the advent of Software Defined Networking (SDN) provides a unique opportunity to effectively detect and mitigate DDoS attacks. Unlike sampling-based approaches for anomaly detection and limitation of proprietary software at routers, we use the SDN infrastructure to relax the sampling-based ADS constraints and collect traffic flow statistics which are maintained at each SDN-enabled switch to achieve high detection accuracy. In order to implement our idea, we discuss how to mitigate DDoS attacks using the features of SDN infrastructure.**

I. Introduction

Internet of Things (IoT) devices are cheaper and readily accessible internet-enabled embedded computing devices, which in these days are ubiquitous. The IoT are internet-enabled embedded computing devices such as digital home thermostats, smart TVs, security cameras, car systems (entertainment, navigation, and engine management computers), networking devices, smart watches, activity trackers and many more. However, this shift from desktop PC to mobile, and now to IoT devices poses huge security risks of all sorts. Consider for instance, the most recent and massive distributed denial of service (DDoS) attack in October 2016 targeting the Dyn server, which knocked offline major websites including Twitter, Spotify, Amazon, Reddit, Netflix, and The New York Times [1]. The following reasons could contribute to the success of a DDoS attack: 1) Due to the exponential growth of IoT devices around the globe, these networks continue to be poorly managed, e.g., lack of stringent security measures. Gartner predicts that by 2020 there will be over 26 billion connected devices, while other analysts believe the number will exceed 100 billion [2]. 2) The unprecedented traffic volumes which could be observed on contemporary enterprise networks and the stringent memory and complexity constraints of network devices, it is not feasible for an in-line ADSs to examine every packet in detail [3]. As a result, packet and flow sampling approaches are relied on to reduce the amount of data to be analyzed by a real-time in-line ADS resulting in bypassing malicious traffic from the compromised IoT devices towards the victim servers/hosts. 3) The malware "Mirae" which continuously scans the internet for IoT devices protected by factory default usernames and passwords, e.g., the list of 68 usernames and passwords is available in "Mirai" source code [4].

Packet forwarding in traditional network infrastructure rely on routing tables lookup routines to route packets from input ports to their respective destinations output ports. As discussed above, due to stringent memory and computational complexity constraints, it is not possible for in-line real-time ADSs to examine every packet in detail. It is therefore task of the destination network's firewall/ADS to examine traffic flows in detail and drop malicious packets/flows. In a typical DDoS flooding attack, significant amount of network bandwidth is consumed by the compromised IoT devices trying to inject huge number of malicious packets into the network targeting a particular victim server. Additionally, when such malicious packets arrive at the destination network might overwhelm the firewall/ADS making it unavailable to serve legitimate users. It is therefore significantly important to seek for network architecture allowing to examine network traffic flows on packet path and drop malicious flows/packets before they accumulate to create an avalanche effect at the endpoint firewalls/ADSs. Two major aims are achieved from such network architecture: Firstly, it reduces the network load/congestion, secondly, it prevents malicious IoT devices from amplifying the attack at the victims server/host.

In this paper, we argue that the advent of SDN provides a unique opportunity to effectively detect and mitigate DDoS attacks. By leveraging SDNs features such as software-based traffic analysis, logical centralized control, dynamic flow insertion and deletion at remote switches, and global view of the network, the malicious flows contributing to the DDoS flooding attack can be detected and effectively mitigated. Consequently, significant amount of bandwidth could be saved by dropping malicious traffic flows resulting in reduced network load and latency. The SDN architecture decouples the network control from the data (forwarding functions) which enables the network control to become directly programmable. The OpenFlow is a protocol that allows OpenFlow-enabled

This work was partly supported by the ITRC (IITP-2016-R0992-16-1006), and the IITP (No.R-20160222-002755). M. E. Ahmed and Hyoungshick Kim are with College of Software, Sungkyunkwan University (SKKU), Suwon, Korea (e-mail: ejaz629@skku.edu, hyoung@skku.edu).

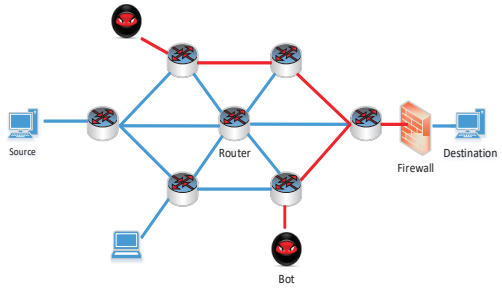

Fig. 1. Routing in traditional networks

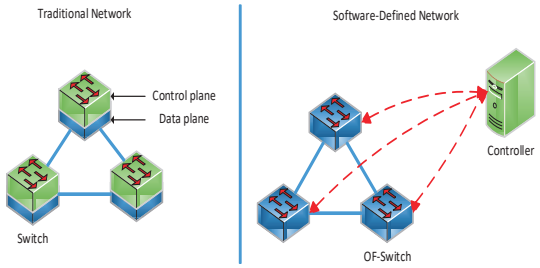

Fig. 2. Traditional network architecture Vs SDN network architecture.

switches (OF-switch) containing internal flow tables to be managed by an external controller [5]. If a flow entry exists in a flow table of OF-switch, it is forwarded in a normal way otherwise the incoming packet is sent to the controller for further analysis. The controller may perform the following actions: Add flow entry for traffic flow in the OF-switch, drop packets from malicious sources, and mirror any port to the external server for thorough packet analysis. The SDN infrastructure motivates the possibilities to detect and mitigate DDoS attacks in the internet.

The organization of this paper goes as: Section II present the background of packet forwarding in traditional networks and the problem statement. In section III, we discuss the SDN inter-domain innovation and propose our idea of DDoS mitigation using those innovations. Section IV explain the unique features of SDN. Section V discusses our proposal for DDoS mitigation. Section VI concludes this paper.

II. Background and problem statement

It is very important to understand how packets are forwarded in traditional network infrastructure before dealing with the subject of mitigating DDoS attacks. Today's network devices program their forwarding tables locally, meaning these devices make their own decisions internally about how to forward traffic. Traditional internet traffic forwarding uses routing tables to forward incoming packets to destinations by performing routing table lookups. For example, a packet destined for a particular destination arrives at a router, the router consults its routing table to find the destination address entry in its routing table. After matching the network address of destination node in its routing table, the packet is sent via the outgoing interface which leads to the destination, as illustrated in Fig. 1. In this way, packets are forwarded in traditional network infrastructure.

On the other hand, the emerging SDN infrastructure separates the control plane from the data plane (network devices), as illustrated in Fig. 2. The complex control intelligence is held on a centralized controller that understands the complete topology of the network [6]. Such network architecture imparts ability to the controllers to dynamically manipulate traffic flows or packets flowing in the network. For instance, controllers can, depending on the situation, drop flows/packets at remote OF-switches, if they are detected to be malicious.

Conventional out-of-band (outside direct line of communication) ADSs are typically deployed at network edges where detailed packet payload data is available. However, an inline ADSs deployed in packet's path rely on sampling-based approaches for anomaly detection. Therefore, it remains the job of out-of-band ADSs to examine each arriving packet (after traversing network devices in the packet path) to their network in detail, and correspondingly decide about its legitimacy.

We do not consider this line of argument to be continued and therefore propose that deep packet examination should be performed at the intermediate network devices to decide about the validity of packets before they arrive at their respective destination networks. We propose that packets flowing from source to destination, as illustrated in Fig. 1, should be examined and discarded at intermediate routers if they belong to malicious traffics. For that, we propose an SDN infrastructure which provides a unique opportunity to dynamically monitor traffic flows at network devices (data plane) and install traffic flow rules at the intermediate OF-switches remotely.

III. Inter-Domain SDN Innovations for DDoS Mitigation

Here we discuss the validity of our proposal about DDoS mitigation using SDN inter-domain information exchange. It is claimed that the Internet is managed and controlled by owners of different administrative domains, so the centralized control model of SDN may not be applicable to outside the domain. Which means that it is not possible to monitor malicious/legitimate traffic flows outside of an administrative domain. Thus, examining and dropping malicious traffic flows at the intermediate routers can not be realized. This argument is based on the fact that since SDN architecture separates the network control plane from the network data plane, and moves it to a centralized controller, the controller manages and controls only intra-domain network devices and information, such as OF-switches, bandwidth, routes, etc. For instance, Fig. 3 represent northbound logical structure of a single domain SDN network. Here, the SDN domain refers to the administrative SDN domain. One SDN domain may include multiple ASs (autonomous systems). The architecture in Fig. 3 shows the network devices in the data plane are closed and can only be manipulated from intra-domain controllers, not from outside or other domains.

Another argument is that to avoid computational complexity on routers, the task of deep packet inspection[1] is performed

[1]Note that here we refer to thorough inspection of every packet, not sampling-based packets inspection.

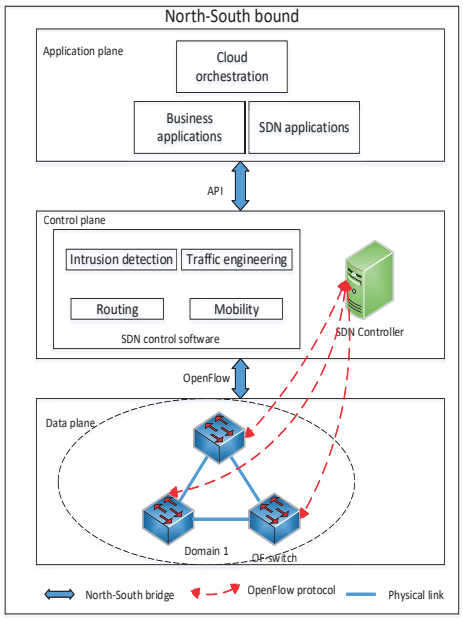

Fig. 3. Single domain SDN logical structure.

at out-of-band ADSs, not at the intermediate routers. Thus the network performance would degrade if such computationally expensive tasks are performed at in-line routers/ADSs. This argument is based on the fact that to deal with increasing link speed, sampling techniques are deployed at intermediate routers, for example, Cisco's Anomaly Guard [8] and Junipers Traffic Sampling [9], to reduce overhead in terms of router CPU, memory, and bandwidth. Commercial in-line ADS products are integrating sampling and anomaly detection algorithms in the routing fabric in order to achieve high-speed and truly-inline anomaly detection in real-time. In practice, sampled traffic flow data is used by the network providers for traffic engineering (TE) and capacity planning tasks. However, in recent years, it has also been used as an input for anomaly detection, e.g., detecting distributed DDoS attacks or worm scans. Therefore, to achieve efficient network performance and to cope with the computational overhead at routers, the task of detecting DDoS attacks is delegated to the out-of-band ADSs so as to deeply inspect each packet before passing it to the destination local network.

We do not agree with the above statements. For the first argument, SDN was initially proposed to run experiments in real networks such as campus networks [10], however, further development in SDN extended SDN to SDNi [7] to interface between multiple SDN domains so as to enable SDN controllers to share information among each other. This extension to SDNi plays critical role in mitigating DDoS attacks due to the fact that malicious packets travelling across multiple domains can be tracked and dropped by using this extension. It is responsible for coordination between SDN controllers to exchange control and application information across multiple SDN domains. SDNi is an "East-West" protocol between SDN controllers, as an analogy to OpenFlow being a "North-South" protocol between controller and Network devices, as illustrated in Fig. 3. Inspired from SDNi, researcher have designed new inter-domain mechanisms named West-East Bridge (WE-Bridge) for inter-domain SDN, which is used for different SDN administrative domains to collaborate [11], [12]. Fig. 4 shows that various administrative domains are connected to each other via WE-bridge interface. This feature of SDNi enable controllers to monitor traffic statistics across various domains and very useful to detect DDoS attacks. WE-Bridge itself is a platform to exchange basic network information between different administrative domains, and enable applications to carry out SDN inter-domain innovations.

In practice, Link Layer Discovery Protocol (LLDP) is used by SDN controllers to discover the underlying network topology. The controller of each SDN domain instructs the connected OF-switches to broadcast the LLDP packets out over all the ports (the LLDP packet contains the out-port, the source OF-switch identity, and other capabilities) [11]. The neighbor OF-switches upon receiving the LLDP packets send them directly to the SDN controller. Then the controller analyzes the information from the LLDP packets to determine if the source OF-switch's identity belongs to its administrative domain and the LLDP packet received by a neighbor is the same as the one sent out from the source OF-switch. If this condition is met, the controller will then create a direct intra-domain link between the source switch and this neighbor. For the inter-domain link, the LLDP is extended in the controller by adding a network view driver [11], i.e., if the source OF-switch's identity does not belong to its domain, then the controller can infer that this packet is from another domain, and will correspondingly create an inter-domain link according to the source OF-switch identity, source switch out-port, and the destination switch (who received the LLDP packet in its SDN domain) identity with the in-port. The inter-domain links should be stored in both the neighbor domains local network views. Using such inter-domain communication flows can be monitored to detect anomalies in traffic behavior under DDoS attacks.

For the second argument, OF-switch send the packets to the SDN controller if no entry in the flow table is found. Then the controller install the flow entries in the OF-switch after analyzing the packets. Once the flow installation, traffic flows are routed in a normal way, so no extra computational requirement except monitoring flow statistics at each OF-switches. Also, SDN controllers have reasonable computational power to monitor traffic flows.

IV. SDN Features

SDN provides unique features that plays an instrumental role in detecting and mitigating DDoS attacks. We summarize the features of SDN as follows:

1) *Separation between control and data planes*: SDN decouples the data plane and control plane thereby providing grounds to establish large scale attack and defense experiments. Progressive deployment of innovative ideas can be realized via seamless transition from an experimental phase to an operational phase. Moreover

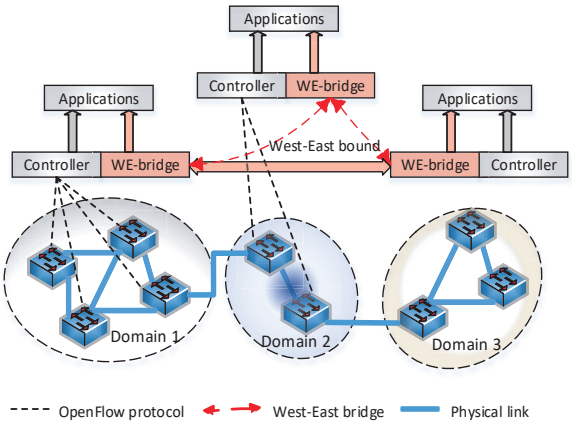

Fig. 4. Multi-domain SDN logical structure.

it enables innovation and evolution by providing a programmable network platform to implement, experiment, and deploy new ideas, new applications. By using this feature, DDoS attack detection and mitigation is realized with much convenience.

2) *Logical centralized controller and view of the network*: The controller has network-wide knowledge of the system and global views to build consistent security policies and to monitor or analyze traffic patterns for potential security threats [13]. Centralized control enables the centralized SDNi controllers to monitor traffic patterns generated by various sources with in their corresponding networks and identify potential anomalies in the behavior of sources.
3) *Cooperation among SDN domains via SDNi*: As discussed earlier, the inter-domain cooperation is paving path for the efficient detection and mitigation of DDoS attacks initiated from locations across the globe. The WE bridge for SDN inter-domain peering serves as a good motivation for inter-domain peering. The DDoS attack is launched from an army of the compromised sources, geographically distributed across the globe, the malicious packets from such sources traverse through many administrative domains before converging at the target victim host making it unavailable. Such malicious packets are identified while traversing through various administrative network domains using SDNi protocol, the DDoS attacks could be early detected and mitigated without collapsing the victim server/host. Such early detection via SDNi serves for two major purposes, first, reduces the network load/congestion by dropping malicious packets at the intermediate SDN domains, second, the impact of DDoS can be significantly reduced or totally eliminated by imparting intelligence to network entities by exploiting the features provided by SDN/SDNi. For instance, the WE bridge developed and implemented in [11] first defines what network information could be exchanged and how such information is efficiently exchanged among inter-domain SDN peers. With SDNi protocol, they built an intercontinental SDN testbed with four SDN networks: CERNET, CSTNET, Internet2, and SURFnet. To verify the WE-Bridge platform, they designed, implemented, and deployed two SDN inter-domain routing innovations which could not be achieved from the traditional network architecture.

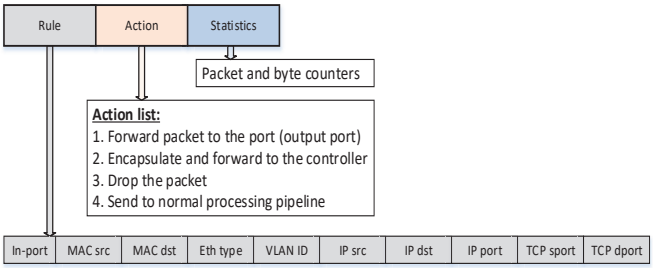

Fig. 5. Flow table entry.

4) *Dynamic updating of forwarding rules and flow abstraction*: The OpenFlow protocol provides a common interface to control how packets are forwarded by accessing OF-switch's internal flow tables and the statistics associated with these flows. A flow is a 12-tuple entry in a flow table in OF-switches with the fields that are matched against the incoming packets. Each flow entry has an action, a priority, and statistics fields, as illustrated in Fig. 5. Each flow entry contains a header against which the incoming packets are matched and the corresponding action is applied on the packet. On the other hand, if no matching entry is found in the flow table of an OF-switch, the packet is forwarded to the controller. After analyzing those packets, the controller may decide to install flows in the OF-switch or just tell the OF-switch to drop the packet. Additionally, controllers can perform the following after analyzing the packet: 1) Add flow entry in the switch, 2) Drop packets from the flow it is malicious, 3) Mirror any port to external server for flow analysis.
5) *Software-based traffic analysis*: Software-based traffic analysis greatly enables innovation, as it is possible to improve the capabilities of a switch using any software-based technique. Traffic analysis can be performed in real time using machine learning algorithms (Support Vector Machines (SVMs), Gaussian Mixture Models (GMM), Artificial Neural Networks (ANNs)), databases and any other software tool. Traffic of interest can be explicitly directed to intrusion prevention systems (IPSs) for Deep Packet Inspection (DPI).

V. SDN as a Potential Defense Mechanism

Here we discuss our proposals for utilizing SDN to guard networks against DDoS attacks and provide evidence from the most recent literature.

A. Source-based attack mitigation in SDN

SDN controllers are able to detect anomaly in traffic behavior and can correspondingly filter the malicious packets

near the ingress of the network[2] Therefore, all the connection requests from the spoofed IP addresses originating from the reference SDN network can effectively be detected and dropped so as to reduce the impact of such malicious connections contributing in a DDoS attack. For instance, the authors in [15] demonstrate that a programmable home network router can provide an ideal platform and location in the SDN network for detecting security threats in SOHO (Small Office/Home Office) networks. Four prominent traffic anomaly detection algorithms, i.e., threshold random walk with credit based rate limiting (TRW-CB), rate-limiting, maximum entropy detector, and NETAD are implemented in an SDN context using OpenFlow compliant switches and NOX as a controller. Threshold random walk with credit based rate limiting (TRW-CB) is a classification method using sequential hypothesis testing (i.e., likelihood ratio test) to classify whether or not the internal host has a scanning infection. It is based on the observation that the probability of a connection attempt being a success should be much higher for a benign host than a malicious one. Rate Limiting uses the observation that an infected machine has different connection characteristic to limit new connection rate. The Maximum Entropy detector estimates the benign traffic distribution using maximum entropy estimation. Unlike TRW-CB and Rate Limiting, Maximum Entropy relies on examining every packet in order to build packet class distributions every *t* seconds. NETAD operates on rule-based filtered traffic in a modeled subset of common protocols. The filter removes uninteresting traffic based on the premise that the first few packets of a connection request are sufficient for traffic anomaly detection [15]. Experiments indicate that these algorithms are significantly accurate in identifying malicious packets in the home networks as compared to the ISP, and that the anomaly detectors can operate at line rates without introducing any performance penalties for the home network traffic.

B. Network attack mitigation in SDN

Controllers, also called as network operating systems (NOS), allow global watch and control of the SDN network from the packet flow perspective. The controller can view/control all the network devices within its administrative domain. Thus any deviation from the normal behavior in network traffic flows among the network devices will be directly observed by the controller, and corresponding preemptive measures are taken to prevent potential threats. The controller can be implemented using NOX (written in C++), POX (written in Python), OpenDayLight (written in Java), where OF-switches [5] keep Flow Tables with statistics about all active flows. All the features information required is easily accessible by means of SDN controllers (POX, NOX, OpenDayLight), and then processed by an intelligent algorithms to decide about the maliciousness of traffic flows. Therefore, those attacks which are being launched from or converges at the SDN network domain can easily be detected and mitigated by exploiting the features offered by the SDN.

For instance, Braga et al. [16] proposed a lightweight method for detecting DDoS attacks based on the traffic flow features, in which the extraction of such information is made with a very low overhead compared to traditional approaches. This approach is divided into three main parts, first, since the controller can ask for flow table statistics maintained by each OF-switch, the controller periodically request for flow statistics from flow tables of OF-switches in the network. Second, the features are extracted from the flow table statistics obtained from the first step. These features include: Average of Packets per flow (APf), Average of Bytes per flow (ABf), Average of Duration per flow (ADf), Percentage of Pair-flows (PPf), Growth of Single-flows (GSf), and Growth of Different Ports (GDP). The Feature Extractor module gathers them in 6-tuples to be passed to the classifier. Third, the classifier analyzes whether or not a given 6-tuple data entry corresponds to a DDoS flooding attack or to legitimate traffic.

Self Organizing Maps (SOMs) are used as the classification method. SOM, like other machine learning algorithms, needs to be trained with a sufficiently large data set of 6-tuple records collected during normal and attack traffics. SOM is able to create a topological map where various regions represent different traffic types. When the trained SOM is provided with a 6-tuple feature point, it will be able to classify it either as normal traffic or attack traffic. To compute the topological neighborhood, a Gaussian function is used due to the fact that it induces the SOM algorithm to converge more quickly than a rectangular topological neighborhood does. Thus SDN provides an efficient platform to observe network devices in a realtime and respond correspondingly in a timely fashion which is not possible in traditional networks.

Flow-based monitoring in SDN is based on the information in packet headers, so flow-based IDSs have to handle a considerably lower amount of data compared to the payload-based IDSs. Tuan et al. [14] proposed deep learning approach for network intrusion detection in SDN. They constructed a simple deep neural network with an input layer, three hidden layers and an output layer. The input dimension is six and the output dimension is two. The six input features are duration, protocol type, source bytes, destination bytes, count, and service count. However, the results are not yet encouraging enough to be adopted in any commercial product, but the approach still has significant potential and advantages for further development.

C. Cross-domain attack mitigation in SDN

Transitioning from the widely deployed Internet infrastructure to SDN presents creative ideas for incremental deployment of SDN networks. During this transition, the SDN networks are required to coexist with the traditional IP networks and any SDN deployment must cooperative in the following manners, to exchange reachability information, forward traffic, and express routing policies with existing IP networks. In

[2]Ingress filtering in traditional networks is usually not enabled. Additionally, there are a large number of open DNS resolvers that could be exploited for launching DDoS attacks. Consequently, the DDoS attack can easily be launched from the compromised devices with spoofed IP addresses in such networks.

this way, the DDoS attacked launched from the compromised hosts/IoT devices geographically located across the globe can not only be efficiently detected but also mitigated. Unlike traditional networks, it is possible with inter-domain SDN information exchange. SDN domains inter-connected with each other need to seamlessly peer with IP networks.

Crosss-domain transitioning approach is proposed and implemented in [11], [12]. Lin et al. [11] implemented the WE-Bridge (discussed earlier), and for deployment, an international federal SDN testbed is made functional, as shown in Fig. 4. Four SDN networks are covered in this testbed: Internet2 (United States open national research and education network), CERNET (China Education and Research Network), CSTNET (China Science and Technology Network), and SURFnet (the national research and education network of the Netherlands). The WE-Bridge has successfully connected those four SDN networks and tested for experimentation. Thus, network-wide attacks consuming network resources such as computation, memory, and bandwidth could easily be tracked down using inter-domain SDN infrastructure. Moreover, DDoS attacks emanating from large number of compromised IoT devices or hosts could be detected and mitigated based on the information exchange among various SDN domain controllers.

D. Fast packet examination for legitimacy in SDN

We propose that SDN offers a unique opportunity to delegate network security tasks to the host networks while sparing the home user from complex security management tasks. Each SDN network has the ability to programmatically control the forwarding structure of the OF-switches at line rates using open controller software. Commercial in-line ADSs deployments in the network core suffer from two limitations: 1) *Low detection rates*: It is because in-line ADSs are integrating sampling and anomaly detection algorithms in the routing fabric in order to achieve high-speed and truly-inline anomaly detection in real-time. However, packet sampling is inherently a lossy procedure which as a result provides an incomplete and biased approximation of the underlying traffic flows. 2) *Inability to achieve line rates for detection in the network core*: To alleviate large memory and CPU requirements at routers and to maximize network bandwidth utilization, packet/flow sampling approaches are mostly deployed at in-line ADSs in the network core. However, to achieve line rates, sampling-based approaches are inaccurate.

The problems mentioned above can easily be mitigated if anomaly detection is performed close to the anomalous sources, i.e., at end points in home networks [15]. The main obstacle in realizing this concept is that traditional routers are running proprietary software which make them closed to be manipulated from outside. Thus we believe that the advent of SDN can provide an appealing solution to such problems. The *standardized programmability* feature of SDN offers to share the burden on in-line ADSs among home and intermediate networks which will add to the detection performance.

VI. Conclusion

Deployed in-line ADSs in networks cores against potential threats is an attractive choice for fulfilling the security needs of applications. However, to maintain wire rates and higher detection accuracy, a transition from sampling-based in-line ADSs to thorough flow/packet examination approach is required. Moreover, with exponential growth of IP traffic and limited computational/memory resources, the burden over in-line sampling-based ADSs needs to be shared in an efficient way. Keeping these challenges in view, we believe that the emerging SDN infrastructure using OpenFlow can potentially be a viable network infrastructure to overcome recent challenges faced by traditional networks. The DDoS attack emanating from large number of compromised IoTs/hosts from different geographical regions can efficiently be defended with-in the core network before converging and making the victim server unavailable. We discussed possibilities to guard networks using inter-domain SDNi which is an extension of the SDN. We propose that by examining packets/flows at the intermediate SDN domains, the DDoS attacks could be efficiently mitigated before it arrives at the victim network.

References

[1] Filip Jelic, "Analysis: Record DDoS Attacks by Mirai, IoT Botnet," *Deep.Dot.Web,* Nov. 2016.

[2] Christy Pettey, "The Internet of Things and the Enterprise," *Gartner,* Aug. 2015.

[3] S. Ali, I. U. Haq, S. Rizvi, N. Rasheed, U. Sarfraz, S. A. Khayam, and F. Mirza, "On mitigating sampling-induced accuracy loss in traffic anomaly detection systems," *ACM SIGCOMM Computer Communication Review,* vol. 40, no. 3, pp. 4-16, 2010.

[4] Brian Krebs, "Who Makes the IoT Things Under Attack?," *KrebsonSecurity,* Oct. 2016.

[5] OpenFlow, "OpenFlow Switch Specification, Version 1.5.1," *Open Networking Foundation,* Mar. 2015.

[6] Network Services/Service Providers, "Software Defined Networking (SDN) Explained," *COMMSBUSINESS,* May 2016.

[7] H. Yin, H. Xie, T. Tsou, "SDNi: A Message Exchange Protocol for Software Defined Networks (SDNS) across Multiple Domains," *Internet Research Task Force,* Internet-Draft, June 2012.

[8] Cisco Anomaly Guard Module. www.cisco.com/en/US/products/ps6235/.

[9] Juniper Networks: JUNOS 7.2 Software Documentation.

[10] M. Nick, et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69-74, 2008.

[11] P. Lin, J. Bi, S. Wolff, Y. Wang, A. Xu, Z. Chen, H. Hu, Y. Lin "A west-east bridge based SDN inter-domain testbed," *IEEE Communications Magazine*, vol. 53, no. 2, pp. 192-197, 2015.

[12] P. Lin, et al.,"Seamless interworking of SDN and IP," *ACM SIGCOMM computer communication review*, vol. 43, no. 4, pp. 475-476, 2013.

[13] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 1, pp. 602-622, 2016.

[14] T. A. Tang, L. Mhamdi, D. McLernon, S. A. R. Zaidi, and M. Ghogho, "Deep Learning Approach for Network Intrusion Detection in Software Defined Networking," *IEEE International Conference on Wireless Networks and Mobile Communications, WINCOM,* pp. 258-263, Fez, Morocco, Oct. 2016.

[15] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," *Springer International Workshop on Recent Advances in Intrusion Detection,* pp. 161-180, Berlin Heidelberg, 2011.

[16] R. Braga, M. Edjard, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," *IEEE 35th Conference on Local Computer Networks,* pp. 408-415, Denver, Colorado, U.S.A., Oct. 2010.