Blocking Intrusions at Border using Software Defined-Internet Exchange Point (SD-IXP)

Mauro Conti
Department of Mathematics
University of Padua, Italy
conti@math.unipd.it

Ankit Gangwal
Department of Mathematics
University of Padua, Italy
ankit.gangwal@math.unipd.it

***Abstract*—Servers in a network are typically assigned a static identity. Static assignment of identities is a cornerstone for adversaries in finding targets. Moving Target Defense (MTD) mutates the environment to increase unpredictability for an attacker. On another side, Software Defined Networks (SDN) facilitate a global view of a network through a central control point. The potential of SDN can not only make network management flexible and convenient, but it can also assist MTD to enhance attack surface obfuscation.**

In this paper, we propose an effective framework for the prevention, detection, and mitigation of flooding-based Denial of Service (DoS) attacks. Our framework includes a light-weight SDN assisted MTD strategy for network reconnaissance protection and an efficient approach for tackling DoS attacks using Software Defined-Internet Exchange Point (SD-IXP). To assess the effectiveness of the MTD strategy and DoS mitigation scheme, we set two different experiments. Our results confirm the effectiveness of our framework. With the MTD strategy in place, at maximum, barely 16% reconnaissance attempts were successful while the DoS attacks were accurately detected with false alarm rate as low as 7.1%.

***Index Terms*—Denial of Service, Software Defined Networks, Moving Target Defense, Network Security**

I. Introduction

The Internet has become an integral part of our everyday life, and it strongly affects the economy, politics, etc. While several network security issues are still open, network-based attacks such as large-scale port scan, DoS attacks are becoming increasingly powerful and frequent. In 2016, on average, 58.3% websites were targeted more than once, and 13.1% were targeted more than ten times[1]. A critical aspect of any attack tackling mechanism is to properly classify attack traffic from legitimate traffic. Misclassification of the traffic may lead to customer dissatisfaction causing a substantial impact on revenue/cost and especially, on the reputation of service providers.

The first step in almost all exploits and attacks (except zero-day vulnerabilities) is to identify vulnerabilities and weakness in the target. Investigating the attack surface may include (but not limited to) finding saturation points through network mapping, finding the next victim for worm propagation through network address scanning, or recognizing the version of software running on the target to exploit version-specific vulnerabilities. MTD has gained significant attention from both security experts and research community, as it can help in degrading the effectiveness of an attack by preventing or at least delaying the network reconnaissance.

On another side, SDN is a recently emerging networking paradigm. SDN offers a flexible network management by giving network operators a direct control over network functioning and allows them to perform a variety of actions. Networking experts believe that SDN will shape the architecture of the future Internet. Since SDN enables matching on multiple header fields, there is growing interest in applying the concepts of SDN in the wide-area network to make its management easier. An Internet Exchange Point (IXP) can be an interesting place to begin because it plays a central role in interconnecting many networks. An IXP is a network location where multiple independently operated networks, also known as Autonomous Systems (ASs), exchange internet traffic with one another. An SD-IXP is an IXP that is governed by the principles of SDN.

Contributions: In this paper, we propose an effective framework for detection and mitigation of flooding-based DoS attacks. Our framework employs an SDN assisted MTD strategy as the first line of defense, which is reinforced by a change-point detection technique for DoS detection and mitigation. In particular, the major contributions of our work are as follows:

1) We propose a framework to tackle DoS attacks using SD-IXP. To the best of our knowledge, our work is the first proposal that explores DoS mitigation using SD-IXP.
2) We implemented a light-weight MTD technique that utilizes the global-view available to the SDN controller.
3) We emulated our solution and evaluated its effectiveness using CAIDA [1] and DARPA intrusion detection evaluation dataset [2].

Organization: The remainder of this paper is organized as follows. Section II thoroughly explains MTD and its typical implementation approaches for network security followed by a summary of related works. Section III elaborates threat model. In Section IV, we give a detailed description of our framework for DoS prevention, detection, and mitigation. Section V elucidates our experimental setup and results. Finally, Section VI concludes the paper.

II. Preliminaries and Related Work

In this section, we elucidate MTD and its standard practices for network security, followed by a brief overview of the main research studies related to denial-of-service attacks in SDN.

[1] http://www.govtech.com/blogs/lohrmann-on-cybersecurity/online-denial-of-service-attacks-a-growing-concern.html

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

A. *Moving Target Defense*

MTD believes that perfect security is unattainable. MTD intends to morph the target, so the attacker is compelled to learn the target over and over again. Such mechanisms increase the complexity and uncertainty of the system to reduce the window of opportunity for an attacker while increasing the cost of attack attempts. It enables the system to continue a safe operation even in a compromised environment [3]. MTD can be broadly classified into three categories: network-level MTD, host-level MTD, application-level MTD.

1) Network-level MTD: It focuses on changing the topology of the network to deceive the attacker at the network reconnaissance and mapping phase. It includes imitating fake listening hosts, IP hopping/mutation, extra closed/open ports, randomized port numbers, obfuscating port traffic.
2) Host-level MTD: It includes faking information about the host, its OS version/type. Broadly, it focuses on the alteration to the OS and host-level naming, resources, and configuration.
3) Application-level MTD: The primary goal of such techniques is to change the environment in which an application executes. It includes shuffling memory layout of the application, randomly changing application version and type, modifying the source code at each compilation, and/or altering the programming languages and settings to compile the source code [4].

Irrespective of the category, the major idea of any MTD strategy is to mutate the environment to prevent or delay the attacks on the system.

B. *Related Work*

In this section, we provide a summary of studies related to MTD and DoS attacks in SDN. For SDN, we focus only on the works proposing MTD-based solutions for DoS. Rowe et al. in [5] evaluate the security of an MTD attempt and also quantify its effectiveness based on mitigation costs. Here, the MTD approach includes IP address and memory randomization along with a heavyweight stateful machine for protocols such as DHCP. Dunlop et al. [6] propose Moving Target IPv6 Defense (MT6D) that implements MT6D tunneled packets to rotate and hide IPv6 assignments. In order to make the tunnels, MT6D requires a nonce, a secret key, and the endpoint's interface identifier, which makes it impractical in the existing networks. Yackoski et al. [7] use Linux hypervisor to furnish similar functionality. Colbaugh et al. [8] propose a Game Theory-based solution to model adversary's activity and correspondingly optimize mitigation strategy. Here, the solution assumes that an attacker always optimizes its actions for an extreme percussion, which might not always be correct.

The fundamental principles of SD-IXP are described in [9]. Conti et al. in [10] raise the concern of the possibility for an attacker to obtain critical information about an SDN network. Jafarian et al. in [11] present an OpenFlow-based mutation scheme for SDN that exploits OpenFlow capabilities to protect against network reconnaissance by changing the identity of hosts. The approach in [12] presents a similar idea that adopts random route mutation to optimally randomize the path between a pair of hosts. Kampanakis et al. in [13] focus specifically on network mapping and reconnaissance protection. MacFarland et al. [14] introduce a concept of allowing the defenders to discriminate between untrustworthy and trustworthy clients using a trusted computing base. To provide access control to legitimate clients, it relies on cryptographic MACs, pre-shared keys, or at least embedded passwords. The work presented in [15] employs a multi-controller system to solve the problem of saturation. However, the approach has several limitations. On one side, it uses random packet transmission delay to protect from scanning attacks, which in fact, affects the data transmission for legitimate users. On another side, synchronization of prolonged route tables among multiple controllers is overlooked.

Our work is different from the state-of-the-art on many dimensions: (1) to the best of our knowledge, it is the first proposal that implements DoS mitigation at SD-IXP; (2) since it functions at SD-IXP, it minimizes the operation overheads for MTD; and (3) it does not depend on any additional infrastructure such as trusted computing base.

III. Threat Model

The target of the attacker is a server, i.e., victim server, which provides services to the hosts. The victim and the attacker reside in different ASs. The attacker has no information about the topology of victim's network. But, the attacker knows the IP prefixes that the router of target's AS is announcing. None of the hosts in the entire system is aware of mutations (for mutation details, please refer to Section IV-A). The SD-IXP controller has prior information about the victim server to be protected. When the attacker launches a DoS attack, it certainly passes through SD-IXP switch before reaching victim's AS. To avoid excessive queries to DNS and thus detection, an attacker uses network scanning techniques to scan a whole range of IP addresses in the network.

IV. Proposed Approach

In this section, we present our framework for preventing, detecting, and blocking DoS attacks. Here, we elucidate the fundamental principles of our system, followed by its comprehensive implementation details.

A. *SDN-based Random Host-IP Mutation for Reconnaissance Protection*

We chose random host-IP mutation as the MTD strategy. The fundamental concept of random host-IP mutation is to regularly change the identity of the hosts in a network. In our system, the controller regularly assigns a fresh random IP address to every network host. A fresh IP address is allotted to a host under the following circumstances:

1) On a predefined interval of time.
2) When a host has received a predefined maximum number of connections.

With an aim to minimize the burden on the controller, we use a scheme of virtual IP addresses and real IP addresses. To reach a host within the same network, the source must use the real IP address of the destination. While to reach a host outside the network, the source must use the current virtual IP address of the destination. As shown in Figure 1, a virtual IP address is translated to the real IP address at the edge of the network. The translation occurs due to the flow-rules installed by the SD-IXP controller in the SD-IXP switch. Any request to the real IP address, from a host outside the network, is dropped at the edge. Mutating host-IP address has following considerations:

1) Preserving integrity of the network configuration,
2) Minimizing operational costs,
3) Preventing disruption of the existing connections while IP addresses are changed.

The controller keeps a mapping of the virtual IP addresses to the real IP addresses. Since the controller has a global view of the network, the mapping is always consistent and updated. The biggest advantage of such address translation scheme is that when the virtual IP address of host changes, then only the SD-IXP switch needs to be updated, which minimizes the operational overheads. At the same time, hosts need not care about the mutations.

We explain management of existing connections with the help of Figure 2. Let host H_A be a server that provides services to clients. The real IP address of H_A is $real\_IP_A$, and at time T_x a virtual IP address vIP_1 is assigned to H_A. When H_B attempts to send a request to H_A using its current virtual IP address, i.e. vIP_1, the SD-IXP switch first sends the packets to the controller to obtain necessary flow-rules. Since the packets from H_B request access to the valid virtual IP address of H_A, the controller sends proper forwarding rules to the switch, where $real\_IP_A$ replaces vIP_1. The flow entries are installed with both HARD_TIMEOUT and IDLE_TIMEOUT so that they expire and are removed from the flow-table of the switch. It also allows individual flows to persist even after a host's IP address changes.

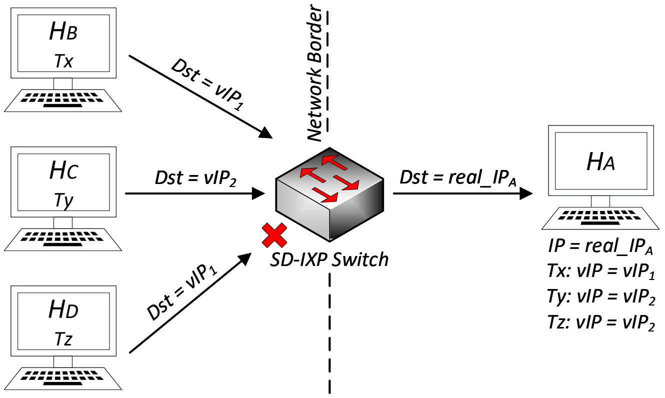


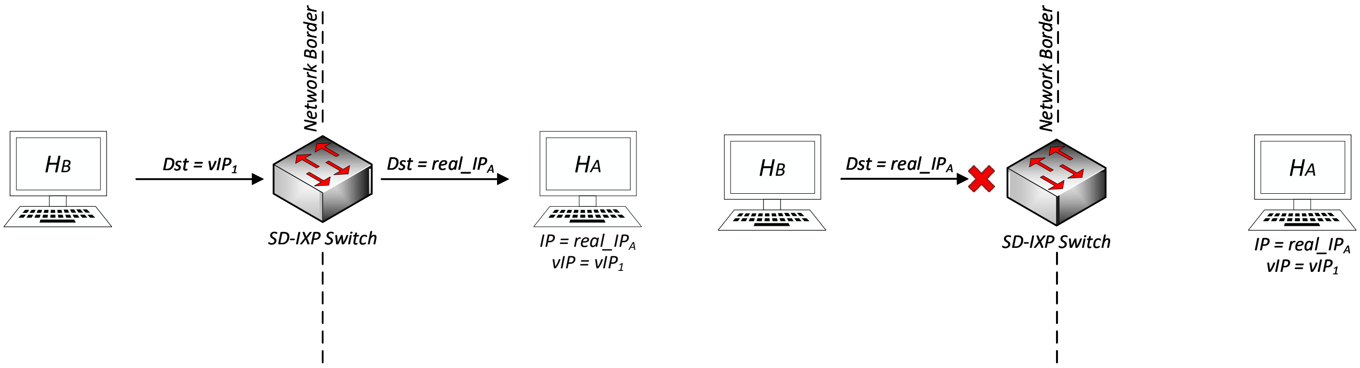
Figure 2: Connection management

Once the virtual IP address of H_A changes at time T_y, any other host will be permitted to connect with H_A only via its new virtual IP address, i.e., vIP_2, e.g., H_C in Figure 2. Importantly, if the flow-table entry in the switch between H_A and H_B that utilizes vIP_1 has not yet expired then H_B's connection would still be valid despite the virtual IP address of H_A has been changed. At this point, if a different host H_D attempts to connect with H_A via the expired virtual IP address, i.e., vIP_1, the controller instructs the switch to drop the traffic from H_D. It is important to note that if the flow-table entry between H_A and H_B has not yet expired, H_B would still be able to reach H_A using the expired vIP_1.

The current implementation of the mutation scheme requires the controller to frequently update Domain Name System (DNS) with the newly generated virtual IP addresses.

B. DoS Detection and Mitigation

A general characteristic of a denial of service attack is that the network observes abrupt changes in the intensity of the traffic when an attack is launched. In the event of such attacks, the statistical properties of network traffic also observe abrupt changes. Hence, the problem of attack detection can



(a) Requests to a valid virtual IP address are translated and forwarded

(b) Requests to the real IP address are denied at the edge

Figure 1: Translation of a virtual IP address to the real IP address at the edge

be formulated as a change-point detection problem [16]. The fundamental idea behind change-point detection approaches is to detect alterations in the statistical properties of the observed parameters with minimal latency and false positive rate.

Statistical Process Control (SPC) techniques have been widely utilized for controlling and monitoring the quality of manufacturing processes. SPC techniques can either be multivariate or univariate. SPC techniques can prominently detect changes in mean shifts (process mean), variance changes (process variance), counter-relationship among multiple variables [17]. Our work focuses on detecting significant changes in the network traffic intensity for DoS detection. The traffic intensity is a single variable that measures the amount of traffic flowing in the network. Hence, we consider only univariate SPC techniques to monitor mean shifts in the traffic intensity to detect possible DoS attacks. CUSUM control charts, Exponentially Weighted Moving Average (EWMA) [18] control charts, and Shewhart control charts are the typical univariate SPC techniques that are widely employed to detect mean shifts. The EMWA control charts are robust to non-normality and are almost perfectly non-parametric (distribution-free) procedures [19]. Since the normality of network traffic cannot be guaranteed, we choose EWMA control charts. A detailed description of EWMA control charts is given in the work [19]. In our work, we compute EWMA of packet arrival rate as shown in Eq. (1).

$$S_t = \begin{cases} N_t^{pk}; & \text{if } t = 1, \\ \alpha * N_t^{pk} + (1-\alpha) * S_{t-1}; & \text{otherwise,} \end{cases} \tag{1}$$

where t represents the time of current observation, $t-1$ represents the time of previous observation, S_t represents EMWA of packet arrival rate at time t, N_t^{pk} represents the number of packets arrived between $t-1$ and t. α is the smoothing factor that varies between zero and one, i.e., $0 < \alpha < 1$. A higher value of α discounts older observations faster. For an N-period moving average system, α is typically calculated as shown in Eq. (2).

$$\alpha = \frac{2}{N+1}. \tag{2}$$

The μ_S and σ_S of S_t are:

$$\mu_S = \mu_{N^{pk}}, \tag{3}$$

$$\sigma_S^2 = \sigma_{N^{pk}}^2 \cdot \left(\frac{\alpha}{2-\alpha}\right). \tag{4}$$

$\mu_{N^{pk}}$ and $\sigma_{N^{pk}}$ can be estimated by observing historical data. The Lower Control Limit (LCL) and Upper Control Limit (UCL) for the EWMA control chart are:

$$UCL_S = \mu_S + L \cdot \sigma_S, \qquad LCL_S = \mu_S - L \cdot \sigma_S. \tag{5}$$

For a 5% significance level, L = 1.96. If S_t drifts outside UCL and LCL then an anomaly is detected, and the controller overrides the forwarding rules for the violating flows with DROP entries. To summarize, the controller has following responsibilities:

1) It coordinates the mutations in the network.
2) It manages connections between hosts by installing appropriate flow-rules in the SD-IXP switch.
3) Using FLOW_STATS messages, it periodically obtains traffic statistics from the SD-IXP switch to detect and mitigate DoS attacks.

A DROP entry has only IDLE_TIMEOUT, which means that the DROP rules for the violating flows persist in the switch until such flows become idle for the specified IDLE_TIMEOUT.

V. Evaluation

In this section, we explain the details of our experiment setup followed by results and their analysis.

A. *Experiment Setup*

We evaluated our framework through emulation. Considering the suggestion in [20] to build our prototype as reliable and realistic as possible. Figure 3 shows the evaluated network topology. The network consists of three ASs, namely, AS1, AS2, and AS3. Each AS has one router, i.e., A1 in AS1, B1 in AS2, and C1 in AS3. Each AS connects to the SD-IXP switch through its router. The SD-IXP controller governs the SD-IXP switch. The route server is based on ExaBGP[2]. The routers run *bgpd* and *zebra* daemons, Quagga[3] routing suite. The network topology is created using MiniNext[4] emulation tool. MiniNext is an extension of Mininet[5] that allows each node in the network to execute a separate version of routing software. The IP addresses 172.0.*.* refer to the interfaces that the SD-IXP controller/route server and the routers use to connect with one another. The IP prefixes with "/24" indicate the IP prefixes that each router announces to its neighboring ASs using BGP. The target server resides in AS1, the malicious and genuine hosts reside in AS2 and AS3.

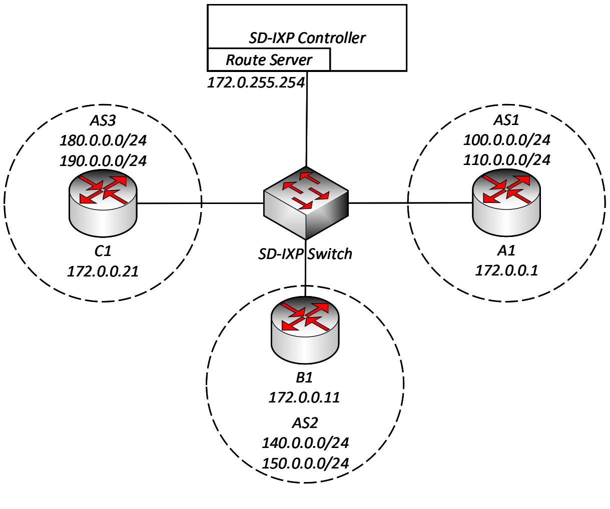


Figure 3: Evaluated network topology

[2] ExaBGP - https://github.com/Exa-Networks/exabgp/
[3] Quagga - http://www.nongnu.org/quagga/
[4] MiniNext - https://github.com/USC-NSL/miniNeXT/
[5] Mininet - http://mininet.org/

Since SDN is a recently emerging concept, no DoS attack dataset for SDN was publicly available at the time of evaluation. To assess the effectiveness of our system we orchestrated two different experiments. In the first experiment, we evaluated the mutation scheme where we used Nmap[6] to imitate an attacker's behavior. Nmap is a free and open-source tool for network mapping and security auditing. Nmap can reveal which hosts are reachable in a network, what services (their name and version) are available on those hosts, which operating system (its name and version) they are running, and several other important information.

To evaluate the versatility and efficiency of our solution for DoS detection and mitigation, we set another experiment. Here, we considered DARPA intrusion detection dataset as well as traffic traces provide by CAIDA. The DARPA dataset contains over 200 instances of more than 50 types of attacks [2]. Table I describes some of the attacks from the DARPA dataset. It is worth mentioning that these attacks are fundamentally different and work at different layers. As a representative example, "Neptune" works at the transport layer while "IPsweep" works at the network layer. On another side, "IPsweep" and "Portscan" can overload an SDN controller by generating a huge number of traffic flows. The CAIDA traffic traces help us to understand the average transmission rate for a genuine source in a network. After following the suggestions in [21], we crafted a 25% attack traffic where the attack traffic has a higher rate than the normal traffic. The network traffic was generated using Scapy[7].

Attacks	Descriptions
Neptune	A flooding of SYN packets on one or many TCP ports.
IPsweep	A surveillance sweep to identify which hosts are listening.
Portscan	A surveillance sweep to discover which services/ TCP-ports are open on the target machine.

Table I: Some attacks from DARPA dataset

B. Results and Analysis

In this section, we present and discuss the results from our experiments. With the SDN-based random host-IP mutation scheme enabled, we performed ten experiments where we ran twenty-five consecutive aggressive (OS detection, version detection, script scanning, and traceroute) Nmap scans against the target network. Figure 4 shows the percentage of correct scan reports against the actual report. It is clear that at maximum barely 16% attempts were successful.

Now we discuss the results from the second experiment. We used Detection Rate (DR) and False Alarm Rate (FAR) to assess our solution. DR measures the percentage of correctly detected attacks over all the real attacks and is computed using Eq. (6).

$$DR\ (\%) = \frac{TP}{TP + FN} * 100. \tag{6}$$

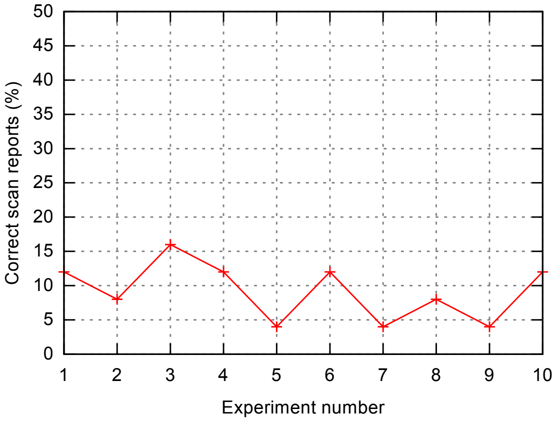


Figure 4: Correctness of consecutive Nmap scans

FAR measures the percentage of benign traffic incorrectly detected as attack over the entire benign traffic and is computed using Eq. (7).

$$FAR\ (\%) = \frac{FP}{FP + TN} * 100. \tag{7}$$

Figure 5 shows DR and FAR for various values of α. A smaller value of α gives more weightage to historical values as compared to the current observation. Hence, the attacks are misclassified leading to lower DR. On the another side, the misclassified attacks still influence the EMWA control chart values, which possibly leads to misclassification of subsequent high-intensity benign traffic. Hence, higher FAR. With the increasing value of α DR and FAR improves. Our results show that our mechanism can perfectly detect the attacks with FAR as low as 7.1%.

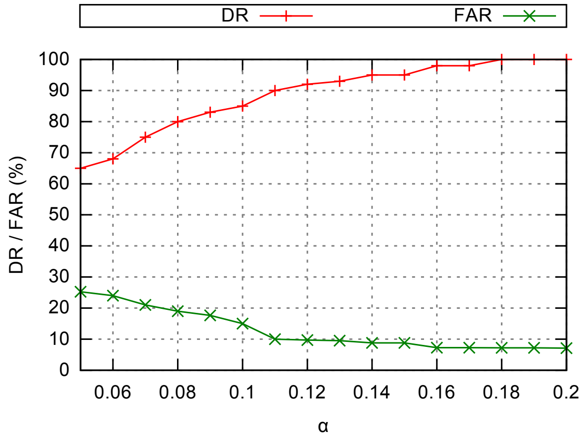


Figure 5: Influence of α on performance

Overheads: We can scrutinize the overheads of our system in terms of the size of address space required for the mutations and CPU usage. To understand the address space requirement, we define two terms M_i and U_i. M_i denotes the rate of mutation for host h_i while U_i defines the time interval during which a virtual IP address must not be reassigned to the same

[6] Nmap - https://nmap.org/

[7] Scapy - http://www.secdev.org/projects/scapy/

host h_i. Considering M_i and U_i, the total number of virtual IP addresses required for any host h_i must be at least $\lceil \frac{U_i+M_i}{M_i} \rceil$. Hence, for a system of n hosts, the least size of address space must be $\sum_{i=1}^{n} \lceil \frac{U_i+M_i}{M_i} \rceil$. For CPU usage, we set an experiment where each host has a different M_i and U_i, and the measured CPU overhead for the controller was less than 5% on a system with Intel Core i5-7200U CPU @ 2.50GHz x 4 processor.

VI. Conclusion and Future Work

SDN provides a simple and flexible network management compared to traditional networks. Applying the concepts of SDN at IXP could make some aspects of wide-area network management easier. In this work, we have proposed an effective framework for the prevention, detection, and mitigation of DoS attacks. As shown by the results, our framework is not only effective to prevent network reconnaissance through moving target defense, but it can also efficiently detect and mitigate DoS attacks. Moreover, our framework has subtle operation overheads. In the future, we will extend our framework to detect and mitigate Distributed Denial of Service (DDoS) attacks. We hope to perform a thorough analysis of our extended framework on a physical testbed.

Acknowledgments

Mauro Conti is supported by a Marie Curie Fellowship funded by the European Commission (agreement PCIG11-GA-2012-321980). Ankit Gangwal is pursuing his Ph.D. with a fellowship for international students funded by Fondazione Cassa di Risparmio di Padova e Rovigo (CARIPARO). This work is partially supported by the EU TagItSmart! Project (agreement H2020-ICT30-2015-688061). This work is partially supported by the grant n. 2017-166478 (3696) from Cisco University Research Program Fund and Silicon Valley Community Foundation, and by the grant "Scalable IoT Management and Key Security Aspects in 5G Systems" from Intel. This work is also partially funded by the project CNR-MOST/Taiwan 2016-17 "Verifiable Data Structure Streaming".

References

[1] The CAIDA UCSD Statistical information for the CAIDA Anonymized Internet Traces, http://www.caida.org/data/passive/passive_trace_statistics.xml, 2016.

[2] MIT Lincoln Laboratory, "Intrusion detection attacks database," http://www.ll.mit.edu/ideval/docs/attackDB.html.

[3] S. Venkatesan, M. Albanese, G. Cybenko, and S. Jajodia, "A moving target defense approach to disrupting stealthy botnets," in *Proceedings of the ACM Workshop on Moving Target Defense*, 2016, pp. 37–46.

[4] L. Ge, W. Yu, D. Shen, G. Chen, K. Pham, E. Blasch, and C. Lu, "Toward effectiveness and agility of network security situational awareness using Moving Target Defense (MTD)," *SPIE*, vol. 9085, p. 90850Q, 2014. [Online]. Available: dx.doi.org/10.1117/12.2050782

[5] J. Rowe, K. N. Levitt, T. Demir, and R. Erbacher, "Artificial diversity as maneuvers in a control theoretic moving target defense," in *National Symposium on Moving Target Research*, 2012.

[6] M. Dunlop, S. Groat, R. Marchany, and J. Tront, "Implementing an IPv6 moving target defense on a live network," in *National Symposium on Moving Target Research*, 2012.

[7] J. Yackoski, H. Bullen, X. Yu, and J. Li, "Applying self-shielding dynamics to the network architecture," in *Moving Target Defense II*. Springer, 2013, pp. 97–115.

[8] R. Colbaugh and K. Glass, "Predictive moving target defense," in *National Symposium on Moving Target Research*, 2012.

[9] A. Gupta, L. Vanbever, M. Shahbaz, S. P. Donovan, B. Schlinker, N. Feamster, J. Rexford, S. Shenker, R. Clark, and E. Katz-Bassett, "SDX: A software defined internet exchange," *ACM SIGCOMM Computer Communication Review*, vol. 44, no. 4, pp. 551–562, 2015.

[10] M. Conti, F. De Gaspari, and L. V. Mancini, "Know your enemy: Stealth configuration-information gathering in SDN," in *GPC*, 2017, pp. 386–401.

[11] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "OpenFlow random host mutation: Transparent moving target defense using software defined networking," in *ACM SIGCOMM Workshop on HotSDN*, 2012, pp. 127–132.

[12] J. H. Jafarian, E. AlShaer, and Q. Duan, "On the random route mutation moving target defense," in *National Symposium on Moving Target Research*, 2012.

[13] P. Kampanakis, H. Perros, and T. Beyene, "SDN-based solutions for moving target defense network protection," in *IEEE WoWMoM*, 2014, pp. 1–6.

[14] D. C. MacFarland and C. A. Shue, "The SDN shuffle: Creating a moving target defense using host-based software defined networking," in *the 2nd ACM Workshop on Moving Target Defense*, 2015, pp. 37–41.

[15] D. Ma, Z. Xu, and D. Lin, "Defending blind DDoS attack on SDN based on moving target defense," in *SecureComm*. Springer, 2014, pp. 463–480.

[16] A. G. Tartakovsky, A. S. Polunchenko, and G. Sokolov, "Efficient computer network anomaly detection by changepoint detection methods," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 1, pp. 4–11, 2013.

[17] T. P. Ryan, "Statistical methods for quality improvement," *John Wiley and Sons, New York*, 1989.

[18] J. S. Hunter, "The exponentially weighted moving average," *Journal of Quality Technology*, vol. 18, no. 4, pp. 203–210, 1986.

[19] D. C. Montgomery, "Introduction to statistical quality control," *John Wiley & Sons, New York*, 1997.

[20] J. Mirkovic, S. Fahmy, P. Reiher, and R. K. Thomas, "How to test DoS defenses," in *Cybersecurity Applications & Technology Conference for Homeland Security*, 2009, pp. 103–117.

[21] J. Sommers, V. Yegneswaran, and P. Barford, "Toward comprehensive traffic generation for online IDS evaluation," *Technical Report, University of Wisconsin*, 2005.