DOI: 10.3969/j.issn.1673-5188.2017.03.003
http://kns.cnki.net/kcms/detail/34.1294.TN.20170912.1054.002.html, published online September 12, 2017

DDoS Attack in Software Defined Networks: A Survey

XU Xiaoqiong, YU Hongfang, and YANG Kun
(School of Communication & Information Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract

Distributed Denial of Service (DDoS) attacks have been one of the most destructive threats to Internet security. By decoupling the network control and data plane, software defined networking (SDN) offers a flexible network management paradigm to solve DDoS attack in traditional networks. However, the centralized nature of SDN is also a potential vulnerability for DDoS attack. In this paper, we first provide some SDN-supported mechanisms against DDoS attack in traditional networks. A systematic review of various SDN-self DDoS threats are then presented as well as the existing literatures on quickly DDoS detection and defense in SDN. Finally, some promising research directions in this field are introduced.

Keywords

software defined networks; SDN security; DDoS; detection method; defense mechanism

1 Introduction

Flooding-based Distributed Denial of Service (DDoS) attacks are the most threatening challenge to Internet security today [1]. The DDoS attacker relies on sending an overwhelming number of fake packets to exhaust the resources of victims, such as CPU, memory and network bandwidth. Therefore, requests from benign users cannot be handled because of unavailable system resources. To cope with this kind of attacks, tremendous mitigation techniques have been proposed [2]–[8]. However, few of them have been extensively implemented because of their deployment complexities as well as prohibitive operational costs. One of the main reasons is that such approaches usually require placing large network connection state tables and high-end equipment at routers or switches and sometimes even requires human intervention, which increases extra storage and computational costs. As a consequence, it is desirable to design some automated, lightweight and scalable DDoS mitigation methods.

Software defined networking (SDN) is a new promise networking paradigm that radically changes the network architecture. The separation of control and data plane in SDN allows us to program the control logic and instruct the forwarding plane to behave accordingly. Furthermore, switches can be made lighter and cheaper since they no longer require computing intelligence to perform control plane processing [9].

However, the centralized control and programmability of SDN introduce new fault and attack points [10], [11]. That is to say, SDN creates new threats that are harder to avoid. For instance, a successful DDoS attack on the SDN controller may cripple the entire network. In this paper, we aim at providing an up-to-date overview of DDoS attack in SDN by presenting detection methods and defense solutions related to individual SDN components, i.e. the controller, switch and data-to-control channel.

This paper is organized as follows. Section 2 describes proposals for DDoS attack in tradition networks addressed by the concepts of SDN and introduces SDN-self DDoS attack challenges. In Section 3, attack detection methods for SDN-self DDoS attack are presented. DDoS attack solutions for each of the attack challenges in SDN are discussed in Section 4. The paper is concluded in Section 5.

2 SDN-Supported vs. SDN-Self DDOS Attack

SDN security, especially DDoS attack, has become a popular research field since software defined network was proposed. There is a contradictory relationship between SDN and DDoS attack. On the one hand, the characteristics of centralized control and programmability of SDN make it easy to detect and react to DDoS attack in tradition networks, i.e. SDN-supported security. On the other hand, the same centralized structure is

This work is supported in part by the "973" Program of China under Grant No. 2013CB329103, the National Natural Science Foundation of China under Grant No. 61271171 and No. 61401070, National Key Research and Development Program of China No. 2016YFB0800105, and the "863" Program of China under Grant No. 2015AA015702 and No. 2015AA016102.

considered vulnerable. Consequently, SDN itself may be a target of DDoS attacks.

2.1 SDN-Supported DDoS Attack

SDN-supported security uses new techniques in SDN to deal with DDoS attacks in traditional networks, from DDoS detection [12]–[14] to DDoS defense [15]–[17] (**Table 1**).

2.1.1 DDoS Detection

At present, there are many studies to detect DDoS attacks [5]–[8]. Hong Jiang, et al. [5] proposed a two-stage detection strategy which combining superpoints and flow similarity measurement. By describing the behaviors of DDoS flooding attacks with superpoints, the suspicious flows are located in a better detection strategy. In [6], a more sophisticated anomaly-based system was proposed for detecting DDoS attacks. The proposed system is designed to solve the detection problem from the perspective of computer vision. Tan, et al. [7] put forward a DDoS attack detection system that adopts multivariate correlation analysis. Moreover, with correlation analysis, another method to detect DDoS attacks against data centers was present in [8]. However, these proposed methods can be hardly applied in online detection.

The advantages of SDN give some flexible DDoS detection methods. In [12], leveraging the global flow monitoring capability of SDN, a quickly and precisely method was proposed to adaptively balance the coverage and granularity of attack detection. Based on dynamically scaling the range of detected IP addresses, this method can achieve the most granularity IP address monitoring, and complete the victim and attacker location as well. In [13], an SDN framework for data centers named FlowTrApp was proposed, which performs DDoS detection and mitigation using some bounds on two per flow based traffic parameters (flow rate and flow duration). It attempts to detect attack traffics ranging from low rate to high rate as well as long-lived to short-lived attacks using an SDN engine. CloudWatcher [14] uses SDN to build a framework to efficiently monitor services in large and dynamic cloud networks. The framework enables the network administrator to protect their network easily by writing a simple policy script.

▼Table 1. SDN-supported DDoS attack

SDN-supported	Solution	SDN capabilities exploited	Description
DDoS Detection	Sequential& concurrent method [12]	Global monitoring	Scaling the range of detected IP addresses
	FlowTrApp [13]	Traffic analysis	Using some bounds on two per flow based traffic parameters
	CloudWatcher [14]	Programmability	Protect network by writing a simple policy script
DDoS Defense	SDN/NFV security policy [15], [16]	Centralized-control or programmability	Combining SDN and NFV
	Collaborative framework [17]	Centralized-control or programmability	A self-management scheme

DDoS: Distributed Denial of Service SDN: software defined networking
NFV: network function virtualization

2.1.2 DDoS Defense

SDN separates the control plane from the data plane and hence allows the network operator to automatically steer individual flows via a central programmable interface [18]. This allows a fine-grained security policy enforcement and thus improves overall network security.

Hence, using SDN, Fayaz et al. [15] proposed a DDoS defense system named Bohatei. This system is scalable because its resource management algorithm controls the network to avoid control and data plane bottlenecks. In addition, it exploits network function virtualization (NFV) [19] capability to flexibly place the defense virtual machine (VM) resources at the locations where they are needed. In addition to that, based on SDN and NFV, a scalable security solution was provided for enterprise networks with greater flexibility and lower operational costs [16].

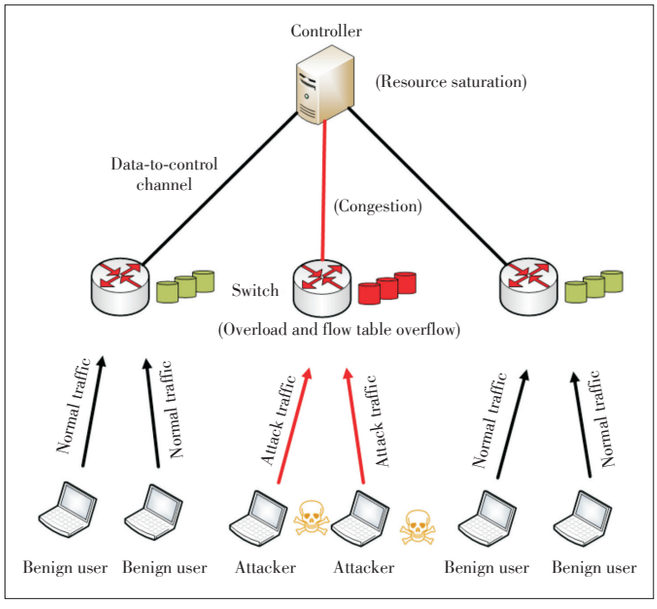
Leveraging the programmability and centralized control offered by SDN, Sahay et al. [17] proposed a self-management scheme, in which an Internet service provider (ISP) and its customers cooperate to mitigate DDoS attack. The ISP collects threat information provided by customers, then it uses this information to enforce security policy and update flow tables in the network accordingly. If a flow is treated legitimate by customers, the ISP controller will mark it with a high priority. Flows with higher priority will get better quality paths.

2.2 SDN-Self DDoS Attack

Despite the advantages of SDN (e.g. programmability, logical centralized control and flexibility) make it easy to detect and defense DDoS attacks in traditional networks, the separation of the control plane from the data plane in SDN introduces new DDoS attack threats. For example, in OpenFlow-based SDN, when a switch receives a new packet, it first checks whether there is an installed flow rule in its Ternary Content Addressable Memory (TCAM) flow table matched this packet or not. If a match is found, the packet is forwarded through the flow rule. Otherwise, the switch buffers this packet and transfers a packet-in message to the controller requesting a new flow rule. The controller then responds with a flow-mod message to instruct all the involved switches with the rules to handle this new packet [20]. An attacker can make use of this characteristic of SDN to launch DDoS attack against the switch, data-to-control channel, and controller, as illustrated in **Fig. 1**.

2.2.1 Switch Overload and Flow Table Overflow

The DDoS attacker can send a large number of table-miss packets to the victim switch. The victim switch should buffer them and generate flow requests sending to the controller since it cannot find matching rules for them. Because of limited re-



▲Figure 1. DDoS attack process in SDN.

sources (CPU and memory), the switch can only generate a limited number of flow requests. Wang et al. [21] show that a hardware switch can only generate less than 1000 requests per second. Thus, the switch may be overloaded, and as a result, flows from benign users may be delayed or dropped. Furthermore, if the controller processed flow requests successfully, a huge number of flow rules should be distributed to the victim switch. Since TCAM is a scarce resource, it only supports a small number of flow rules. For example, the Pronto-Pica8 3290 switch can only hold 2000 rules [22]. Thus, the flow table of victim switch will be filled up quickly and eventually overflow. These two threats in the switch have a local impact as they reduce the throughput of the victim switch.

2.2.2 Data-to-Control Channel Congestion

Following the instructions in 2.2.1, the flow requests flooded by the victim switch are send to the controller though the data-to-control channel, with a lot of bandwidth requirements. In addition, if the buffer of the victim switch fills up, the switch sends an entire packet instead of just a packet header to the controller, resulting in even higher bandwidth consumption. This can overwhelm common bottleneck links, and normal flow requests will experience congestion. The channel congestion affects all hosts with flow requests traversing the congested links.

2.2.3 Controller Resource Saturation

Finally, if the flooded flow requests arrive at the controller, they will consume the controller's resource (i.e. CPU, memory, and bandwidth) for flow rule computation and installation. Without any protection, the controller's resource can be saturated by the flooded requests, and legitimate requests may be dropped. Because the most crucial part of the SDN is the controller, it is also a single point of failure in the entire SDN [23].

3 Attack Detection Methods for SDN-Self DDOS Attack

The most basic but essential task in DDoS research is the detection problem. DDoS attack detection is just the first stage for withstanding DDoS attack. Attack detection mechanisms [8]–[11] used for traditional networks can be adopted in SDN [12]–[14]. Due to the centralized control, the detection for controller DDoS attack is slightly different. This section summarizes the new detection methods for controller DDoS, and then classifies the existing detection algorithms.

3.1 Detection for Controller DDoS

Because only enormous packet-in messages may exhaust controller resources, the simplest detection method is once the amount of the packet-in messages exceeds a predetermined threshold, a DDoS attack against the controller is identified. However, this method may lead to a high false detection rate. In order to improve the detection accuracy, some improved detection mechanisms are proposed [24]–[27].

Considering two facts that new flows can trigger packet-in messages to the controller and low-traffic flows are of high-efficiency for such an attack, an efficient detection method for a novel DDoS attack against SDN controllers was designed by measuring vast new low-traffic flow [25]. This method is based on Sequential Probability Ratio Test (SPRT), a statistical tool which has bounded false negative and false positive error rates. Similarly, S. M. Mousavi et al. [24] proposed an early detection method for DDoS attack against the SDN controller. It assumes that the destination IP addresses are almost evenly distributed in the normal flows, while the malicious flows are destined to a small amount of IP hosts. However, these two methods are not available when attackers generate lots of new low-traffic flows with their destination IP addresses evenly distributed.

In order to detect DDoS attacks against the SDN controller, a lightweight method for DDoS attacks detection based on traffic flow features is presented in [26]. The method contains three modules, i.e. Flow Collector, Feature Extractor and Classifier. The Flow Collector module is responsible for periodically requesting flow entries from all flow tables of OpenFlow switches, then the Feature Extractor module receives the collected flows and extracts features that are important to DDoS attack detection. At last, the Classifier module analyzes whether or not the features correspond to attack or legitimate traffic. This study makes use of the flow statistics character of the OpenFlow switch, but it does not take the controller's overhead caused by flow entries collecting process into consideration. In [27], combining with the sampled flow (sFlow) protocol, the authors reduced flow data gathering by sampling and reduced the required communication between the OpenFlow

switches and controller, thus easing the control plane's overload in the large network traffic condition. Moreover, the authors designed a work-wide anomaly mitigation using OpenFlow. However, the authors did not study if the flow sampling may affect the accuracy of anomaly detection.

3.2 Classification of Detection Methods in SDN

According to the detection algorithms used, we can classify the existing detection methods into the machine learning based [26], [28], [29], the entropy based [24], [30] and the graphic based [31], [32].

Machine learning based techniques for handling DDoS attacks have received much attention in the computational intelligence community. DDoS attacks can be detected by using the machine learning algorithm that was trained with attack and normal patterns. Braga et al. [26] classified network traffic by using Self-Organizing Maps (SOM) [33]. In [28], the intrusion detection system utilizes SVM classifier to detect DDoS attacks. In [29], different machine learning algorithms such as Naive Bayes, K-Nearest neighbor and K-means were used in advanced signature-based intrusion detection system (IDS) to find the sets of hosts that have normal or anomalous behavior. However, the machine learning based methods require a large number of training sets and spend a long time for training.

The entropy-based detection mechanism has a relative low calculation overload. The entropy is used to measure the randomness change of the incoming flows during a given time period. Moreover, the flow-based feature of SDN makes it more convenient to calculate the entropy value. Based on the entropy variation of destination IP address of the incoming packets, an early detection method was proposed in [24] for DDoS attacks against the SDN controller. Moreover, an entropy-based anomaly detection model for DDoS flooding attack in SDN is present in [30]. Differing from the mentioned other methods, this detection algorithm runs in the OpenFlow edge switch. By doing statistics and analysis on the network traffic coming to the OpenFlow network, it achieves detecting the attack locally. Although the entropy-based methods are more flexible, they need to combine with other technologies to make threshold determination and multi-element weight assignment.

Among the graphic model based detection methods, a graphic model based on an attack detection method that deals with a dataset shift problem was proposed by Wang et al. [31]. It saves known traffic patterns as a relational graph. If new traffic is generated, the system can determine whether the traffic is malicious by comparing the graphs. SPHINX [32] was proposed to detect both known and potentially unknown attacks on network topology and data plane forwarding originating within a software defined network. SPHINX leverages the novel abstraction of flow graphs, which closely approximates the actual network operations. It enables incremental validation of all network updates and constraints. SPHINX dynamically learns new network behavior and raises alerts when it detects suspicious changes to the existing network control plane behavior. Graphic models are an effective tool to validate normal and abnormal network behavior. However, if the network topology dynamically changes very frequently, several of the learned invariants may be interrupted, resulting in false detection.

3.3 Others

The abovementioned methods only consider a single IDS in software defined networks. With multiple IDSs, the detection performance highly depends on the way by which the suspicious traffic flows are distributed among the multiple IDSs. In [34], considering the detection of malicious attacks against SDN with multiple IDSs, the proposed algorithm distributes the flows to multiple IDSs according to their routing paths. If two flows are close to each other in terms of the routing path, they are forwarded to the same IDS. Moreover, it uses a gravity clustering algorithm to group the flows, and the cluster size is inversely proportional to the sum of the data rates in each group for load balancing.

4 Defense Mechanisms for SDN-Self DDOS Attack

Once a DDoS attack is detected, a timely and effective defense method is required to restore the network function and reduce the network loss. In this section, we summarize the corresponding defensive measures against the three types of DDoS attacks mentioned in Section 2.2 (**Table 2**).

4.1 Reacting Against Switch Overload and Flow Table Overflow

To mitigate such DDoS threats, Dao et al. [35] present a so-

▼**Table 2. An overview of DDoS countermeasures in SDN system**

Defense techniques	DDoS treats		
	Switch overload	Channel congestion	Controller resource saturation
IP filtering [35]	√		
Scotch [21]	√		
Lightweight [36]		√	
FlowSec [37]		√	
FloodDefender [44]	√	√	√
MLFQ [38]			√
FRESCO [45]	√	√	√
FloodGuard [40]			√
FlowRanger [39]			√
Avant-Guard [42]			√
SDNShield [41]			√
SDN-Guard [43]	√	√	√

DDoS: Distributed Denial of Service SDN: software defined networking
MLFQ: Multilayer Fair Queueing

lution to protect software defined networks based on IP filtering technique. The proposed scheme analyzes user behavior and uses it to assign the timeouts for the flow entries. Short timeouts are assigned for malicious users flows and long timeouts are used for trusted ones. This solution forces entries of malicious traffic to be quickly removed TCAM tables of the switches. However, this may lead to new packet-in messages to be sent to the controller if the flow duration is higher than the set timeout. Furthermore, this solution drops all malicious traffic, which may be problematic for false positive flows.

Scotch [21] uses an overlay network of software switches as a complement to hardware switches. Since software switches can run on more powerful CPUs, they can generate many more flow requests compared to hardware switches. New flows received by hardware switches would be redirected to software switches, which are responsible for generating flow requests. Data plane traffic can still be forwarded by hardware switches for large throughput. Indeed, Scotch can increase the number of new flows that a switch can handle in benign settings; however, it may not be enough for adversarial settings, where an attacker can flood at a rate even higher than a software switch can handle.

4.2 Reacting Against Data-to-Control Channel Congestion

In [36], a lightweight information hiding authentication mechanism was proposed to prevent DDoS attacks in the SDN control channel. In [37], by enforcing a rate limit on the number of packets sent to the controller, FlowSec was introduced to mitigate an attack on the controller bandwidth. FlowSec collects the switch statistics and computes controller bandwidth dynamically. If there is an attack, FlowSec uses the Floodlight module to collect switch statistics and instructs the switch port to slow down. Although this method can mitigate DDoS attacks in the SDN system, it also hinders other switches and normal traffic.

4.3 Reacting Against Controller Resource Saturation

To our understanding, the most vulnerable component in the SDN architecture is the centralized controller. As a result, in recent years, researchers have proposed a variety of strategies to mitigate controller resource saturation attacks in software defined networks. Among them, P. Zhang et al. [38] proposed a novel queue management method that allows dynamic queue expansion and aggregation named Multilayer Fair Queueing (MLFQ). This method is based on enforcing fair sharing of a controller's resources among switches and hosts in the network. When attacks take place, the controller expands the corresponding queues into multiple lower-level queues to isolate flooded requests. In this way, the controller in general only needs to maintain a small number of queues. Despite its advantages, when the number of attack streams is large, this approach is poorly handled.

L. Wei, et al. proposed FlowRanger [39], a flow prioritizing algorithm which is implemented at the controller side to enhance the Quality of Service (QoS) of regular users. In this scheme, a ranking algorithm is first used to identify regular normal users based on their past requests to the controller. Then, the execution of requests prioritized by using multiple priority buffers. Finally, the packets are processed according to a weighted Round Robin strategy, i.e. the packets in a higher priority buffer are handled with higher priority than those in lower priority buffers. FlowRanger can reduce the impact of DDoS attacks on network performance by guaranteeing that legitimate flows are served first in the controller. However, the flows satisfying these criteria are not necessarily malicious. They may also be benign flows that happen to appear with the attack traffic for their first visit. Therefore, simply blocking these requests is not a good solution.

FloodGuard [40] can defend against general flow request flooding attacks. Once the controller detects an attack, it installs a default rule at the victim switch to redirect all new flows to a data plane cache. The data plane cache is responsible for generating flow requests to the controller. At the same time, the controller proactively generates rules by symbolically executing controller applications, and installs these rules at the victim switch to suppress future flow requests. One possible problem with FloodGuard is that symbolic execution may not exhaust all possible execution paths for complicated controller applications. In addition, FloodGuard needs to deploy extra devices on the data plane.

SDNShield [41], a combined solution towards more comprehensive defense against DDoS attacks on SDN control plane. It uses specialized software boxes to improve the scalability of ingress SDN switches to accommodate the control plane workload surge. It further incorporates a two-stage filtering scheme to protect the centralized controller. It statistically distinguishes legitimate flows from forged ones at the first stage, and recovers the false positives of the first stage with in-depth TCP handshake verification at the second stage.

Avant-Guard [42] is an extension to the existing OpenFlow data plane with the addition of Connection Migration (CM). The Avant-Guard responds to handshake packets if no matching flow entries are found. Only when a connection is established, the packet is sent to the controller to ask for a routing path. The purpose of Avant-Guard is to fight against DDoS attacks based on IP spoofing, by effectively reducing the amount of data to the control plane under DDoS attacks. However, a weakness of Avant-Guard is its implementation on switches. All switches need to be Avant-Guard equipped, otherwise the entire network is still vulnerable.

4.4 Others

In [43], a novel SDN application named SDN-Guard is proposed to protect SDN system against DDoS attacks, and simultaneously mitigate DDoS impact on the SDN controller, data-to-control bandwidth and switch. It can dynamically manage flow

routes, rule entry timeouts and the aggregate flow rule entries based on the flow threat probability provided by an IDS.

FloodDefender [44] is a scalable and protocol-independent defense system for protecting OpenFlow networks against SDN-aimed DDoS attacks. It consists of four functional modules: the attack detection, table-miss engineering, packet filter, and flow rule management modules. When no attacks are detected, FloodDefender forwards the packet in messages, actions, and flow rules between the controller platform and controller apps. When attacks occur, FloodDefender detours table-missing packets to neighbor switches with wildcard flow rules to protect the communication link from being jammed, filters out attack packets from the received packet in messages to save the computational resources, and constructs a robust flow table in the data plane by separating the flow table into "flow table region" and "cache region" to save the TCAM of OpenFlow switches.

In [45], an OpenFlow security application development framework FRESCO is proposed. As an OpenFlow application, it offers a programming framework that enables security researchers to implement, share, and compose many different security modules and also exports a scripting API that enables security practitioners to code security monitoring and threat detection logic as modular libraries.

5 Conclusions

The emergence of SDN provides a new paradigm to solve DDoS problem in traditional networks by introducing separate layers for routing and data forwarding. At the mean time, SDN DDoS threat has become an open research field for researchers. In this article, we summarize how a traditional network can incorporate the concept of SDN to solve the issue of DDoS attacks. Then we describe SDN-self DDoS attacks followed by a comprehensive survey of proposed detection methods and defense countermeasures.

Although many methods and systems have been developed by the research community, there are still many open research issues that are not well investigated and need to be addressed by future research efforts. For detection, controller modules often aggregate flow rules to conserve switch TCAM. After the flow table is compressed, the switch reports coarse-grained statistics. How to effectively detect attacks in SDN networks with flow table compression is a problem. Meanwhile, in a software defined network with multi-controllers, how to design efficient detection algorithms to balance the overhead and detection accuracy is another problem. For defense, combining with many other promising technologies in next-generation networks, such as NFV and Information Centric Networking (ICN), may bring in some research opportunities. Furthermore, most of the existing mitigation methods only handle abnormal flows, such as discarding or limiting ones. There is barely a complete method to resolve the problems from the attack source. The location of attack sources and victim hosts is also a relatively new research point.

Acknowledgment

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions.

References

[1] S. Stuart, "Akamai releases prolexic Q2 2014 global DDoS attack report," *Database & Network Journal*, vlo. 44, no. 4, Aug. 2014.

[2] C. Jin, H. Wang, and K. G. Shin, "Hop-count filtering: an effective defense against spoofed DDoS traffic," in *Proc. ACM Conference on Computer and Communications Security* (CCS 03), Washington, USA, Oct. 2003.

[3] Z. S. Taghavi, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 4, pp. 2046–2069, 2013. doi: 10.1109/SURV.2013.031413.00127.

[4] Q. Liao, D. A. Cieslak, A. D. Striegel, and N. V. Chawla, "Using selective, short-term memory to improve resilience against DDoS exhaustion attacks," *Security and Communication Networks*, vol. 1, no. 4, pp. 287–299,2008. doi: 10.1002/sec.22.

[5] H. Jiang, S. Chen, H. Hu, and M. Zhang, "Superpoint-based detection against distributed denial of service (DDoS) flooding attacks," in *IEEE International Workshop on Local and Metropolitan Area Networks*, Beijing, China, 2015, pp. 1–6. doi: 10.1109/LANMAN.2015.7114724.

[6] Z. Tan, A. Jamdagni, X. He, et al., "Detection of denial-of-service attacks based on computer vision techniques," *IEEE Transactions on Computers*, vo. 64, no. 9, pp. 2519–2533, 2015. doi: 10.1109/TC.2014.2375218.

[7] Z. Tan, A. Jamdagni, X. He, P. Nanda, and R. P. Liu, "A system for denial-of-service attack detection based on multivariate correlation analysis," *IEEE Transactions on Parallel & Distributed Systems*, vol. 25, no. 2, pp. 447–456, 2014. doi: 10.1109/TPDS.2013.146.

[8] P. Xiao, W. Qu, H. Qi, and Z. Li, "Detecting DDoS attacks against data center with correlation analysis," *Computer Communications*, vol. 67, no. C, pp. 66–74, 2015. doi: 10.1016/j.comcom.2015.06.012.

[9] N. McKeown, T. Anderson, H. Balakrishnan, et al., "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008. doi: 10.1145/1355734.1355746

[10] M. Antikainen, T. Aura, and M. Särelä, "Spook in your network: attacking an SDN with a compromised OpenFlow switch," in *Proc. 19th Nordic Conference on Secure IT Systems (NordSec14)*, Tromsø, Norway, Oct. 2014. doi: 10.1007/978-3-319-11599-3_14.

[11] D. Kreutz, F. M. Ramos, and P. Verissimo, "Towards secure and dependable software-defined networks," in *Proc. Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking, ser. HotSDN '13*. New York, USA, 2013, pp. 55–60. doi: 10.1145/2491185.2491199.

[12] X. Yang and Y. Liu, "DDoS attack detection under SDN context," in *Proc. IEEE International Conference on Computer Communications IEEE(INFOCOM16)*, San Francisco, USA, 2016. doi: 10.1109/INFOCOM.2016.7524500.

[13] B. Chaitanya and N. Medhi, "FlowTrApp: an SDN based architecture for DDoS attack detection and mitigation in data centers," in *Proc. 3rd International Conference on Signal Processing and Integrated Networks*, Noida, India, 2016. doi: 10.1109/SPIN.2016.7566750.

[14] S. Shin and G. Gu, "CloudWatcher: network security monitoring using OpenFlow in dynamic cloud networks," in *IEEE International Conference on Network Protocols*, Austin, USA, 2012, pp. 1–6. doi: 10.1109/ICNP.2012.6459946.

[15] S. K. Fayaz, Y. Tobioka, V. Sekar, and M. Bailey, "Bohatei: flexible and elastic DDoS defense," in *24th Usenix Conference on Security Symposium*, Washington, D. C., USA, 2015, pp. 817–832.

[16] C. Lorenz, D. Hock, J. Scherer, et al., "An SDN/NFV-enabled enterprise network architecture offering fine-grained security policy enforcement," *IEEE Communications Magazine*, 2017, vol. 55, no. 3, pp. 217–223. doi: 10.1109/MCOM.2017.1600414CM.

[17] R. Sahay, et al. "Towards Autonomic DDoS Mitigation using Software Defined Networking." *NDSS Workshop on Security of Emerging networking Technologies*, 2015. doi: 10.14722/sent.2015.23004.

[18] M. Jarschel, T. Zinner, T. Hossfeld, P. Tran-Gia, and W. Kellerer, "Interfaces, attributes, and use cases: a compass for SDN," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 210–217, 2014. doi: 10.1109/MCOM.2014.6829966.
[19] Q. Duan, N. Ansari, and M. Toy. "Software-defined network virtualization: an architectural framework for integrating SDN and NFV for service provisioning in future networks," *IEEE Network*, vol. 30, no. 5, pp. 10–16, 2016. doi: 10.1109/MNET.2016.7579021.
[20] S. Shin and G. Gu, "Attacking software-defined networks: a first feasibility study," in *ACM SIGCOMM Workshop Hot Topics Software Defined Network (HotSDN13)*, Hong Kong, China, 2013, pp. 165–166.
[21] A. Wang, Y. Guo, F. Hao, T. V. Lakshman, and S. Chen, "Scotch: elastically scaling up SDN control-plane using vSwitch based overlay," in *ACM International Conference on Emerging NETWORKING Experiments and Technologies*, Sydney, Australia, 2014, pp. 403–414. doi: 10.1145/2674005.2675002.
[22] N. Katta, O. Alipourfard, J. Rexford, and D. Walker, "CacheFlow: dependency-aware rule-caching for software-defined networks," in *ACM Symposium on SDN Research*, Santa Clara, USA, 2016, article no. 6. doi: 10.1145/2890955.2890969.
[23] 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 & Tutorials*, vol. 18, no.1, pp. 602– 622, 2016. doi: 10.1109/COMST.2015.2487361.
[24] S. M. Mousavi and M. St-Hilaire. "Early detection of DDoS attacks against SDN controllers," in *Proc. IEEE International Conference on Computing, Networking and Communications*, Anaheim, USA, 2015, pp. 77–81. doi: 10.1109/ICCNC.2015.7069319.
[25] P. Dong, X. Du, H. Zhang, and T. Xu, "A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows," *IEEE International Conference on Communications*, Kuala Lumpur, Malaysia, 2016, pp. 1–6. doi: 10.1109/ICC.2016.7510992.
[26] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *35th Annual IEEE Conference on Local Computer Networks*, Denver, USA, 2011, pp. 408–415. doi: 10.1109/LCN.2010.5735752.
[27] K. Giotis, C. Argyropoulos, G. Androulidakis, D. Kalogeras, and V. Maglaris, "Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments," *Computer Networks*, vol. 62, no. 5, pp. 122–136, 2014. doi: 10.1016/j.bjp.2013.10.014.
[28] R. T. Kokila, S. T. Selvi, and K. Govindarajan, "DDoS detection and analysis in SDN-based environment using support vector machine classifier," in *Proc. IEEE Sixth International Conference on Advanced Computing*, Chennai, India, 2015. doi: 10.1109/ICoAC.2014.7229711.
[29] L. Barki, A. Shidling, and N. Meti, "Detection of distributed denial of service attacks in software defined networks," in *Proc. IEEE International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Jaipur, India, 2016, pp. 2576 – 2581. doi: 10.1109/ICACCI.2016.7732445.
[30] R. Wang, Z. Jia, and L. Ju, "An entropy-based distributed DDoS detection mechanism in software-defined networking," *IEEE Trustcom/BigDataSE/ISPA*, Helsinki, Finland, 2015, pp. 310–317. doi: 10.1109/Trustcom.2015.389.
[31] B. Wang, Y. Zheng, W. Lou, and Y. T. Hou, "DDoS attack protection in the era of cloud computing and software-defined networking," *Computer Networks*, vol. 81, pp. 308–319, 2015. doi: 10.1109/ICNP.2014.99.
[32] M. Dhawan, R. Poddar, K. Mahajan, and V. Mann, "SPHINX: detecting security attacks in software-defined networks," in *Network and Distributed System Security Symposium*, San Diego, USA, 2015. doi: 10.14722/ndss.2015.23064.
[33] S. Ostermann, B. Tjaden, and M. Ramadas, "Detecting anamalous network traffic with self-organizing maps," in *Proc. 6th International Workshop on Recent Advances in Intrusion Detection*, Pittsburgh, USA, 2003, pp. 36– 54. doi: 10.1007/978-3-540-45248-5_3.
[34] T. Ha, S. Yoon, A. C. Risdianto, J. Kim, and H. Lim, "Suspicious flow forwarding for multiple intrusion detection systems on software defined networks," *IEEE Network*, vol. 30, pp. 6, pp. 22–27, 2016. doi: 10.1109/MNET.2016.1600106NM.
[35] N. N. Dao, J. Park, M. Park, and S. Cho, "A feasible method to combat against DDoS attack in SDN network," in *International Conference on Information Networking*, Cambodia, Cambodia, 2015, pp. 309–311. doi: 10.1109/ICOIN.2015.7057902.
[36] O. I. Abdullaziz, Y.-J. Chen, and L.-C. Wang, "Lightweight authentication mechanism for software defined network using information hiding," in *IEEE Global Communications Conference (GLOBECOM)*, Washington, D.C., USA, 2016. doi: 10.1109/GLOCOM.2016.7841954.
[37] M. Kuerban, Y. Tian, Q. Yang, et al., "FlowSec: DOS attack mitigation strategy on SDN controller," in *IEEE International Conference on Networking, Architecture and Storage*, Long Beach, USA, 2016, pp. 1– 2. doi: 10.1109/NAS.2016.7549402.
[38] P. Zhang, H. Wang, C. Hu, and C. Lin, "On denial of service attacks in software defined networks," *IEEE Network Magazine*, vol. 30, no. 6, pp. 28–33, 2016. doi: 10.1109/MNET.2016.1600109NM.
[39] L. Wei and C. Fung, "FlowRanger: a request prioritizing algorithm for controller DoS attacks in software defined networks," in *IEEE International Conference on Communications*, London, UK, 2015, pp. 5254– 5259. doi: 10.1109/ICC.2015.7249158.
[40] H. Wang, L. Xu, and G. Gu. "FloodGuard: a DoS attack prevention extension in software-defined networks," in *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks*, Rio de Janeiro, Brazil, 2015, pp. 239–250. doi: 10.1109/DSN.2015.27.
[41] K. Chen, A. R. Junuthula, I. K. Siddhrau, Y. Xu, and H. J. Chao, "SDNShield: towards more comprehensive defense against DDoS attacks on SDN control plane," in *Proc. IEEE Conference on Communications and Network Security (CNS)*, Philadelphia, USA, 2016. doi: 10.1109/TPDS.2013.146.
[42] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks," in *ACM Sigsac Conference on Computer & Communications Security*, Berlin, Germany, 2013, pp. 413–424. doi: 10.1145/2508859.2516684.
[43] L. Dridi and M. F. Zhani, "SDN-guard: DoS attacks mitigation in SDN networks," in *IEEE International Conference on Cloud Networking IEEE*, Pisa, Italy, 2016. doi: 10.1109/CloudNet.2016.9.
[44] S. Gao, Z. Peng, B. Xiao, A. Hu, and K. Ren, "FloodDefender: protecting data and control plane resources under SDN-aimed DoS attacks," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, Atlanta, USA, 2017, pp. 1–9.
[45] S. Shin, P. Porras, V. Yegneswaran, et al., "FRESCO: modular composable security services for software defined networks," *In Proc. Network & Distributed Security Symposium*, San Diego, USA, 2013, pp. 319–332.

Manuscript received: 2017-06-03

Biographies

XU Xiaoqiong (xiaoqiongxu@std.uestc.edu.cn) is currently a Ph.D. student in University of Electronic Science and Technology of China, China. Her research interests include software defined networking and cloud computing.

YU Hongfang (yuhf@uestc.edu.cn) received her B.Sc. degree in electrical engineering in 1996 from Xidian University, China her M.Sc. degree and Ph.D. degree in communication and information engineering in 1999 and 2006 from University of Electronic Science and Technology of China, respectively. From 2009 to 2010, she was a visiting scholar at the Department of Computer Science and Engineering, University at Buffalo (SUNY), USA. Her research interests include network survivability and next generation Internet, and cloud computing.

YANG Kun (kunyang@uestc.edu.cn) received his Ph.D. from the Department of Electronic & Electrical Engineering of University College London (UCL), UK, and M.Sc. and B.Sc. from the Computer Science Department of Jilin University, China. He is currently a Chair Professor in the School of Computer Science & Electronic Engineering, University of Essex, leading the Network Convergence Laboratory (NCL), UK. He is also an affiliated professor at University of Electronic Science and Technology of China, China. Before joining in University of Essex at 2003, he worked at UCL on several European Union (EU) research projects for several years. His main research interests include wireless networks and communications, future Internet technology and network virtualization, mobile cloud computing. He manages research projects funded by various sources such as UK EPSRC, EU FP7/H2020 and industries. He has published 100+ journal papers. He serves on the editorial boards of both IEEE and non-IEEE journals. He is a senior member of IEEE (since 2008) and a Fellow of IET (since 2009).