Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2020) 28: 757 – 772

doi:10.3906/elk-1904-179

An Inter-Domain Attack Mitigating Solution

Gökhan AKIN[1,*], Ozan BÜK[2] Erdem UÇAR[1]
[1]Department of Computer Engineering, Institute of Science, Trakya University, Edirne, Turkey
[2]Satellite Communication and Remote Sensing Program, Information Institute, İstanbul Technical University, İstanbul, Turkey

Received: 25.04.2019 • **Accepted/Published Online:** 27.12.2019 • **Final Version:** 28.03.2020

Abstract: Online services on the Internet are increasing day by day, and in parallel, the number of cyber-attacks is rapidly increasing. These attacks are not always about data theft, but they can cause severe damage by denial of service attacks. Intrusion Prevention System products that many organizations use at the border of their enterprise networks are not strong enough to protect against DoS attacks. The typical way to mitigate such attacks is to get support from a service provider. However, a service provider only provides solutions for the traffic originating from itself. If the source of attack is in another ISP domain, it is possible to inform that ISP via phone or e-mail. As a result, the source of the attack is blocked by the manual intervention of the service provider whose domain hosts it. Border Gateway Protocol (BGP) based solutions are also available for automating a blocking system, but not all enterprise networks support BGP. In this research, we have developed a centralized automation solution for software defined network (SDN) environments that is capable of preventing cyber-attacks at the source of attack. This solution does not require any BGP support. Non-SDN environments can also use this attack mitigation and notification system. In the long run, we may use this system to create a national protection shield in order to mitigate Cybersecurity attacks.

Key words: Denial of service, Cyber, Attack, Software defined network, Openflow, Flowspec

1. Introduction

With the increase of public and private sector services provided on the Internet, the availability of Internet access has become even more critical. Global Internet access outages cause not only severe financial loss, but also an interruption in public services such as health care, security, and justice systems. [1]

Besides, with the concept of "Internet of Things" (IoT), services provided using the Internet have reached a whole new level. With this new concept, we can create smart cities, intelligent buildings, and smart houses with many physical and virtual devices connected to the Internet. In the long run, the number of such devices will increase and we have to protect them from cybersecurity attacks. [2].

Today, cybersecurity threats are the most serious national security issue. "Cyber-attacks" and "Data fraud or theft" are in the top risks list of the World Economy Forum's 2018 report [3]. One of the most critical cybersecurity threats is "Distributed Denial of Service" (DDoS) attacks which pose a severe threat to the existing Internet infrastructure [4, 5]. Moreover, DDoS attacks can exhaust the network and computer resources almost without any warning [6].

S. T. Zargar et al. classifies DDoS mitigation systems by location. The classes consist of Source-based DDoS Defense Mechanisms (solution at the source of attack), Network-based DDoS Defense Mechanisms (at

*Correspondence: gokhan@agyoneticileri.org

ISP network), Destination-based DDoS Defense Mechanisms (at victim's site) [2]. Blocking volumetric DDoS attacks at the border of a network (destination-based defense) cannot prevent the victim network's service outage. When bandwidth saturation occurs at the victim site, the attacker accomplishes its goal and the victim service becomes unavailable.

ISPs mostly prefer proprietary solutions that cannot drop the attack traffic at the source. Such solutions can mitigate attacks at the directly connected site (network-based defense). As a result, the attack traffic continues to saturate the available bandwidth throughout the traffic path, and the network service may become unavailable. In February 2016, an attack was detected on Turkey's DNS servers. Since Turkish National Research and Education Network Authority (ISP of the Turkish Universities, ULAKNET) dropped the attack, the connection to its peer ISP continued to be saturated. Due to the saturated connection, the domain name servers went out of service and ".tr" sites became inaccessible. This case has shown that individual responses of the ISPs to such cyber-attacks remain inadequate.

Industry-standard mitigation solutions mostly use the BGP protocol (RFC3882, RFC5575, RFC5635). With the BGP protocol, the victim site sends blocking requests to its peer ISP. Upon the receipt of such a request, the network admin manually implements the BGP configuration. This blocking configuration is also manually removed after the attack. However, not all the enterprise network authorities use BGP. Therefore, this solution is not applicable to every network. They often communicate with the ISP via e-mail or telephone and request a solution for DDoS mitigation. This process extends the response time before the actual intervention.

Even in the case that both the ISP and customer use BGP, the ISP cannot detect if the attack exists or not and have to trust the customer's request. With proprietary or BGP based solutions, ISPs can mitigate the attacks, but there is no automatic mechanism to inform the attacker site. Thus, the source which was infected by malware may continue to attack other sites.

Furthermore, these types of solutions mostly focus on DDoS attacks. Such systems do not block single-source or low-volume attacks (mostly targeted applications such as SSH service). Moreover, the amount of these kind of attacks cannot be neglected (see section 2). Since they could be dropped by an intrusion prevention system (IPS) at the victim's site. However in this case the system does not notify the source site and the attack traffic continues.

In this research, we propose an inter-domain attack mitigating solution called Attack Blocking System (ABS) that provides solutions to these problems. With this system, we can share cyber-attack information with other network authorities, validate attack occurrence, inform the network authority which acts as the source of attack, and drop the attack at the closest point to the source. This system can be used even for DDoS attacks and low-volume application-centric attacks. There is no need to use the BGP protocol, and the system can easily work with the Software Defined Network architecture.

2. Background and related work

DDoS protection systems are grouped into two categories: attack detection systems and attack mitigation systems. There are several studies on these subjects. Most of these studies focused on a specific type of traffic. Wang proposed a defense system for DNS infrastructure [7]. Singh et al. worked on HTTP-GET flood DDoS attacks [8]. Kurt et al.'s study was about detecting SIP-based DDoS attacks [9]. Saravanan et al. worked on a system for the detection of application layer DDoS attacks [10]. In addition, several studies have been conducted in attack detection and mitigation techniques on fog [11, 12] and cloud computing [13, 14]. Agrawal et al. and Bhushan et al. worked on low-rate DDoS attack in cloud computing environment [15, 16]. Besides there are

studies on attack detection and prevention in 5G mobile networks [17–19]. Demir et al. proposed an intrusion detection system by combining different classification models, but their study did not have a mitigation system [20]. Patil et al. and Behal et al. worked on DDoS just for early detection [21, 22]. Previous studies have almost exclusively focused on a specific traffic type or have only attack detection mechanism.

Today, the industry-standard mitigation infrastructures which target DDoS attacks work with the BGP Protocol. One of them is the Destination-based Remotely Triggered Black Hole (D/RTBH) solution defined in RFC 3882. In this system, the victim site makes a BGP announcement with community value 666 and triggers the ISP to add a Black Hole route for its server's IP address. The ISP drops the attack traffic, but this causes a service outage. Only after the servers' IP addresses and DNS record are changed, the server can continue its service. This system requires preconfiguration of the discard route on all the edge routers.

Another mitigation solution that works with BGP is the Source-based Remotely Triggered Black Hole (S/RTBH) solution defined in RFC 5635. It is a network-based solution which can stop the attack traffic at the connected ISP, but not at the source of the attack. It requires configuration of the discard route and the source address validation mechanism (Unicast Reverse Path Forwarding) on the edge routers.

Flowspec, which also requires BGP (RFC 5575), is an attack information sharing system between ISPs. However, major ISPs do not prefer sharing these kinds of information with their competitors [23]. Sharing such information is considered a privacy leakage [24]. This system can use access-list rules to drop the traffic and can check if the source IP address is in the routing table or not. All these three solutions are network-based defense mechanisms, and they do not inform the source of the attack. In addition, sites which are not using BGP cannot use these techniques. You can find the differences between these industry-standard mitigation techniques and Attack Blocking System (ABS) in Table 1.

Routers today mostly work as distributed devices and do not have a central decision-making mechanism. One of the first studies that centralize this mechanism is the Route Control Platform (RCP) system that works with BGP [25]. This system focuses on the BGP protocol to centralize forwarding and shorten the convergence time. The OpenFlow protocol, developed later to centralize the network infrastructure, benefits from the existing control plane (network application) and data plane (switching application) architectures of network devices [26].

Control plane operations use a general purpose CPU. The data plane has switching operations that are carried out with ASICs. A new architecture called Software Defined Network (SDN) moves the control plane to an external computer which is called "controller". The OpenFlow protocol transmits flow information between network devices and this controller. OpenFlow controller software can handle multiple network devices. A new plane called application plane is added for customizing the network decisions [27]. With the use of SDN, management and monitoring have become much easier [28–30]. SDN also provides many advanced security solutions [31].

This architecture can be used as an intrusion detection and mitigation solution. One of the studies with SDN is STRIDE, an attack prevention system for DDoS, which operates by assigning trust values from clients' network activity behaviors [32]. Moving Target Defense (MTD) is another OpenFlow (SDN) security solution, which assigns random IP addresses to clients [33]. This system provides protection against reconnaissance attacks. Giotis et al. worked on another security solution with SDN. It is a system that uses the advantages of SDN for easy traffic management, monitoring and firewalling [34]. This system uses sFlow to detect and mitigate the attack. Mantur at al. worked on developing a signature-based firewall and statistical-based network intrusion detection system with the SDN infrastructure [35]. They built a centralized firewall and

Table 1. Mitigation techniques

System	For all Traffic Type	BGP Independent	Works with SDN	Block at the Attacker's ISP (at other AS)	Inform the Attacker Site	Verify Attack Traffic at Source
D/RTBH (RFC 3882)	Yes	No	No	No	No	No
S/RTBH (RFC 5635)	Yes	No	No	No	No	No
Flowspec (RFC 5575)	Yes	No	No	**Yes**	No	No
Wang [7]	For DNS Traffic Only	Yes	No	No	No	No
Singh et al.[9]	For SIP Traffic Only	Yes	No	No	No	No
Saravanan et al. [10]	For Layer 7 Traffic Only	Yes	No	No	No	No
Zhou et al. [11] Priyadarshini et al. [12]	For Fog Computing Only	Yes	No	No	No	No
Yusop et al. [13] Miao et al. [14]	For Cloud Computing Only	Yes	No	No	No	No
Agrawal et al. [15] Bhushan et al. [16]	For Low-rate Traffic at Cloud Computing Only	Yes	No	No	No	No
Serrano et al. [17] Sotelo et al. [18] Serrano et al. [19]	For 5G Mobile Networks Only	Yes	No	No	No	No
[20],[21], [22]	For Detection Only w/o SDN	Yes	No	No	No	No
[32–36]	For Detection Only w/ SDN	Yes	Yes	No	No	No
Wang et al. [37]	Yes (For low density)	Yes	Yes	No	No	No
Sahoo et al. [38] Carvalho et al. [39]	For Data Centers Only	Yes	Yes	No	No	No
Yuan et al. [40] Wang et al.[41]	For Cloud Computing Only	Yes	Yes	No	No	No
Sahay et al. [42]	Yes	Yes	Yes	No	No	No
ABS	Yes	Yes	Yes (Also works w/o SDN)	**Yes**	Yes	Yes

network intrusion detection system which works on the destination site. Joldzic et al. worked on a local DoS detection method which provides scalability with balancing algorithms [36].

Wang et al. created a system called "Woodpecker" for detecting and mitigating low density link flooding attacks via SDN [37]. Sahoo et al., Carvalho et al., and Yuan et al. also worked DDoS attack detection and mitigation system for data centers with SDN [38–40]. Wang et al. studied on cloud computing DDoS attack protection system with software defined networks [41]. ArOMA is a SDN based DDoS mitigation framework [42]. ArOMA can mitigate the attack at the victim site and can also mitigate at the local ISP.

All these security solutions are designed to mitigate mostly low rate or high rate DDoS attacks. Attacks which do not try to denial a service, such as SSH login attempts, are not blocked by such systems. Local IDS / IPS systems can block SSH attempts, but the attack traffic reaches the entrance of the network. Besides, the network administrators at the attacker's site are not informed about this activity. If the source of the attack is an infected PC with malware, it continues its activity. To get an idea of the volume of such attacks, we reviewed all unauthorized ssh request attempts to a server on a Turkish University for 90 days(Jun 14th-Sep 11th, 2019) which has a global SSH service enabled. You can find the details at Figure 1. In ninety days, attackers made 693,597 unauthorized SSH connection attempts. On average 7707 attempts have occured per day.

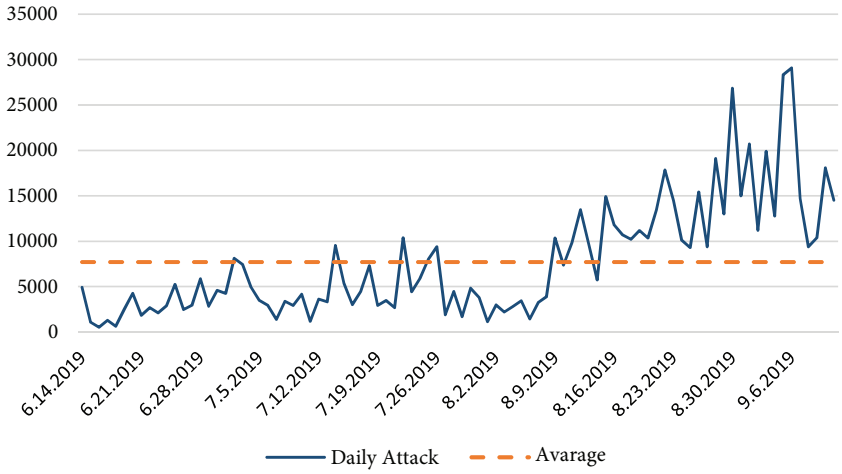

Figure 1. Daily distribution of attacks.

3. Operation and structure of ABS (Attack Blocking System)

3.1. Goals of ABS

The primary objective of this research is to create a communication system between the network authority for the attacker site and that for the victim site and inform these authorities after any detection of attack. With this communication system, the authority for the source originating the attack is informed about the details of the attack, and this authority, after verifying the issue, can drop the attack traffic at a point as close as possible to the source. Unlike any other existing solution, ABS has the following features:

1. The proposed system works independently of any routing protocol. Yet, it can operate alongside any routing protocol, like BGP, etc. That is to say, this system does not require BGP, but the routers that use the BGP protocol can be beneficial in certain aspects. Global AS (Autonomous System) numbers can be used to identify a network authority. Private AS numbers can also be used for the same purpose instead of globally registered AS numbers. The details are provided in Section 4.
2. The attack can be stopped at the source, not at the service provider's network. Unlike the popular technique "Source-based Remotely Triggered Black Hole" (which adds a black hole (Null 0) route to layer 3 devices and drops the victim's server traffic), ABS ensures that the server continues its operation.
3. ABS has a trust level mechanism. With the trust level mechanism, even if the system does not confirm a full-trust relationship between the victim and the attacking site, it still informs the attacker site with

the details of the attack traffic. Hence, the attacker site's ABS can warn the IT staff on a dashboard, via SMS or e-mail.

4. If we integrate this system with the SDN, it can detect whether there is an actual attack or not. After receiving an attack notification, the ABS can block the attack traffic with the help of the SDN controller. This system can also keep monitoring to verify if the threat continues. Once it is confirmed that there is no threat, the system can automatically remove the blocking rules.
5. With the current solutions, the attack information sharing occurs between the enterprise network and its peer ISP with the help of BGP. A victim enterprise network which does not use BGP may not be part of any attack mitigation system currently in use. ABS does not require BGP, and any system can be part of this notification network.
6. Even if an ISP does not join ABS, corporate networks can still be part of this system. The victim site can send alerts to other corporate network authorities if they are identified as a source of attack traffic. The ISP's not being part of the detection and blocking process saves its resources in terms of time and workload.
7. This remote warning system can help reduce future attacks. ABS sends attack information to the enterprise network administrators of the attacker site and informs about the attacker PC which is possibly infected by malware. They can take necessary precautions to make sure that the computers are clean before they are permitted to reaccess the network. In the long run, this system can help to reduce the size of botnets.
8. Existing attack mitigation systems mostly provide solutions to DDoS attacks. We can also use ABS for single-source attack attempts (e.g. SSH easy password attacks). Even if the system admins are not sure if it is an attack or not, with the help of the ABS they can send notifications like "possible attack" to inform the peer ABS.
9. Since the system is very transparent to current network infrastructure, it is suitable for closed network systems like military networks as well.
10. With ABS, attack prevention will no longer be only the ISPs' responsibility. Any network authority can provide feedback to cybersecurity teams.

3.2. Operation and structure of ABS

The ABS attack control system has two goals. The first one is to inform the source of the attack from the victim system. The second goal of the project is to stop this attack at the nearest point to the attacker site. ABS is not an intrusion detection system. It is a distributed attack notification and prevention system. There are many studies and solutions on attack detection with SDN [43]. We can integrate ABS software into an existing intrusion detection system (IDS) such as Snort [44]. ABS can also get information from a honeypot system [45, 46].

With the SDN infrastructure, it is also easy to track packet statistics and detect any anomaly. IDS can use this information to detect attacks and share attack details with the ABS. Thus, if we use ABS in conjunction with SDN, we can get more benefits.

ABS can collect attack information from systems like IDS or Honeypot via an API. The attack information collected by the network administrator can also be specified manually in the system by a custom GUI. In this research, we examined how to prevent those attacks that are manually defined by network admins.

We primarily focused on using ABS software with Software Defined Network infrastructure. Our software is designed to receive related information from the SDN controllers and issue blocking orders. ABS can work with multiple SDN controllers. In other words, an institution may have an ABS with multiple SDN controllers connected to it (Centralized design, Figure 2). This model is most suitable for campus networks.

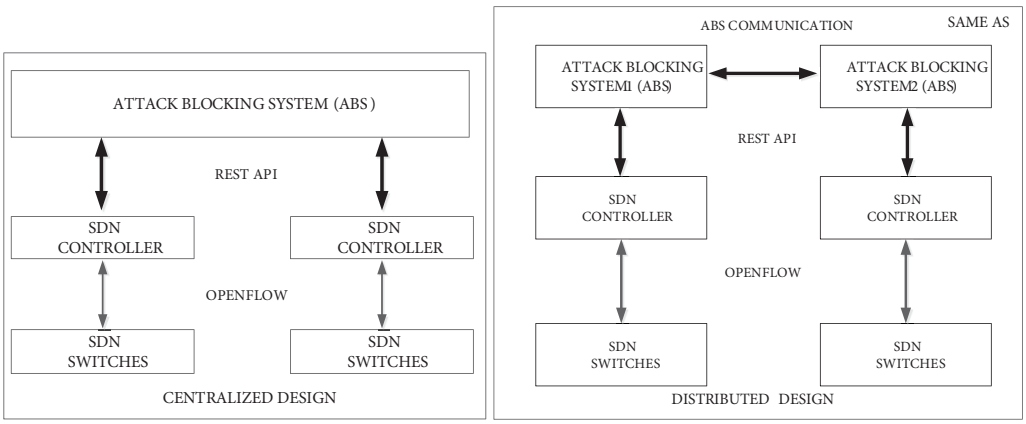

Figure 2. Centralized and distributed design.

It is also possible to have more than one ABS within the same institution or under the same AS. ISPs could prefer this sort of design (Distributed design, Figure 2).

There are four different communication channels in the ABS. The first communication channel is between the ABS software and the system which provides the attack information (IDS, a honeypot or local admin's manual entry, shown as (1) in Figure 3). This channel is for getting the local attack information to local ABS.

The second channel is for communication between two ABS software. After the system receives the attack information, communication starts between ABS systems and local ABS sends attack information to the remote ABS. Details of this communication are given in chapter 4 (Shown as (2) in Figure 3).

Then the remote ABS can verify this attack information with the SDN integrated architecture in the third communication channel. For this purpose, the attack traffic flow is added to the SDN switch by the SDN controller to trace the attack traffic (Figure 4). We use REST API for this communication shown as (3) in Figure 3. In other words, this third communication is between the ABS software and the SDN controller. Later on, the SDN controller collects statistical information related to this flow. If there is matching traffic with the attack flow, the system concludes that the attack continues.

The fourth and the last communication channel uses OpenFlow protocol between the SDN controller and the SDN switch. Even OpenFlow v1.0 is sufficient for this communication channel (Shown as (4) in Figure 3).

Once the system confirms the attack and if the ABS peers form "notification only relationship", the attacker site's ABS only informs the local network admin. If ABS peers have built "full-trust relationship", ABS informs the local network admin and blocks the attack traffic. Then the attacker site's ABS informs the victim site's ABS about the process.

Another important point is to remove this blocking rule after the attack is over. The network admin can remove this blocking rule manually. However, with an SDN-integrated ABS, the system can follow the matching packet traffic statistics from the SDN switch at a specific frequency. Blocking remains as the number of attacking packets continues to increase. If the number of packets for that flow does not increase, it automatically removes the block after a predefined expiration period. With this feature, ABS does not leave any forgotten drop rules.

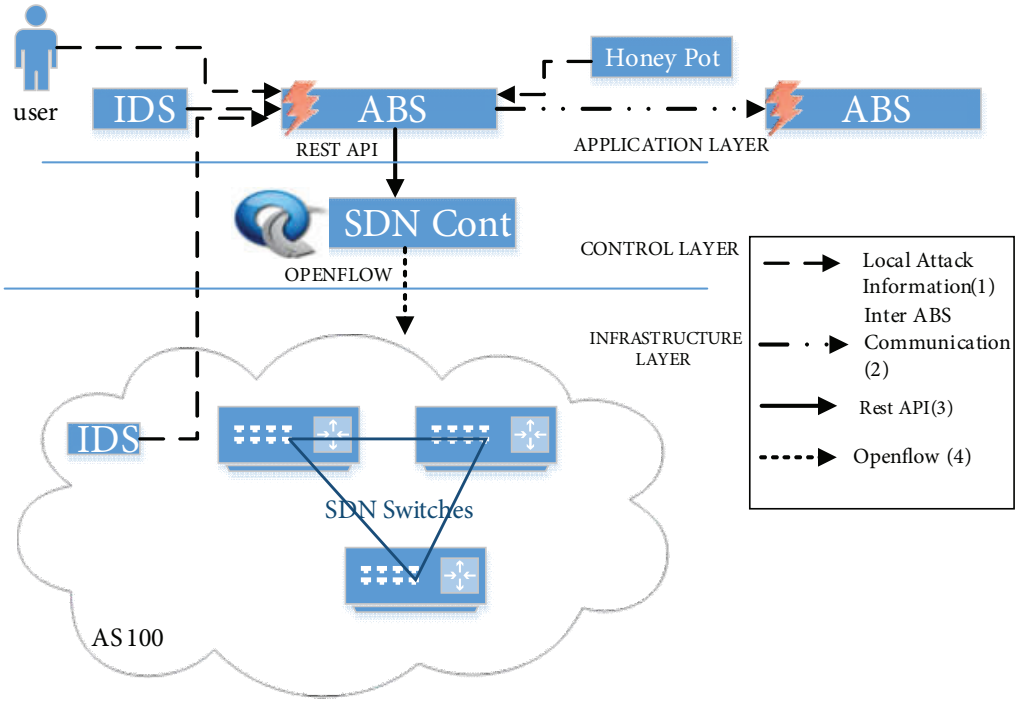

Figure 3. Communication channels in the ABS.

Figure 4. Attack verification.

We can use ABS in a non-SDN system. When it is configured in a non-SDN environment, attack traffic can be manually blocked without verifying the attack. In both cases, the attacker site's network administrator can be aware of the attack and can identify the attacker device. The network admin examines the attacker PC and removes the malware if it is infected. In addition to this, ABS can block attacks with the help of S/RTBH, D/RTBH or Flowspec techniques. However, this requires BGP running systems and pre-configuration of edge routers. In some scenarios, both parties may be an enterprise network which does not use BGP. With ABS, it is not necessary to get support from an ISP. The network authority can take action manually, or if they have an SDN-integrated ABS, it can prevent the attack automatically.

4. Design details of ABS

4.1. System configuration process

Step 1: Basic configuration of ABS system

In an infrastructure with SDN architecture, the SDN controller can block the attack automatically. If the system does not have an infrastructure with SDN architecture, we specify this in the ABS software as a legacy network. The system can then only send notifications instead of blocking. This notification can be configured

to alert network admins via e-mail, SMS, or similar systems like notification dashboards. In this research, we focused mostly on SDN-based attack blocking systems.

In case the SDN-based blocking system is selected, we have defined the details of the SDN controllers in the ABS (IP address and TCP port number of the SDN Controller). Optionally, security credentials can be applied to enable secure communication with the controller and we define IP prefixes that are under the management of the configured controller. After this phase, the ABS can receive data from the controllers and send a blocking request through the REST API.

The system can also be configured to manage multiple SDN controllers within the same organization with a single ABS (Centralized Design). This means multiple SDN controllers can be added to a single ABS server.

Step 2: Defining other ABS systems

We have configured the local ABS with the IP addresses, security credentials, trust levels of peer ABS systems, and blocking mechanisms agreed between peers. A certification-based authentication can be used for the local ABS. There are two trust levels in the system. The first one is the "notification only relationship" which only notifies the peer network authority when an attack alert is received, and the second one is the "full-trust relationship" which does not only notify the neighbor ABS but also enables the peer ABS to block the attack at the source. In this sense, the process of authentication of the ABS systems is highly critical because a rogue request from an unauthorized source can block the legitimate traffic. We recommend using out-of-band communication between ABS systems where communication is critical. Thus, the network communication between the ABS peers does not have to be affected by security issues or high-volume DDoS attacks.

Another information required by the system is the networks/prefixes that are under the management of the peer ABS systems. IP prefix definition can be done with various methods and is a very critical process. An ABS should not issue a blocking order for unauthorized IP prefixes.

The most secure method is manual definition by the local admin of the prefixes that are under the control of other peer ABS systems. Alternatively, each ABS can teach its authorized prefixes to its peer ABS systems. In both cases, ABS can check this information from the global BGP tables for control purposes. However, there cannot be such control if the organization is not using BGP. In this case, the values entered by the authorized persons have to be trusted. Once the ABS peers complete their mutual information exchange, they form a neighborship.

From now on, if ABS detects an attack from a peer prefix which belongs to a peer ABS server, the system can now notify the peer ABS of the attack. The notification message sent to the ABS peer should include the attacker's source IP address, destination IP addresses and message type. We can also include the protocol number, source and destination port numbers in the message. With these details, the remote ABS device can drop the flow more precisely compared to the case where only the IP addresses are used.

In this research, we have used three different message types and additional subcodes in the ABS. The message type and code sections use 4 bytes in total. The first message type is for the "hello" communication used for establishing the neighborship between the ABS systems. Hello messages also control the communication status between them. This message type is defined as Type 0 and has two codes. Code 0.1 is a "hello packet" and Code 0.2 is an "ack packet".

The second message type is defined as Type 1 and has two codes. ABS sends Type 1 messages in case of an attack. Code 1.1 is an "intrusion prevention request" message. Code 1.2 is an "intrusion notification" message sent in case of ABS peers which has formed a "Notification Only Relationship". The last message type

is defined as Type 2 and has four codes. ABS sends Type 2 messages for reply purposes. These are "request confirmed" and "request rejected" packets. You can find message details in Figure 5. In further studies, we can add more message types and code numbers if required.

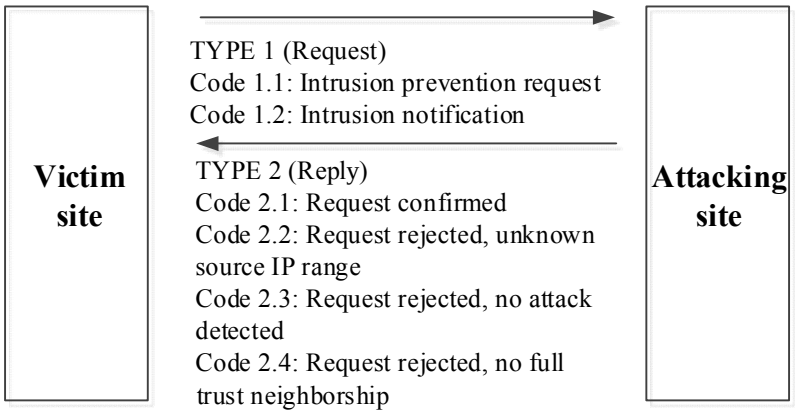

Figure 5. Message types used for communication between ABS servers.

4.2. Flowcharts of the system and software components

The ABS software has 3 different modules for mitigating attacks. First one is the "Attacker Location Identification" module represented with (1) as shown in Figures 6 and 7. This module works after the attack information is received and identifies which ABS the attacker computer belongs to. If the module detects that the source of the attack is a device under the authority of its own ABS, it stops the attack with the help of module two without communicating with the other ABS systems. If the source IP of the attack is not associated with any ABS, the system creates a log and notifies the local system admin.

If the attack traffic comes from a peer ABS, it triggers the second module. The second module of the system is called "Communication Between ABS Peers". This module is responsible for the communication between the ABS systems as defined in Figures 6 and 7. Module two establishes a neighborship between the ABS peers. After this phase of neighborship establishment, each site periodically sends hello packets within the hello interval to verify that the peer ABS is alive. If no hello packet arrives within the hold time interval, the local ABS determines that the communication with the peer ABS has been lost and sends a notification to the system admin. In that case, the ABS falls back to the phase of neighborship establishment.

The other task of the second module is to inform a peer ABS that it has an attacking computer (Figure 6). When the ABS sends an attack notification message to a peer ABS, the peer replies with an ack message and confirms receipt of the attack information. In that phase, it compares its IP prefixes with the attacker's IP address. (Figure 7) If they do not match, the peer system sends a Code2.2: error message "Request rejected, unknown source IP Range". If the attacker's IP address matches the peer systems' IP range, module 3 performs an attack verification. At this step, the current traffic flow is checked to see if there is such traffic defined from the victim ABS. The name of this third module is "Attack Verification and Blocking System".

In Module 3, the ABS sends related flow information with FORWARD action to the SDN Controller, and SDN Controller adds this flow to the switch. Then the packet number of this flow is checked to see whether it is increasing or not. If the relevant traffic does not match the added flow, the local ABS sends a Code 2.3 error message (Request rejected, no attack detected) to the victim ABS. If the traffic matches the flow, the

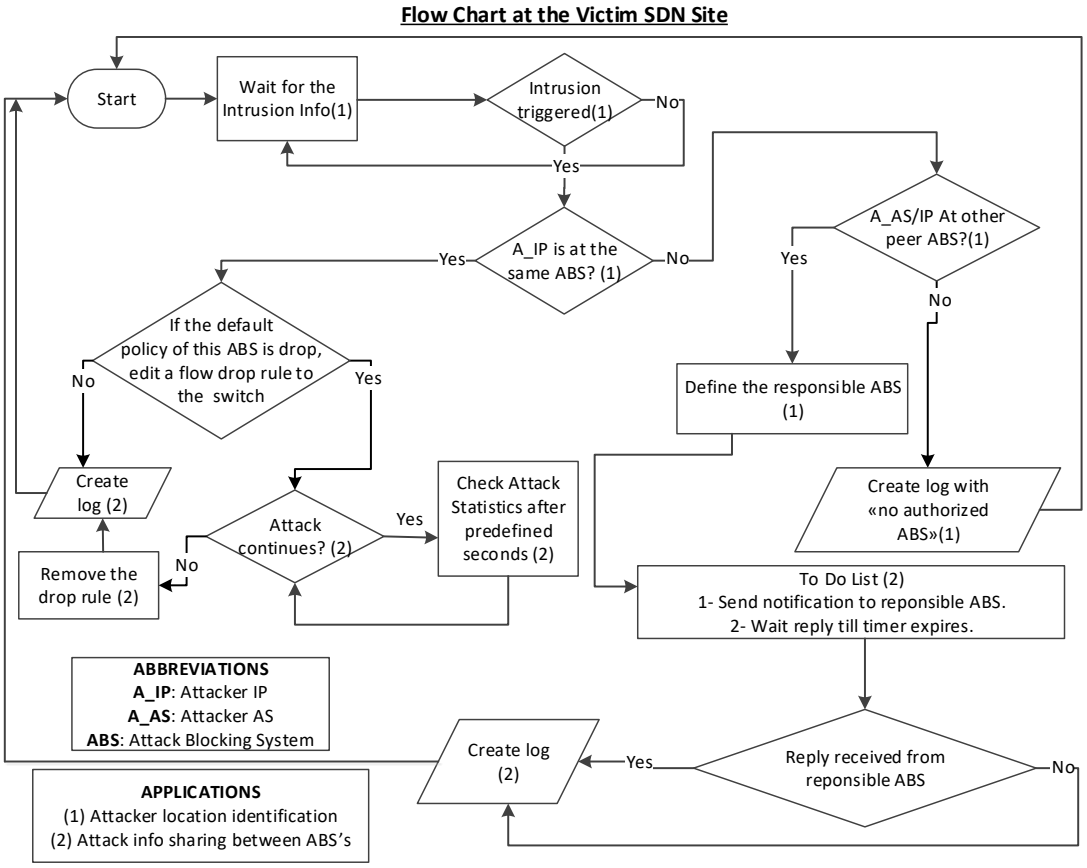

Figure 6. Flow chart at the victim site.

ABS checks the trust level with the peer ABS. If the ABS neighborship is "Full", ABS sends the related flow information with DROP action to the SDN Controller, and the SDN Controller adds this flow to the switch. Then ABS sends a Code 2.1 message ("Traffic is Blocked") to its peer. If the trust level is not "Full" and the remote site requested the flow to be blocked, the ABS sends a Code 2.4 message ("Request rejected, full-trust neighborship is not established") to its peer.

In this stage, the ABS blocks the attack traffic, but the flow added to the SDN controller must be removed after the attack is over. The ABS checks the number of dropped packets at a specific frequency. If the number of matching packets stops increasing, it removes the corresponding flow with the help of the controller.

4.3. Test environment and achieved goals

In this study, we established a Mininet network simulator, which is based on Ubuntu Linux distribution. We installed Open Virtual Switch (OVS) and Open Daylight (ODL) SDN Controller software. Our testbed composed of three virtual servers. We installed Mininet/OVS, ODL Controller and ABS software to each server. First server represents the victim site, second server represents the attacker site with full-trust ABS relationship and the third server represents the attacker site without full-trust ABS relationship. We installed all servers on the same computer on a Virtual Box virtualization platform. The computer has an Intel Core i5 processor with 16 GB of memory.

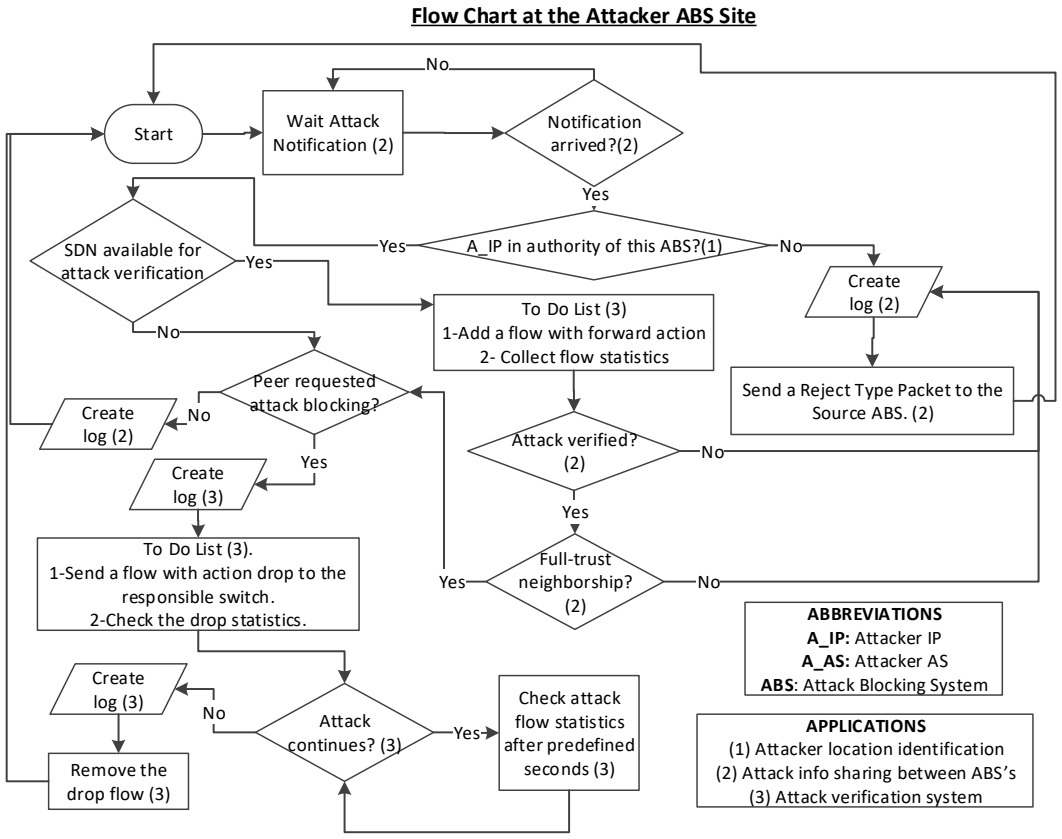

Figure 7. Flow chart at the attacker site.

As indicated in Table 1, the only technique that operates multidomain as well as ABS is BGP/Flowspec. We compared the two techniques in order to obtain performance values. Since Flowspec only works on carrier grade routers, we got support from a national ISP to build the test environment. In the GNS3 emulation environment, we created an testbed with a Cisco ASR9K series router and a Nokia 7750 series router. Both systems were implemented between two directly connected domains. According to the measurements in the test environment, the attack blocking durations are shown in Table 2.

Table 2. Attack blocking durations (The average of ten trials)

System	Requires BGP	Blocking duration	Blocking attack traffic with attack verification	Remove blocking rule by verifying that attack traffic stopped
Flowspec	Yes	0,547 Sec	No	No
ABS	No	0,051 Sec	0,111 Sec	0,110 Sec

Flowspec is a feature added to the BGP protocol. The main purpose of the BGP protocol is to create global routing tables. For stability purposes, BGP has to wait for advertising timers to expire before making a new advertisement. This occurs 30 seconds after the last advertisement in the eBGP protocol (RFC4271). We changed this timer to zero for a fair comparison. Since the Flowspec does not support "Blocking attack traffic with attack verification" and "Remove blocking rule by verifying that attack traffic stopped" features, relevant

values could not be measured. The ABS system which we developed in Python accomplished the following goals in our testbed:

- The victim site was able to send attack notifications to the attack sources.
- Attacker sites were able to validate the specified attack information from the OVS.
- Attacker site which has full-trust ABS relationship have completely stopped the attack. Other attack site which does not have full-trust ABS relationship, only notified the network admins from the dashboard.
- The system monitored the attack traffic and automatically deactivated the blocking rules when the attack stopped.
- The system worked independently from ISP and BGP as intended.

5. Conclusions and Discussion

In conclusion, with the ABS, ISPs and enterprise network administrators can now directly be involved in the detection and prevention of global cyber-attacks. This system can verify and block the attack with the help of the SDN architecture. In addition, these kinds of systems can help spread the use of SDN. Even if the system is not using SDN, this system provides a global attack information environment between network administrators. This feature is even more important.

The ABS aims to block the data traffic of attacking computers and minimize the effects of DDoS attacks on the victim system. Most of the time, a malicious code causes this kind of attacks. For example, if malware infects a PC, IPS and IDS systems do not concentrate on cleaning it, they only try to block its traffic. As more ISPs and corporate networks join the ABS infrastructure, more malware-infected computers can be detected. Therefore, this system also helps reduce the number of botnet members.

In the case of NAT, the system administrator can only access the public IP address information used by the attacking PC. Even in this case, this system can block attacks using the attacker's public IP address and port number. We can also check NAT translation tables to determine the attacker PC. In future studies, a module can be developed that integrates NAT translation tables into this system. The same problem does not occur on systems that use public IP addresses. With IPv6 migration, this situation will be eliminated.

Neighborship between the ABS systems must be full mesh. In the case of a large number of installations, ABS can cause problems due to the large volume of neighborships. For example, if we install this system for 100 network authorities, 4950 ABS neighborships are required. Later studies can focus on implementation of designated trust points for scalability purposes. With this feature, any ABS can be registered as this designated ABS only.

In this study, we have implemented an attack verification and blocking system with SDN or manual intervention. Future studies can add DDoS attack mitigation techniques, such as S/RTBH, D/RTBH or Flowspec, to the ABS.

References

[1] Skowyra R, Bahargam S, Bestavros A. Software-defined IDS for securing embedded mobile devices. In: 2013 IEEE High Performance Extreme Computing Conference; Waltham, MA, USA; 2013. pp. 1-7.

[2] Zargar ST, Joshi J, Tipper D, Member S. A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks. IEEE Communications Surveys & Tutorials 2013; 15 (4): 2046-2069. doi: 210.1109/SURV.2013.031413.00127

[3] World Economic Forum. The global risks report 2018, 13th edition. In: World Economic Forum; Geneva, Switzerland; 2018. pp. 1-20

[4] Mirkovic J, Reiher P. A taxonomy of DDoS attack and DDoS defense mechanisms. ACM SIGCOMM Computer Communication 2004; 34 (2): 39-54. doi: 10.1145/997150.997156

[5] Cui Y, Yan L, Li S, Xing H, Pan W et al. SD-Anti-DDoS: Fast and efficient DDoS defense in software-defined networks. Journal of Network and Computer Applications 2016; 68: 65-79. doi: 10.1016/j.jnca.2016.04.005

[6] Douligeris C, Mitrokotsa A. DDoS attacks and defense mechanisms: Classification and state-of-the-art. Computer Networks 2004; 44 (5): 643–666. doi: 10.1016/j.comnet.2003.10.003

[7] Wang Z. An elastic and resiliency defense against DDoS attacks on the critical DNS authoritative infrastructure. Journal of Computer and System Sciences 2019; 99: 1–26. doi: 10.1016/j.jcss.2017.05.012

[8] Singh K, Singh P, Kumar K. Application layer HTTP-GET flood DDoS attacks: Research landscape and challenges. Computers and Security 2017; 65: 344–372. doi: 10.1016/j.cose.2016.10.005

[9] Kurt B, Yıldız Ç, Ceritli TY, Sankur B, Cemgil AT. A Bayesian change point model for detecting SIP-based DDoS attacks. Digital Signal Processing: A Review Journal 2018; 77: 48–62. doi: 10.1016/j.dsp.2017.10.009

[10] Saravanan R, Shanmuganathan S, Palanichamy Y. Behavior-based detection of application layer distributed denial of service attacks during flash events. Turkish Journal of Electrical Engineering & Computer Sciences 2016; 24: 510–523. doi: 10.3906/elk-1308-188

[11] Zhou L, Guo H, Deng G. A fog computing based approach to DDoS mitigation in IIoT systems. Computers and Security 2019; 85: 51–62. doi: 10.1016/j.cose.2019.04.017

[12] Priyadarshini R, Barik RK. A deep learning based intelligent framework to mitigate DDoS attack in fog environment. Journal of King Saud University - Computer and Information Sciences 2019; 1: 1-15. doi: 10.1016/j.jksuci.2019.04.010

[13] Yusop ZM, Abawajy JH. Analysis of insiders attack mitigation strategies. Procedia - Social and Behavioral Sciences 2014; 129: 611–618. doi: 10.1016/j.sbspro.2014.06.002

[14] Miao R, Yu M, Jain N. NIMBUS: Cloud-Scale Attack Detection and Mitigation. In: ACM conference on SIGCOMM; August 2014; Chicago, IL, USA. pp. 121–122.

[15] Agrawal N, Tapaswi S. Low rate cloud DDoS attack defense method based on power spectral density analysis. Information Processing Letters 2018; 138: 44–50. doi: 10.1016/j.ipl.2018.06.001

[16] Bhushan K, Gupta BB. Hypothesis Test for Low-rate DDoS Attack Detection in Cloud Computing Environment. Procedia Computer Science 2018; 132: 947–955. doi: 10.1016/j.procs.2018.05.110

[17] Serrano Mamolar A, Salvá-García P, Chirivella-Perez E, Pervez Z, Alcaraz Calero JM et al. Autonomic protection of multi-tenant 5G mobile networks against UDP flooding DDoS attacks. Journal of Network and Computer Applications 2019; 145: 1-12. doi: 10.1016/j.jnca.2019.102416

[18] Sotelo Monge MA, Herranz González A, Lorenzo Fernández B, Maestre Vidal D, Rius García G et al. Traffic-flow analysis for source-side DDoS recognition on 5G environments. Journal of Network and Computer Applications 2019; 136: 114–131. doi: 10.1016/j.jnca.2019.02.030

[19] Mamolar AS, Pervez Z, Calero JMA, Khattak AM. Towards the transversal detection of DDoS network attacks in 5G multi-tenant overlay networks. Computers and Security 2018; 79: 132–147. doi: 10.1016/j.cose.2018.07.017

[20] Demir N, Dalkiliç G. Modified stacking ensemble approach to detect network intrusion. Turkish Journal of Electrical Engineering & Computer Sciences 2018; 26: 418–433. doi: 10.3906/elk-1702-279

[21] Patil NV, Rama Krishna C, Kumar K, Behal S. E-Had: A distributed and collaborative detection framework for early detection of DDoS attacks. Journal of King Saud University - Computer and Information Sciences 2019. doi: 10.1016/j.jksuci.2019.06.016

[22] Behal S, Kumar K, Sachdeva M. D-FACE: An anomaly based distributed approach for early detection of DDoS attacks and flash events. Journal of Network and Computer Applications 2018; 111: 49–63. doi: 10.1016/j.jnca.2018.03.024

[23] Chen Y, Hwang K, Ku WS. Collaborative detection of DDoS attacks over multiple network domains. IEEE Transactions on Parallel and Distributed Systems 2007; 18 (12): 1649–1662. doi: 10.1109/TPDS.2007.1111

[24] Zhu L, Tang X, Shen M, Du X, Guizani M. Privacy-preserving DDoS attack detection using cross-domain traffic in software defined networks. IEEE Journal on Selected Areas in Communications 2018; 36 (3): 628-643. doi: 10.1109/JSAC.2018.2815442

[25] Caesar M, Caldwell D, Feamster N, Rexford J, Shaikh A et al. Design and implementation of a routing control platform. In: 2nd Conference on Symposium on Networked Systems Design; Boston, MA, USA; 2005. pp. 15–28.

[26] Mckeown N, Anderson T, Peterson L, Rexford J, Shenker S et al. OpenFlow: enabling innovation in campus networks. ACM SIGCOMM Computer Communication Review 2008; 38 (2): 69-74. doi: 10.1145/1355734.1355746

[27] Farhady H, Lee H, Nakao A. Software-defined networking: A survey. Computer Networks 2015; 81: 79–95. doi: 10.1016/j.comnet.2015.02.014

[28] Sezer S, Scott-Hayward S, Chouhan PK, Fraser B, Lake D et al. Are we ready for SDN? Implementation challenges for Software-Defined Networks. IEEE Communications Magazine 2013; 51 (7): 36–43. doi: 10.1109/MCOM.2013.6553676

[29] Vissicchio S, Vanbever L, Bonaventure O. Opportunities and research challenges of hybrid software defined networks. ACM SIGCOMM Computer Communication Review 2014; 44 (2): 70–75. doi: 10.1145/2602204.2602216

[30] Ding AY, Crowcroft J, Tarkoma S, Flinck H. Software defined networking for security enhancement in wireless mobile networks. Computer Networks 2014; 66: 94–101. doi: 10.1016/j.comnet.2014.03.009

[31] Yoon C, Park T, Lee S, Kang H, Shin S et al. Enabling security functions with SDN: A feasibility study. Computer Networks 2015; 85: 19–35. doi: 10.1016/j.comnet.2015.05.005

[32] Jantila S, Chaipah K. A Security analysis of a hybrid mechanism to defend DDoS Attacks in SDN. Procedia Computer Science 2016; 86: 437–440. doi: 10.1016/j.procs.2016.05.072

[33] Jafarian JH, Al-Shaer E, Duan Q. Openflow random host mutation: Transparent moving target defense using SDN. In: Hot Topics in Software Defined Networks (HotSDN 2012); Helsinki, Finland; 2012. pp. 127–132.

[34] Giotis K, Argyropoulos C, Androulidakis G, Kalogeras D, Maglaris V. Combining OpenFlow and sFlow for an effective and scalable anomaly detection and mitigation mechanism on SDN environments. Computer Networks 2014; 62: 122–136. doi: 10.1016/j.bjp.2013.10.014

[35] Mantur B, Desai A, Nagegowda KS. Centralized control signature-based firewall and statistical-based network Intrusion Detection System (NIDS) in Software Defined Networks (SDN). Emerging Research in Computing, Information, Communication and Applications 2015; 1: 497-506. doi: 10.1007/978-81-322-2550-8

[36] Joldzic O, Djuric Z, Vuletic P. A transparent and scalable anomaly-based DoS detection method. Computer Networks 2016; 104: 27–42. doi: 10.1016/j.comnet.2016.05.004

[37] Wang L, Li Q, Jiang Y, Jia X, Wu J. Woodpecker: Detecting and mitigating link-flooding attacks via SDN. Computer Networks 2018; 147: 1–13. doi: 10.1016/j.comnet.2018.09.021

[38] Sahoo KS, Puthal D, Tiwary M, Rodrigues JJPC, Sahoo B et al. An early detection of low rate DDoS attack to SDN based data center networks using information distance metrics. Future Generation Computer Systems 2018; 89: 685–697. doi: 10.1016/j.future.2018.07.017

[39] Carvalho LF, Abrão T, Mendes L de S, Proença ML. An ecosystem for anomaly detection and mitigation in software-defined networking. Expert Systems with Applications 2018; 104: 121–133. doi: 10.1016/j.eswa.2018.03.027

[40] Yuan B, Zou D, Jin H, Yu S, Yang LT. HostWatcher: Protecting hosts in cloud data centers through software-defined networking. Future Generation Computer Systems 2017; 1: 1-20. doi: 10.1016/j.future.2017.04.023

[41] Wang B, Zheng Y, Lou W, Hou YT. DDoS attack protection in the era of cloud computing and Software-Defined Networking. Computer Networks 2015; 81: 308–319. doi: 10.1016/j.comnet.2015.02.026

[42] Sahay R, Blanc G, Zhang Z, Debar H, ArOMA: An SDN based autonomic DDoS mitigation framework. Computers and Security 2017; 70: 482–499. doi: 10.1016/j.cose.2017.07.008

[43] Porras P, Shin S, Yegneswaran V, Fong M, Tyson M et al. A security enforcement kernel for OpenFlow networks. In: First workshop on hot topics in software defined networks - HotSDN '12; Helsinki, Finland; 2012. pp. 121-126.

[44] Xing T, Huang D, Xu L, Chung CJ, Khatkar P. SnortFlow: A OpenFlow-Based IPS in Cloud Environment. In: Second GENI Research and Educational Experiment Workshop; Salt Lake, UT, USA; 2013. pp. 89–92.

[45] Li L, Sun H, Zhang Z. The research and design of honeypot system applied in the LAN security. In: ICSESS 2011; Beijing, China; 2011. pp. 360–363.

[46] Baykara M, Das R. A novel honeypot based security approach for real-time intrusion detection and prevention systems. Journal of Information Security and Applications 2018; 41: 103–116. doi: 10.1016/j.jisa.2018.06.004