8th International Conference from "Scientific Computing to Computational Engineering"
8th IC-SCCE
Athens, 4-7 July, 2018

2-STAGE SOFT DEFENDING SCHEME AGAINST DDOS ATTACK OVER SDN BASED ON NB AND SVM

Chaiyaporn Khemapatapan

College of Innovative Technology and Engineering (CITE)
Dhurakij Pundit University (DPU), Bangkok, Thailand
chaiyaporn@dpu.ac.th

Keywords: Naive Bayesian, SVM, DDoS, SDN, Soft Defense

Abstract. *Software Defined Network (SDN) is a new legend for future Internet including Cloud computing and Fog computing due to its smart packet flows along applications. In this paper, we proposed 2-stage scheme to soft defend against Distributed Denial of Service (DDos) attack over SDN. Normally, with classical defending scheme implementation such as Web App Firewall (WAF), some good traffic will be unfortunately dropped. However, response from an application server especially for good users should be possibly served. Thus, in the proposed scheme, an incoming traffic will be first detected by the first stage defending function using Naïve Bayesian (NB) classification in which next state load factor is involved. Thus, the traffic that is classified as good will be passed to the next service point, e.g., computer server or customer site network. With SDN features, the rest traffic from first stage can be immediately passed to second stage. The second stage deployed Support Vector Machine (SVM) classification. Normally, SVM has more accuracy than NB. After SVM classified, legitimate traffic will be sent back to next service point if its load factor is less than a threshold, whereas attack traffic will be finally dropped. As a result, next service point can serve without denial of service caused from the attack. Definitely, cascade classification from different algorithms will not much improve overall accuracy. However, service rate for legitimate traffic is much improved about 5%, 8.5% and 29% in comparison with single stage SVM, NB and WAF systems, respectively.*

1 INTRODUCTION

Currently, Internet is widely used in human daily life. SDN [1] had been proposed for working in an Internet networking. SDN is a new paradigm to Internet that facilitates network management and enables network configuration based on software programming. Normally, SDN composes of three operation planes: data plane, control plane and application plane. Data plane is functioned for transferring data which is normally working by Open vSwitch (OVS). An OVS has programmatically flow, group and meter tables that are used for defining data paths and managing data flow or traffic. An SDN controller is a main device on control plane and used to add and edit tables of OVS via Southbound using OpenFlow protocol. Application plane enables user to program data path or manage traffic via SDN controller. Normally, northbound is an interface point, .e.g., REST/RESTFUL web services. Applications can use northbound to communicate with an SDN controller.

Unfortunately, Internet is under risk from threats. There are many attacks on Internet. Dos and DDoS, denial of service from computer server or target to customers, are hazard threats that are used by hackers to attack victims and target. Attacking using DDoS technique are not difficult to deploy [2] and can be found as a service from Internet [3]. The impact of DDoS causes a highly value loss in business view. The revenue loss as a business impact of DDoS attacks is annual increasing. Many DDoS attack techniques can be implemented by tools on Internet, for example, hping3, LOIC, HOIC, Slowloris, DDOSIM, Apache JMeter and etc. However, volumetric DDoS attacks based on layer 7 (application) and layer 3/4 (network) have a tendency for a few years ago [4]. Thus, this study mainly focuses on the attack based on flooding huge traffic volume of HTTP GET/POST messages from various victims to a target. Normally, victims can be either user's devices or computer server that infected malicious programs such as botnet, malware or computer virus while the target is finally attacked by victims controlled by the hacker. Normally, business services on Internet such as web servers use many tools such as firewall, IDS/IPS and WAF to defend the attack. However, use of these tools caused more investment and complexity of networking. Additionally, these tools possibly made a wrong decision and dropped much legitimate traffic.

In the past, many contributions proposed how to detect and mitigate DDoS attacks on Internet by several methods as described in [5], [6] and [7]. However, these methods are only suitable for specific conditions and have limitations due to theirs definite processes. To detect or classify DDoS traffic from legitimate traffic is difficult as stated in [6] and [8] especially for HTTP GET/POST flooding. Too much of a legitimate traffic

suffers from wrong decision of detection and protection tools, yielding detection with high false positive rate. Intelligent based methods with applying machine learning such as Naïve Bayesian, SOM, SVM and ANN methods as presented in [9], [10], [11] and [12] had been proposed for defending DDoS attacks with more flexibility and accuracy. However, DoS and DDoS attacks in early studies are mostly based on TCP Sync flood, TCP fragment flood, ICMP flood and UDP flood.

For SDN environment, there are contributions related to the survey of security as described in [13] and [14]. Obviously, SDN can be either under risk from DDoS attacks or enhanced to mitigate the DDoS attacks. Possibly, an SDN controller may be under risk from DDoS attacks due to flooding huge volume traffic through an OVS caused load and bandwidth saturation at the SDN controller. To enhance security, making a stateful firewall application in cooperation with SDN controller as presented [15], [16] and [17] that can simply mitigate DDoS attacks based on TCP Sync flood, TCP fragment flood and ICMP flood. In addition, a lightweight DDoS Flooding attack detection on SDN using self organization map (SOM) had been proposed by [18]. The method to mitigate DDoS in an SDN environment with taking care of dropping legitimate packets was proposed by [19] which applied a threshold-based decision modified from NETCONF and YANG model. However, it provided disappointment results for detection rate of bad traffic originated from devices infected a botnet. Thus, in order to solve a problem of dropping legitimate traffic from false positive decision, soft defending scheme to detect the HTTP flood DDoS attack using NB and SVM methods as classifier in an SDN environment will be proposed in this paper.

2 RELATED WORKS

2.1 SDN

Today, SDN is extensively adopted to be implemented in many autonomous systems especially for Internet providers, mobile networks and data centers. It can seamlessly work with Network Functions Virtualization (NFV). Basically, SDN framework defines 3 operation planes: application plane, control plane and data plane as shown in Figure 1.

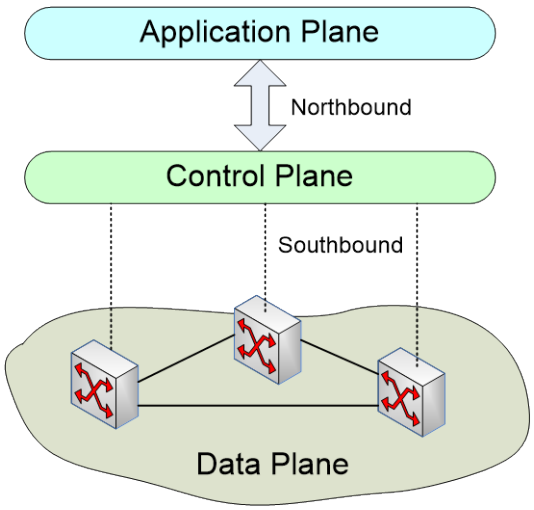

Figure 1. SDN architecture

From application plane's view, key benefit of SDN is that applications can programmatically demand to an SDN controller for setting theirs data paths and making theirs appropriate traffic management. Normally, applications can be from either user site or administrator site. At the control plane, an SDN controller is a major device used to operate. However, in order to operate with more reliable and complex, cluster of SDN controllers is possible. An SDN controller can communicate with applications via northbound interfaces which are normally REST/RESTFUL web services. However, communication between applications and SDN controllers is usually a secure channel using HTTPS/TLS framework. There are many SDN controllers in the market such as NOX, POX, OpenDaylight, Floodlight, Ryu, etc. In data plane, OVS switches will transfer data and manage flows as it is programmed in flow, group and/or meter tables stored in the memory themselves. The tables can be programmatically defined from an SDN controller via southbound which is normally an OpenFlow protocol. However, sFlow and VxLan can be used besides OpenFlow. Thus, SDN can do more functions in compatible with a traditional router. Some networking functions that are needed by administrators such as firewall, QoS

management, policy control and load balancing can be simply implemented by SDN controllers working with applications.

2.2 HTTP flood DDoS attacks

Figure 2. shows the concept of attacks by flooding DDoS traffic using botnets. Possibly, victim's devices which can be computers, smart phones, and even computer servers infected a malicious program such as botnets will be remotely controlled by hackers to attack a target. Botnets may be infected to either OS or browser as stated in [4]. Impact of the attack caused overwhelming traffic near the target site networks and loaded the memory and CPU utilizations of the target. Legitimate user traffic should not be responsible both from data transferring through the networks and from request servicing of the target. HTTP flood is one of botnet based DDoS attack and is famously used by hackers. HTTP flood is volumetric DDoS attack and is the most common attack that targeting application layer. Obviously, to classify HTTP flood DDoS attack traffic is difficult because its request traffic seem to be legitimate.

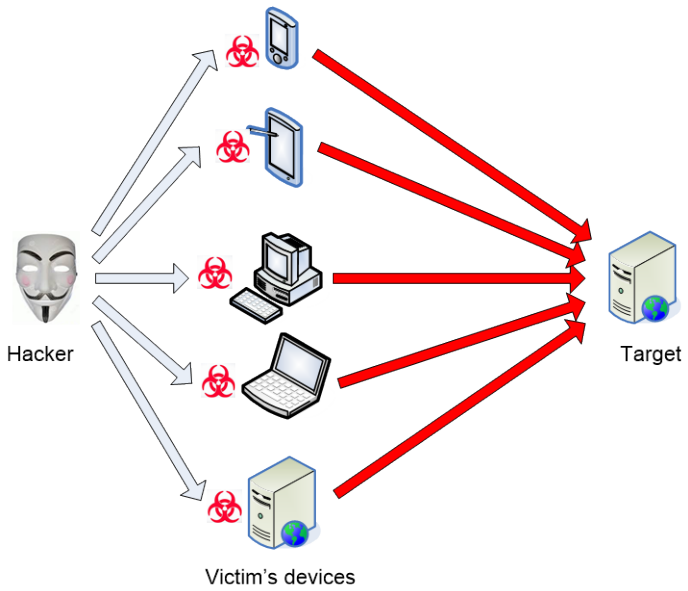

Figure 2. Botnet based DDoS attack

2.3 Classifiers

Classification is a kind of machine learning. There are many methods which can be used for classification. Normally, machine learning can be typed into 2 categories: supervised learning and unsupervised learning. Supervised learning needs known example having inputs and desired outputs or labels for teaching learning algorithms. The example methods typed as supervised learning algorithm are NB, SVM and ANN. On the other hand, unsupervised learning is no needed label to the learning algorithms. Hence, the algorithms find structure in theirs input on theirs own. SOM and deep learning are typed as unsupervised learning algorithm. Normally, supervised learning algorithms need much iteration in order to converge to the optimum solution.

NB and SVM have been selected for the proposed scheme in cascade operation because NB works on probabilistic knowledge and has an attractive advantage in less complexity and fastest processing while SVM is a powerful classifier and provides high accuracy with moderate complexity and processing. Moreover, although both algorithms have to train theirs own with known samples, but affordable time is needed.

3 PROPOSED SCHEME

The proposed scheme is shown in Figure 3. The objective of this proposal is to minimize false positive detection. Thus, possibly, the proposed scheme may serve incoming HTTP requests as much as possible while the web server still stands in accepted operation. It can be noted from the figure that detection using NB is firstly applied because it is quite simple algorithm and can support volumetric DDoS attacks of an incoming traffic. Thus, as its performance, POX-NB can be assumed to be a coarse detector with soft decision depending on web

server's load utilization. The second stage, POX-SVM, is dealt with SVM algorithm. Of course, as its accuracy, it can be noted that a fine detector placed here. Thus, only suspicious traffic that cannot be classified will be passed to the second stage. Moreover, the second stage detector will have less detection processes than the first stage because most of DDoS attack traffic will be dropped in the first stage and most of legitimate traffic had been already passed to the web server. In practical implementation, both SDN controllers can be placed on the same host.

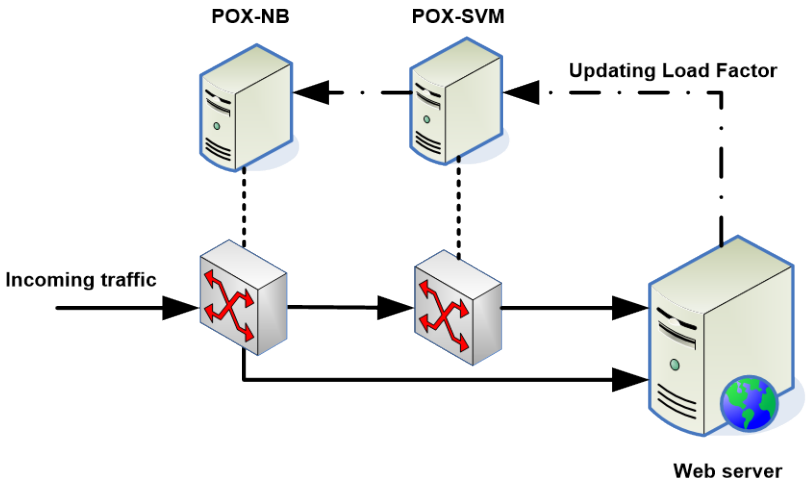

Figure 3. Proposed scheme

In details, the SDN controller using POX platform on Mininet simulator is deployed in this study. POX is a networking software platform written in Python language. It is easy to gather flow parameters by using Python. In addition, count of source ports per flow, source address frequency, packet size, inter arrival time or packet rate in the same flow, User-Agent frequency, header hash frequency of HTTP request message except User-Agent and body hash frequency of HTTP request message will be used as additional features to defend flooding from HTTP GET/POST messages. All features that are measured by frequency calculated from previous interval. The interval is normally set to be 2 seconds. In addition, at every interval, web server will automatically update its last load factor metric, L_F, to SDN controllers via northbound interfaces for using in detection processes. L_F will be used to make a soft defending scheme against DDoS attacks at the first stage while it is used to manage a viability of web server at the second stage. However, L_F is neglected when detectors are trained.

At the first stage detector, NB works on probabilistic model based on Bayes' theorem. Thus, the conditional probabilities of the request message, *M*, will be an attack state and normal state are represented in equations (1) and (2), respectively.

$$P(M|Attack) = \frac{P(Attack)}{Z}\prod_{i=1}^{n} P(x_i\,|Attack) \tag{1}$$

$$P(M|Normal) = \frac{P(Normal)}{Z}\prod_{i=1}^{n} P(x_i\,|Normal) \tag{2}$$

where x_i indicates the *i*th feature from total *n* features and

$$Z = P(Attack)\prod_{i=1}^{n} P(x_i\,|Attack) + P(Normal)\prod_{i=1}^{n} P(x_i\,|Normal)\,. \tag{3}$$

The detection process can be expressed in logarithm space as shown in equations (4) and (5).

If $L_F \leq 0.5 - T_1$,

$$y = log\left\{\frac{P(M|Normal)}{P(M|Attack)}.\frac{(1-T_1-L_F)}{(L_F+T_1)}\right\} \quad (4)$$

where T_1 is a reserved safety factor and is less than 0.5. Thus, first stage detector will pass the request message to the server if $y \geq 0$, otherwise the request message will be dropped. It can be noticeable that when load factor is still at low, i.e., web server has not much load, the features characterized the legitimate request messages will be enhanced yielding decreasing the false positive detection. On the other hand, the features characterized the attack messages will be diminished. Thus, if the detection value *y* is less than zero, the features characterized the attack behaviors must be outstandingly numerous and the message should be definitely dropped.

For $0.5 - T_1 < L_F \leq 0.5$, the first stage detector will also pass the request message to the server if $y \geq 0$, otherwise the request message will be passed to the second stage one. This event occurs when the server has moderate load utilization. Thus, the features characterized the legitimate request messages will be slightly diminished in order to make sure that a request message having numerous values of legitimate features will be served, while suspicious request messages will have a second chance to be checked at the second stage detector having higher accuracy.

If $L_F > 0.5$,

$$y = log\left\{\frac{P(M|Normal)}{P(M|Attack)}.\frac{L_F+1}{2L_F}\right\} . \quad (5)$$

This actions try to slightly enhance the features characterized the legitimate request messages when the server has moderate load utilization in order to decrease a positive false detection classified by the first stage. However, if the server achieves heavily load, enhancement features is not needed. Thus, first stage detector will pass the request message to the second stage one if $y \geq 0$, otherwise the request message will be dropped.

At the second stage detector, an SVM algorithm is applied. Basically, SVM is 2-state classification algorithm used a high dimension space to find a linearly margin on hyper-plane. From a given *n* sample of training data set as presented in form as follows:

$$\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \quad (6)$$

where x_i is any feature and known label $y_i \, \epsilon \, \{-1,1\}$ of the sample. There is a hyper-plane among set of the sample which satisfies

$$\boldsymbol{w}^T\boldsymbol{x} - b = 0 \quad (7)$$

where $\boldsymbol{w}$ is a normal vector and *b* is an offset. The distance between planes is $\frac{b}{\|\boldsymbol{w}\|}$. Thus, to maximize the distance, minimization of $\|\boldsymbol{w}\|$ is needed. So that

$$y_i(\boldsymbol{w^T x_i} - b) \geq 1 \quad , \forall_{i=1}^{n} . \quad (8)$$

Therefore, the detect function of any testing data $\boldsymbol{x}$ is as follows:

$$f = sgn(\boldsymbol{w^T x} - b) . \quad (9)$$

If the detector promotes the legitimate request message, it will be positive. Thus, the rest of them will be negative. Moreover, in order to let the server to be viable, threshold T_2 is defined. Thus

$$f = \begin{cases} sgn(\boldsymbol{w^T x} - b) , & \text{if } L_f \leq T_2 \\ -1 , & \text{if } L_f > T_2 \end{cases} \quad (10)$$

Hence, if the output from detector is negative, the request message will be assumed as the attack while if the output is positive, the message will be passed to the server.

4 SIMULATION AND RESULTS

The simulation applied 4 scenario protection systems: NB detector, SVM detector, WAF and the proposed scheme. First three scenarios are used as baseline. WAF is implemented by enabling Apache ModSecurity with OWASP core rule set version 3. The simulation topology as shown in Figure 4. deployed on Mininet environment. All client devices locate in 4 subnets. In order to simulate close to real situation, 3 subnets compose of victim and normal devices while last subnet has only normal devices. Each subnet connects to an external network via a router enabling NAT address function. Thus, source addresses of victim and normal devices from the same subnet will be the same addresses after NAT function. In addition, Apache JMeter is used to emulate 12 victim devices infected botnets and 16 normal devices. An Apache JMeter host emulated 4 NICs in order to enable 4 spoofing IP addresses, so that each host emulates 4 different devices. Each victim device generated 1,000 HTTP request messages within 3 minutes obtaining about 5.55 HTTP request messages per second per device, while normal devices randomly access the web server with rate 10-second accessing for 5 minutes.

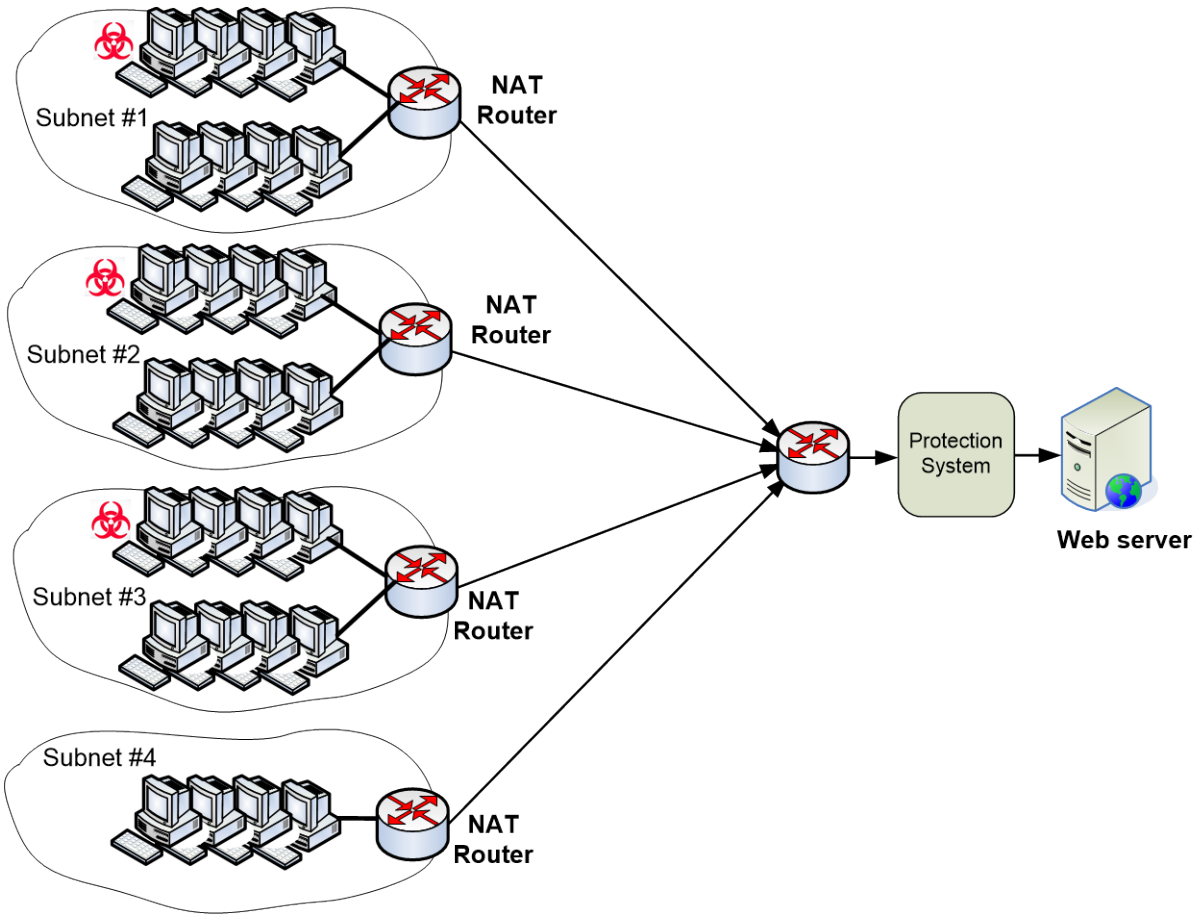

Figure 4. Simulation topology

The proposed scheme thresholds are chosen to be 0.2 and 0.9 for T_1 and T_2, respectively. Hence, with these thresholds, a web server can run at normal utilization about 30% and has a viable utilization at 90%. The testing results are shown in Table 1. It can be noted from the testing results that detection rate accuracies of all scenarios are not different. Furthermore, the proposed scheme can slightly improved detection rate in comparison with SVM detector. In case of taking care of servicing to the legitimate users, WAF detector provides worst false alarm rate at 34.85% due to definite rule of the firewall. NB detector can improve the false alarm rate in comparison with WAF. However, SVM detector can perform better than NB detector. Finally, the proposed scheme can achieve the best false alarm rate at 5.82%, i.e., the proposed scheme improved the false alarm rate or service rate of legitimate traffic in comparison with single stage SVM, NB and WAF detectors about 5%, 8.5% and 29%, respectively.

Scenario	Attack count	Legitimate count	False Positive count	False Negative count	False Alarm Rate (%)	Detection Rate (%)
WAF	12,000	482	168	447	34.85	96.28
NB	12,000	482	69	297	14.32	97.53
SVM	12,000	484	52	231	10.74	98.08
Proposed scheme	12,000	481	28	227	5.82	98.11

Table 1. Simulation results

5 CONCLUSION

In this paper, defending schemes to mitigate an HTTP DDoS attack is mainly focused. Obviously, to detect HTTP DDoS attack from legitimate HTTP request is difficult because the similar behavior between them. Hence, legitimate users suffer from wrong decision of detectors. The two-stage soft defending scheme using NB detector and SVM detector is proposed in comparison with single stage NB, SVM and WAF detectors. The proposed scheme also included load utilization factor of next service point, e.g., computer server or customer site network. The factor is used to create soft detection at the first stage detector in order to gain the service rate of legitimate users. From the testing results, it can be noted that the proposed scheme can accomplish the objective of this study. It can absolutely improve false alarm rate or service rate for legitimate users.

However, currently, HTTPS is widely adopted as a main protocol for web and web services. In addition, hackers have evolved themselves by modifying theirs attack style. For example, multi-vector attacks or the combo of DDoS attack methods, e.g. HTTP/HTTPS flood plus UDP flood plus TCP Sync flood, is the new style used to attack efficiently. This new attack style can overcome current defending systems. Thus, study on how to defend HTTPS DDoS attack and multi-vector attack is interesting.

REFERENCES

[1] Kreutz D. Et al. (2015), “Software-Defined Networking:A Comprehensive Survey”, Proceedings of the IEEE, Vol.103, pp.14-76.

[2] Elleithy K.M., Blagovic D., Cheng W.K. and Sideleau P. (2005), “Denial of Service Attack Techniques: Analysis, Implementation and Comparison”, Journal of Systemics, Cybernetics and Informatics, Vol. 3, No. 1, pp. 66-71.

[3] Santanna J.J. et al. (2015), “Booters - An Analysis of DDoS-as-a-Service Attacks”, *Proceedings of IFIP/IEEE International Symposium on Integrated Network Management, Ottawa Canada, 11-15 May,* pp.243-251.

[4] IMPERVA (2015), “Top 10 DDoS Attack Trends, Discover the Latest DDoS Attacks and Their Implications”, Imperva white paper, www.imperva.com (accessed on Apr 1, 2018).

[5] Xu J. and Lee W. (2003), “Sustaining Availability of Web Services under Distributed Denial of Service Attacks”, IEEE Transactions on Computers, Vol. 52, pp.195-208.

[6] Gupta S., Grover D., and Bhandari A. (2014), “Detection Techniques against DDoS Attacks: A Comprehensive Review”, International Journal of Computer Applications, Vol. 96, No.5, pp. 49-57.

[7] Bonguet A. and Bellaiche M. (2017), “A Survey of Denial-of-Service and Distributed Denial of Service Attacks and Defenses in Cloud Computing”, Future Internet, Vol.9.

[8] Gao Z. and Ansari N. (2006), “Differentiating Malicious DDoS Attack Traffic from Normal TCP Flows by Proactive Tests”, IEEE Communications Letters, Vol. 10, No. 11, pp.793-795.

[9] Mitrokotsa A., and Douligeris C. (2005), “Detecting Denial of Service Attacks Using Emergent Self-Organizing Maps”, *Proceedings of the Fifth IEEE International Symposium on Signal Processing and*

Information Technology, Athens Greece, 21-21 December, pp.375-380.

[10] A. Sahi et al. (2017), "Efficient DDoS TCP Flood Attack Detection and Prevention System in a Cloud

[11] Li J., Liu Y. and Gu L. (2010), "DDoS attack detection based on neural network", *Proceedings of 2nd*

[12] Wang C., Zheng J. and Li X. (2015), "Research on DDoS Attacks Detection Based on RDF-SVM", *Proceedings of 10th International Conference on Intelligent Computation Technology and Automation, Changsha China, 9-10 October,* pp.161-165.

[13] Ahmad I. et al. (2015), "Security in Software Defined Networks: A Survey", IEEE Communication Surveys & Tutorials, Vol. 17, pp.2317-2346.

[14] Scott-Hayward S., Natarajan S. and Sezer S. (2015), "A Survey of Security in Software Defined Networks", IEEE Communication Surveys & Tutorials, Vol. 18, pp. 623-654.

[15] Zhu S. et al. (2015), "SDPA: Enhancing Stateful Forwarding for Software-Defined Networking", *Proceedings of IEEE 23rd International Conference on Network Protocols, San Francisco, CA, USA, 10-13 November,* pp.323-333.

[16] Gupta V., Kaur S. and Kaur K. (2016), "Implementation of Stateful Firewall using POX Controller", *Proceedings of 3rd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 16-18 March,* pp.1093-1096.

[17] Zerkane S., Espes D., Le Parc P., Cuppens F. (2016), "Software Defined Networking Reactive Stateful Firewall", *Proceedings of 31st IFIP TC 11 International Conference SEC 2016, Ghent Belgium, May 30 - June 1,* pp.119-132.

[18] Braga R., Mota E. and Passito A. (2010), "Lightweight DDoS Flooding Attack Detection Using NOX/OpenFlow", *Proceedings of 35th Annual IEEE Conference on Local Computer Networks, Denver, CO, USA, 10-14 October,* pp.408-415.

[19] Hyun D. et al. (2017), "SDN-based Network Security Functions for Effective DDoS Attack Mitigation", *Proceedings of International Conference on Information and Communication Technology Convergence (ICTC), Jeju, South Korea, 18-20 October,* pp.834-839.