ISSN 2278-3091
Volume 9, No.4, July – August 2020
International Journal of Advanced Trends in Computer Science and Engineering
Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse278942020.pdf
https://doi.org/10.30534/ijatcse/2020/278942020

Experimental Study on Software-Defined Network Implementation for DDoS Attack Detection and Mitigation

Nurzuriana Mohd Zahri[1], Zul-Azri Ibrahim[2], Fiza Abdul Rahim[3], Yunus Yusoff[4]
[1]College of Computing and Informatics, Universiti Tenaga Nasional, Malaysia, zuwien98@gmail.com
[2]Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional, Malaysia, zulazri@uniten.edu.my
[3]Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional, Malaysia, fiza@uniten.edu.my
[4]Institute of Informatics and Computing in Energy, Universiti Tenaga Nasional, Malaysia, yunusy@uniten.edu.my

ABSTRACT

In this Industrial Revolution 4.0 era, the ability to access digital resources anywhere and anytime is essential to increase the level of user's trustworthiness. For many years, Distributed Denial of Service (DDoS) attacks has become the biggest problem for service providers in making their services available to end-users. This attacks can take down any servers or hosts available online by flooding the available resources such as bandwidth and memory with a false request to prevent legitimate users from accessing the servers. There are various categories of DDoS which are volumetric attack, protocol attack and application-layer attack, and it is challenging to detect by conventional security devices as it mimics the regular network traffic. In combatting DDoS attacks, Software-Defined Network (SDN) is one of the latest technologies used. SDN has a separated data plane and control plane, which make it more flexible in controlling the network with a central controller that has the capabilities to detect any malicious activities within the SDN network. This research aims to study the DDoS attack mitigation in SDN Local Area Network (LAN). This study simulated three types of DDoS attacks to see the effects on the CPU of the victim and RAM usage compared to a previous study using open-source network simulator.

Key words: DDoS attacks, Entropy, Impact DDoS, Mininet, PCA, SDN techniques

1. INTRODUCTION

Nowadays, the Internet has turned into demand with the arises of Industry Revolution 4.0. The existing Internet architecture is more focus on the performance rather than security. Novice users can be easily compromised with their systems' vulnerabilities as they use a general password, leaving design features in default mode, switching off firewalls and many others. All those deficiencies give opportunities for an attacker to exploit them by using a different type of attacks that exist in the network.

Distributed Denial of Service (DDoS) attack is a malicious attempt to interrupt regular traffic of a targeted server, service, website, and network resource by overwhelming the target or its surrounding infrastructure with a flood of requests or malformed packets from distributed sources. Due to the astounding amount of parallel applications, it forces the system to slow down or even crash and shut down, thereby denying service to authorized users such as employees and customers or systems itself [1]. Ever since DDoS turn out to be one of the most popular and long-lasting problems in the network and Internet security. According to the study, the most common form of DDoS attack results from damage to the limited resources of legitimate user connections and services as well as network and server resources [2].

Most famous DDoS attacks in recent history happened on 28th February 2018 [3] where a popular developer platform which is known as GitHub being hit with a sudden onslaught of traffic with a scale of 1.35Tbps at peak. The amount of traffic not only massive but surprisingly it is record-breaking. Even they had doubled their transit capacity during the time, which has allowed them to withstand specific volumetric attacks; still, they require the help of the third party with more extensive transit networks. That incident shows that, even though the DDoS attack mechanism is widely understood, the solutions are not yet discovered. Not just that, the problems are become more frequent recently due to the common characteristics between the DDoS attack and the regular traffic.

The emergence of Software-Defined Network (SDN) is here to detect and mitigate DDoS attack. It is a new architecture that improves network control management by separating the control plane and data plane [4][5]. Many techniques that

were used by SDN in detecting the attack such as Entropy, PCA, Time-based Detection, Low-Traffic Flows Detection, FlowRanger and Scheduling-based Architecture Flows [6].

The objective of this study is first to review the current DDoS attack and SDN technique used in mitigating DDoS attack. Reviewing the existing literature also had been done. Next is to detect DDoS attack based on CPU and RAM utilization and to compare the result with past research. The last one is to examine the efficiencies of Entropy and PCA technique in mitigating DDoS attack.

This paper is organized as follows. Section 1 is about introducing what this project about by telling the problem statements and objectives of this project. Section 2 is about the literature review, which is the research studies on a past research paper by reviewing it. Next is section 3, which is listing all the processes involved throughout this project. Section 4 is the experiment set up, which is what to be used in the experiment. Followed by Section 5, which is the finding and analysis from the data that had been collected and section 6 is the conclusion.

2. LITERATURE REVIEW

2.1 DDoS Attacks

The DDoS attack has become a significant concern for security experts because it can make its target slow down or even collapse by sending a large number of malicious packets at a particular time. This is possible by using infected workstation or zombie node which will work together to transmit DDoS attacks packets or zombie packet that have the same destination and port addresses [7]. This flood of internet traffic become worsen due to the development of emerging technologies such as cloud computing, Internet of Things (IoT) [8] and artificial intelligence (AI). From that, the attacker can launch a vast volume of DDoS attack with lower cost.

DDoS attack consists of three categories; volumetric, protocol, and application-layer [9]. Volumetric attacks most likely happen at the transport layer as it includes UDP floods, ICMP floods as well as other spoofed-packet floods. The intention of this attack is to make a bottleneck situation whereby significant quantities of data and requests are forwarded to a specific target.

While, a protocol attack, also popular as a state-exhaustion attack focuses on exploiting server resources which by means can cause a service disruption. It includes SYN flood, Ping of Death, fragmented packet attacks, Smurf and many more.

An application-layer attack referred to as a layer 7 DDoS attack since it is at the 7th layer in the OSI model. The attack is focused on web applications and is considered the most sophisticated and severe type of attacks. The goal of this attack is to exhaust the resources of its victim. As mentioned in the previous section, it is hard to protect, as traffic can be hard to mark as malicious as it is legitimate as regular traffic.

2.2 SDN Techniques

SDN is a new architecture that improves network control management as it can programmatically control network behaviors, such as changing, inserting and updating rules. This is because SDN separates the control plane and the data plane. The feature of separation is an approach to designing, building and managing the network. Due to its capability to reprogram the data plane at any time, this function can motivate, promote or e-network related security applications. SDN was introduced to overcome the traditional networking limitations [5] which are time-consuming and vulnerable to error, high skills needed for Multi-Seller environments as well as conventional architectures that complicate the network segmentation.

Three underlying architectures of SDN are application layer, control layer, as well as infrastructure layer [10]. In the application layer, a new function is applied to help traffic management. It collects information from the controller for decision-making purposes. This application will be forwarded to control plane of the SDN for the controller to take the appropriate steps and advise them on the data plane.

Control layer, also can be referred to as control plane acts as a brain of the SDN as its centralized SDN controller software which controls traffic policy and flow through the network. It interconnects the applications on the top and bottom of the architecture. So, after the controller gives the instructs, the network device will start applying it.

Next is the infrastructure layer or named as data plane that is made up of the physical switchers such as Juniper Junos MX-series and virtual switches such as OpenvSwitch in the network. It consists mainly of Forwarding Elements (FEs) that monitor network transmission and data processing capabilities including forwarding and data path processing.

SDN has many unique characteristics that are essential to identifying DDoS. There are several mitigation techniques employed by SDN. For example, time-based detection [6]. It is to test for a valid target address that is handled by the controller. Next is the Entropy detection algorithm where a particular limit value is calculated as well as Entropy value for each window. If the calculated entropy for five windows is smaller than a limit, an assault will be identified.

Apart from that is Principle Component Analysis (PCA). This technique is a well-known statistical method used in multivariate optimization problems that help to reduce the dimensionality of data without losing the most information in the large set.

2.3 Related Work

Several research papers have reviewed based on some criteria to make it relevant to the research topic. First, the paper selection is between 2008 until 2019. The type of paper selection is from conferences proceeding and journal articles in a digital library such as IEEExplore, ScienceDirect, ResearchGate and Indexing Service ISI.

Table 1 below shows the summarization of past research papers that consist of information that has been extracted, such as the topology for simulation, simulator and SDN technique used in the experiment. The idea of the topology used is inspired by [11][4][12]. Apart from that, Mininet can be seen as the most used tool by the researcher to simulate the experiment. Hence, in this project, the simulator used is Mininet. As for the SDN mitigation technique, Entropy had been chosen as the table shows 3 out of 23 paper using Entropy as their mitigation method against DDoS attack, which is the most. And most of them look into a technique that focuses on performance, followed by effectiveness, accuracy and efficiency. This project is highlighted about the efficiencies of the Entropy technique by observing the time taken needed for it to detect the attack.

Table 1: Summary of past research papers

No	Publication	Technique	Topology	Simulator	DDoS Attack	Research Aims
1	2019 [11]	Entropy-based	1 controller 7 switches 48 hosts	Mininet	Network layer (IP spoofing) Transport layer (TCP & UDP)	Performance (calculating different window size) Effective (reduce computational overhead)
2	2019 [13]	Advancement of Finite State Machine (FSM) to build load balancer (use module)	Not shown	Mininet	Transport layer (TCP)	Accuracy (percentage that detect what type of attacks it is and true positives showed)
3	2019 [14]	Use module (traffic collection, attack identification, flow table delivery)	1 controller 2 switches 1 server 2 normal hosts 2 attackers	Mininet	Application layer (HTTP)	Effectiveness (calculating the TP, FP, FN, TN of various type of KDD99 Data Set)
4	2019 [2]	Using four modules and the detection incorporating with support vector machine classification algorithm	1 Floodlight controller 1 OF Switch 10 hosts	Mininet	UDP DDoS flood attack	Performance (evaluate the Detection Rate and False Positive Rate)
5	2019 [15]	Modified Decision Tree Algorithm (MODET) & DETPro (decision based on tree method)	2 switches Web server 3 hosts 1 controller Attack by botnet	Mininet	Transport layer (SYN Flood) Application layer (HTTP Post & Get)	Performance (detection rate)
6	2018 [16]	Dynamic and adaptive threshold by classifying the packet	Not mentioned	Mininet	Transport layer (SYN Flood)	Effective (determine the threshold value to decide the attacks)
7	2018 [17]	Use algorithm of SVM & DNN	1 switch 1 controller 4 hosts	Mininet	Transport layer (TCP)	Accuracy (measure how accurate the classifier classifies both positive and negative instances)
8	2018 [18]	Entropy-based	1 controller 2 switches *n* hosts	Mininet	Transport layer (UDP) Network layer (IP spoofing)	Accuracy (get optimal entropy threshold by ltesting various window sizes)
9	2018 [19]	Defence mechanism (ingress, egress, pushback)	Not mentioned	Mininet	Network layer (Domain system)	Performance (measure detection rate and false positive)
10	2018 [20]	Deploys rule in controller based on packet & byte average	1 controller 1 switch 1 web server 2 normal users 2 malicious users	Mininet	Not stated	Performance (evaluate traffic and attack flows with proposed algorithm and without proposed algorithm)
11	2018 [21]	TenserFlow (mitigate) & Deep Reinforcement Learning Algorithm	1 controller 5 switches 1 server 1 mitigation host 2 traditional routers 3 normal users 4 attackers	Mininet	Transport layer (TCP, SYN, UDP) Internet layer (ICMP)	Effectiveness (benign traffic still flows despite the attack occurs)
12	2017 [22]	Detect and eliminate DDoS attack in time by predict the new unmatched packet	2 switches 20 hosts 1 controller 2 security gateways	Not mention	IP address proofing	Accuracy (run the test many time and calculating the average of false positive and negative values)
13	2017 [23]	StateSec (stateful SDN concepts)	1 switch 1 controller	Mininet	Transport layer (UDP Flood)	Accuracy (limiting control plane overhead and controller load)

No	Publication	Technique	Topology	Simulator	DDoS Attack	Research Aims
			3 hosts (server [victim], malicious user, genuine user)			
14	2017 [8]	Threshold-based method	9 switches 1 firewall 1 DDoS attack mitigation 1 DPI 4 switches	Mininet	Not stated	Effective (total packet drop ratewhich decrease by incresing policy set)
15	2017 [24]	Hybrid defense mechanism using IDS and Firewall	Not mentioned	Ubuntu Server	Application Layer HTTP DDoS attack	Efficiency (quick detection time) Performance (less false positive rate)
16	2017 [5]	Third party application which is iftop	2 switches 1 controller 1 server 5 nodes 4 attacker nodes	Mininet	Internet layer (ICMP request)	Performance (decide if a computer, network, software program or system is speedy or adequate)
17	2016 [25]	FlowTrApp algorithm	20 switches 1 controller 16 hosts (Fattree topology)	Mininet	Transport layer (UDP & TCP) Internet layer (ICMP)	Performance (can detect and mitigate both high and low rate DDoS attack)
18	2016 [26]	Firewall	5 switches 3 routers Firewall 30 physical systems in 3 clusters of 10 computers each	DDoS Testbed (DDOSTB)	UDP floods and HTTP	Effectiveness (measur network, application and server level metrics)
19	2015 [27]	Entropy-based	7 switches (1 victim) 20 hosts 1 controller	Mininet	Transport layer (TCP & UDP)	Performance (measure how many numbers of attack that suitable to be detected) Efficiency (calculate the attack detection time)
20	2015 [4]	Define temple table in controller (T table) to store source IP	1 switch 1 controller 1 server 3 host (frequent user, DDoS user, malicious user)	OPNET	Network layer (IP spoofing)	Effectiveness (measuer the minimum number of packets per connection with the average of connections of frequent users to detect the attack occurs)
21	2015 [12]	Flow collector (store non-valid packet for further inspection)	1 controller 4 switches 12 hosts	Mininet	Not mention type of attack	Performance (evaluate Detection Rate (DR) and False Alarm Rate (FA))
22	2015 [28]	ACL Threshold Limit Reverse Path Forwarding Network Load Balancing	1 router 2 attackers 1 victim 1 monitoring machine	Not mentioned	UDP flood using Hping3	Effectiveness (measure throughput value after attack occurs)
23	2008 [29]	Principal Component Analysis (PCA)	Not mentioned	Not mentioned	Profile Attack and AAS	Performance (comparing between CLP and PCA in term of false positive and false negative value during profile attack and AAS attack)

3. METHODOLOGY

In completing this study, there are five main phases involved, including initial study, topology design, experiment, analysis and report.

Phase 1: Initial study

In this phase, an initial study by reviewing past research papers had been done to get the basic concept and idea clearly regarding the research topic.

Phase 2: SDN topology design

The design topology that needed to be used during the simulation had been decided in this phase.

Phase 3: Conducting an experiment

The experiment is run, and the data had been collected to analyze and compare. The main experiment is to see the increasing CPU and RAM utilization during the attack. And the next experiment is to see the efficiencies of the proposed SDN techniques, which are Entropy and PCA.

Phase 4: Compare and analyze the result

This phase is basically to show the findings by analyzing the data collected and comparing the results based on the existing studies.

Phase 5: Report and presentation

In this phase, findings and results of the analysis of the conducted experiments were being shared.

4. EXPERIMENT DEVELOPMENT

The experiment is conducted based on the test case shown in Table 2, which consist of 3 test cases. The result of the test case is being discussed in Section 4.

Table 2: Test case throughout the experiment

Test Case	Description
Test Case 1	Run the normal traffic in the topology that does not has a controller to see the normal CPU and RAM utilization
Test Case 2	Initiate three types of attack which are UDP flooding, FTP flooding and Telnet DDoS at topology that does not have a controller to see the DDoS impact towards CPU and RAM utilization
Test Case 3	Initiate three types of attack which are UDP flooding, FTP flooding and Telnet DDoS at topology that has controller implemented with Entropy and PCA to compare the efficiencies of the SDN techniques

4.1 Topology

To run the experiment, the topology design is shown in Figure 1. In this topology, there are no SDN techniques applied; hence there is no controller exist. It consists of 9 switches and 24 hosts. The topology is being used to calculate the impact of the DDoS attack on the victim's resources.

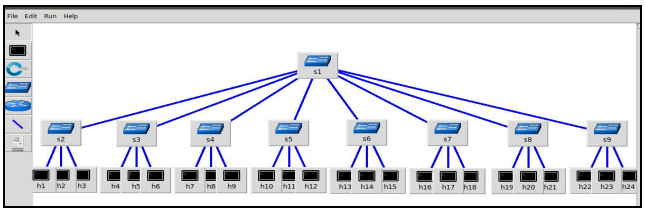


Figure 1: Topology without a controller

There is another topology that is used in the experiment, as shown in Figure 2. SDN concepts are applied with a controller, nine switches as well as 24 hosts. This topology is being used to compare the SDN techniques since the techniques can be implemented inside the controller. Those techniques are Entropy and PCA.

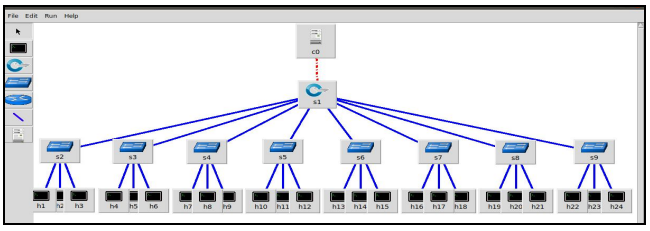


Figure 2: Topology with a controller

4.2 Simulator

Mininet is being used to simulate the SDN topology on a computer with an Intel(R) Core (TM) i5-7200U CPU 2.50 @ GHz. By default, when the Mininet is being run, there will be one switch, two hosts and one controller embedded in that. This situation is applied on a virtual machine running Ubuntu 18.04.2 LTS. This virtual machine comes with 4 GB of RAM and a two-core processor.

4.3 DDoS Attacks

There are three types of different attacks that being launch which are UDP flooding, FTP flooding as well as Telnet DDoS. The environment to launch the attack is the same. The attacks being captured by the Wireshark as it is a network protocol analyzer which it can see what packet that being send in that network. The attacks were launch by using a tool named Scapy. Before the attack reaches the specific target, which is the host 24, the switch will forward the packet to the controller first. Then the controller will make a decision whether to allow it to be forwarded to the network or deny and drop the packet. Once the controller detects it as an attack, it will drop the packet before it reaches the victim. But during the attack occurred, the impact can be seen based on the reading of CPU and RAM utilization.

4.4 SDN Techniques

To mitigate the attack occurs, SDN had being used. Two major techniques that being implemented inside the controller are Entropy and PCA. Both are really useful in mitigating the attack. But in term of efficiencies, it shows slightly different between one another.

i) Entropy

A technique that evaluates the randomness of flow data and highlights the likely attack. Apart from that, it is a lightweight method [27]. It can use between these two components to determine the attack, which is window size and threshold.

ii) PCA

A multivariate optimization technique because it reduces the dimensionality of the attacks attribute [18][29]. There are three fundamental properties of this technique which are:

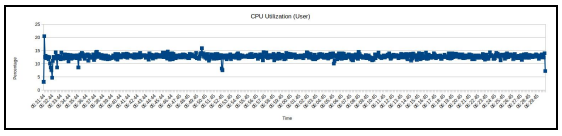
- First, it reconstructs compression of high dimension to become optimal linear.
- Second, the data can directly get from the model parameter
- Lastly, it can perform by matrix multiplication.

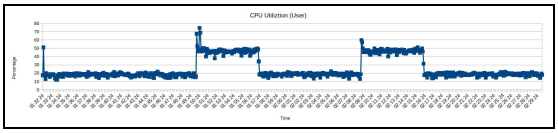
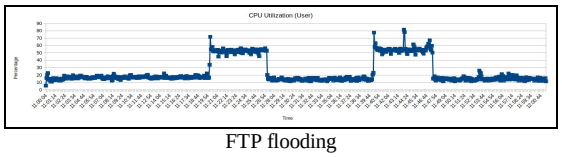
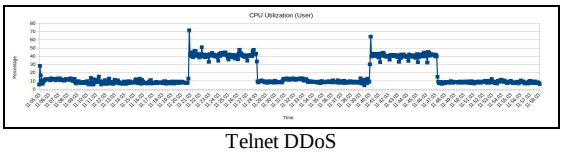
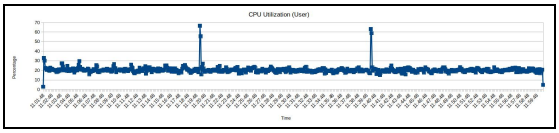
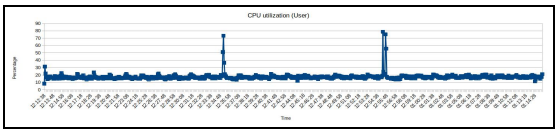
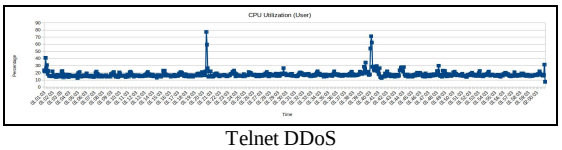
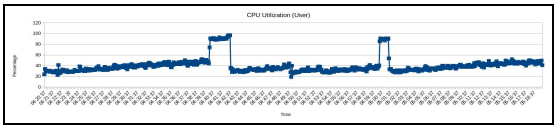
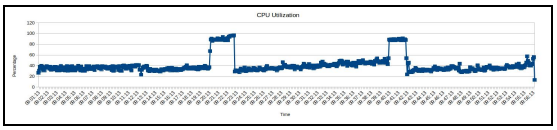
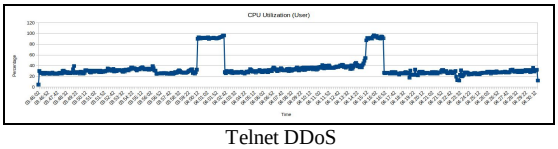
5. FINDING AND ANALYSIS

5.1 Overall Result from the Test Cases

Based on the test cases being given in the previous section, the outcome is as Table 3 below. It shows the reading of CPU utilization of the packet flows. From that, the attack is captured based on the spike exist in that line graph. The attacks that were tested are UDP flooding, FTP flooding as well as Telnet DDoS.

Table 3: Result based on test cases

Test Case	Description
Test Case 1	

Test Case	Description
Test Case 2	 UDP flooding  FTP flooding  Telnet DDoS
Test Case 3 (Entropy)	 UDP flooding  FTP flooding  Telnet DDoS
Test Case 3 (PCA)	 UDP flooding  FTP flooding  Telnet DDoS

5.2 Impact of DDoS Attacks

Two parameters of impact on the victim's resources had been highlighted, which are CPU and RAM utilization.

i) CPU utilization

The impact of DDoS attacks on the victim's CPU is shown in Table 4. The table shows the type of attack that is tested to see the increase of the CPU utilization when the attack occurs by taking the reading of the CPU before and after the attack.

The first two types of attack in the table are from past studies which are HTTP DDoS [24] and UDP Flooding [26]. They show the increasing of CPU utilization up into 20%. And from the experiment that is tested using UDP flooding, FTP flooding and telnet DDoS attack, they show the increasing can achieve more than 20%. This proves that the attacks give a great impact on CPU utilization.

Table 4: CPU utilization before and after the attack

Type of Attack	Before Attack	After Attack	Differences
HTTP DDoS (S.Acharya, N.Pradhan, 2017)	70%	90%	↑ 20%
UDP Flooding (S.Behal, K.Kumar, 2016)	5%	25%	↑ 20%
UDP flooding	13%	47%	↑ 34%
FTP flooding	13%	55%	↑ 42%
Telnet DDoS	13%	41%	↑ 28%

ii)RAM utilization

The value of free space in the RAM of the victim's machine is being compared before and after the attack to examine the RAM utilization. As shown in Table 5, the first two attacks are also being taken from previous studies [24][26]. The outcome from the experiment and the previous studies not likely the same as the previous studies shows that the attack used the RAM up until 2400Mb and 1233Mb. But in the experiment, it just takes up only 494Mb, 1123Mb and 789Mb respectively.

Table 5: RAM utilization before and after the attack

Type of Attack	Before Attack	After Attack	Differences
HTTP DDoS (S.Acharya, N.Pradhan, 2017)	3900 Mb	1500 Mb	↓ 2400 Mb
UDP Flooding (S.Behal, K.Kumar, 2016)	3500 Mb	2300 Mb	↓ 1233 Mb
UDP flooding	2848 Mb	2446 Mb	↓ 494 Mb
FTP flooding	2848 Mb	1930 Mb	↓ 1123 Mb
Telnet DDoS	2848 Mb	2121 Mb	↓ 789 Mb

5.3 Comparison of SDN Techniques

In this section, the SDN techniques are being compared in the term of efficiencies by seeing the time taken needed for the techniques to detect the attack. Table 6 shows a comparison between the techniques chosen, which are Entropy and PCA. The duration of attacks lasts for 6 minutes.

As for the Entropy, the time taken for it to detect and stop the attack is just 5 seconds for all the attacks launch which are UDP flooding, FTP flooding and Telnet DDoS. But, using PCA techniques, it takes a longer time than Entropy which is 2 minutes 25 seconds for UDP Flooding, 2 minutes 35 seconds for FTP Flooding and 2 minutes 20 seconds for Telnet DDoS. This is due to the PCA technique is a machine learning base. Hence, it needs more time to recognize the attack. But, after 2 minutes, it detects it as an attack.

Table 6: Comparison between Entropy and PCA at first attack attempt

DDoS Attack / SDN Technique	Time Taken to Detect Attack	
	Entropy	***PCA***
UDP flooding	5 seconds	2.25 minutes
FTP flooding	5 seconds	2.35 minutes
Telnet DDoS	5 seconds	2.20 minutes

During the experiment, the attack is launched two times within the 1-hour time. This shows a difference in time for the PCA to detect it as an attack at the second attempt, but as for the Entropy, the result is still the same. Results are as shown in Table 7.

Based on the result, this can be concluded that Entropy is more efficient compared to PCA since the time taken for Entropy to detect the attack is lesser than PCA for both the first and second attack attempt even the PCA showing it's improving in detecting the attack. The lesser time is taken in detecting the attack, the more efficient the technique is.

Table 7: Comparison between Entropy and PCA at second attack attempt

DDoS Attack / SDN Technique	Time Taken to Detect Attack	
	Entropy	***PCA***
UDP flooding	5 seconds	1.00 minutes
FTP flooding	5 seconds	1.50 minutes
Telnet DDoS	5 seconds	1.30 minutes

5.4 Future Studies for PCA

As mentioned before, PCA is a machine learning technique that takes time to detect attacks. However, when the next attack occurs, the time taken to detect it as an attack is much lower than the first detection.

Figure 3 shows the time taken for Entropy to detect attacks on the first and second attempts. It clearly shows that there is no difference between the two techniques for detecting both attacks.

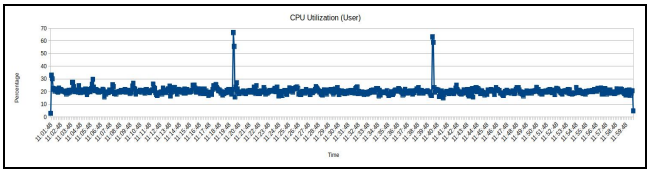


Figure 3: Entropy technique

For the PCA technique, the results of the detection of attacks on the first and second attempts are shown in Figure 4. It is clearly shown that both attempts took a much shorter time to detect the attack than the first attempts. This is because the PCA has been aware of attacks from previous attempts. Therefore, for future research, a variety of style attacks can be studied to find out what type of attack is appropriate for PCA to detect it efficiently.

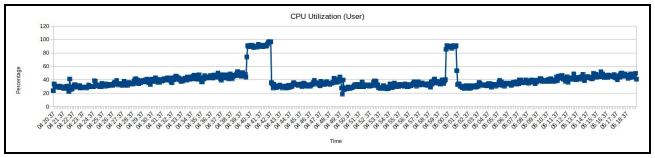


Figure 4: PCA technique

6. CONCLUSION

A secure network is needed to protect our valuables network services from contemporary attacks, such as DDoS, by employing suitable protection technique. Based on the conducted experiments, we were able to identify that the SDN Entropy technique is suitable to mitigate the ever disturbing DDoS attacks.

ACKNOWLEDGEMENT

The authors would like to thank the Ministry of Higher Education Malaysia (MoHE) and Universiti Tenaga Nasional (UNITEN) for funding this study under the Fundamental Research Grant Scheme (FRGS) of Grant No. FRGS/1/2017/ICT03/UNITEN/03/1.

REFERENCES

1. V. R. Sayoc, T. K. Dolores, M. C. Lim, L. Sophia, and S. Miguel, "**International Journal of Advanced Trends in Computer Science and Engineering Available Online at http://www.warse.org/IJATCSE/static/pdf/file/ijatcse68832019.pdf Computer Systems in Analytical Applications**," vol. 8, no. 3, pp. 195–200, 2019.
2. J. Cui, M. Wang, Y. Luo, and H. Zhong, "**DDoS detection and defense mechanism based on cognitive-inspired computing in SDN**," *Futur. Gener. Comput. Syst.*, vol. 97, pp. 275–283, 2019.
3. "**5 Most Famous DDoS Attacks | A10 Networks.**" [Online]. Available: https://www.a10networks.com/blog/5-most-famous-ddos-attacks/. [Accessed: 15-Sep-2019].
4. N. N. Dao, J. Park, M. Park, and S. Cho, "**A feasible method to combat against DDoS attack in SDN network**," *Int. Conf. Inf. Netw.*, vol. 2015-Janua, pp. 309–311, 2015.
5. R. M. Thomas and D. James, "**DDOS detection and denial using third party application in SDN**," *2017*

Int. Conf. Energy, Commun. Data Anal. Soft Comput. ICECDS 2017, pp. 3892–3897, 2018. https://doi.org/10.1109/ICECDS.2017.8390193

6. H. D. Zubaydi, M. Anbar, and C. Y. Wey, "**Review on Detection Techniques against DDoS Attacks on a Software-Defined Networking Controller**," *Proc. - 2017 Palest. Int. Conf. Inf. Commun. Technol. PICICT 2017*, pp. 10–16, 2017.
7. B. H. Rasheed, M. Sivaram, D. Yuvaraj, and A. Mohamed Uvaze Ahamed, "**An improved novel ANN model for detection of DDoS attacks on networks**," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 8, no. 14, pp. 9–16, 2019. https://doi.org/10.30534/ijatcse/2019/0281.42019
8. D. Hyun, J. Kim, D. Hong, and J. Jeong, "**SDN-based network security functions for effective DDoS attack mitigation**," *Int. Conf. Inf. Commun. Technol. Converg. ICT Converg. Technol. Lead. Fourth Ind. Revolution, ICTC 2017*, vol. 2017-Decem, pp. 834–839, 2017.
9. "**What Is a Distributed Denial-of-Service (DDoS) Attack? | Cloudflare**." [Online]. Available: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/. [Accessed: 27-Aug-2019].
10. "**Software-Defined Networking (SDN) Definition - Open Networking Foundation**." [Online]. Available: https://www.opennetworking.org/sdn-definition/. [Accessed: 28-Aug-2019].
11. R. Swami, M. Dave, and V. Ranga, "**Defending DDoS against Software Defined Networks using Entropy**," *2019 4th Int. Conf. Internet Things Smart Innov. Usages*, pp. 1–5, 2019.
12. N. I. G. Dharma, M. F. Muthohar, J. D. A. Prayuda, K. Priagung, and D. Choi, "**Time-based DDoS detection and mitigation for SDN controller**," *17th Asia-Pacific Netw. Oper. Manag. Symp. Manag. a Very Connect. World, APNOMS 2015*, pp. 550–553, 2015.
13. S. Gangadhara, S. N. Hasyagar, and U. Damotharan, "**Deployable SDN architecture for network applications: An investigative survey**," *2019 5th Int. Conf. Adv. Comput. Commun. Syst. ICACCS 2019*, pp. 43–49, 2019. https://doi.org/10.1109/ICACCS.2019.8728347
14. L. Yang and H. Zhao, "**DDoS attack identification and defense using SDN based on machine learning method**," *Proc. - 2018 15th Int. Symp. Pervasive Syst. Algorithms Networks, I-SPAN 2018*, pp. 174–178, 2019.
15. Y. Chen, J. Pei, S. Member, D. Li, and S. Member, "**DETPro : A High-efficiency and Low-latency System against DDoS Attacks in SDN Based on Decision Tree**," *ICC 2019 - 2019 IEEE Int. Conf. Commun.*, pp. 1–6, 2019.
16. S. Murtuza and K. Asawa, "**Mitigation and Detection of DDoS Attacks in Software Defined Networks**," *2018 11th Int. Conf. Contemp. Comput. IC3 2018*, pp. 265–270, 2018.
17. M. Vizv, "**Mitigation of DDoS Attacks in Software Defined Networks**," *2018 Elev. Int. Conf. Contemp. Comput.*, no. January, pp. 123–127, 2015.
18. M. M. Najafabadi, T. M. Khoshgoftaar, C. Calvert, and C. Kemp, "**User behavior anomaly detection for application layer ddos attacks**," *Proc. - 2017 IEEE Int. Conf. Inf. Reuse Integr. IRI 2017*, vol. 2017-Janua, pp. 154–161, 2017. https://doi.org/10.1109/IRI.2017.44
19. B. Pande, G. Bhagat, S. Priya, and H. Agrawal, "**Detection and Mitigation of DDoS in SDN**," *2018 11th Int. Conf. Contemp. Comput. IC3 2018*, pp. 1–3, 2018.
20. C. Gkountis, M. Taha, J. Lloret, and G. Kambourakis, "**Lightweight algorithm for protecting SDN controller against DDoS attacks**," *Proc. - WMNC 2017 10th Wirel. Mob. Netw. Conf.*, vol. 2018-Janua, pp. 1–6, 2018.
21. Y. Liu, M. Dong, K. Ota, J. Li, and J. Wu, "**Deep Reinforcement Learning based Smart Mitigation of DDoS Flooding in Software-Defined Networks**," *IEEE Int. Work. Comput. Aided Model. Des. Commun. Links Networks, CAMAD*, vol. 2018-Septe, pp. 1–6, 2018.
22. X. Huang, X. Du, and B. Song, "**An effective DDoS defense scheme for SDN**," *IEEE Int. Conf. Commun.*, pp. 1–6, 2017.
23. V. I. Arkhipenko, V. A. Dlugunovich, V. Z. Hreben, and S. M. Zgirouski, "**Laser reflectometry of the cathode surface of glow discharge in helium at the atmospheric pressure**," *Light. 2002 Metrol. Test. Tech. Using Light*, vol. 5064, no. D, p. 146, 2003. https://doi.org/10.1117/12.501415
24. S. Acharya and N. Pradhan, "**DDoS Simulation and Hybrid DDoS Defense Mechanism**," *Int. J. Comput. Appl.*, vol. 163, no. 9, pp. 20–24, 2017.
25. C. Buragohain and N. Medhi, "**FlowTrApp: An SDN based architecture for DDoS attack detection and mitigation in data centers**," *3rd Int. Conf. Signal Process. Integr. Networks, SPIN 2016*, pp. 519–524, 2016.
26. M. Sachdeva, G. Singh, K. Kumar, and K. Singh, "**Measuring Impact of DDOS Attacks on Web Services,**" *Int. J. Comput. Sci. Inf. Secur.*, vol. 5, no. 9, pp. 392–400, 2010.
27. R. Wang, Z. Jia, and L. Ju, "**An entropy-based distributed DDoS detection mechanism in software-defined networking**," *Proc. - 14th IEEE Int. Conf. Trust. Secur. Priv. Comput. Commun. Trust. 2015*, vol. 1, pp. 310–317, 2015.
28. S. S. Kolahi, K. Treseangrat, and B. Sarrafpour, "**Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13**," *2015 Int. Conf. Commun. Signal Process. Their Appl. ICCSPA 2015*, pp. 1–5, 2015.
29. H. Sun, Y. Zhaung, and H. J. Chao, "**A principal components analysis-based robust DDoS defense system**," *IEEE Int. Conf. Commun.*, pp. 1663–1669, 2008. https://doi.org/10.1109/ICC.2008.321