Detection Of UDP Attacks In Software Defined Networks Using Fuzzy Logic

Ogunleye G.O., Fasan Oluwabukola

ABSTRACT- Software Defined Network (SDN) is a new network technology in which the control plane and data plane are separated to make the network more flexible and dynamic. Security is one of the biggest challenges in SDN. Due to the controller which serves as the single point of failure, the network can be easily disrupted by Distributed Denial of Service (DDoS) attack. The existing detection techniques have concentrated more on other DDoS attacks but focus less on UDP attack. In the proposed work, we examine UDP based DDoS protocol, prototyped Software-Defined network using mininet and POX controller. We further experiment how UDP attack can be performed in SDN, design a fuzzy model using Fuzzy logic library and detect the UDP attack in a software-defined network using fuzzy logic model. The packets are classified into normal and attack packets. Confusion matrix is used to calculate the accuracy of the model.

Index Terms: SDN, DDoS, Fuzzy Logic, Attacks

1. INTRODUCTION

Over the years, attackers have developed different strategies to launch attacks on computer and network infrastructure. This attack has continued to be on the increase every year and can take the form of browsers, brute force, Denial of Service, worm, malware, web scan and other form of attack. Software components participating in the service of global networks such as Personal Digit Assistant (PDA), supercomputers, home and internet network are usually attackers’ target. Denial of Service (DOS) is the third most common form of attack across the network, it is a coordinated attack usually performed by attackers to overwhelm the computer resources to prevent communication service or prevent legitimate system users from using specific resources. Mitrocosta and Douligeris [7] stated that the two main goals of the attacker are: Bandwidth depletion and Resource Exhaustion. In Distributed Denial of Service (DDOS), attackers use a cluster of hundreds of computers to carry out their coordinated invasion rather than making use of one host machine. Attackers/hackers prefer DDOS attack to DOS because no special skill is needed once attackers can generate enough network traffic. Sager, [12] explained that the attackers’ carry out DDOS attacks for Extortion, Business Competition, and Hacktivism. In Software-Defined Network (SDN), DDOS is the major challenge in SDN where attackers send large numbers of packet to exhaust the network controller. Software-Defined Network (SDN) is a modern technology for computer networks that uses open protocols for controlling switches and routers placed at the network edges by using specialized open programmable interfaces [18]. In SDN, there is a centralized view of network which makes it easy to manage the network and also a clear separation between the forwarding and processing planes. Instead of switches processing the incoming packets, they look for a match of the incoming packet in their forwarding tables and if there is none, it will be sent to the controller for processing. The controller represents the operating system of SDN, the packets are processed and a decision whether the packet will be forwarded to the switch or will be dropped. If the source address is spoofed by the attacker, the switch will not find a match in the flow entry table, the packets are then forwarded to the controller by the switch. The collection and continuous processing of the normal traffic and attack traffic can exhaust the controller, causing the controller to be unreachable to newly-arrived legitimate packets and result in losing the SDN architecture. DDOS attacks are of different types: Volume-based attacks, protocol attacks, and Application layer attacks. User Datagram Protocol (UDP) attack is a form of volume-based attack in which the attacker sends a large number of UDP packet to a random port destination of the target server. It is a rule that the server must respond to all the entire packets sent, but first, the server needs to be replied to thereby exhausting the server connectivity and resources. The attacker may also forge the IP address of the data packets to ensure that the return Internet Control Message Protocol (ICMP) packets do not reach the hosting machine. This causes a huge traffic, making connection to the server impossible and exhausting the network until it goes offline. UDP flooding is one of the most pursued DDOS attacks among the attackers and due to the stateless nature of UDP, the attack is very difficult to detect and can effectively flood the victim with unwanted traffic [6]. Several detection techniques have been used to detect UDP attack such as entropy, deep neural network, support vector machine and genetic algorithm but fuzzy logic has proved to have an advantage over them because it provides a way to describe any definitive conclusions from the vague, ambiguous and imprecise information[16]. Fuzzy logic is a rule-based system that uses parameters from the packets, this parameter has certain features that are peculiar to each packet and can be used to determine whether a packet is normal or an attack packet. Fuzzy logic doesn’t just determine if a packet is good or bad but chose a percentage of attack in each packet. Fuzzy logic uses understandable linguistics term for classification which makes it easy for anyone to understand in this project, we are going to examine a form of Volume-based attack called UDP attack using Fuzzy Logic to detect this form of attack in a Software Defined Network (SDN). According to Debojith et al.[6], UDP attack can effectively flood the victim with unwanted traffic and also makes the detection of the attack to be very difficult because of the stateless nature of the protocol. Therefore it is necessary to detect the attack before it exhausts the network resources and to allow legitimate packet to be processed. Several techniques have been used to detect UDP attack but the fuzzy logic technique has an advantage over them because of its high
2. LITERATURE REVIEW

2.1 Overview of DDOS Attack

The growing use of internet service in the past few years have promoted an increase in the Denial of Service (DOS). Denial of Service (DOS) is the most common method and the most basic application of Distributed denial of Service (DDOS) which is used to disrupt a service in the network. According to Shiaeles et al. [15], DDOS attacks are recognized to be part of cyber warfare tactics which are used primarily for financial gain purposes, it is usually employed for blackmail and extortion. In DDOS attack, the application layer, transport layer and network layers of the network model are the layers involved where the server increases its computational power when a huge number of requests are sent and starts to process every request. The server has a limited capacity to process the number of user request at a time. So, when a huge several spoofed requests or queries are sent to the server, the server gets busy and the legitimate request cannot be processed at that period. Imperva's Q1 2017 global DDOS threat landscape recorded that TCP, UDP, and SYN flood attacks are the most common types of non-amplified DDOS attack in the year 2017.

There are two main classes of DDOS attacks which are bandwidth depletion and resource depletion attacks [17].

- **Bandwidth depletion attack**: is an attack to flood the victim's network with unwanted traffic that blocks all legitimate traffic from reaching the victim. They are characterized as flood attacks and amplification attacks such as UDP attack, smurf attack, ICMP attacks, and Fraggle attack.

- **Resource depletion attack**: is an attack to exhaust the resources of a victim’s system to ensure that the victim is unable to process legitimate service requests. They are characterized as protocol exploits attack and malformed packet attacks such as TCP SYN attack, IP address attack, and PUSH+ACK attack.

There are several types of DDOS Attack, but the most common types of DDOS are:

1. TCP SYN attack

TCP SYN flood attack is a type of protocol exploit based attack in which an attacker can attack either by not sending the expected ACK code back to the client-server that is requesting for connection or by spoofing the source IP address while sending SYN request, the client-server will have to wait for the acknowledgment for some time causing a simple network congestion due to the missing ACK [14].

2. Smurf attack

Smurf attack is a type of amplification-based attack in which attackers sends large ICMP echo request packets directed to IP broadcast addresses with a spoofed IP source address, when all the machines at the intermediary's site respond to the ICMP echo requests, they send replies to the victim's machine and the machine is subjected to
network congestion that could make the network inaccessible [1].

3. ICMP attack
ICMP is a type of flood based attack occurs when the zombies send large volumes of ICMP_ECHO_REPLY packets to the victim system. During the attack, the source IP address of the ICMP packet may be spoofed and The ICMP packets signal the victim system to reply and the traffic exhaustion of the victim’s network connection (Specht and Lee, 2004).

4. HTTP flood
HTTP FLOOD is an application flood based attack that do not require spoofed addresses or high amount of data to be sent in order to attack a server but a simple HTTP requests of GET and POST are sent that require huge amount of data are sent to consume a large amount of bandwidth, taking down the server [5].

5. UDP attack
UDP attack is a type of flood based attack in which an attacker sends a large amount of UDP packets to random ports on the host’s machine and the host constantly checks for application on that port. It replies with ICMP destination unreachable packet if there is no listening application on that port and consumes more resources, making the host unreachable [5].

2.2 Overview of UDP Attack
User datagram protocol (UDP) is a connectionless protocol and sessionless networking protocol which doesn't require a three-way handshake to establish a connection, unlike TCP. It runs with lower latency and is ideal for traffic that doesn't need to be acknowledged, this characteristics makes UDP to be vulnerable to attacks. Anand and Patel [2] grouped UDP attacks as a type of protocol attacks and also claimed that a UDP attack is similar to ICMP flood attacks. Meanwhile, Specht and lee [17] also grouped UDP attacks as bandwidth depletion attacks which is characterized as a flood attack. UDP attack occurs at the transport layer and network layer of the network model. At the transport layer, the port numbers of the source and destination is wrapped around the data to form segments, then the transport layer uses UDP as an engine to transport the segments to the network layer, where the IP addresses are added to the segment to form IP packets. The attackers create a large size of packets sometimes above 8kb because UDP does not specify the limits of the sizes of data packets and send these packets to the host machines. According to Debojit et al [6], UDP attack is one of the most common DDOS attacks among the attacker and due to the stateless nature of UDP, it can effectively flood the victim with unwanted traffic and also makes the detection of the attack to be very difficult.

2.3 Overview of Software Defined Network (SDN)
Software-Defined Network is a new networking technology that has been trending in the networking industry because over the years, they have experienced stagnation to innovation due to limitations of the existing networks. The networking industry can no longer meet the present business and technical requirements with the use of existing networks because of the static nature of the network, which had made it difficult to manage and maintain as more advanced technologies emerge such as mobile devices, cloud services, 4G wireless communication and server virtualization etc. SDN was presented to overcome the limitations of the existing networks by providing a dynamic, flexible, adaptable network to improve the performance and monitoring which also allows administrators to respond quickly to requirement changes with the use of a centralized control console.

Sezer et al [13] stated that the term SDN has been coined in recent years and has been evolving since 1996 but most recently, Ethane (2007) and OpenFlow (2008) have brought the implementation closer into reality. Some of these ideas of the past early effort (such as Ethane, 4D, RCP, DCAN, NETCONF, etc.) in SDN have brought about the characteristics of present-day SDN. However, Kreutz et al [10] reported that SDN was originally defined as a network architecture where the forwarding state in the data plane is managed by a remote control decoupled from the former. SDN was also described in this article, according to Open Networking Foundation (ONF) “In the SDN architecture, the control and data planes are decoupled, network intelligence and state are logically centralized and the underlying network infrastructure is abstracted from the application.” This is the most acceptable definition worldwide. But there is no established definition for SDN in the networking industry, the term given by different network vendors has always evolved around these four characteristics that the network architecture must possess which are:

- The data and control plane are decoupled i.e. the decision on how to forward the packet is separated from the operations that will accomplish the decisions.
- A centralized controller.
- Open interfaces between the devices in the control plane and those in a data plane that allows operations from a single point.
- The network is programmable by external software applications.

The SDN architecture consists of three layers which are: the application layer (Management plane), the control layer (control plane) and the infrastructure layer (data plane). Bawany et al [3] stated that the OpenFlow Protocol was the first and most widely deployed protocol in SDN and OpenFlow protocol defines the communication mechanism that enables SDN controller to directly interact with the data plane.

2.4 Overview of Fuzzy Logic
During the past several years, Fuzzy logic has found numerous applications in fields and research has shown that there are many misconceptions about fuzzy logic. But, one of the principal contributions of fuzzy logic is its high power of precision of imprecision and approximate reasoning. Kahraman et al [9] stated that the founder of fuzzy set theory is Lofti Asker Zadeh who published his first paper on his new theory in 1965 and that Zadeh broaden the foundation of fuzzy set theory by establishing fuzzy similarity relations, linguistic hedges and fuzzy decision
making between 1965 and 1975. The authors also stated that Mamdani developed the first Fuzzy logic controller in 1970 and the success of fuzzy logic was first observed in Japan in the early 1980s which led to the revival of fuzzy logic in the US in late of 1980s. The term “fuzzy” refers to things that are not clear or vague, fuzziness can be found in many areas of daily life especially those that deals with human judgment, decision making and where approximate reasoning are relevant. Fuzzy logic is used basically to handle uncertainties in computer-based problems or making of decisions. Fuzzy logic also introduces the possibility of resolving problems from the human perspective and may take intermediate conditions to provide several solutions satisfactory to the problem presented instead of a single solution from the “false” or “true,” [8]. An example was cited by Ozturk [11] to differentiate between a classical logic and a fuzzy logic, “if the limit for the tallness is 180cm, then a person which is 179cm is short and 181cm is tall in classic logic but in fuzzy logic, we can define a person both as tall and short and short according to the membership value of a membership function between 0 and 1.” Fuzzy logic-based system uses a collection of IF-THEN rules and fuzzy membership functions to make decision on the input data parsed into the Fuzzy Logic Controller (FLC). A fuzzy rule consists of two components: an “if” clause, which is the antecedent of the rule to give condition of application domain; and a “then” clause, which is the consequent of the rule to give an action of control given to the process. Different fuzzy logic controller has been developed over the years but the two most basic fuzzy logic controllers are “Mamdani fuzzy inference system” and “Takagi-Sugeno fuzzy model”. Behroz et al [4] also stated that the Mamdani type is used for all types of the systems while the Sugeno type is mainly used for dynamic non-linear systems.

3.0 METHODOLOGY

1 Algorithm

Fuzzy rules

Fuzzy rules relating to detect the UDP based denial of service attack is given below:

1. if packetRate is Low and ByteRate is Low and Utilization is Low then implication is False Attack
2. if packetRate is Low and ByteRate is Medium and Utilization is Medium then implication is False Attack
3. if packetRate is Low and ByteRate is Medium and Utilization is High then implication is False Attack
4. if packetRate is Low and ByteRate is Low and Utilization is Medium then implication is False Attack
5. if packetRate is Low and ByteRate is Low and Utilization is High then implication is False Attack
6. if packetRate is Low and ByteRate is Medium and Utilization is Low then implication is False Attack
7. if packetRate is Medium and ByteRate is Low and Utilization is Low then implication is False Attack
8. if packetRate is Medium and ByteRate is Low and Utilization is Medium then implication is False Attack
9. if packetRate is Medium and ByteRate is Low and Utilization is High then implication is False Attack
10. if packetRate is Medium and ByteRate is Low and Utilization is High then implication is False Attack
11. if packetRate is Medium and ByteRate is Medium and Utilization is Low then implication is False Attack
12. if packetRate is Medium and ByteRate is Medium and Utilization is Medium then implication is False Attack
13. if packetRate is Medium and ByteRate is Medium and Utilization is High then implication is False Attack
14. if packetRate is Medium and ByteRate is High and Utilization is Low then implication is Attack
15. if packetRate is Medium and ByteRate is High and Utilization is Medium then implication is Attack
16. if packetRate is Medium and ByteRate is High and Utilization is High then implication is Attack
17. if packetRate is High and ByteRate is Low and Utilization is Low then implication is Attack
18. if packetRate is High and ByteRate is Low and Utilization is Medium then implication is Attack
19. if packetRate is High and ByteRate is Low and Utilization is High then implication is Attack
20. if packetRate is High and ByteRate is Medium and Utilization is Low then implication is Attack
21. if packetRate is High and ByteRate is Medium and Utilization is Medium then implication is Attack
22. if packetRate is High and ByteRate is Medium and Utilization is High then implication is Attack
23. if packetRate is High and ByteRate is High and Utilization is Low then implication is Attack
24. if packetRate is High and ByteRate is High and Utilization is Medium then implication is Attack
25. if packetRate is High and ByteRate is High and Utilization is High then implication is Attack
26. if packetRate is High and ByteRate is Medium and Utilization is Medium then implication is False Attack
27. if packetRate is High and ByteRate is Medium and Utilization is High then implication is Attack

Fuzzy Rules Training Parameters are:

Input parameters:
- Packet rate
- Utilization
- Byte rate

Output parameter:
- Implication

The fuzzy rules are a combination of input parameters. These rules may result in a normal mode or attack mode.

- **Normal mode**: An attack has not occurred when the network traffic detected as normal traffic.
- **Attack mode**: An attack has occurred when we have an abnormal traffic.

The linguistic values for the basic subsystem regarding detecting system are LOW, MEDIUM and HIGH. These input and output parameters were chosen according to the IG (Information Gain) algorithm to infer the UDP attack which is based on the concept Entropy. In order to generate the optimized fuzzy rules and fuzzy sets, the Rule Base Extension using Default Set Shapes (RBE-DSS) method was applied.
3.4.3 Membership Functions
Trapezoid membership function was used in this project. It describes the level of impact of UDP attack on network, where 3 membership functions of network traffic are low, medium and high.

Figure 2.0: Membership functions of the input parameters. The value being set for the membership functions for each parameters are:
- **Packet rate**
  - Low = 0.000 - 100.0
  - Medium = 101.0 - 600.0
  - High = 601.0 - 1000.0
- **Utilization**
  - Low = 0 - 150000.0
  - Medium = 150001.0 - 100000000.0
  - High = 100000001.0 - 200000000.0
- **Byte rate**
  - Low = 0 - 400000.0
  - Medium = 400001.0 - 10000000.0
  - High = 100000001.0 - 300000000

The outcome of the Input variables that result to the implication (False attack and attack), the values are being set as:
- **Implication**
  - False attack = 0.000, 0.5
  - Attack = 0.6, 1.0

3.4.4 Performance Analysis
Performance of the system will be evaluated using the following criteria: precision, recall, and overall accuracy.
- Precision = \( \frac{TP}{TP+FP} \)
- Recall = \( \frac{TP}{TP+FN} \)
- Overall accuracy = \( \frac{TP+TN}{TP+TN+FN+FP} \)

Where,
- True Positive (TP): These are the number of attack packets that are correctly identified.
- True Negative (TN): These are the number of normal packets that are correctly classified.
- False Positive (FP): These are the number of packets that are incorrectly classified as an attack whereas they are normal packets.
- False Negative (FN): These are the number of attack packets that are incorrectly classified as normal packets.

3.4.5 Process Analysis

3.5 Design
3.5.1 System Architecture
The system architecture shows the conceptual view of the proposed system showing the various components and how they are interconnected.
The datasets were used to analyze the behavior of attack detection. We have normal traffic dataset, attack traffic dataset collected from the software-defined network in Figure 4.0. The dataset is divided into two subsets which are the training data and testing data. In the training phase, the data are first classified into classes of subsets where attack traffic and normal traffic are separated. Features are selected and identified from both attack data and normal data of the input data. Then, the fuzzy rules are given to the fuzzy system manually. These fuzzy rules are used to train the data and also to help the fuzzy system to apply the rules, in order to improve the effectiveness of the system. In this project, three input parameter and a single output parameter of Mamdani fuzzy inference system was used for this purpose. Then in the fuzzy system, each input and output fuzzy set will be defined with the three membership (low, Medium and High) functions using trapezoid membership function for fuzzification scheme. In the testing phase, test data from each of the datasets and is given to the designed fuzzy logic system. The fuzzy rules given are fed to the fuzzy rule base for learning the system. The Fuzzifier converts the input parameter into linguistic variable using the membership function, the output of the fuzzifier is then sent to the inference engine where the fuzzy rules are matched with the test input data to detect whether the test data is an attack data or a normal data. The output of inference engine contains the linguistic variable with the membership function, the defuzzifier converts the linguistic variable to crisp value. The crisp value obtained from the fuzzy inference engine is varied between 0 to 1.0, where ‘0.0 to 0.5’ denotes that the data is normal, and ‘0.6 to 1.0’ denotes attack.
Flowchart
The flowchart of the system is shown below:

Start

Simulate SDN traffic flow

Capture UDP packets

Is source IP blacklisted

No  Deny packets

Yes

Parse packets as input into the fuzzy controller

Match a fuzzy rule

No  Drop packets

Yes

Classify packets

Stop

Figure: 5.0: Flow chart diagram
Figure 5.0 shows how the flow chart starts by stimulating the Software-Defined network traffic flow, then the UDP packets are captured and stored as PCAP file. If the source IP address is blacklisted, it denies the packets but if the packets are not blacklisted, it parsed the packets into the fuzzy logic controller. Once it matches with a fuzzy rule, the packets are then classified into normal packet or attack packets, if it doesn’t matches with any of the fuzzy rules, it drops the packets.

4.0 Implementation, Results and Discussion

4.1 NETWORK SETUP

The experiment was done on a HP notebook laptop with a Celeron processor, 2.3 GHz of processor speed, 64-bit operating system, 4GB of ram, and local disk of 464 GB. The network was set up in a virtual environment with the use of Virtualbox, a virtual machine (VM) was created on the Virtualbox where the Mininet was installed. The operating system of the host system is Windows 10 pro and the guest system (VM) is Linux Ubuntu 16.04 LTS. Mininet version 2.2.2 was to run native on Linux and supports Openflow protocol version 1.0. Using Mininet, a single-type network topology of an Open Virtual Switch (OVS) of IP address (127.0.0.1) and 4 hosts with the IP addresses of 10.0.0.1, 10.0.0.2, 10.0.0.3 and 10.0.0.4 was created, which was connected to the remote POX controller of IP address (127.0.0.1). OVS is a software switch that runs both on hardware and software. Fig 6.0 shows the topology of the network which was built with miniedit, a graphical user interface for simple network editor for Mininet. Where "co" stands for the controller, “s1” stands for the switch and “h1-h4” stands for the hosts.

Host h1 and host h2 generates or sends the normal network traffic while host h3 is the attack host and host h4 is the victim target to generate attack network traffic. Wireshark is used to capture the UDP packets from the SDN. These datasets from the SDN are used for training, to generate the fuzzy logic model and to detecting attack.

Below are the screenshots of the setup of software-defined network in a virtual environment: Figure 7.0 shows how software defined network was created using mininet running on the VirtualBox. The network consists of four hosts, a switch and a remote controller.
Figure 7.0: software defined network using Mininet.

Figure 8.0 shows how software defined network is connecting to the remote controller POX.

Figure 8.0: connecting to POX.
Figure 9.0 shows how Hping3 tool is used to flood the software defined network with UDP attack packets with a spoofed source IP address by running Hping3 command on host3.

```
root@bookee-VirtualBox:~ # sudo hping3 10.0.0.4 --udp --flood --spoof 10.0.0.8
hping 10.0.0.4 (h3-eth0 10.0.0.4): udp node set, 28 headers + 0 data bytes
hping in flood mode, no replies will be shown
[send_ip] send; Operation not permitted
```

Figure 9.0: generating UDP attack packets.

Figure 10.0 shows how Wireshark is used to capture the UDP attack packets with a spoofed source IP address from software defined network and also store the packets as PCAP file.

Figure 10.0: capturing the UDP attack packets.

Figure 11.0 shows how Hping3 tool is used to generate normal UDP packets by running Hping3 commands on host1.
Figure 11.0: generating UDP normal packets.

Figure 12.0 shows how wireshark is used to captured the UDP normal attack packets being sent from host1 to host 2 from software defined network and also store the packets as PCAP file.

4.1.2 System Graphical User Interface

Figure 13.0 shows the administrator login panel which is used by administrator to login into the system. The panel requires a username and a password to prevent unauthorized user to have access to the system. Wrong details supplied by the administrator will result to error.
The Extract PCAP file tab shown in figure 14.0 enables the user to import UDP dataset collected from the software defined network by selecting a PCAP file and extracting it. The raw form of the PCAP file will be displayed after extraction.

The process features tab shown in figure 15.0 enables the user to process the features selected which are the packet rate, Byte rate and utilization from each UDP data packets for classification.
Figure 15.0: Process Features Tab.

Figure 16.0 shows the fuzzy rules tab which view the fuzzy rules which are being set up for analysis with their membership functions (low, medium and High) in order to classify the UDP packets into normal packet or attack packet.

Figure 16.0: ViewFuzzy Rules Tab.

The classifier tab shown in figure 17.0 enables the user to start the fuzzy classifier to classify the packets into attack or normal packets based on the fuzzy rules and also display the result of the classifier.
4.3 Testing

To test the correspondence of this system with the stated system requirements in my chapter three, the quantitative testing method was used where metric such as accuracy, precision and recall generated using confusion matrix.

The Table 1 below shows the result of the classification of the predicted class using confusion matrix against the actual class.

<table>
<thead>
<tr>
<th>Actual Class</th>
<th>Predicted Class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Normal</td>
</tr>
<tr>
<td>Anomaly</td>
<td>TP</td>
</tr>
<tr>
<td>Normal</td>
<td>FP</td>
</tr>
</tbody>
</table>

Accuracy = \frac{TP + TN}{TP + FN + FP + TN} = \frac{141 + 32}{141 + 13 + 14 + 32} = \frac{173}{200} = 86.5%

Precision = \frac{TP}{TP + FP} = \frac{141}{141 + 14} = \frac{141}{155} = 0.9096774193

Recall = \frac{TP}{TP + FN} = \frac{141}{141 + 13} = \frac{141}{154} = 0.9155844155

CONCLUSION AND FUTURE WORKS

Distributed denial of service (DDOS) attack is one of the bottlenecks of software-defined network. The attack can easily be done by flooding the controller of the software-defined network (SDN) with data packets. Here in this paper, we examined a form of Volume-based DDOS attack called UDP attack, it showed how UDP attack can be performed in a software-defined network and also reviewed existing detection techniques that have been used to detect UDP attacks. Furthermore, a model was designed to detect UDP attack in software defined network using fuzzy logic. This paper used the prototyping of software defined network using Mininet as the network emulator, generating UDP packets using Hping3 tool, capturing the normal traffic and the attack traffic using a network protocol analyzer called Wireshark and building a fuzzy based model using Jfuzzylite library to classify the packets as a normal packets or an attack packets.

5.1 Future Works

There are some works that could be implemented in the future to improve the security of the Software-defined network.

For further works, the following enhancements could be implemented:

1. Mitigation of the UDP flood attack
2. Implementation of Software Defined Network (SDN) in a real world environment with real network traffic in order to check the effectiveness of our technique.
3. Detection of the attack in a multi-controller software-defined network
REFERENCES


