Mitigating Address Spoofing Attacks in Hybrid SDN

Fahad Ubaid
Computer Science Department
University of Engineering and Technology, Taxila
Taxila, Pakistan

Faisal Bin Ubaid
Computer Science Department
Chongqing University, Chongqing
Chongqing, China

Rashid Amin
Computer Science Department
University of Engineering and Technology, Taxila
Taxila, Pakistan

Muhammad Muwar Iqbal
Computer Science Department
University of Engineering and Technology, Taxila
Taxila, Pakistan

***Abstract*—Address spoofing attacks like ARP spoofing and DDoS attacks are mostly launched in a networking environment to degrade the performance. These attacks sometimes break down the network services before the administrator comes to know about the attack condition. Software Defined Networking (SDN) has emerged as a novel network architecture in which date plane is isolated from the control plane. Control plane is implemented at a central device called controller. But, SDN paradigm is not commonly used due to some constraints like budget, limited skills to control SDN, the flexibility of traditional protocols. To get SDN benefits in a traditional network, a limited number of SDN devices can be deployed among legacy devices. This technique is called hybrid SDN. In this paper, we propose a new approach to automatically detect the attack condition and mitigate that attack in hybrid SDN. We represent the network topology in the form of a graph. A graph based traversal mechanism is adopted to indicate the location of the attacker. Simulation results show that our approach enhances the network efficiency and improves the network security**

Keywords—Communication system security; Network Security; ARP Spoofing Introduction

I. INTRODUCTION

Software Defined Network (SDN) is a new paradigm shift in a networking environment that brings a lot of new innovations and revolutions in traditional networking techniques. It aims to resolve the several limitations of the traditional networks by decoupling the control plane from the data plane. In SDN, network devices i.e., switches, routers have become simple forwarding devices which only implement the data plane logic [19]. The control or network intelligence is implemented in a centralized unit called controller. Different applications for routing, load balancing, network measurement etc. are implemented on the controller [1][2].

Although there are lots of benefits of SDN, yet it is not widely adopted by the organizations due to budget constraints, the effectiveness of traditional routing and some other reasons. An organization has to establish a new network from scratch to adopt SDN paradigm. Recently, a new network architecture is proposed that is based on a limited number of SDN switches deployed among legacy switches. This type of network is called Hybrid SDN. If an organization wants to update its traditional network to SDN, it needs to change the entire network devices to SDN-based devices, which requires a lot of money to buy new devices. In order to save this extra cost, a Hybrid SDN paradigm is adopted to take complete advantages of SDN [3][4].

In SDN network, security mechanisms are adopted to protect users from a different type of attacks. New kinds of attacks like (Link Flooding Attack) LFA [5] and other DDoS [6] attacks can be launched in the network through (Address Resolution Protocol) ARP Spoofing [7] or IP Spoofing method. ARP or IP Packets are usually used to know the MAC address or the IP address of the system in the network. These packets can be modified easily by an adversary party and the MAC address or the IP address can be changed to a particular host from the adversary party. Authors in [8-10] discuss the techniques to prevent these attacks in SDN. However, currently in hybrid SDN, no proper mechanism to deal with these types of attacks. These attacks further lead to Man-in-the-middle attack, eavesdropping, modification attack and masquerade attack.

In this paper, we propose an automatic ARP spoofing detection and mitigation mechanism for hybrid SDN. This new mechanism prevents the LFA, ARP Spoofing and DDoS attack in hybrid SDN. Our solution adds a separate module (server) in the network where ARP packets are received. Topology information of the whole network is collected at the proposed server and flows are installed on devices to get ARP traffic. Furthermore, ARP packets are analyzed for a possible attack in the hybrid SDN. In this new mechanism, SDN controller is protected from attackers by diverting unnecessary processing to the proposed server. Furthermore, a graph based traversal method is adopted to detect the proper location of the attacker. Our research contributions are

- We are considering a newly emerging network architecture called hybrid SDN.
- To the best of our knowledge, we are the first to deal with this problem in Hybrid SDN.
- For hybrid SDN, we identified the problem that ARP spoofing can poison the network topology. Due to these attacks, different types of applications running on the controller are badly influenced. Furthermore, it may

result in the form of entire network failure. We address this problem as follows.

- We automatically get the network topology information from legacy switches, SDN switches and also from DHCP server at proposed server.
- We construct a graph for the network topology having connectivity information of all users.
- We installed flow rules on the SDN switches and configure the legacy switches to forward ARP packets to the server.
- At the server, we analyze the ARP packets to detect the possible attack condition.

Rest of paper is organized as Section II presents the related work. Problem definition is explained in Section III. The proposed solution is described in Section IV. Implementation and performance evaluation is presented in Section V and Section VI concludes the paper.

II. Related Work

Masoud [9] describes two different mechanisms to handle ARP spoof attacks, one is SDN_Dynamic and the other is SDN_static. These two mechanisms are used to detect ARP spoof packets in the network but this scheme creates an overhead at the controller and can decrease the performance of the network. For example, in this case, if adversary party continuously launches an attack then controller analyze all the packets and this will increase the load of the controller and decreases the performance of the controller. In this situation, the controller cannot block malicious traffic at the switch. Figure 1 shows system design for this approach.

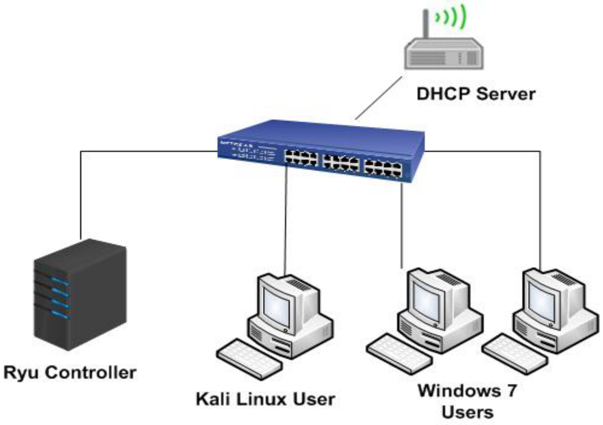

Fig. 1. System Design for [9]

Huan Ma et al. [18] in data centers ARP broadcast storm can be handled in the SDN environment by using SDN switches. Because for every packet received on the SDN switch, flow rule is checked if flow entry is not found then this packet is forwarded to the controller. In a traditional environment, ARP broadcast storm cannot be controlled and this creates a lot of traffic in the network and may become the cause of traffic congestion. Data centers consist of many VMs and multiple network domains. If a VM is moved from one network domain to another network domain then One of the VMs launches ARP packet for the moved machine then this creates ARP broadcast storm in the network and a lot of overhead. To prevent this overhead one can use SDN technology and can control this extra type of traffic from the controller [8] [13].

Roberto di Lallo [14] presents the features of SDN that ARP packets can be controlled through SDN switches in multiple subnets, limiting the ARP traffic at the edge switch of the subnets. The controller keeps the information of all the network devices and from this information, ARP request packet can be controlled at the edge of the network. For this purposes, controller installs the required flow rules at the switch. Controller also keeps track of the network devices in the table called CAT (Controller ARP Table). This table is updated time to time when new requests for the resources of the network arrived.

Fabian Schneider [15] describes how to handle ARP traffic in SDN. ARP traffic is a big problem in SDN environment if it is not controlled in a proper way. It may also be generated if network devices are not configured properly. This bulk of traffic created by ARP packets causes an unnecessary overhead on the network. This issue has been tackled by properly configuring CAT (Controller ARP Table) table and installing flow entries in the SDN switch properly.

Sezer et al. [22] discuss issues of performance, security, scalability and interoperability when deploying carrier grade network based on software defined networking. After analyzing performance vs programmability tradeoff in detail, the author in [22] concludes that hybrid SDN is suitable for old traditional networks. The scalability issues with respect to the communication overhead between the switches and controller, the communication between the controller in the multi-controller environment, maintenance of the backend database in controller analyzed and conclude with the suggestion of hybrid approach where SDN node may share some load to reduce communication and processing overhead of the SDN controller. Security issues in SDN investigated as the centralized controller and the switches may be attacked through DOS attack so the security model must be defined to secure the SDN controller and switches by using currently available security mechanism. The issue of interoperability reviewed in SDN deployment which is desirable, because the complete transition from traditional to SDN paradigm is not possible in most of the cases.it is suggested that the protocol and standard should be made for interoperability between the SDN and legacy devices.

Lei Wang [5] describe Link Flooding Attack is a new type of DDoS (Distributed Denial of Service) attack. In DDoS attack, legitimate or authorized user cannot gain access to the network resources. In this case, adversary party attacks the target server to cut down the resources. LFA is an advanced type of DDoS attack in which selected group of connectivity links to the server is under attack with a different type of malicious traffic. In this attack, the server cannot distinguish the malicious traffic from regular traffic. Due to this attack, the performance of the network and the server affected very badly and the legitimate user cannot gain the access to the server [16].

According to Michael [17] Man-in-the-Middle Attack against Open Daylight SDN Controller, exploit many vulnerabilities of the SDN Network. The author raised many security issues of the SDN environment and showed that controller is a single point failure of the network. The author performed an experiment of Man in a Middle attack by using ARP spoofing method. The author succeeded to launch an attack and intercepted the traffic between a client and the Open Daylight controller [20].

For the large scale SDN enterprises the unified virtual monitoring function (SuVMF) middlebox architecture is introduced in [21]. The objective of the SuVMF is to monitor traffic and resources of the large enterprise network to ensure the effective use and security of resources. SuVMF architecture composed of three main components namely Filtering and Common Processing (FCP) Module, Transformation and Adaptation (TA) Module, Basic Common Monitoring (BCM) and User Defined Monitoring (UM) Module as shown in Figure 2. Filtering and Common Processing Module is responsible for collecting network events, event mitigation function, packet and flow filtering, time stamping, anomaly traffic detection, host detection and other related functions. Transformation and Adaptation Module provides communication between remote managements and controllers by supporting OpenFlow and SNMP protocols. OpenFlow Statistics collections Proxy (OSP) is responsible for the collection of statistics from the OpenFlow switches and provides it to the controller. Detection and Mitigation Abnormality (DMA) component is responsible for the detection of abnormal behavior of different components in the network. The proposed middlebox architecture provides integrated services for the hybrid SDN network and reduces the load on the controller.

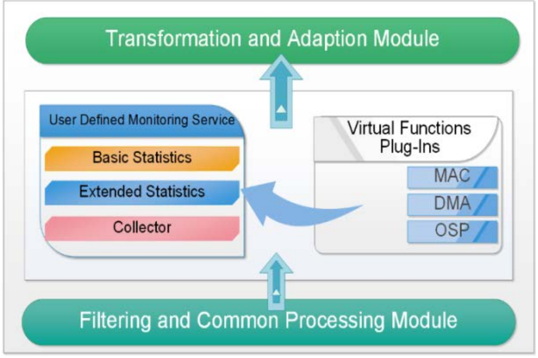
Transformation and Adaption Module
User Defined Monitoring Service
Basic Statistics
Extended Statistics
Collector
Virtual Functions Plug-Ins
MAC
DMA
OSP
Filtering and Common Processing Module

Fig. 2. SuVMF basic and User Defined Monitoring Functions

Ahmed et al. [10] describe different traditional network threats like ARP Spoof attack or Distributed Denial of Service attack can affect the whole network badly. In traditional networks, these kinds of threats cannot be eliminated completely due to lack of centralized control of the network. But there are third party tools to mitigate such threats. For example, Dynamic ARP Inspection (DAI) is a Cisco Device protocol and it can be used to check ARP spoofing packet but for this tool, the network must be configured with all cisco devices having same protocol. But there is also a chance of false positive attack that affects the network performance. The author presents the solutions for mitigating such attack by monitoring port level packets. But this solution is limited to the only single controller and for simple local Area Network (LAN).

From the above literature, it is clear that ARP spoofing and DDoS attacks have not been discussed in hybrid SDN. It is a big issue because, in a communication network, ARP protocol is mostly used to get IP/MAC information. Due to these attacks, packets may be traveled to an unauthorized node. Thus, security of the network is at risk as shown through some examples in Problem Statement. To mitigate these attacks in hybrid SDN, there are following challenges.

- Getting network topology information from legacy switches using customized mechanism
- Identification of legacy switches and their interfaces forming a hybrid SDN
- Getting ARP packets from legacy devices at controller and at proposed server need to be customized technique.
- Analysing the ARP packets for possible threats at proposed server.
- Identification of devices that are generating malicious traffic
- Blocking the malicious devices for further processing against controller

III. Problem Definition

SDN controller is the main component of the SDN network, due to his reason SDN controller becomes more vulnerable to several types of attacks. Most common type of attack is ARP spoofing attack in which malicious node sends ARP packets. Successful attacks can effectively poison the network topology information and a fundamental building block for core SDN components. With the poisoned network visibility, the upper layers services and application of SDN controller may be completely misconfigured and badly influenced. This situation leads to serious hijacking, denial of service attacks and network failure in some cases. Several SDN studies show that all current major SDN controllers (e.g., Floodlight, Open Daylight, Beacon, and POX) are affected by these attacks. if such fundamental network topology information is poisoned then all the dependent network services become immediately affected and causing catastrophic problems. For example, the routing services/apps inside the controller can be manipulated to incur a black hole route or man-in-the-middle attack.

Suppose there is an enterprise network for an organization as shown in Figure 3. There are four legacy switches l1, l2, l3, l4 and two SDN switches SDN switch A and SDN switch B. A controller is connected to these SDN switches. Eight users PC1-PC8 are connected to legacy switches as per requirement. An attacker’s PC is connected to the network when an attack is launched.

Ideal Condition

In Ideal condition as shown in figure 3, PC1 with IP address 10.0.0.1 wants to communicate with PC 5 with IP address 10.0.0.6. PC1 does not have the MAC address of PC5. PC1 send the ARP packet to the legacy Switch and legacy switch broadcast this packet. This packet gets received at SDN switch A. SDN switch A checks the flow entries for received ARP packet. If it does not find the flow entry for that packet, then the packet is forwarded to the controller. The controller checks the packet and finds its path to the destination and generates the flow rules for this packet. Now packet moves according to flow entries installed on the corresponding switches and receives the destination MAC address.

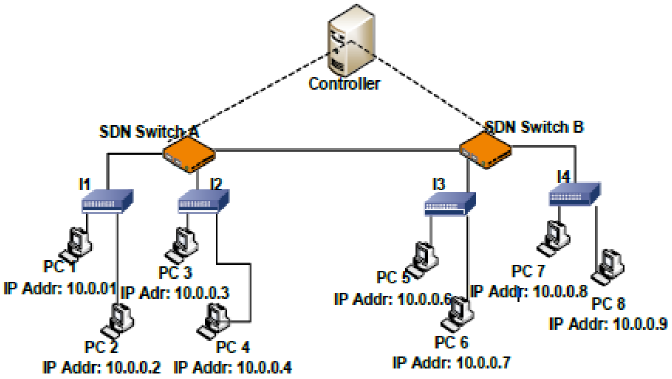

Fig. 3. Ideal Condition

A. Attack Condition

In Attack phase, Kali Linux user as an attacker launches broadcast Gratuitous ARP message with the IPv4 address of PC5 i.e 10.0.0.6. Gratuitous ARP is a broadcast packet that is used by network devices to announce any change in their IPv4 address or MAC address. By sending a Gratuitous ARP message with the IPv4 address of PC5, attacker deceives as PC5 and captures all the network traffic of PC5 as shown in Figure 4.

Once the attack is successful, network information kept by the controller is poisoned and the adversary can take control of the network and capture all the network traffic. After getting network information Kali Linux user with IP address 10.0.0.6 launches a DDoS attack. Due to this attack, the controller continuously remains busy with PC5, while all other users are waiting for a response to their queries. In this way, the whole network is affected due to these attacks. Consequently, controller performance is degraded and legitimate users are unable to get a response from the controller.

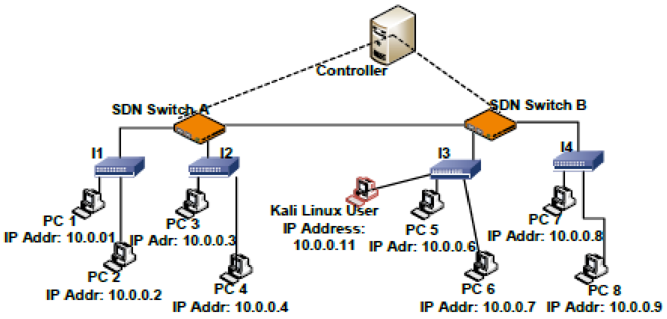

Fig. 4. Attack Condition

These problems of ARP spoofing and DoS attacks occur in hybrid SDN. SDN controller only controls the data flow through SDN switches. In addition to these, the legacy switches use traditional network protocols to forward the data. In order to configure legacy devices in hybrid SDN, customized mechanism is required to be implemented by the SDN controller. To mitigate these attacks in hybrid SDN, an intelligent attack detection and network recovery mechanism is required.

IV. Proposed Solution

In order to handle the problems of ARP spoofing and DDoS attack as discussed in the problem statement, we proposed an automatic network device identification mechanism, which detects the ARP spoofing attacks in hybrid SDN and mitigates these attacks with the help a proposed server. We model the Hybrid SDN (HN) as HN = (L, D), where L is a set of the undirected edges and D denotes the set of nodes (devices). D is subdivided into two subsets; T consists of traditional (legacy) switches, and O consists of both Openflow based SDN switches and a controller. Thus, $D = T \cup O$. A path from source $s \epsilon D$ to destination point $t \epsilon D$ such that $s \neq t$ is represented as a list of traversed links, the mathematical path is represented as $r(s, t) = \{s, v1, v2...vk, t\}$ and where v1, v2 ...vk $T \cup O$.

In Figure 5, we have shown the overall system in which an individual server is used to handle the ARP requests. We have implemented our proposed solution on this server. This solution consists of multiple components. The first component is used to get topology information from SDN switches and legacy switches. A customized algorithm is used to get topology information from legacy switches through SDN switches. The second component installs the flow rules on the switches and configures the legacy switches so that ARP traffic is forwarded to proposed server. The third component is consisting of the modules that deal with ARP requests generated by different users.

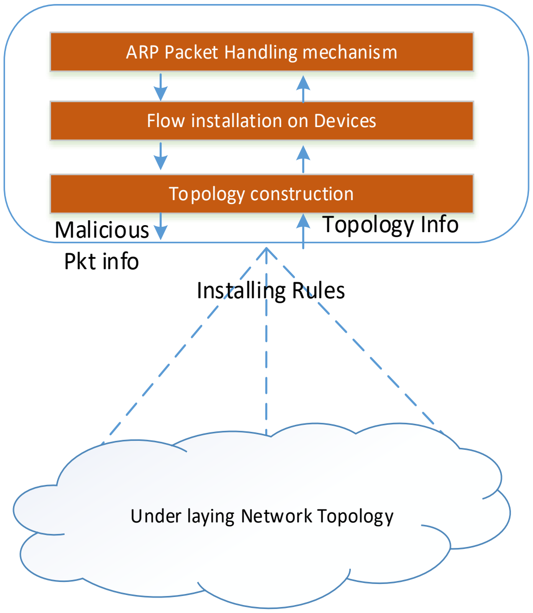
ARP Packet Handling mechanism
Flow installation on Devices
Topology construction
Malicious
Pkt info
Topology Info
Installing Rules
Under laying Network Topology

Fig. 5. Overall System Design

A. Topology Information

We get the network topology information of SDN and legacy devices at the proposed server. An Openflow device exchanges its link state information with proposed server after fix time interval. Link state information of legacy devices is collected from the remote log information of legacy devices switches. Thus, after getting the link state information from all forwarding devices, the edges are stored in a set E and the nodes are stored in a set V. We construct an undirected graph G where forwarding devices are represented as nodes, and links are represented as edges.

In graph construction algorithm, an edge from E and its respective vertices are selected and added to graph G. Then next edge and its respective vertices are selected, and then added to G. This process is repeated till all edges and vertices are added to G. Algorithm 1 explains the graph construction.

Algorithm 1: Graph Construction
Input: L is No of Links, N is No. of Nodes (Devices)
Output: An undirected Graph R

```
R= {0}
While (Links or Nodes are presents)
    Select the Link from the L and Node from N
    if Link connects two Nodes in different subsets then
        merge the subsets;
        add the link to R;
    end if
    if all the subsets are merged then
        the instance is solved
    end if
end while
```

B. Installation of Flow rules on switches

After getting topology information from all the devices in the network, we need to install flows on the switches so that ARP traffic may be directed towards the proposed server for analysis. In order to install flow rules on all SDN switches, we instructed the controller to install flow rules at the switches. Legacy switches are configured to forward ARP traffic towards the SDN switch. Once we have got ARP traffic on proposed server then further analysis is performed on it. The following algorithm explains the flow rule installations for ARP packets as follows:

Algorithm 2: Installation of Flow rules
Input: Number of Packets,
Output: Route to forward packet

```
Controller gets switches information
Controller installs flow rules on switches for Packets
if (Pkt belongs to ARP) || (Pkt.dest == FF:FF:FF:FF:FF:FF)
    Pkt sent to the Specific Port for verification
else
    Pkt Sent to Controller || Forward according to Flow
end if
```

C. Detecting ARP Spoofing attack

After getting topology information from all devices in the network and installation of flow rules, we formed a graph that stores the whole network information. This information is used in verification of APR request generator. In order to detect the attack condition, proposed server checks the packets of a particular host. At first step, it checks that the packet either belongs to our network or not. Secondly, it checks that ARP request belongs to this network or not. If ARP request belongs to the corresponding network, then appropriate action is taken. Furthermore, we explained it in following scenarios.

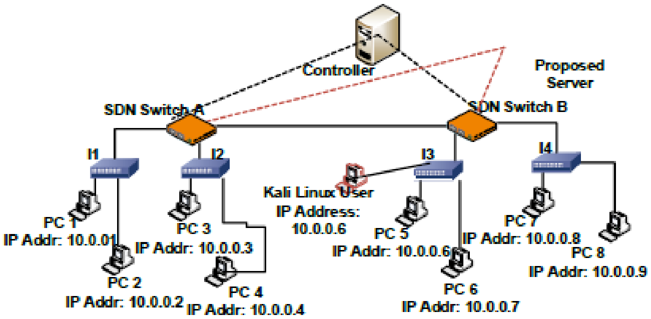

Fig. 6. Proposed Solution Scenario

In the first scenario, Whenever ARP packet is generated by a user in the network and if the user is attached to a legacy switch then this packet is forwarded to SDN switch. When the packet reaches an SDN switch, flow entries for that packet are checked. If the flow entries or rules did not match at the switch, then SDN switch sends the packet to the proposed server as shown in Figure 6. This packet is analyzed by the proposed server for possible attack scenario as describe in algorithm 3. If this packet belongs to our network, then it would be forwarded with a response by the proposed server otherwise, it will be dropped.

For example, when PC1 with IP address of 10.0.0.1 launches an ARP packet according as shown in Figure 6. The legacy switch receives the ARP packet and it forwarded to nearest SDN switch. SDN switch checks the type of received packet and if it is ARP request then it is forwarded to our proposed server. The server examines the packet whether it belongs to our network or from outside. if the packet belongs to our network then it will be entertained with ARP reply message to PC1 via SDN switch and further communication is possible. On the other hand, if the packet does not belong to our network means its IP and MAC addresses are not matched with the database then it is dropped. Because if this ARP packet is not dropped then it may be get modified and used by any adversary party to launch an attack.

In the second Scenario, if an adversary party sends a packet to the network and pretends like a legitimate user by spoofing the IP address of the other user. Then a packet of this user looks like our network and in that case over server checks the IP address with the all recorded MAC to IP mapped table. If the entry is found, then check the source MAC address of the packet with the mapped MAC address. If the entry is matched with MAC address, then server responses with appropriate MAC address otherwise server will drop the packet. If multiple numbers of packets are generated from sending node, then the corresponding port also be blocked.

Algorithm 3: Detection of ARP attack
Input: ARP Packets or Broadcast Address, n nodes,
Output: Get flow rules

```
Initialize CAT [] table
for i in range(1,n)
    Add IP address and MAC Address in CAT[]
end for
if (Pkt.src not in CAT[] and Pkt.dest not in CAT[])
        drop the Pkt
    else
    if (Pkt contain ARP)
        Check IP and MAC Addresses match in CAT[]
        send IP/MAC address
    end if
    if(Pkt.dest == broadcast Pkt && Pkt.Src in CAT[])
        install rule for broadcast
    endif
end if
```

D. *Attacker's location using graph traversal*

In order to mitigate the ARP attack in hybrid SDN when attacker pretend to be a legitimate user by using both IP and MAC address. A graph based traversal mechanism is used to detect the actual location of legitimate user and attacker's location. On the base of this location information, we can block the malicious user's port. In Hybrid SDN, the controller has the overall network view and topology information of all nodes. This information also indicates the connections between users and respective switches. We generate the graph for the whole topology after a fixed time interval. This graph has all the connectivity information of all devices in the network. Whenever a malicious device sends ARP request to the server and tries to spoof the network then graph traversal is used to detect actual location of the attacker. A modified depth-first search (DFS) mechanism is adopted to track the attacker's location. At first stage, we have the original topology of the network and after attacker's ARP requests, topology gets modified. By using graph traversal mechanism, attacker' location is identified and respective port is blocked for further communication.

Algorithm 1: Graph Traversal
Input: graph G, attacker IP(A)
Output: Location of Attacker IP address

```
enqueue (G, m)
While (queue is not empty)
    do dequeue (h, i)
    if (h is unchecked) then
        mark i
        add compare IP(i) with IP(A)
        if (IP(A) == IP(i) then
            generate alarm, return location
        else
            parent(i) ← h
        end if
    end if
    for each link (i, j)
        do enqueue (i, j)
    end for
end while
```

In Figure 7, a flow diagram for the whole system is shown. It represents the step by step procedure of proposed solution.

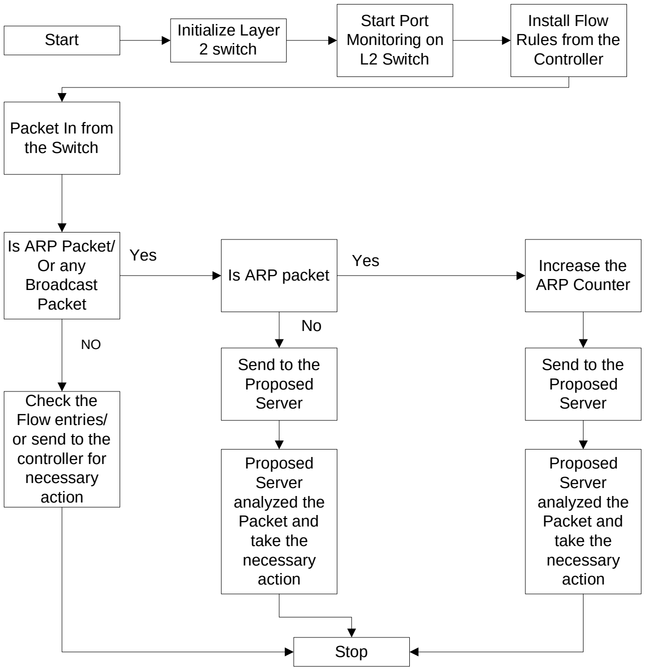
Start
Initialize Layer 2 switch
Start Port Monitoring on L2 Switch
Install Flow Rules from the Controller
Packet In from the Switch
Is ARP Packet/ Or any Broadcast Packet
Yes
Is ARP packet
Yes
Increase the ARP Counter
No
NO
Send to the Proposed Server
Send to the Proposed Server
Check the Flow entries/ or send to the controller for necessary action
Proposed Server analyzed the Packet and take the necessary action
Proposed Server analyzed the Packet and take the necessary action
Stop

Fig. 7. Flow Diagram

V. Implementation and Performance Evaluation

We have used the following simulation set up and implemented our proposed solution. We conducted our experiments on Ubuntu virtual Machine with 4 core and 4 GB RAM running on hypervisor server consisting of 16 GB RAM with 32 cores. Mininet [23] Simulation tool is used to create a virtual environment in which different tests are conducted. In Mininet we add multiple SDN switches, legacy switch, hosts and controller according to our scenarios discussed in problem statement and in proposed solution. Links are created between the switches, host, and controller. To enable a switch as a legacy switch we disconnected it with controller and OpenvSwitch (OVS) fail mode to be "standalone". POX [24] controller is used to install flows on the switches and to control the entire system. We have compared our results with the technique explained in [10]. Although, this technique is used in pure SDN and we are considering hybrid SDN where both type of network devices are present i.e. legacy and Openflow. Yet there is no mechanism available to deal with ARP spoofing attack in hybrid SDN.

Our proposed solution topology is shown in Figure 6 in which one controller, two SDN switch and three legacy switches are used with 9 host machines. The one machine with IP address 10.0.0.11 is used for the attacking purpose, where "kali Linux" OS is installed. This machine is used to generate ARP and other spoofed packets to poison the network topology. Our proposed server is connected to the controller for the data exchange and rules installation on the SDN Switches.

A. *Implementation of Proposed Scenario*

To evaluate our proposed solution, we measured several parameters like attack detection time, attack mitigation time and load on the CPU and throughput of our proposed algorithm using different attack scenarios. We used the several attacks like spoofed ARP request, ARP request attack, ARP reply attack and DDoS attacks. Each of these attacks is discussed below.

1) *Discussion on Spoofed ARP*

In Spoofed ARP request attack victim's cache table is poisoned with the fake entry of the host. This type of attack is usually used to intercept the traffic of the victims. This attack can be achieved by injecting thousands of spoofed ARP request packet into the network and victim PC cache is updated with wrong entries. This type of attack can be mitigated by our proposed solution in hybrid SDN. There are two types of spoofed ARP request attack. First, ARP request attack is same as the Spoofed ARP request attack. In ARP request attack an adversary party launches an ARP request packet by using the IP address of the other legitimate user and other users update their cache with this request. Consequently, communication between the legitimate users is not possible and the adversary party can get the traffic of legitimate user. To avoid such situations in hybrid SDN, SDN controller can handle the ARP request packet by installing the flow entries on switch for our proposed server. Second, ARP Reply attack is launched by an adversary party launches a Gratuitous ARP packet or an ARP reply by itself with the fake IP address or the MAC address in the network. Other users update their cache with wrong entries and the communication between the legitimate users is halted and the adversary party can get the traffic of the entire network host.

2) *Discussion on DDos attack*

DDoS uses the technology of ARP spoofed method to launch a DDoS attack on the network. This type of attack is usually launched to degrade or cut down the performance of the network and the legitimate users fail to access network resources.

To evaluate our proposed solution, we used a different network parameter like CPU load, attack detection time, attack mitigation time and throughput.

- Attack Detection time is the total time in which adversary party launch attack on the network and the controller detect the attack on the network.

- Attack mitigation time is a time to mitigate an attack after the detection of the attack in the network.

Figure 8 shows the comparison of attack detection time and attack mitigation time for the proposed algorithm and the existing technique. From Figure 8, one can conclude that our proposed algorithm performs better against malicious attacks than the existing approach. We can also secure our traditional network using hybrid SDN technology with limited investment in term of SDN switches deployment.

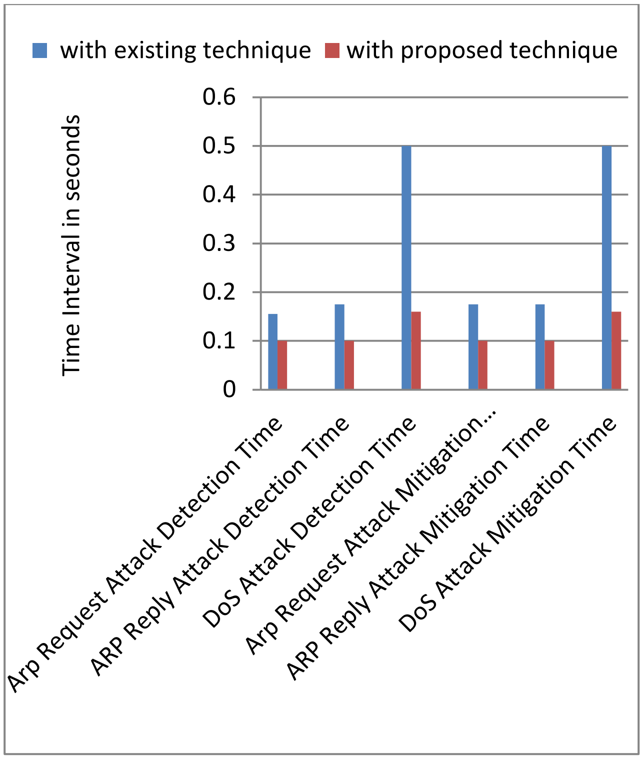

Fig. 8. Time Experiment

CPU load is a parameter to check the load of CPU when attacker launches an attack and controller run the algorithm to mitigate the attack. In our case, CPU utilization is a little bit higher than with the existing approach but this is the normal utilization of CPU. It didn't affect the performance of the network because this utilization is at proposed server not at the controller. The graphs of the CPU utilization are shown in Figure 9.

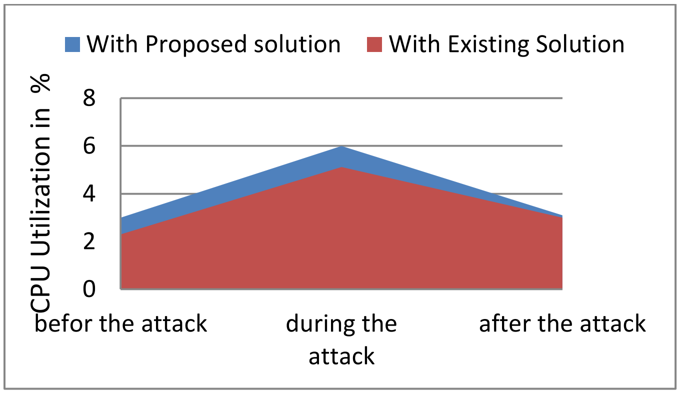

Fig. 9. CPU Utilization

Another factor to measure network performance is throughput. Throughput is the maximum utilization of the resources of a network system. In our case, we take the throughput of the link between the host and the controller before the attack and after the attack. We compare the throughput of our proposed algorithm with the existing approach as shown in Figure 10.

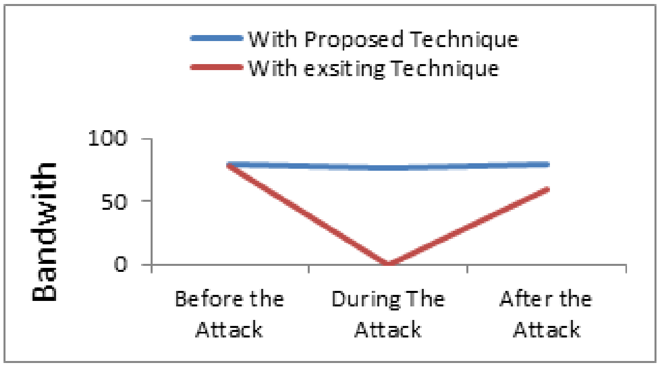

Fig. 10. Throughput between the links

We measure the successful packet delivery ratio. Figure 11 shows the results of successful packet delivery with respect to the time interval. The results indicate that successful delivery ratio is much better for proposed solution as compared to existing mechanism. When an attack is launched then our system automatically detects the attack and minimizes its effect on the system.

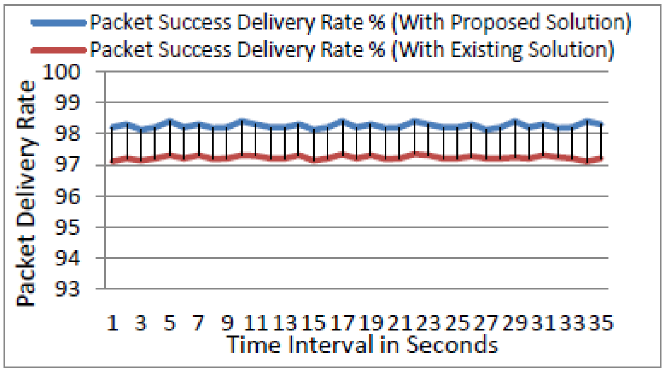

Fig. 11. Packet Success Delivery Rate

VI. Conclusion

In this paper, we examine layer 2 attacks in hybrid SDN and proposed a novel attack detection and mitigation technique. ARP spoofing and DDoS attacks are the most common attacks that affect the network performance very badly. In communication networks, most of the attacks are launched by spoofing the packet and poisoning the network topology by using ARP spoofing method. Our proposed solution consists of an individual server and customized mechanisms to get the network topology information. After this step, flow rules are installed on the switches for ARP packet to be forwarded to the server. We detect the attacker by analyzing ARP request from the source. We also used graph based traversal mechanism to detect the attacker location by verifying legitimate users. Experimental results showed that these threats have been resolved by using our mechanism. Furthermore, our solution supports multiple controllers in the network and can be used in pure SDN network also.

References

[1] Sezer, Sakir, et al. "Are we ready for SDN? Implementation challenges for software-defined networks." IEEE Communications Magazine 51.7 (2013): 36-43.

[2] Levin, Dan, et al. "Logically centralized?: state distribution trade-offs in software defined networks." Proceedings of the first workshop on Hot topics in software defined networks. ACM, 2012.

[3] Vissicchio, Stefano, Laurent Vanbever, and Olivier Bonaventure. "Opportunities and research challenges of hybrid software defined networks." ACM SIGCOMM Computer Communication Review 44.2 (2014): 70-75.

[4] Levin, Dan, et al. "Panopticon: Reaping the benefits of partial sdn deployment in enterprise networks." TU Berlin/T-Labs, Tech. Rep (2013): 1436-9915.

[5] Wang, Lei, et al. "Towards mitigating Link Flooding Attack via incremental SDN deployment." Computers and Communication (ISCC), 2016 IEEE Symposium on. IEEE, 2016.

[6] Shin, Seungwon, and Guofei Gu. "Attacking software-defined networks: A first feasibility study." Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013.

[7] Whalen, Sean. "An introduction to arp spoofing." Node99 [Online Document], April (2001).

[8] Cho, Hyunjeong, Saehoon Kang, and Younghee Lee. "Centralized ARP proxy server over SDN controller to cut down ARP broadcast in large-scale data center networks." 2015 International Conference on Information Networking (ICOIN). IEEE, 2015.

[9] Masoud, Mohammad Z., Yousf Jaradat, and Ismael Jannoud. "On preventing ARP poisoning attack utilizing Software Defined Network (SDN) paradigm." Applied Electrical Engineering and Computing Technologies (AEECT), 2015 IEEE Jordan Conference on. IEEE, 2015.

[10] AbdelSalam, Ahmed M., Ashraf B. El-Sisi, and Vamshi Reddy. "Mitigating ARP Spoofing Attacks in Software-Defined Networks."

[11] Abad, Cristina L., and Rafael I. Bonilla. "An analysis on the schemes for detecting and preventing ARP cache poisoning attacks." Distributed Computing Systems Workshops, 2007. ICDCSW'07. 27th International Conference on. IEEE, 2007.

[12] Xing, Wenjian, Yunlan Zhao, and Tonglei Li. "Research on the defense against ARP Spoofing Attacks based on Winpcap." Education Technology and Computer Science (ETCS), 2010 Second International Workshop on. Vol. 1. IEEE, 2010

[13] Hwang, Ren-Hung, Huei-Ping Tseng, and Yu-Chi Tang. "Design of SDN-Enabled Cloud Data Center." 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity). IEEE, 2015.

[14] di Lallo, Roberto, et al. "How to handle ARP in a software-defined network." NetSoft Conference and Workshops (NetSoft), 2016 IEEE. IEEE, 2016.

[15] Schneider, Fabian, Roberto Bifulco, and Anton Matsiuk. "Better ARP handling with InSPired SDN switches." Local and Metropolitan Area Networks (LANMAN), 2016 IEEE International Symposium on. IEEE, 2016.

[16] Kandoi, Rajat, and Markku Antikainen. "Denial-of-service attacks in OpenFlow SDN networks." 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM). IEEE.

[17] Brooks, Michael, and Baijian Yang. "A Man-in-the-Middle attack against OpenDayLight SDN controller." Proceedings of the 4th Annual ACM Conference on Research in Information Technology. ACM, 2015.

[18] Ma, Huan, et al. "SDN-Based ARP Attack Detection for Cloud Centers." Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom), 2015 IEEE 12th Intl Conf on. IEEE, 2015.

[19] Scott-Hayward, Sandra, Gemma O'Callaghan, and Sakir Sezer. "Sdn security: A survey." Future Networks and Services (SDN4FNS), 2013 IEEE SDN For. IEEE, 2013.

[20] Dhawan, Mohan, et al. "SPHINX: Detecting Security Attacks in Software-Defined Networks." NDSS. 2015.

[21] Taesang Choi, Saehoon Kang, Sangsik Yoon, Sunhee Yang, Sejun Song, and Hyungbae Park. 2014. SuVMF: software-defined unified virtual monitoring function for SDN-based large-scale networks. In Proceedings of The Ninth International Conference on Future Internet Technologies (CFI '14). ACM, New York, NY, USA, , Article 4 , 6

pages. DOI=10.1145/2619287.2619299 http://doi.acm.org/10.1145/2619287.2619299

[22] Sezer, Sakir, et al. "Are we ready for SDN? Implementation challenges for software-defined networks." IEEE Communications Magazine 51.7 (2013): 36-43.

[23] http://mininet.org/

[24] https://github.com/noxrepo/pox