Security of Software Defined Networks (SDN)

Levent Ertaul, Krishnakumar Venkatachalam
Department of Computer Science, CSU East Bay, Hayward, CA, USA
levent.ertaul@csueastbay.edu,  kvenkatachalam@horizon.csueastbay.edu

Abstract - Software defined networking (SDN) is a developing technology which completely changes the architecture and working of traditional networks. There has been growing urgency among networking developers to come up with a new set of devices to support Software Defined Networks and to implement better services in SDN compared to traditional networks. During these activities, security of the packets in the SDN networks comes into debate. In this paper, we describe the security of the SDN networks by implementing the SDN topology using the Mininet tool and explain the security vulnerabilities in the SDN networks. We describe the implementation of the DDOS attack using IP spoofing and Man in the Middle attack (MITM) using Arpspoof on the SDN topology and explain the behavior of SDN networks against these security attacks. We proposed the security method to provide security for SDN networks against the security attacks like DDOS, Man in the Middle (MITM), ARP spoofing, Replay, IP spoofing and the modification of information attacks.

Keywords: Software Defined Networks (SDN), IPsec VPN, Software Defined Security (SDSecurity).

1. Introduction

Software Defined Networking (SDN) has been the burning issue in the networking world since its development about 6 years ago. SDN is an architecture intending to be dynamic, manageable, cost-effective and adaptable, suitable for the high-bandwidth, dynamic nature of today's applications. [1] SDN deals with the abstraction of the control plane from the data plane and it controls the flow entries programatically using a single high level program.

OpenFlow is the protocol which separates the control plane from the data plane and it is based on the Ethernet Switch, with an internal flow table and a standardized interface to add and remove flow entries. [2] OpenFlow switch is used to perform the packet forwarding based on entries on the flow table. Flow rules are always installed by the controller, Controller can install the proactive and reactive rules in the OpenFlow network based on the type of notification from the switch.

Software Defined Security (SDSecurity) [3] is a next generation security category which is being developed for SDN environments. It is being built for providing protection for the controller, to make the controller secure, and to establish the trust for the controller on the operating entities and applications. It is also needed to create the robust framework for the policy. The security architecture which is being developed on the above criterions should be easy to implement on the dynamic SDN networks and it should also be secure and affordable.

To find the security vulnerabilities in the SDN networks, the security attacks like Distributed Denial of Service (DDOS) [4] using IP spoofing, Man in the Middle (MITM) [5] using Arpspoof are performed on the SDN network that runs on the Mininet [6] tool and also the behavior of the SDN networks using Wireshark [7] are recorded for better understanding.

In this paper, we designed an experimental SDN network using the Mininet tool and implemented DDOS and MITM attacks on this network. During the implementation of the attacks we studied the behavior of the SDN networks against these attacks using Wireshark. After examining the SDN networks’ capabilities against these attacks, we came up with a security method to protect the packets flowing; between the switch and the controller. We designed the security method on SDN networks to prevent the packets flowing in the clear form in the SDN networks. We added the authentication and encapsulation headers to provide security for the OpenFlow packets. That way we prevented the DDOS, MITM, IP spoofing, Replay and modification of information attacks.

Section 2 of this paper describes the SDN networks operations and security vulnerabilities in the SDN networks. Section 3 describes the implementation of DDOS attacks and MITM attacks on the SDN networks. Section 4 describes the IPsec VPN security method which is used to give protection for the SDN networks against the security attacks. Section 5 concludes the research paper.

2. SDN networks Operation

In this section, we explain briefly about the structure of the SDN networks and describe the operation of SDN networks The security vulnerabilities in the SDN networks are also described.
2.1 Operation of SDN Networks

Software Defined Networking (SDN) consists of three layers: Application Layer, Control Layer and Data or Infrastructure Layer. [10] Figure 1 depicts the architecture of the SDN networks.

Data layer or Infrastructure Layer: It is the forwarding plane which embraces the network elements used for forwarding the data packets on the network using the OpenFlow protocol. It is connected to the controller using the southbound API such as OpenFlow.

Control Plane: It is the heart of the whole network. It embraces the controller which is used to perform the network management operations and control operations for the SDN networks.

Application Layer: It is connected to the controller using the northbound API. This layer contains all the applications that runs on the SDN networks.

<table>
<thead>
<tr>
<th>APPLICATION LAYER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Applications: OpenStack etc.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CONTROL LAYER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Controller: OpenDaylight, POX, North Star</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DATA LAYER</th>
</tr>
</thead>
<tbody>
<tr>
<td>OPENFLOW</td>
</tr>
</tbody>
</table>

| NETWORK ELEMENTS |

Fig 1. SDN Architecture.

The OpenFlow protocol is used to communicate to the controller in the control layer with the network elements in the infrastructure layer. Using this OpenFlow, the network device will hold its flow table which contains all the flow entries. Figure 2 depicts the flow entries fields in the flow table for the network device.

<table>
<thead>
<tr>
<th>Match Fields</th>
<th>Priority</th>
<th>Counters</th>
<th>Instructions</th>
<th>Timeouts</th>
<th>Cookie</th>
</tr>
</thead>
</table>

Fig 2. Flow Table Entries Fields.

2.2 Security Vulnerabilities in the SDN Network

This section gives the detailed explanation about the security issues in the SDN networks. The proposed attacks are DDOS, IP spoofing, MITM, Replay and modification of information.

2.2.1 DDOS attacks

DDOS attack is a major attack on the SDN networks. It mainly occurs due to the flooding of traffic packets to the desired host on the SDN networks and makes it unavailable for further communication. The attacker will send the flood...
of TCP SYN messages [13] or UDP SYN messages [14] or ICMP messages [15] to the victim host using the IP address which can be found using IP spoofing. It causes a table-miss in the flow table in the switch. Table-miss will occur if the received packet values do not match with the flow entries in the flow table. These flow table miss packets are collected and notified to the controller using the Packet in message. Then the controller will process the packet information and modify the flow table. It adds the value to the flow table and acknowledges the packet. If there are enormous amounts of Packet-in messages consistently flowing from the same node, then the controller will experience huge amounts of traffic. By that, it pushes the controller to become unavailable for other nodes to make routing decisions. This reduces the performance of the SDN networks.

### 2.2.2 MITM attacks

MITM occurs in the forwarding link in the SDN networks. It occurs easily in the SDN networks because of the lack of authentication header in the OpenFlow packets flowing through the switch. If the attacker is listening to the SDN networks in between the controller and switch connection, the attacker can make duplicate traffic to its host and dumps the flows into the SDN networks with the rules to hijack the switches and capture the traffic flowing through it. And these will initiate other attackers to attack the SDN networks. This will affect the performance and credibility of the SDN networks.

### 2.2.3 IP spoofing

IP spoofing is mainly used in the DDOS attacks. It is the process of using forged IP addresses to inject packets into the SDN networks. It also occurs due to improper network configuration.

### 2.2.4 Replay attacks

Replay attacks occurs in the forwarding-control link because of the lack of time stamping in the flowing packets and because the packets are sent in the clear form in the communication channel. The packets are not authenticated by any mechanisms. These allow the attackers to sniff the packets flowing through the OpenFlow channel and resend the packets or delay the packets fraudulently.

### 2.2.5 Modification of Information

Tampering of information [16] occurs during the transfer of packets through the SDN networks. Since, the packets are not encrypted or authenticated, data packets are sent in the SDN networks without any protection. It leads to the tampering of data packets by the attackers.

### 3. Security Analysis of SDN networks

In this section, we explain about the creation of SDN networks topology using the Mininet tool. We describe briefly about the implementation of the DDOS attack using the hping3 [17] tool and MITM attack using the arpspoof [18] tool. And we explain the behavior of the SDN networks against these attacks. These attacks are implemented on a Lenovo laptop with features of windows 64-bit OS, Intel i7 processor, 2.59 GHz and the Mininet tool is installed on the Ubuntu OS running on Virtual box.

### 3.1 SDN topology

Mininet is the software emulator which runs on the VM machine on Ubuntu which is used to create large SDN network prototypes. Wireshark tool is the packet analyzer which is used to analyze and troubleshoot the SDN networks.

We have created a custom topology using the Mininet software with a switch, a controller and three hosts. Mininet creates a realistic virtual network, running a real kernel, switch and application code on a single machine. [19] It is the SDN software which is used to establish the OpenFlow on the network topology. A Python script with the imported Mininet library is used to create the SDN topology. Self.addHost is used to add host and self.addSwitch is used to add switches to the SDN topology. Self.addLink is used to form the link connection between the network devices and controller. Figure 5 illustrates the SDN topology which is created using the python script which is shown in the figure 4.

```python
class MyTopo(Topo):
    def __init__(self):
        Topo.__init__(self)
        S1 = self.addSwitch('S1')
        S2 = self.addSwitch('S2')
        H1 = self.addHost('H1', ip="10.0.2.1", defaultRoute = "10.0.2.1")
        H2 = self.addHost('H2', ip="10.0.3.1", defaultRoute = "10.0.3.1")
        self.addLink(H1,S1)
        self.addLink(H2,S2)
        self.addLink(S1,S2)

topo = { "mytopo" : (lambda : MyTopo()) }
```

Fig.4 Python code for SDN topology
The communication is established between the host and controller using the OpenFlow messages [20]. It starts with the *Hello* message to setup the connection between the switch and the controller. Then the controller will send the *Echo* request message to the switch and the switch will respond using the *Echo reply* message to denote the presence in the network. The switch then shares the features with the controller using the *Feature Request* and *Feature reply* messages. Once the switch shares the features, the controller will set the configuration for the switch and notifies the switch using the *set config*. When the new packet comes inside the network, the switch will send the packet information using the packet-in message and the controller will add the flow entry into the existing flow table and replay back using the *flow mod* message. Then the flow entry for the packet is added into the switch containing the flow table. This process is continued for all the new packets.

### 3.2 DDOS Attack

The *hping3* tool is used to perform DDOS attack by sending flood of TCP packets or UDP packets or ICMP packets. For running DDOS attack on SDN networks, we took “10.0.2.2” host as the zombie host and we performed DDOS attack on the “10.0.3.2” network. To begin the experimentation, we sent out flood of TCP packets with different IP addresses from the “10.0.2.2” to the “10.0.3.2” host. In the process there will be large amount of traffic flowing towards the switch with the destination name as H2. When the switch checks in the flow table for matching of flow entries, there could be a table miss. Then the switch will send the packet-in message to the controller with the packet information. Since, there are enormous amounts of packets in messages from the switch, the controller will experience huge amounts of traffic from the same node. This eventually brings down the controller due to large amounts of traffic and it makes the hosts unreachable in the SDN networks. This attack can easily be done in the SDN networks because of the lack of authentication header and the messages are sent in a clear form. The figure 6 shows the screenshot of the DDOS attack in the SDN network which was captured using *Wireshark*.

### 3.3 MITM Attack

The MITM attack is an attack which is used to spoof the information from the victim host during its communication with other hosts. It occurs in the forwarding-control link in the SDN network. This attack can be done using the *Arpspoof* tool. We used H2 as the victim node and H1 as the attacker node. We spoofed the host H2 IP address using the IP spoofing method. The IP address of the H2 is found by sniffing the traffic flow in the SDN network using *Wireshark* capture since there is no authentication header for the packet flowing through the OpenFlow communication channel. Then we did the ARP spoofing using the *arpspoof* tool on the host H1. The commands are,

```
sarpspoof -t 10.0.3.2 10.0.3.1
sarpspoof -t 10.0.3.1 10.0.3.2
```

By doing this, Host H1 can see the activities of the host H2 as it is illustrated in the figure 7. Whenever, H2 contacts any Host in the network or transfers any information it can be seen by the Host H1.
4. Security Method Against the Attacks in SDN Networks

We have discussed the security issues in SDN networks by implementing DDOS and Man in the Middle attacks on the SDN networks topology. It is clear from the above experimentation that the packets were sent in a clear form in the SDN networks and anyone can spoof the data packets flowing through the OpenFlow communication channel. To overcome these security issues, we have come up with the security method which uses IPsec VPN [21] method with the integration of AES [23] encryption and TLS for the controller-switch communication on the SDN networks.

4.1 IPsec Tunnel mode VPN using ESP header

In the tunnel mode of communication, Encapsulation (ESP) header [22] is used to provide both confidentiality and authentication to the data packets. IPsec tunnel mode VPN is used between the gateways. The communication channel between the OpenFlow switches and controller are encrypted using the symmetric key encryption which uses AES algorithm. This can be done by integrating the AES algorithm into the function of the controller by creating the separate security control layer on the top of Control layer. Since it is a symmetric key encryption, it uses the same key for encryption and decryption, which is send by the controller to all its connected switches. The controller sends the secret key payload with the TLS protection to improve the security of the key exchange between switches. The secret key is shared by the controller with the OpenFlow switch immediately after the Echo Request (Type -3) and Echo Reply (Type-4) messages are exchanged. By doing that, all the communications that occur between the controller and the switch are encrypted. The firewall [24] is built on the gateway interface(s) on S1 and S2 through which H1 and H2 are connected. Using the topology showed in figure. 8, when host H1 wants to setup the connection with H2, it will first send the message with the pre-shared key label along with firewall gateway address of H2. Since, there is no existing flow entry for this flow in the flow table present in the switch, the switch will send the Packet-in message to the controller to update the flow. The controller will reply with the flow_mod message and the flow entry is updated in the flow tables of the two gateway switches.

Controller uses packet-out message to send the message packet to H2. Thus, the pre-shared key is stored at the Firewall gateway of the host H2. It forms the encrypted channel between S1 and S2. The firewall gateway of H2 is provided with the access-list of Host H1 network IP address so that, the firewall in S2 allows all the traffic from H1. Then the data packet from the H1 to be transferred to H2 is encrypted along with OpenFlow header, IP header and TCP header using the ESP header and trailer. Each encrypted packet is stamped with a sequence number to provide protection against the replay attacks.

Then the entire encrypted packet is authenticated using the authentication trailer and it will calculate the Integrity Check Value (ICV). It is then placed in the authentication trailer which provides both authentication and integrity for the data packets. The encapsulated packet is attacked with the new IP header which is nothing but the IP addresses of the H1 and H2 gateway. The packet structure is shown in figure. 9. Then the packet is received at the firewall of the S2 and the authentication and integrity of the data is checked using ICV and it uses the pre-shared key to decrypt the packet. Then the packet is delivered to the host H2.
whose IP address is present in the original IP header. This method provides protection against IP spoofing, replay, MITM, ARP spoofing and DDoS attacks by providing data confidentiality, integrity and authentication functionalities. This method has the constraint of IP addresses. Since, the IP addresses are unique, there is a slight possibility for the shortage of IP address in the network, because this method needs a minimum of four IP address for communication between two devices. To reduce the usage of number of IP addresses for the VPN connection between two hosts, IPsec Transport mode VPN can be used.

4.2 IPsec Transport mode VPN using ESP header

It is like Transport mode [25] but here it uses the original IP address in sending the packets rather than attaching the new IP address in Tunnel mode. It is designed to provide an encrypted channel between the hosts. Here we are not using the firewall between the gateways. Using the topology shown in figure.10, in which the host H1 tries to setup a connection with H2, it will first send the message with the pre-shared key label along with IP address of H2.

![Fig 10. IPsec Transport mode VPN topology](image)

It follows the same packet forwarding method as explained in the Tunnel mode. Then the pre-shared key is stored in the host H2. It forms the encrypted channel between H1 and H2. The packet to be sent is first encrypted along with IP header TCP header and OpenFlow header using the ESP header and trailer. Next the encrypted packet is authenticated using the ESP authentication trailer which provides data authentication and data integrity. Then the original IP header is attached at the front of encapsulated packet which is the mirror image of IP header packet which is authenticated and encrypted. The packet structure is shown in figure. 11.

![Fig 11. IPsec Transport mode using ESP header.](image)

Then the packet is transferred through the encrypted channel between the H1 and H2. When the packet reaches H2, the ICV is checked to find the data integrity and authentication of the transit data packet and it is decrypted using the pre-shared key. The data packet is then delivered to the host H2. This method provides protection against the MITM, replay, Arpspoof, IP spoofing and DDoS attacks and for the data modification. This mode might become the victim of IP spoofing attacks because the original IP header is sent in the network without any protection. But we can improve the protection for the data and OpenFlow header by integrating TLS [26] protection. This will provide extra protection for the data packets during the IP spoofing attacks.

5. Conclusion

In this paper, we have discussed the security issues in SDN networks. We implemented the DDoS and MITM attacks in the SDN networks topology and explained the behavior of the SDN networks against these attacks. We explained about the security method which uses AES symmetric key encryption and TLS for the switch-controller communication and IPsec VPN method for the communication between the hosts. It provides protection against the IP spoofing, MITM, DDoS, ARP spoofing and Replay attacks and it also provides protection for the data modification. From the above discussion, it was proved that it is possible to provide desired security to the SDN networks using the existing IPsec VPN method with the integration of AES encryption and TLS. In future, we plan to implement this security method in the real-time environment and use OpenStack for managing the flow tables in the cloud based environment.
6. References


