Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 171 (2020) 2581–2589

Procedia Computer Science

www.elsevier.com/locate/procedia

Third International Conference on Computing and Network Communications (CoCoNet'19)

# Solutions to Vulnerabilities and Threats in Software Defined Networking (SDN)

Aayush Pradhan[a], Rejo Mathew[b]*

[a]IT Department, Mukesh Patel School of Technology Management and Engineering, Mumbai 400056, India
[b]IT Department, Mukesh Patel School of Technology Management and Engineering, Mumbai 400056, India

**Abstract**

In today's advancing world, there is an increase in the size and requirements of networks which can be a burden since moving around with switches is quite chaotic. SDN helps switches to be programmed and implemented independently. SDN is a way of providing programmability for network application development by separating the control plane from the data plane. Security of Software Defined Networking (SDN) is an open subject. Separating the control plane from the data plane opens up a number of security challenges such as a man-in-the-middle attack (MITM), a service denial (DoS), overload saturation attacks, etc. In this paper, we have focused on the overview of software defined network (SDN), it's challenges, it's issues, and their solutions. First, we have discussed about the architecture of SDN, followed by elaborating the threats and at the last, proposing some solutions to improvise the security.



*Keywords:* Software Defined Networking (SDN), DDoS, packet sniffing, ARP spoofing, brute force, API exploitation

* Aayush Pradhan. Tel.: +91-9820730884
*E-mail address:* aayushpradhan29@gmail.com


10.1016/j.procs.2020.04.280

## 1. Introduction

SDN is a networking technology that provides a programmable control planes and data plan abstraction, and explicitly manages its own virtualized networks without perceiving comprehensive hardware technologies. SDN supports the isolation of control plane from data plane as well as separation of the logically hierarchical controller from the interaction of data plane and control applications running at the top of the network [1].

SDN consists of 3 layers [1]:

- Infrastructure Layer: It acts as the foundation for an SDN architecture. This layer consists of physical and virtual network devices such as switches and routers and various other networking apparatus which becomes an underlying network to forward network traffic.
- Control Layer: It consists of a centralised control plane where business logic is being written in controller to fetch and maintain different types of network information such as state details, topology details, statistics details, etc.
- Application Layer: It consists of various network devices, network tools and business applications that interact with the control layer via controller.

SDN architectures generally have three components [2]:

- SDN Applications: These are programs that communicate the required resources with the controller with the help of some application programming interfaces (APIs).
- SDN Controller: Controller is a logical entity that receives commands or requirements from the application layer and then depending upon the type of action needed, it transmits them to the networking components in that particular network.
- SDN Networking Devices: These devices control the forwarding and data processing capabilities of the network as shown in figure 1.

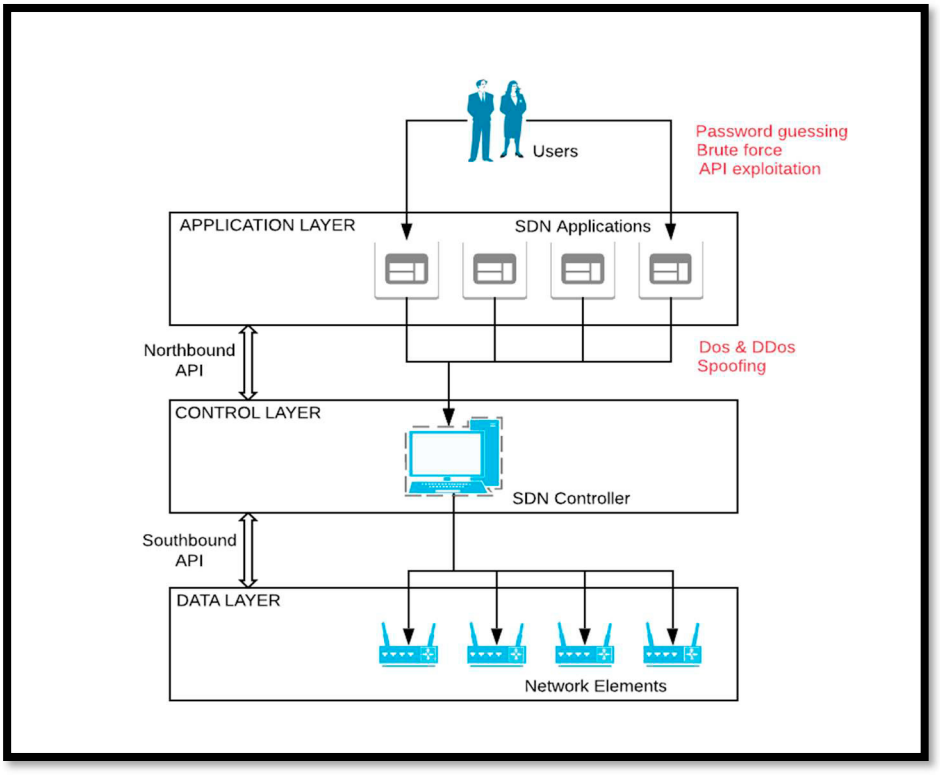


Figure 1. SDN architecture along with possible attack vectors

## 2. Vulnerabilities in SDN

The vulnerabilities that may reside on the controller can be as follows–

- Weak authentication:
  Low authentication defines any situation where the authentication process capacity is relatively weak compared to the value of the secured property.
- Incomplete encryption:
  Encryption uses complex mathematical equations to hide information from outsiders, converting documents, messages and files into incomprehensible gibberish. Typically, encrypted documents need a key to decrypt, but in some situations, hackers may circumvent authentication mechanisms and access privileged information. It may lead to unauthorized access to intruders.
- Information disclosure
  Disclosure of information is when an application does not adequately protect sensitive and confidential information from entities that in normal circumstances are not expected to have access to the subject matter. It that can further lead to taking control of the network traffic.

The controller is the core component in SDN, hence if one does not manage the controller present in the control plane right, then it may be a potential single point of attack and failure. The controller defines the data flow that takes place in the data plane. Hence, when the controller is attacked and compromised, it will definitely affect the network badly. Many works have been done to compare the SDN controllers with least number of vulnerabilities, amongst which Floodlight and OpenDaylight were two controllers with almost similar results [16].

## 3. Threats on SDN

The list of possible threats in SDN can be broadly categorized based on targeted area:

- SDN controller.
- Networking devices.
- The data communication between controller and various networking devices.
- The communications between several controllers in different domains or different networks.

SDN security threat vectors listed down by [4] are:

- Faked traffic flows.
- Attack on the weaknesses present in switches.
- Attacks on the communications involved in control plane (between controller and other networking devices, as well as between the controller and APIs).
- Deficiency of strong security mechanisms to ensure the trust between controller and other applications.

All the types of potential attacks that are possible while implementing SDN can be classified down into threat on each of the layers as depicted in table 1.

### *3.1 Control Plane –*

The control plane defines the network topology [3]. It is a major concept in the technology of network routing. It is responsible for linking routers and for exchanging information about the protocol. A range of routing protocols are used to define and manage connections. The following are the types of attacks possible on this layer.

#### *3.1.1 Network Manipulation*

In this type of attack, an intruder compromises the controller by constructing a false data on the network and then simultaneously initiates other attacks on the network.

### *3.2 Data Plane –*

It is also known as forwarding plane [3]. It transfers traffic in line with the control plane logic to the next hop along the route towards the selected target network. The router goes through the data plane packets. The routers / switches use the control plane configured for the entry and exit of frames and packets. The following are the types of attacks possible on this layer.

#### *3.2.1 Traffic Diversion*

Such kind of attacks happen to the various elements in the network, especially at the data plane wherein the attacker compromises the network element to redirect the flow of traffic, followed by exploring further vulnerabilities by methods such as eavesdropping.

#### *3.2.2 Side Channel Attack*

Another attack that targets the network elements at the data plane is side channel attack. To successfully carry out this attack, the attacker takes the help of specific network information such as timing information (i.e. details of how long it took to create a new connection) to analyse if a flow rule is present or not.

### *3.3 Application Plane –*

The application plane on a network is the tier of applications and services involving the control plane and the data plane's network functions. In this layer reside conventional network devices [3]. The functionality of these devices is largely the same in software-defined networking, but their delivery is abstract, centralized and often virtualized. The following are the types of attacks possible on this layer.

#### *3.3.1 App Manipulation*

In such attacks, the attacker could gain access (sometimes with high privilege) to any SDN application and can be successful in accomplishing illegal actions over the network. Exploitation of application vulnerability can cause adverse effects such as malfunction of network, disruption of service, or eavesdropping of data.

Table 1. Attacks on different layers in SDN

| Security Attacks | Application Layer | Control Layer | Data Layer |
|---|---|---|---|
| Unauthorized access | | | |
| • Controller hijacking | | ✓ | |
| • Unauthorized application | ✓ | ✓ | ✓ |
| Data leakage | | | |
| • Credential management | | ✓ | |
| • Forwarding policy discovery | | ✓ | |
| Data modification | | | |
| • Data flow modification | | ✓ | ✓ |
| Denial of service | | | |
| • Controller flooding | | ✓ | ✓ |
| • Switch flooding | | | ✓ |
| Configuration issues | | | |
| • Lack of TLS adoption | ✓ | ✓ | ✓ |
| • Policy enforcement | ✓ | ✓ | ✓ |

### *3.4 Attacks that can affect all parts of SDN –*

#### *3.4.1* Distributed Denial of Services (DDoS)

Steps involved in DDoS are:

- Initially, the attacker creates a series of fake packets where their source address is spoofed such that they appear to be a valid packet.
- Packets pass through the switches after they are introduced into the network. The switches will assume that those packets are legitimate.
- Switches then sends these packets to the controller requesting for a new flow rule. This is where the problem arises. The controller will now identify the packets and will instantly create a flow entry for the spoofed packets. This process is time consuming.
- The attacker will then continuously keep sending fake packets that will occupy resources of the controller to process the fake packets. Legitimate packets will not have any chance to be directed by the controller, leading the work done by the controller be completely useless.
- This long process will exhaust the resources of the controller. These fake packets will make the controller unreachable for a certain period of time. The loss of controller will jeopardize the whole network services in SDN [5].

#### *3.4.2 Packet Sniffing*

Sniffing packets are a method of traffic gathering the network in order to evaluate the traffic for potential use. This method calls for a device, computer or hardware packet sniffer. In addition to how simple it is to get tools, the functionality of software which can capture all in and out traffic encourages attackers to use the technique to their benefit. Due to the inactiveness of the tools, they can simply collect information without modifying the network as shown in figure 2.

When we talk about non-switched network environment, the network traffic is publicized to each and every device connected in that particular network. Thus, any device can sniff the packets. On the other hand, talking about switched network environment, the packets will be directly sent to the destination address only. In such cases, all the other devices will not be able to see the traffic except the destination host device [6].

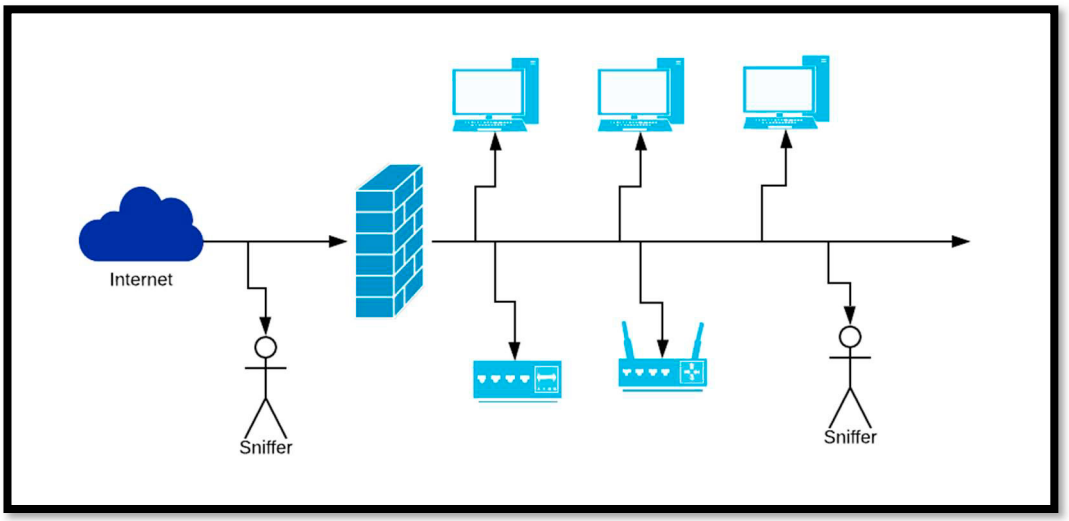


Figure 2. Packet sniffing attack

#### *3.4.3 ARP Spoofing Attack*

Common example of ARP Spoofing attack is Man-in-the-middle attack (MITM). In man-in-the-middle attack, the attacker secretly relays and maybe alters the communications between two parties.

Eavesdropping is a type of man-in-the-middle attack, wherein the intruder makes autonomous communications with the victims, sending messages to them to make them believe that they speak to each other directly on a private basis. This type of attack corrupts the network topology information and those applications which are based on topology.

#### *3.4.4 API Exploitation*

A programmer makes a request to the operating system with the aid of APIs. In any computer element, the Application Program Interface (APIs) may be vulnerable to the ability of a hacker to conduct an unauthorized data release.

#### *3.4.5 Password guessing or Brute Force*

This attack can happen on any SDN component or a non-SDN element. An unauthorized user can access the SDN with password guessing or brute force. A brute force attack is basically hit and try error method used by software to decode encrypted data such as passwords or encryption keys by exhaustive effort.

## 4. Solutions

### *4.1 Solution for DDos- To analyze user's behavior via detection through Entropy*

One way to identify the characteristic attack of DDoS attack is by using the method of entropy [5]. Entropy tests DDoS randomness. The higher the ratio of randomness, the higher is the quantity of entropy. The more randomness. The entropy is determined after a collection of packets is received. Relative to the threshold, the outcome of measurement of entropy is calculated. DDoS is observed if the entropy has been higher or lower than the threshold. With the early detection, it can be used more often by the network manager.

### *4.2 Solution for Packet Sniffing – Encryption using SSL Certificates*

SDN can also be secured using SSL encryption in the same way as the traditional networking. In order to secure communication between network devices and the controller, SSL Certificates are implemented on the data plane. Two stages are included in SSL: the handshake and data transfer processes. [7].

The primary requirement of the handshake phase is to check whether the switch and the controller are both ready for the data transfer or not.

Steps involved in this handshake are-

- In the first step, the switch requests for a public key from the controller.
- The controller then will send a certificate that contains the public key. This certificate validates the controller and the identifies the authenticity of the switches. To obtain the certificate, the controller must send its public key to the Certificate Authority (CA). CA generates the certificate and creates a fingerprint respectively.
- CA will then encrypt this fingerprint with its own private key to create a certificate signature.
- In order to get the fingerprint, the switch must decode the newly created signature with the CA public key. The switch shall also automatically determine the fingerprint of the certificate.
- The certificate will be authenticated by comparing both of the fingerprints. If they don't match, the certificate has been tampered with.
- Using a public key algorithm, the controller and the switch evaluate secret key information once the authentication is completed on both sides
- The handshake phase is finally finished when both the controller and the switch are ready to use the secret key.

Talking about the data transfer phase, the outgoing information will be divided into fragments and annexed to the Message Authentication Code (MAC). MAC is the fingerprint computed from data with its own key. This key is not the secret key generated during the handshake process.

### *4.3 Solution for ARP Spoofing Attack – Using various methods such as patching, tools, mapping etc.*

To solve ARP spoofing attacks, precautionary solutions that can be chosen are as follows [8]:

#### *4.3.1 ARP authentication*

ARP protocol will be modified in such a way that it will not process ARP packets unless and until they hold a predefined cryptographic code. Some of the common examples are S-ARP (Secure-ARP) [9] and TARP. This method has some disadvantages of its own. This approach causes some conflict with devices that still use the primitive generic ARP protocol.

#### *4.3.2 Operating system patching*

The aim of this method is to patch the operating system to avoid spoofing of ARP. But for various kernels of operating systems, this strategy requires a different patch. A patched operating system may not interact properly with other operations that are not patched. Anticap and Antidote mechanisms are some solutions for ARP spoofing problems that repair the OS kernel.

#### *4.3.3 Dynamic ARP Inspection (DAI)*

Some switches have a Dynamic ARP Inspection (DAI) security feature[ 10]. Switches use DAI to avoid ARP spoofing attacks by intercepting ARP requests and replies, and dropping invalid IP-to-MAC address packets. The main issue with this approach is that the DAI function is not included in all switches.

#### *4.3.4 ARP mitigation tools*

Several software tools to mitigate spoofing attacks have been developed. There are good detection mechanisms for free detection systems such as ARPWatch and XArp, but they cannot provide full protection. AntiNetcut [11], NoCut [11] and AntiARP [11] are additional software tools. However, these tools have many problems such as high rate of false positives and false negatives and high rate of ineffectiveness.

#### *4.3.5 Static ARP mappings*

ARP spoofing attacks can be better managed by manually configuring the ARP entries. In so doing, the attacker has no chance of spoofing ARP cache from the other host. Nonetheless, for large networks this approach is not feasible. But this solution is not viable for large scale networks. It can't be used in networks that use DHCP addressing. In addition, it will be error prone and requires a great overhead on the network administrator. Several solutions have been proposed in attempt to overcome these drawbacks such as DAPS [12] and NIDPS [13].

### *4.4 Solution for API Exploitation – Using Bots for robust security*

A bot management solution that protects APIs from automated attacks and guarantees that only true users can access APIs is important. While considering a solution like this, consider the extensive coverage and identification of threats, detailed reporting, review and versatile options to implement.

Precautionary steps can be taken to avoid misuse of the API:

- Monitoring and management in the handling of API calls from automated scripts (bots)
- Dropping rudimentary authentication

- Implementing measures to prevent the use of sophisticated human-like bots to block API access
- Using token-based rate limits to limit the number of IPs, sessions and tokens based on the total API usage.
- Robust encryption is a must have

### *4.5 Solution to Password guessing or Brute Force*

Such types of attacks can be avoided by algorithms like vendor switch, strong passwords, and frequently updating them, etc.

Other possible solutions to prevent brute force include [14]-

- Locking of accounts
- Time bound logins- Allowing selective users to access the network at a particular time.
- Query-based authentication
- One-time password authentication- It is a password that is valid only for one login session on a device.
- Using Captcha- A captcha is a form of mathematical challenge-response test used to decide if the user is human.
- Unique IP address login- Enable the use of some privileged IP addresses that have the power to overwrite any other command on the network.

## 5. Analysis

Table 2. Comparison table for different solutions

| | Entropy analysis | Encryption using SSL | Using Bots | Limiting API access | Using one-time password (OTP) |
|---|---|---|---|---|---|
| Encryption | No encryption of data during transmitted. | User can only access with exact key. | They provide robust encryption. | No encryption of data. | No encryption techniques. |
| Authentication | Entropy does not authenticate the topology applications. | Only SSL certificate provider can be trusted. | It provides drop primitive authentication. | User authentication | User can only login using the correct OTP. |
| Guards against phishing | No protection to phishing, only analyzing user behavior. | Intruders can have a hard time obtaining authentic SSL certificate. | May or may not, depending upon the bots used. | Yes, by limiting the number of users. | Not applicable |
| Access granted | Doesn't support this feature. | Not applicable | Depends upon the type of bot used. | Only limited users based on the number of IPs. | Allows only authorized users to access the network. |

## 6. Conclusion

We can conclude this paper by stating that the central controller is the most vulnerable part of the SDN architecture. From table 2, a brief review on threats and solutiond can be grasped. Although a centralized network control system and the programmability of the network could lead to strong and viable implementation of security, it also presents new challenges for security. Therefore, we have presented most of the security weaknesses in SDN and their proposed solutions in this paper.

With the gradual development of SDN technology, it is highly possible that new security threats will arise. We assume that SDN can be one of the most important technologies with time to drive a number of network security advancements. SDN innovations also make it easier to computerize the cloud [15].

The prominent characteristics supported by SDN are scalablability, highly reliablability, low latency distributed control plane etc, which helps SDN to miragte towards wide area networks and 5G core networks. In order to meet the high reliability and low latency requirements under varying amount traffic, the distributed control plane, needs to be managed dynamically [17].

## References


[1] I. Ahmad, S. Namal, M. Ylianttila and A. Gurtov, (2015) "Security in Software Defined Networks: A Survey," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2317-2346, Fourthquarter

[2] T. D. Nadeau, K. Gray, (2013) "SDN: software defined networks." *O'Reilly Media, Inc.*

[3] S. Hogg, (2014) "SDN Security Attack Vectors and SDN Hardening"

[4] D. Kreutz, V. Ramos, E. R. Christian, (2015) "Software Defined Networking: A Comprehensive Survey", *Proceedings of the IEEE*, Vol. 103, No.1

[5] S. M. Mousavi, M. St-Hilaire, (2015) "Early detection of DDoS attacks against SDN controllers.", *IEEE International Conference on In Computing, Networking and Communications (ICNC)*

[6] R. Spangler, (2003) "Packet sniffing on layer 2 switched local area networks.", *Packetwatch Research*

[7] W. Chou, (2002) "Inside SSL: the secure sockets layer protocol."

[8] M. Jammala1, T. Singha, A. Shamia, R. Asalb, and Y. Lic, (2014) "Software-Defined Networking. State of the Art and Research Challenges" *Computer Networks*, vol. 72, pp. 74–98

[9] D. Bruschi, A. Ornaghi, E. Rosti, (2003) "S-ARP: a secure address resolution protocol", *19th Annual Computer Security Applications Conference*

[10] *https://www.cisco.com/c/en/us/td/docs/switches/lan/catalyst4500/12-2/25ew/configuration/guide/conf/dynarp.html#wp1039201*

[11] A. M. Abdelsalam, A. El-Sisi, V. Reddy, (2015) "Mitigating ARP Spoofing Attacks in Software-Defined Networks", *ICCTA*

[12] X. Hou, Z. Jiang and X. Tian, (2010) "The detection and prevention for ARP Spoofing based on Snort,", *International Conference on Computer Application and System Modeling*

[13] S. G. Bhirud, V. Katkar, (2011) "Light weight approach for IP-ARP spoofing detection and prevention,", *Second Asian Himalayas International Conference on Internet (AH-ICI)*

[14] G. Sowmya, D. Jamuna, M. Venkata Krishna Reddy, (2012) "Blocking of Brute Force Attack", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 1 Issue 6

[15] S. Mishra, M. A. R. AlShehri, (2017) "Software Defined Networking: Research Issues, Challenges and Opportunities", *Indian Journal of Science and Technology*, Vol 10

[16] S. Rowshanrad, V. Abdi, M. Keshtgar (2016) "Performance evaluation of SDN controllers: Floodlight and OpenDaylight", *IIUM Engineering Journal*, Vol. 17, No. 2

[17] B. Görkemli, S. Tatlıcıoğlu, A. M. Tekalp, S. Civanlar and E. Lokman (2018) "Dynamic Control Plane for SDN at Scale", *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 12, pp. 2688-2701