MÄLARDALENS UNIVERSITET
AKADEMIN FÖR INNOVATION, DESIGN OCH TEKNIK
VÄSTERÅS, SVERIGE

Thesis for the Degree of Bachelor of Science in Engineering - Computer Network Engineering högskolepoäng 15.0 hp

MAN IN THE MIDDLE ATTACKS ON SOFTWARE DEFINED NETWORK

Eskil Christensson
Ecn18004@student.mdu.se

Examinator: Hossein Fotouhi
Mälardalens universitet, Västerås, Sverige

Handledare: Václav Struhár
Mälardalens universitet, Västerås, Sverige

2023-02-28

Abstract

Software Defined Networking (SDN) is a method of abstracting the network to ease administration by centralizing it. Doing this removes intelligence from the devices and this might open up local security vulnerabilities. One attack type is a Man in the middle attack, this requires the attacker to have access to the data flow that they want to attack. One type of this is to trick other network devices like hosts into believing the attacker is the one they are supposed to communicate with. Software defined networks might make this type of attack more dangerous since the network administration is centralized. In this thesis we summaries attacks against SDN and propose a method to detect Man in the middle attacks in the SDN network. I used Mininet virtualization to create an SDN to perform an experiment to answer this. The experiment performed is an ARP-poisoning attack on SDN to find out the effect on network latency that the attack causes, this will help to develop an detection method for ARP-poisoning attacks.

Table of content

1. Introduction

Software Defined Networking (SDN) is an approach to network management that centrally controls the network, and thus can configure and administer network devices, e.g., switches and routers [1]. This opens more ways to easily control the network and makes it more flexible by using a central control device for the network known as an SDN controller. This makes it easier to configure the network since the network administrators do not need to change the configuration of all devices in the network when the configuration needs to be changed since the SDN controller customizes network devices' policies in a dynamic manner [2]. This means most of the administrative configuration only needs to be changed on the SDN controller.

However, centralized control of the network might introduce several problems, for example, it introduces a single point of failure, [2] if the controller does not work, the network fails. Moreover, if the controller is compromised in an SDN, it is likely that the entire network will be compromised along with it [3]. For example, if an attacker manages to place itself between the communication of the controller and the other devices, it will be able to analyze the network traffic. The attacker can even trick network devices in the SDN network to believe that the attacker is the SDN controller, and thus, it can modify network configuration. This type of attack is called the Man in the Middle (MITM) attack [4, 5], and it is what will be the focus of this thesis. The aim of the thesis is to analyze SDN regarding a Man in the middle attack. I will identify in the literature methods to detect and protect against MITM, and thus to improve network security. A MITM is a type of attack where the attacker routes the traffic to itself instead of sending it to the true receiver. A common method to perform MITM attack against a SDN controller [3] is called an ARP-poisoning attack. In this type of attack, the attacker sends ARP (Address Resolution Protocol) packets into the network with the attacker MAC-address being the owner of IP addresses that the attacker wants to reroute traffic from. This thesis will focus on ARP-poisoning attacks.

This thesis is done as a literature survey on papers in SDN and MITM attacks and as an experiment to understand the relation of network latency during an ARP-poisoning attack. By doing this we hope to give foundation for security solutions for ARP-poisoning attack and to increase understanding of MITM attacks in SDN.

2. Background

This section will describe Software Defined Networks and Man in the Middle attacks and the tools and programs, Ettercap and Mininet that are used in the experimental part of the thesis.

2.1. Software Defined Network

Software Defined Network (SDN) is a new networking paradigm that enables the configuration and management of networks centrally [1]. This makes the network easier to manage and more flexible since adding or changing network configuration only requires configuration in the controller. This is done by using software to centralize the network management and separating the control plane and data plane in the network. This central controller is called the SDN controller. To be able to administer the traffic within the network the controller creates tables of actions for the network traffic, these are called flow tables. This setup of the flow tables can take time depending on the size of the network and its complexity.

An SDN can be seen as separated into three planes [1][2]. The Application plane is where the applications are run, these applicatitions are programs that communicate or gather resurces from the controller and the network, example an application to monitor general network capacity. The Control plane where the controller is, and the Data plane where the data traffic is. Communications between the planes are called either northbound or southbound and are seen from the controller, i.e. traffic between the application plane and the control plane is northbound and traffic between the control plane and data plane is southbound. The SDN architecture is depicted in Figure 1.

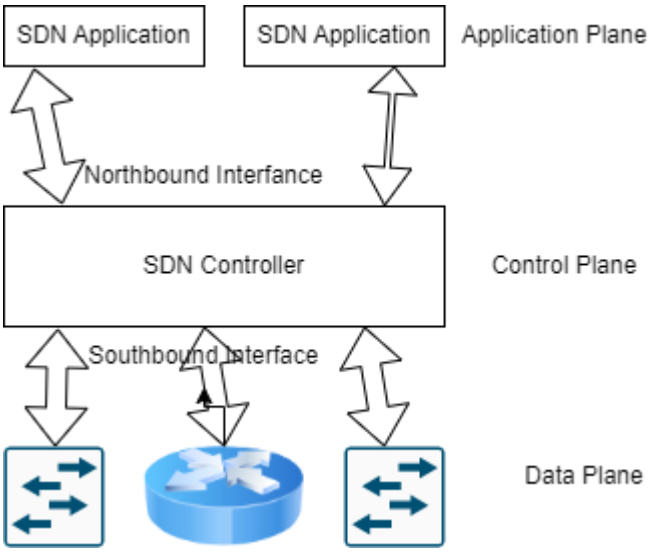

Figure 1: Overview of the planes in SDN.

By using a central controller for the network, the control plane is abstracted away from the rest of the network, creating an easier way to administer it [1]. Since the network administrators only need to tell the controller about the configuration for the network. This especially helps networks with a large number of devices in it. Which is increasing since the number of network capable devices are increasing with Internet of Things [6][7]. And with this increase there is a need for better methods for administrating and managing different types of network capable devices, as well as devices that may be in the network temporarily or changing locations inside the network often.

Since the security configuration can be centralized it is easier to monitor and detect malicious network traffic, but it also means that the controller can be a single point of failure for the security. But using SDN also adds to network traffic overhead in the form of the SDN control protocol that is used to manage the network devices. One example of an SDN protocol is OpenFlow that is described in the following section.

2.1.1. OpenFlow

OpenFlow [8] is an open-source protocol for administering SDN networks traffic in the network. It is the protocol that is used for communication between the devices so that it is possible for the devices to use SDN. OpenFlow is used to control the flow of the network traffic and does not require a specific type of network or hardware to work, more than that the devices used support and is able to run OpenFlow [1]. In the network OpenFlow runs on top of the Transmission Control Protocol (TCP) and the traffic uses Transport Layer Security (TLS).

2.1.2. OpenDaylight

OpenDayLight (ODL) is an open source SDN controller platform developed and supported by the OpenDaylight Project [9]. ODL is designed to allow flexibility when implementing a network with ODL, allowing the user to build a network controller that suits their needs. ODL does this by having a modular design. This modularity allows the use of other services, such as specific vendor solutions like Cisco solutions for ODL integration. and the possibility of creating and incorporating one's own services so that it better suits what the user of the SDN network needs.

2.2. Man in The Middle Attack

For networking, a Man in the middle (MITM) attack is when an attacker places themselves between a sender and receiver and impersonates the other, the sender thinks the attacker is the receiver and the receiver think the attacker is the sender [4][5]. By doing this the attacker can

read traffic or even change what is sent. This, however, requires the attacker to already be between the devices that they want to attack, or redirect the traffic through themself. If this is a local network that means the attacker is already inside the network before the attack starts, if this is true its quite hard to defend against. Especially for unencrypted traffic since the attacker can read most information in plain text, even if the traffic is encrypted it still means the attacker was inside the network, even if they are unable to read the traffic, with access to the network they can still perform other actions for example, increase the network load or analyze the network infrastructure.

2.2.1. ARP-poisoning

One way to perform an MITM attack is by ARP-poisoning.[3] This method uses the Address Resolution Protocol (ARP) that is used in networks to know what MAC address a IP address have. This protocol works by having the device that wants to know a mac-address for a certain IP address send a broadcast into the network asking what device has that IP address and then the device that has that IP address answers with their MAC address. ARP-poisoning is done by constantly sending out false ARP reply packets to the devices an attacker wishes to attack, this means that when the ARP broadcast is sent there is already a reply for it. The information in the ARP-reply is that the attacker's MAC address belongs to the other device, and this will redirect the traffic to the attacker. In this example host 1 will believe the mac-address of the attacker belongs to host2 and the reverse for host 2. This means that all traffic between these hosts will go to the attacker. The attacker can then choose if they want to listen to it, sending the traffic to their real destination.

For SDN MITM attacks can be a problematic if the attacker manages to make the clients believe the attacker is the SDN controller and thus enabling the attacker to be able to change the network.

2.2.2. Ettercap

Ettercap is a program that is used for performing MITM attacks [10]. It is supported and developed by the Ettercap Project. It is a program suite that contains several different types of tools to facilitate MITM attacks to be able perform different types of attacks, for example network sniffing tools and tools for ARP-poisoning attacks.

2.3. Minnet

Mininet is an open-source network emulator for SDN [11]. It supports OpenFlow as a SDN protocol and does not need any special hardware to be run, it can run on a single device and simulate different SDN networks on it. The emulated Mininet networks behave similarly to how a real SDN network would behave so that it is possible to run tests with it and get a feel

how it will work in a real environment. Because of this Mininet is good to use for teaching and testing purposes. Mininet can be installed on regular commercial off-the-shelf computers.

Mininet has several different network configurations and designs pre-configurated, these can easily be changed if needed using the Python programming language [12]. Therefore, Mininet can be used to emulate a wide variety of computer networks. Mininet also supports several different SDN controllers, for example, OpenDayLight, Ryu and Floodlight [13].

3. Related Work

There have been several papers in the area of SDN, for example Al Hayajneh et al. [14] on using SDN to increase security in IoT networks, the paper is about implementing SDN for IoT device networks. This paper's goal is to make it possible to change the security of the network without modification of the IoT devices in the network. For security, the main goal is to protect the IoT devices that uses HTTP from a man-in-the-middle attack, this since IoT devices are especially vulnerable against that type of MITM attack. They implemented their experiment with a Raspberry PI, Kodi media center and with the OpenFlow protocol. The researchers conclude that their implementation was evaluated as more resilient to attacks.

There are papers on how to use SDN to make a network for specific device networks, for instance SD-WISE [6], is a paper to introduce a SDN network that is designed for wireless sensors. This paper proposes a network that has features to abstract network node resources so that it is more flexible and as such able to work better in a wireless sensor network. This makes it easier to handle adding and removing nodes, and handling of nodes changing locations.

There are also general-purpose studies, for example a study by Arahunashi et al. [15] on various SDN controllers done with an emulated environment in Mininet. In this the researchers perform an analysis of several SDN controllers' effect on the network in terms of average delay and throughput. The researchers test the different controllers for several different network topologies, for example linear and mesh topologies.

There are also several works about security against MITM with SDN. For instance, the paper Detection MITM Attack in Multi-SDN Controller [16] in which the authors present several different attacks on SDNs different layers and interfaces and created two MITM attack scenarios to test against different controllers. The focus in this paper is the Open daylight controller. This was simulated with Mininet. Their simulations indicated vulnerabilities and they give several recommendations and solutions for security against MITM attacks.

Another IoT security paper is A survey on emerging SDN and NFV security mechanisms for IoT systems [7], it is a survey on security mechanisms with Network Function Virtualization (NFV) and SDN for IoT systems. In this they also do a general study on security threats against IoT systems and one of the goals with this was to give options and directions for future research on SDN and NFV.

There are also papers on MITM attacks on SDN for instance "New context-based node acceptance CBNA framework for MitM detection in SDN Architecture" [17] the researchers have goals to mitigate threats on SDN networks like MITM. They do this by studying attacks scenarios to reduce the vulnerabilities during authentication, they also propose an algorithm to detect malicious SDN devices.

There is a paper on ARP MITM attacks in SDN by A. M. AbdelSalam et al. [18]. In their paper they propose a solution to ARP-spoofing attacks in SDN, they also touch on the difference between traditional networks and SDN. Their solution is to use SDN to monitor ARP to mitigate the attacks. The researchers concluded that their proposed solution works to reliably mitigate different ARP attacks. Their paper uses packet monitoring in order to detect and mitigate the attacks. In my thesis we aim to detect ARP-poisoning attacks by detecting latency changes caused by the attack

4. Problem Formulation

This thesis will examine the security in SDN in regard to MITM attacks and a method to detect those attacks. The flaws I will focus on are those that were revealed in previous works about SDN. I will focus on finding a method of detecting MITM attacks in SDN. This to increase the knowledge regarding MITM attacks in SDN, so the security for the networks will increase. This thesis will not look at ways to reduce the flaws that are outside SDN, for example using encryption.

The reason I choose to focus on this is that security is important in most networks and the security of SDN will increase if there are security measures that can be used. Especially if those measures do not require other programs or solutions.

The questions I have formulated to answer this are the following:

• What vulnerabilities against MITM attacks does SDN have?

• What is a possible method to detect MITM attacks in SDN?

5. Method

This thesis is done as a literature survey and an experiment [19, ss 85-99.]. The information I gather is from other studies and papers and different technical manuals, thus it is done as a literature survey, but I will also need to check the information from them by testing them so this needs to be done as an experiment.

The main part of the project will be done as a literature survey of other related works to formulate what tests I need to do and how the tests will be done. After that I will familiarize myself with the equipment and software that I will need to use in the tests to understand how they work and how to perform the tests I want to do.

The experiments will be done in a simulation of SDN with Mininet since it will be easier than doing it in a real network environment and with Mininet I do not need the devices themselves since it is possible to simulate them on a computer. This will make it so I will not be able to get real data values, but Mininet's simulation will work for the small experiments planned.

The experiment that will be done is an analysis of latency in the SDN network during an ARP-poisoning MITM attack, this will discover if there are any differences during an attack from during normal traffic in terms of network latency. The difference in latency could then be used to develop methods to detect this type of ARP-poisoning MITM attacks.

The beginning of the thesis started with finding several related papers about SDN and MITM attacks so to better understand the subjects, and to form the experiment. The next step was finding tools that could be used for the experiments. The tool for SDN emulation that was chosen for this was Mininet due to prevalent instructions and that it was widely used in SDN papers. For the tool that would be used for the MITM attack Ettercap was chosen due to being able to perform the wanted MITM attack, ARP-poisoning.

6. Ethical and Societal Considerations

This project will not handle information from or about people, so there is no ethical problem in that regard. There should also be nothing that is secret or needed to be confidential in the project, but if it is used it will be removed before completion. Any other ethical or societal considerations will be considered and kept in mind during this project.

Any finding that I find will add to knowledge, and if I find methods to better protect SDN networks then that will be generally good. Since I cannot speculate on what the networks protects and why it is attacked with MITM, the general idea is that better methods to protect and security one's own networks is good. A point that might be seen as a negative from this is that it gives attackers more information about vulnerabilities, but this information already exists if attackers want to use them.

7. Literature Survey

In this section we summarize methods used in MITM attacks against SDN and countermeasures to mitigate and prevent such attacks. We perform a literature survey and summarize the results.

The literature survey is done in order to gather information about SDN and MITM attacks from different papers. To find the papers that Google Scholar was used, a search engine for academic papers. To find the papers, we used "SDN" and "Man the middle attack." as search terms to find suitable papers. After that we read the title and went through the abstract quickly to check if the papers had relevance to this thesis.

In the paper "New context-based node acceptance CBNA framework for MitM detection in SDN Architecture" [17] the researchers describe Man in the middle attacks as "The MitM attack is a scenario where a hacker listens to the communication between two legitimate users and forges the exchanges in a way that allows him to pass as one of the legitimate parties".

There are several different ways to look at types of MITM attacks one perspective is to divide into 4 types [20]:

1. Spoofing based attacks are an attack where the attacker interjects themself between two hosts to control the traffic between them. This is done via a spoofing attack.
2. SSL/TLS MITM is an attack done by placing themself between two hosts that are using SSL or TLS. One example where SSL is used is that it is often used between a web browser and the HTTP server. This requires the attacker to use SSL/TLS to link themself with the hosts.
3. BGP MITM is an IP hijacking attack where the attacker redirects the traffic through their network so that they have access to it.
4. False Base station MITM the attacker forces the victim to connect to a base station and from it manipulate their traffic.

These methods are just one way to differentiate between MITM attacks, it is also possible to do more strict groupings, for example ARP spoofing attacks are a form of spoofing attack.

Most of the MITM attacks can be protected against by increasing the use of identification and authentication so that the administrator knows what devices there are in the network and that only allowed devices are in the network. This protection, however, can be circumvented if the attacker manages to authenticate themself. For example, SSL and TLS are used for secure authentication, but there are MITM attacks that target that step. One other method is to analyze the network to create a standard for normal traffic in the network and what devices in the network are communicating, this can then be connected with alerts when the traffic no longer is standard. This, however, cannot be done in most network solutions, but knowing what traffic is used in the network is usually good for administrators.

In Table 1 is the categorization of MITM attack types that this thesis is using, together with some protections for that type of attack. One security improvement that is true for all types of attacks is to increase knowledge and the risk of them, no security implementation will work if the users do not know or ignore security risks [4].

Type of MITM attack	Protection Improvements	Description
Spoofing	Protection against this is to have authentication of hosts, to know that the host is who they say they are	Targeting the identification by falsifying what device you are, for example Ip address
SSL/TLS	Handle secure keys securely. Validate certificates. Do not accept certificates from unknown sources	Targets the authentication step, by compromising the security certificate or getting a valid keychain.
BGP	Setting up detection of this type of attack, checking if other peers are hijacking your own Ip and knowing what Ip's your peers are using,	Targets the external routing by compromising the BGP and falsely claiming to have certain Ip-addresses.
False Base station	Protection can be improved be better authentication and validation. Not using one-way authentication for example	Targets traffic that needs to go through a base station and falsifies being that base station.

Table 1: Table over different types of MITM attacks.

The type of MITM attack that is done in the experiment part of the thesis is an ARP-poisoning attack, this type of attack is also called an ARP spoofing attack [18]. A reason it can be categorized as a spoofing attack is that ARP is used for the identification of MAC addresses that an IP address has. As such it is a form of false identification and falls under the spoofing type of MITM attacks. ARP attacks can also be even more separated into request-based attacks and reply-based attacks [18]. A request-based ARP attack is done by the attacker sending several spoofed ARP requests into the network in order to flood the victim's address mapping, this means the victim will not be able to properly communicate in the network. An ARP reply attack is when the attacker spoofs an IP address by sending the victim unsolicited ARP-reply

messages in order to make the victim believe the attacker's MAC address belongs to that IP address, it is the ARP-poisoning attack that this thesis is focusing on in regard of ARP-poisoning.

The usage of SDN can improve some vulnerabilities by introducing a centralization of the network administration. This allows the administrator to easily apply security solutions to the entire network. But introducing SDN also introduces a single point of failure and vulnerabilities that are more dangerous to a network using SDN [16][17].

8. Experiment setup

The goal of the experiment is to find out if there is a noticeable difference in network latency during normal network operation and during an ARP-poisoning MITM attack. The presence of a noticeable delay in latency between regular network usage and that experienced during an ARP-poisoning attack may be used for identifying and detecting such an attack. The experiment is performed in a network environment created using SDN simulation software (Mininet). The latency in the experiment is measured as the time elapsed between the transmitting of an ICMP ping packet and the reception of the answer. The ping is measured by having host 1 ping host 2 20 times and taking the average time from those. The ping command that is used is the inbuilt ping command from the operating system. This latency reading is done in two different network states, during low traffic, and during high traffic. We define a low-traffic network state as a state when the hosts are idle and only necessary control network packets are transmitted. The high-traffic network state adds additional traffic caused by transmitting a large file (Ubuntu ISO file, ~3GB) between host 1 and host 2. The transmission is done using the SCP program.

The experiment's goal is to analyze the network latency during ARP-poisoning attack and that by doing this it might be possible to develop a detection solution for this type of MITM attack.

8.1.1. Detection of ARP MITM attack in SDN

The implementation is done on an Ubuntu Linux environment, I created a virtual machine on a computer with 2 GB memory and 10 GB storage that runs Ubuntu 20.04, and it is created with Oracle Virtual Box. The network speed for the simulation is limited to 100Mb/s bandwidth.

There is a ready Mininet Virtual Machine downloaded from Mininet's website [11] and then Mininet is installed, I also need to install a program for the MITM attacks which is Ettercap, this was installed using a basic Linux package installer APT to download and install it. This installation takes most of the project's time due to several minor problems with both Mininet, Ettercap and the Linux operating system. One major problem I had was that Mininet was installed without an SDN controller, so it was possible to start but no traffic worked in the simulation. This was eventually resolved by reinstalling the operating system and then downloading the files from Mininet's GitHub, instead of installing Mininet via the APT command. The installation was then done by executing the installation file that was in the files that downloaded. One other need before starting was to have a terminal-prompt that would allow X11-forwarding, the ability to open new terminal windows from inside a session, this is needed to be able to SSH to the different devices in the simulation while running Mininet. X11-forwarding is also needed to run Ettercap in graphical mode on the attacker, since a new shell is needed to start it.

For the experiments, I used Mininet to create a simulated wired network with 2 hosts spread over 2 switches, the switches are connected to each other and an OpenFlow controller, see figure 2. And each switch is connected to one host, and switch 1 is also connected to the attacker.

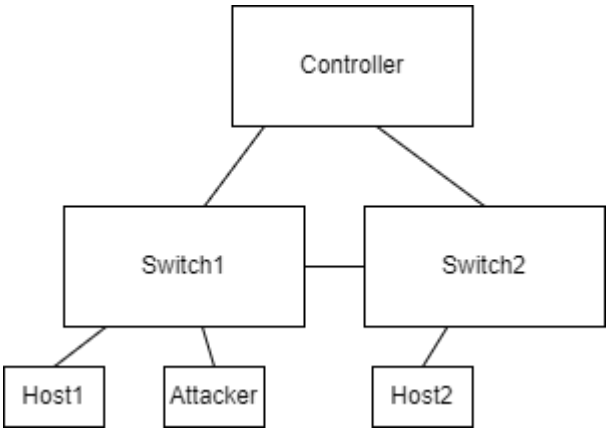

Figure 2: Layout of network.

These devices in the network were created and connected with the network using Mininet. After creating the network, I ran a ping from all devices in the network using Mininet's inbuild pingall command, it runs a ping from all devices in the network and is a simple way to check connectivity. This was done both to see that the network worked and in order to generate some traffic so that the controller could generate flow tables for SDN.

After that I needed to enable IP-Forwarding on the attacker, so that Ettercap would be able to redirect the traffic towards the destination. After this I started Ettercap on the attacker to start the MITM attack. The attack setup was to use an ARP-poisoning attack. The attack was done by selecting the two normal hosts as targets for the attack and sending out ARP messages telling the network that the attacker was the other device. By doing this all traffic from host 1 or host 2 towards the other will be sent through the attacker and then to their destination, instead of directly to its destination. The goal is to examine how the network behaves during the attack. This is examined with ping from host 1 to host 2 and taking the average latency from 20 ping attempts. The goal is that by examining the latency during low traffic, high traffic and during the ARP-poisoning attack there will be enough differences to use latency as a detection method for that type of MITM attack.

For experiment during high network traffic the traffic was done by transferring a large file from host 1 to host 2. The file used was a large file (3 GB). The experiment performed during high traffic is to find if the ARP-poisoning attack is detectable during periods of high network traffic, and a large file was used in order to have a longer period of traffic for the experiment. The time it took to transfer the file was 5.1 minutes, this time is the average from five repetition of the

experiment.
The SCP program was used to copy the file between the hosts. The program was used in host 1 and set to copy the file to /dev/null on host 2, this requires the user on host 1 to be allowed to access host 2 and that SSH is allowed since SCP uses the SSH protocol. The target of the transfer does not need to be /dev/null but the experiment had limited storage, and the path /dev/null on Linux is a path that removes anything sent to it instead of saving it, so it was easier to send it to that path than to remove the transferred file after each experiment. Each experiment took ~1 minute to perform.

9. Results

In this section is the gathered results of the experiment. The type of attack I did was having the attacker trick the devices into believing the attacker was the destination with ARP-poisoning. This was done to either stop the traffic or to act like a switch and send the traffic to their destination but listening to the traffic.

The attacker was able to interject themselves between the two hosts and get them to send traffic to the attacker instead of each other. But for the switch and controller it did not work due to the traffic between them not using IP-addresses, the switch and the controller communicated with MAC addresses without using ARP.

For MITM attacks with the attacker redirecting the traffic through itself and just listening there was an added latency to the traffic, see table 2. This was found out from the latency on Ping packet latency between the hosts during low and high traffic and during the MITM attack. The high traffic was simulated by setting up a large file transfer from host 1 to host 2.

	Average latency for Ping package:	Average latency for Ping package during MITM-attack (ARP-poisoning).	Average latency during high traffic	Average latency for MITM attack during high traffic
Latancy in milisecond (ms):	0.2 ms	12.7 ms	0.3 ms	13.4 ms

Table 2: Table over latency in the test network.

The traffic latency shown in table 2 is the average latency taken from 20 ping packages and rounded down to 1 decimal. This was chosen so that the tests would not take too much time to do, and the result be easier readable. The experiment itself was repeated 10 times to make sure that the test results were correct.

10. Discussion

From the results gained the ARP-poisoning attacks are noticeable in the network by significantly increasing the network latency during the attack. This increase in network latency went from 0.2 ms to 12.7 ms during normal traffic and from 0.3 ms to 13.4 ms in the scenario of high traffic. This finding could be used to design a detector to continuously observe the latency in the network and if the latency increases enough, this detector can trigger some security measures, for example alerting a network administrator.

However, stopping MITM attacks from starting is hard, and even with high authentication and other security it is still possible to be attacked. And most security will increase the difficulty for users in the network, for example, requiring authentication for network use might take up time for the users. There are methods to automate authentication, for example keycards or tie the authentication to the machine itself, but these also open risks, the keycard or machine can be stolen.

From the results we can see that the MITM attack that was done in the experiment, ARP-poisoning, severely increased the network latency. Our hypothesis for the increasing latency is that much of the increase in latency is due to the attacker needing route the traffic between the hosts it is attacking, if the attacker wants to let traffic continues like in the experiment. While Ip forwarding enables the attacker to be able to act as a router it does not actually have the hardware, so the traffic will not be routed as effectively as a real router. In the experiment during normal traffic the packets between host 1 and host 2 need to pass between the switch to reach each other. During the MITM attack the hosts traffic now also pass through the attacker which adds a few more steps.

One limitation the experiment had was that it was done in a limited network environment, a simple topology with just a few devices, another is that the experiment was only done in a simulated wired network. These limitations reduce the conclusions able to be drawn from the experiment. One difference between wired and wireless networks that I believe would affect this experiment is that the medium used handles traffic differently. The air is far more difficult to limit access to than a wired network.

This thesis answer for the question of detecting MITM attacks was to detect the changes in network latency caused by an ARP-poisoning attack, there are however other methods that can be used in order to detect MITM attacks. One method is to use traffic monitoring to detect malicious traffic [18].

It might be possible to reduce the latency difference from between the network during ARP-poisoning attack and standard traffic by changing the settings in Ettercap or using other programs, but that was outside of the scope of the project and there was not enough time to explore that idea.

11. Conclusion

The goal of the thesis was to give an answer to what vulnerabilities against MITM attacks SDN have, and what a possible method to detect MITM attacks in SDN is. The first question regarding vulnerabilities was answered in the form of the literature survey where I identified related works and provided a summary of the types of MITM attacks as well as protection mechanics against these attacks. As for different types of MITM attack, there are different ways to categories the types of attacks, I used the categorization in table 1.

The second research question, what is a possible method to detect MITM attacks in SDN? This is answered by performing an experiment that shows significant increase in network latency in SDN networks during an ARP-poisoning attack. This finding can be used to create a detection system that is able to observe the network latency and notify the network administrator when the system detects an increase in latency.

One limitation with this thesis experiment is that there were no security solutions in the network used, and as such the results might be different if tried in a network that have some security solutions in place. Another limitation is that this attack was only tried with the wired topology in figure 2, the layout of the network might change the results as well. But this thesis has given answers to the questions it set out to answer.

12. Future Work

There are questions that can be answered with future works in the area of SDN and MITM attacks. Those work could focus on MITM attacks where the infected device is part of the SDN. Other areas to focus on could be how this works with an SDN controller that is outside the local network, and with an SDN that contains several controllers.

One topic for future research that can be interesting is a survey for how different types of MITM attacks works in SDN, since this paper focused on an ARP-poisoning attack on a simple wired network topology, a topic for another thesis might be the difference in latency due to MITM attacks in wired and wireless SDN.

Another topic for a future study is to focus on a deeper analysis of network change during an ARP-poisoning attack, there are most likely more efficient methods to perform these types of attacks on a network so there is a need for a deeper dive in this area.

References

[1] M. S. Olimjonovich, "Software Defined Networking: Management of network resources and data flow," 2016 International Conference on Information Science and Communications Technologies (ICISCT), 2016

[2] Nkosi, Mpho, et al. "Classification of SDN distributed controller approaches: a brief overview." 2016 International Conference on Advances in Computing and Communication Engineering (ICACCE). IEEE, 2016.

[3] 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. 2015.

[4] Tunggal, A. What Is a Man-in-the-Middle Attack and How Can It Be Prevented. *UpGuard.com*. Jan 23. 2022 [online] url: https://www.upguard.com/blog/man-in-the-middle-attack

[5] Mallik, Avijit. "Man-in-the-middle-attack: Understanding in simple words." Cyberspace: Jurnal Pendidikan Teknologi Informasi 2.2 (2019)

[6] ANADIOTIS, Angelos-Christos, et al. SD-WISE: a software-defined wireless sensor network. *Computer Networks*, 2019

[7] FARRIS, Ivan, et al. A survey on emerging SDN and NFV security mechanisms for IoT systems. *IEEE Communications Surveys & Tutorials*, 2018

[8] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: enabling innovation in campus networks. SIGCOMM Comput. Commun, 2008

[9] OpenDayLight Project, 'Platform Overview - OpenDaylight' [online], Available: https://www.opendaylight.org/ . [Acceses:15 May 2022]

[10] Ettercap Project, Available: https://www.ettercap-project.org/ [Access. 30 oct 2022]

[11] Mininet Contributors,' Mininet: An Instant Virtual Network on Your Laptop (or Other PC) - Mininet'[online], Available: http://mininet.org/. [Accesses: 15 May 2022]

[12] S. Asadollahi and B. Goswami, "Experimenting with scalability of floodlight controller in software defined networks," 2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT), 2017

[13] A. K. Arahunashi, S. Neethu and H. V. Ravish Aradhya, "Performance Analysis of Various SDN Controllers in Mininet Emulator," *2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, 2019

[14] AL HAYAJNEH, Abdullah; BHUIYAN, Md Zakirul Alam; MCANDREW, Ian. Improving internet of things (IoT) security with software-defined networking (SDN). *Computers*, 2020

[15] A. K. Arahunashi, S. Neethu and H. V. Ravish Aradhya, "Performance Analysis of Various SDN Controllers in Mininet Emulator," *2019 4th International Conference on Recent Trends on Electronics, Information, Communication & Technology (RTEICT)*, 2019

[16] A. Sebbar, M. Boulmalf, M. Dafir Ech-Cherif El Kettani and Y. Baddi, "Detection MITM Attack in Multi-SDN Controller," *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, 2018

[17] SEBBAR, Anass, et al., New context-based node acceptance CBNA framework for MitM detection in SDN Architecture. *Procedia Computer Science*, 2019

[18] A. M. AbdelSalam, A. B. El-Sisi and V. Reddy K, "Mitigating ARP Spoofing Attacks in Software-Defined Networks," *2015 25th International Conference on Computer Theory and Applications (ICCTA)*, Alexandria, Egypt, 2015

[19] K. Säfsten, M. Gustavsson, Forsknings-Metodik, För ingenjörer och andra problemlösare, Lund: Studentlitteratur, 2019

[20] M. Conti, N. Dragoni and V. Lesyk, "A Survey of Man In The Middle Attacks," in IEEE Communications Surveys & Tutorials, vol. 18, no. 3, pp. 2027-2051, thirdquarter 2016, doi: 10.1109/COMST.2016.2548426.