ABSTRACT

SDN, or software-defined networking, is a network strategy that separates the control and data planes. The data plane is dispersed throughout the network parts, but the control plane is conceptually centralised. To maintain dispersed state information of the traffic flows, the real-time network must have distributed controllers. Distributed SDN controllers can handle varying network traffic with the use of software-based solutions, and the controller's configurations are dynamically configured. In this study, a switch is configured as a firewall using a bifold approach. Primarily, the project focuses on configuring the switch without passing the configurations to the SDN controller. The second approach uses SDN controller to implement the set of firewall rules on layer 2 and 3 of the networking architecture. The purpose of the project is to use the features of SDN controllers to furnish a model which provides centralized security features to comply with the scalability requirement of the network. Further, the model’s performance is analysed by considering different aspect of the packet transfer rate. The Mininet emulator was used to establish the analysis' findings. The results of the performance study show that distributed SDN controllers outperform centralised controllers in terms of performance.

Table of Contents

1. INTRODUCTION

Middlebox in networking is referred to as the device integrated to demonstrate functions other than basic packet forwarding. These devices perform some intermediary functions like inspection, filtering, manipulation and transformation of the data packets. As these devices behave opposite to the conventional packet forwarding devices therefore are commonly used for security purposes. These devices can be firewalls, intrusion detection systems and layer 2 switches providing firewall functionality. These middleboxes are majorly used in a variety of enterprise network to introduce security services to the physical networks and the cloud. (Ontiveros, 2019) Examples of middleboxes include firewall, load balancers, NAT (Network Address Translation), and deep packet inspection (DPI). With the increasing complexity of networks and internet, the demand of enhanced security services is rising as perpetually. To implement an enhanced security system the organisations are relying on not only the hardware security tools but also on a plethora of software measures like software defined packet filtering and inspection firewalls. A traditional firewall performs the packet filtering functions by interacting with the IP addresses and the port numbers (Michali,2022). The functionality of the firewall involves analysing the data packets against a set of user defined rules. These rules define the fate of the data packets passing through the security system. The advancement in the complexity of networks has given rise to the need of advanced and complex security systems. Therefore, advanced firewalls and security systems are a common component of the network security.

The structure of the internet is vulnerable in its design. It can attract plenty of threats and backdoors. However, the structure can be altered to impart levels of security in the network. It is generally achieved using different encryption techniques and firewall systems. (Daya, 2013)
Software defined networks has shown promising future in resolving threats by providing strong security to the network. Hence, plethora of research work has been carried out around different usage of SDN. (Xia et al., 2015)
SDN operates by separating the data plane and the control plan which eventually supports the programmable functionality of the network. SDN can be used to configure the hardware switch using OpenFlow protocol (Han and Lee, 2015)
The main component of the network is the scalability which is an ability to add more computer and devices to the network. With adding new devices on the network, the network security can be compromised as the devices can launch attacks and breach into other devices. Therefore, the security of the network is to be treated with utmost importance. This project aims at designing a programmable network switch virtually using SDN - software defined networking to filter the packets being carried through the network and only allow the required packets to pass through while dropping the other packets hence, acting as an internal packet filtering firewall.

According to a survey conducted in 2008, a whooping 80% of the users and organisations relied on antivirus programs and 37% were dependent on firewalls and other security measures (Kumar et al., 2008). Over the last decade, the paradigm has shifted from most of the organisations choosing antivirus software over firewall to almost everyone using firewall as a first line of defence. This shift in reign has inspired a plethora of researchers and security analysts to perpetually develop new and advanced strategies to improve the firewall architecture. Furthermore, along with the use of hardware firewalls, a significant research and

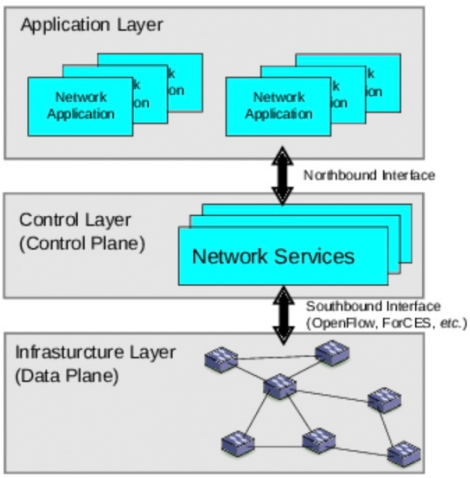
study is conducted in the field of software firewall and other software defined networks to provide complex network security solutions. Similarly, the purpose of this project is to create a software defined network where a switch acts as a firewall providing the packet filtering capabilities on layer 2 and layer 3. The project encapsulates the use of python to develop a software defined network using Mininet. The defined network is a basic architecture of three switches and 6 hosts creating a topology described in the further section of the reports. The two sections of implementation focus on, first, implementing firewall functionality without using a controller and second using a controller to implement the firewall services on layer 2 and 3. The design and implementation is followed by the analysis of the performance of the network with and without the firewall to understand the load generation and bearing capacity of the network. The study is conducted using virtualization to provide a base for the research.

The project aims to answer a few research questions posted by the field of network security.
RQ1: Can SDN be used to create a switch performing the firewall functionality?
RQ2: Are software firewalls effective in providing security to the networks?
RQ3: How can SDN be used to enhance network security?
RQ4: Can a centralized firewall be defined in a network using SDN?

The hypothesis of the report encapsulates that an SDN can be used to define a switch as a firewall. The switch operates as a firewall providing the basic traditional firewall functionality to the system. Furthermore, the SDN controller as a firewall is efficient in providing security to the system.

1.1 WHAT IS SDN?

SDN stands for Software Defined Networking. SDN is a tool to enhance the programmability of the network. The main idea behind software defined networking is to separate the data plane from the control plane. (badotra & Singh, 2017) The segregation of the planes enables the user to program the network as per the requirements. The mentioned technology has a three-layered architecture.



(Braun & Menth, 2014)

The standard SDN architecture defines three different layers, these are, Application layer, Control Layer and Infrastructure Layer. The application layer is the first layer in the architecture. This layer constitutes the network applications used by analysts and engineers. The applications like Load balancers, Intrusion detection systems, firewalls, etc. fall in this

category. SDN can also be used to replace the need of physical firewall and security software by using the controller which can control the Data plane's behaviour. Control plane is the brain behind software defined networking. This is where the centralized SDN software controller resides. The controller is capable to manage the policies and traffic flowing through the network. Finally, infrastructure layer is a more of a physical layer and is made of actual network infrastructure like switches used to forward the packets to the desired destination. Following thee layered architecture, SDN also includes APIs. The northbound API is designed to aid the communication between applications and the controller. However, southbound interface is designed to enable the communication between the controller and the switches. An example of southbound interface is OpenFlow.

1.2 HOW DOES SDN WORK?

The basic SDN model operates on a controller architecture. The basic SDN model follows the open flow architecture. The open flow switches broadly consist of three segments, these are flow table, secure channel and open flow protocol (Hui et al., 2021). All the data packets matching a specific header is added as an entry to the flow table. These flow table entries are further classified into three fields. The first section is the matching field, this field consists of a variety of selected header packets that define the flow. Following that is the action field. All the incoming packets pass through the matching field and based on the criteria the action to be conducted on those packets is then decided. The action field consists of those actions that decide the fate of the packets. The final field is called the field of statistics. This field is essentially a counter which keeps a track of all the packets that pass through the system. The purpose of this field is to keep a count of all the packets and be able to trace back in case of requirement.

The flow table is accompanied by the second section of the architecture called the secure channel. The role of the secure channel is to enable a safe and secure communication channel between the switch and the remote controller.

The final one is the OpenFlow protocol, these are the set of rules that enable the controller to communicate with the switch. These set of rules define how the switch will operate in the network. For the purpose of this project, the mentioned protocol is crucial as it defines the switch to operate as a security firewall and decide the fate of the packet. The role of the controller also includes adding and removing flow entries from the flow table.

1.3 STATELESS VS STATEFULL INSPECTION FIREWALL

The stateless inspection firewalls are the most basic type of firewall which allows all the packets to pass through a common set of rules defined by the network security agent. The rules define the fate of the packets. This implies that all the packets entering the system are checked against the set of rules defined. Furthermore, the firewall system checks the packets against all the rules from the list. If the packet matches any of the rule the actions assigned against the rules are then implemented on the packets. The stateless firewalls decide the actions on the packet based on the static information like the source and destination. The system just checks the source and the destination of the packet, and the decision is based on just this piece of information (able, 2021). This firewall is a conventional firewall which has been used in most systems and devices over many years. These firewalls rely on the packet filtering rules that specify a match conditions. If the conditions are satisfied, the firewall take actions from the list of predefined rules on the packets; otherwise, the packets

are dropped if the conditions are not met. This firewall functionality is essentially pivotal for conditions where each packet must be tested. However, it does not analyse any other aspect of the packets which is the main drawback in stateless firewalls. Following the drawbacks of stateless firewall, the concept of stateful firewall was introduced. These firewalls are designed considering the holistic impact of packet filtering. Therefore, the main aim behind the stateful firewall is to analyse the state of the network traffic flows and packets (CDW, 2022). In this case, the fate of the packet is decided depending upon the pattern of the network traffic flows. This enforces strong security against large attackers. The major advantage of the stateful firewall is the ability to base the filtering decisions on the past and present information (CDW, 2022). Despite of a plethora of advantages of the stateful firewalls, these firewalls are highly vulnerable to attackers trying to take control of the firewall in the absence of the updated version of the software. Furthermore, these firewalls are also susceptible to man in the middle attacks.

Despite the risks, many organisations and security analysts are introducing new and enhanced ways of developing stateful firewalls to improve the overall security of the system. For the scope of this project, a switch is programmed to behave as a stateless firewall implementing the basis packet filtering operations. This is due to the limitation of the time, resources and research. Furthermore, this research can provide the base for future works and experiments that can be carried out to create a stateless firewall using a switch.

1.4 SWITCH

A switch is a basic hardware device known to forward the packets to the desired destination. The data packets usually consist of packet headers. The packet headers contain information about the source and the destination. Gathering the information from the headers, switch forwards the packet through to the destination. The traditional switches can be programmed using Software Defined Networking (SDN). A python script can be used to configure the switch to behave differently. This can be achieved by using an OpenFlow controller. The OpenFlow switch can further operate as a firewall following the pushed configurations. The configured switch then follows the basic OpenFlow architecture as mentioned above. The OpenFlow switch has three sections which are flow tables, set of commands and a secure channel (Khodbhaya et al., 2020). The previous section of the report highlights the importance of all the three sections of the OpenFlow controller. The switch as a firewall follows the same logic as that of the traditional firewall. The firewall rules are integrated with the OpenFlow switch using python; the rules are pushed with other configuration using the code. Each data packets passing through the network enters the switch interface. Primarily, the switch makes an entry in the flow table and then tests the packets against the firewall logic as described in the list of commands. These packets are checked with highest to lowest priority and specific actions are performed on the packets. These actions are also defined in the list of configurations. All these actions take place in a secure environment using a protected communication channel.

2. PROBLEM STATEMENT

The advancement in technology has not only open doors to new and innovative tools and software but has also introduced new threats and vulnerabilities to the networks. As per the recent statistics presented by (Sec, 2022) more than 71 million people fall prey to the increasing cyber crimes every year. With an increasing number of people using the

technology, the number of attacks every year is skyrocketing. A variety of security measures are persistently being developed by the researchers and analysts. The security engineers are focussed to ensure protection by researching and continuously testing the protection and prevention tools (Borky & Bradley, 2018). The security analysts persistently develop or improve new and existing technologies. As mentioned earlier, SDN has been proven to be efficient for cyber security. The prevalent use of SDN in enterprise is influenced by its complex structure. The amalgamation of the global view of the network with the flexibility influenced by programming has been used to improve number of existing Intrusion Prevention Systems (IPS) and Intrusion Detection Systems (IDS) (Scott-Hayward et al., 2013). Owing to the efficiency in security provided by the functions and operability of SDN, emerging analysts and security engineers has used SDN to expand the scope of network security. Therefore, to overcome the bridge created by the attacks being introduced through sophisticated tools and software, it is imperative to enhance the security measures by using highly reliable technology.

The purpose of the project is also inspired by the literature review conducted in the field. The further section of the report highlights a plethora of existing literature which supported the initial idea of this research. Moreover, the experiment conducted by creating a virtual network using Mininet is concluded into analysing the efficiency of the switch as a firewall. The results aim to support the problem statement by providing a centralised firewall using switches to improve the security of the network. To tackle the increasing number and variety of the attacks, these security measures play a pivotal role in providing the base of many cyber security analysts and engineers. Henceforth, the project is aimed at reducing the gap in the technology.

3. LITERATURE REVIEW

The previous section of the report highlighted that the presented problem is faced by a diversity of population. Therefore, rigorous research and studies are persistently contributing towards designing different security measures. One of the most prevalent technologies is the use of SDN to enhance network security. As the paper focuses on the use of SDN for enhancing the security of the network, therefore, the reviewed literature revolves around the security implications of the SDN and SDN controllers.

Research conducted around the impact of SDN's architecture on the security of the network. The authors studied the architecture and features of SDN. The study was aimed at using the provided feature to improve the security. The results also identified the use of SDN architecture for detecting many big attacks. They suggested that SDN can be easily configured and re-configured to modify the behaviour of a packet. This also provides the ability to any SDN network to add new network functionalities. It supported the SDN controller to control the flow of each packet. Moreover, the study also highlighted the ability to observer the traffic patterns using a node which provides the global view of the network. This analysis supported the understanding of the flow of the data packets. The global view of the network also supports in determining and understanding the IP routes, source, destination etc. The information extracted through the experiment is crucial for the development of effective intrusion detection and prevention systems. The authors proposed a DFSA system which is capable to use the features of the SDN network to detect the major attacks known to IP networks. The presented theory implemented GFSA in a SDN network. This module is responsible for alerting the user after encountering a threat. The system is efficiently designed and tested for a reliable threat and intrusion detection system. The study also highlighted the

future work in the area. The authors suggested that the SDNs ability to change the flow tables can prove crucial in real time threat detection and prevention. The flexibility provided by SDN ensures that an automatic reaction can be initiated upon threat encounter. The reaction can be a set of actions defined by the user. These conditions depend upon the desired requirements. The main objective of the project is to ensure filtering the packets based on the historical and present information. This is accomplished using 2 different modules namely, LFSA and GFSA. The LFSA module is executed in the SDN switch and is responsible for matching all the data packet sets with the existing set provided to the system initially while designing the system. All the sets that do not match the provided set are then passed onto the module GFSA. This module is implemented in the SDN controller and decides the fate of these packets along with using them to further train the system. This approach is highly beneficial for analysing the large data packets or the traffic coming in bulk from heavy sources to classify them as potential threats. The described approach was intended to reduce the traffic overhead and minimize CPU cycles. Their research not only reduces the overhead but implements the concept of the distributed firewalls where the SDN controller installs the firewall rules on all the SDN switches described by the topology.

Another study around building a reliable firewall for Software Defined Networks was inspired to address one of the major challenges posed to the software defined network. The security of networks has been a challenge for many analysts and security specialists. With the introduction of SDNs the vulnerabilities have increased which has eventually led to the rise in security measures. This inspired the authors to conduct research to develop a tool called FLOWMON. (Hu et al., 2019). FLOWMON was an open-flow based firewall aimed to provide real time violation detection and resolution. FLOWMON operates on the fundamental of checking the flow path space against the firewall authorization space. The experiment was conducted on a real-world network topology. Despite of the positive results of developing the firewall, the challenges revolved around inducing performance overhead. The benefits of the firewall were not only the packet filtering features but also minimized and managed the performance overhead. The basic functionality of the FLOWMON firewall was inspired by the generic firewall rule <condition, action>. This implies that if a particular packet matches the condition, it performs a specific action on the data packet as described. The condition rule according to the theory is defined by 5-tuple format. The condition includes source IP, source port, destination IP, destination port and finally, the protocol. Furthermore, the action format was inspired by the basic firewall actions of accepting or dropping the packet. Moreover, the firewall also tackled the problem of policy or rule violation. The conditions in which the packet has already matched with one of the firewall policies and matches the condition on of another firewall rule. This condition indicates an overlapping problem which was resolved by the firewall through first-match resolution concluding that if a firewall has already matched a rule, the action depends on the first met rule. Furthermore, their firewall was designed to implement various functionalities like Violation detection, flow path classification, and flow path space analysis. The firewall design demonstrates efficient detection of the firewall space violation and provide real time detection. The major drawback of the study was the requirement of the sophisticated and complex network firewall design and configurations. The model was effective and sustainable but required complicated design algorithms to define the firewall policies.

(IKARASHI et al., 2018) proposed an idea in creating a proactive SDN firewall system to collaborate with the DNS (Domain name resolution server). The aim of the firewall is to detect the sender of the traffic. The detection of the sender focuses on gathering enough information about the sender to analyse the legitimate user from a potential attacker. This

proposed approach focussed on storing the information in the SDN controller's database. The information gathered by DNS is stored in the SDN controller creating a whitelist and a blacklist depending upon the predefined inputs classifying a communication as an attack to cause potential harm. DNS gather information from the nodes and other devices while providing them an IP address. This information includes the address of the user along with the type of packets usually initiated by the sender. The two lists stored in the database works as the base for separating legitimate users from attackers. In the presented scenario, when a communication stream is initiated from a source to the destination network; all the data packets are passed through a system matching the information from the ones stored in the SDN's system. If the information of the packets travelling through the channel matches to the information stored in the whitelist the packets are accepted, otherwise the packets are dropped. This criterion supports in identifying attacks like ICMP flood attack (DDoS). The performance of the proposed firewall is compared to the performance of the traditional firewall and the results conclude that the proposed SDN firewall with the support from Domain Name Resolution proves to be more efficient in identifying and tackling Denial of Service attacks. Despite of the success as an intrusion detection system, the project has a plethora of unaddressed challenges. These challenges are majorly related to the performance overhead and complications in designing a common platform to match the information with the existing information captured and stored by DNS. The list of drawbacks also witnessed deteriorated performance of the designed system in case of some specific DDoS attacks.

Another study conducted on the similar theory is a method that identifies and mitigate flood attacks initiated using the SYN packets of the TCP communication. (Nugraha et al., 2018) The method used a collector called a S-flow collector. The role of the collector is to collect all the packets at the SDN controller switch. The packet rate is compared with the assigned threshold and if the rate exceeds the desired threshold the packets are dropped. This is achieved by using the flow table. Once the condition is met, a flow table entry is initiated which eventually blocks the packet. The theory concluded positive results, however, indicated a massive challenge in the maintenance of the system. The administrator of the system is expected to manage the characteristics of all the OpenFlow switches based on individual attacks. In addition to the system being verbose and exhausting it also demonstrated limited applications to the applicable target.

The extensive research conducted by renown researchers, students and administrator is elongated through the paper presented by (Yoon et al. , 2022) In their system, they tried to implement four different types of functionalities implemented by Software Defined Network. These features included using SDN as firewall and Intrusion Detection Systems, Intrusion detection systems, scan and DDoS detector. In the firewall functionality of the system, the packets are checked by the firewall created and a flow entry is created in the table. The administrator is required to maintain the security policies. The SDN controller is responsible for checking all the security violation. This indicates that a bottleneck can be caused at the controller as the controller is responsible for checking and maintaining the policies. The performance overhead of the controller is the main reason behind the decrease in the throughput and performance of the network.
Many other researchers used deep learning for the development of the sustainable SDN security systems. Using machine learning concepts for developing intelligent intrusion detection and prevention systems is innovative however, falls outside the scope of this research.

Furthermore, inspired by a plethora of existing literature in the area of Software Defined Networking, the purpose of this study is developing a centralized security system using SDN. This paper explores the opportunity of using a SDN switch as a firewall. The firewall rules are designed and analysed using three different approaches. Primarily, the firewall rules are created in the SDN switch without using a controller. This approach reduces the network performance overhead involved in the system and aims to improve the throughput and overall performance of the network. The second approach is designed to implement the firewall rules on layer 2. Finally, the rules are implemented on layer 3 using a SDN controller. The three different approaches are used to draw a conclusion on network performance and the reliability of the threat detection and prevention system. The proposed research can provide a base for a development of a faster and more reliable security system using SDN. Furthermore, the study focuses on creating a scalable security solution. This implies that if any new systems are added to the network, the security system covers the changed topology as well.

4. CONSIDERATIONS

There are many ethical and legal considerations that must be accounted for while conducting research. This section of the report aims to define the ethical and legal considerations involved in this project. The project does not involve human participation/ Intervention; hence, it does not require an ethical approval. SDN pose several vulnerabilities towards DDOS attack, making it easy for an attacker to breach the security of the network using SDN (Ubale & Jain, 2020)
Therefore, this project can also be used by the attackers to test a variety of penetration techniques and exploits to deteriorate the security of the network. The malicious use of the project is a massive ethical implication of this project. “Furthermore, SDN can also be used to infect any network and, eventually firmware, with a malware. It also exposes the network to man in the middle attacks.” (Krishnan, Prabhakar & Najeem, J.S, 2019)
This opens doors to possibility of harm or damage to the network in a variety of possible ways. The code for this project tries to use the switch with basic packet filtering functionality. Thus, the attackers can also manipulate the code for malicious intent. There is plethora of considerations involved in working with SDN which will be described in the Final project report.

The project has critical social implications. The Society on social implications of Technology (SSIT)critically evaluates the impact of technology in advancing the society. (Queenland, 2020). The society focuses on the impact caused by the existing and new technology on the society. It evaluates the positive and negative implications to provide a detailed analysis of any technology’s social considerations. This projected complies with the objective of SSIT. Hence, the project mostly demonstrates positive impact on the society. As mentioned earlier, the idea can fall prey to the illegitimate users. Therefore, the former can cause it to be adversely impact the society. The extent of positive impact can be considered more than the negative implications. Hence, it can be concluded that the project will significantly contribute to produce results in the best interest of the society.

4.1 ETHICAL CONSIDERATIONS

Ethical considerations have recently become a fundamental aspect of computer science. Technology ethics can be summed up as the set of moral principles or ideals that govern the system. A balance between the right to information, privacy, dependency, etc., may be

necessary to uphold these moral goals. Numerous studies have attempted to conceptualise a framework to better identify the ethical difficulties due to the scope of the subject. For instance, (Norberto Patrignani, 2018) tried to develop a framework to describe and address the influence of computers on various societal segments. This demonstrates how crucial moral issues are to the project.

Like this, several authors have attempted to particularly address the ethical issues associated with the use of human subjects in CS research (Computer Science). To prevent any violations, human rights considerations are considered when researching technology that involves people (Buchanan et al., 2011). As a result, different levels of ethical approval may be needed for every project or research that involves human participants. There was no need for human involvement in the project that was proposed. As a result, obtaining university ethical approval was no longer necessary.

After resolving the University's ethical obligations, the initiative raises questions about privacy and intellectual property rights. Illegal users can test the effectiveness of the attack using the firewall emulator. This gives the user the opportunity to identify the attack tools' shortcomings and make the required adjustments to strengthen the attack.

In a similar manner, it also puts intellectual property rights at danger. As the code, if made available, can entice individuals to copy or modify the code for malevolent purposes, resulting in the violation of rights. The Computer Misuse Act of 1990 (CMA), which stipulates that unauthorised access to a system or piece of software with the intent to do harm constitutes a violation, may be broken as a result of falling victim to an unauthorised user.

4.2 LEGAL CONSIDERATIONS

The legal risks that technology poses are made clear by all the bad content, attacks, and tools that are readily accessible to users. Many authors agreed that technology poses a growing threat to security. For instance, the harm that a hostile user could cause to the defence systems, electrical grid, robbery, etc. Due to these, it is necessary to create a legal framework for the effective use of technology (Hathaway et al., 2012).

This project may pose a risk of gathering user information, which is one of the criteria outlined in the Data Protection Act (DPA). Despite this, several laws have been enacted in the subject of cybersecurity. However, a wide range of issues remain unresolved, necessitating the ongoing revision of rules. Some of the legal problems raised by this initiative are also addressed under the Computer Fraud and Abuse Act (CFFAA 1984). The legislation makes it a federal violation to utilise accessible technology for fraud and harm (Veale and Brown, 2020). As a result, if the tool is used to create immoral rules or tools that cause harm to the system, it may be regarded illegal and subject to legal consequences. To reduce legal risk, software distribution can be licenced and monitored, allowing users to be tracked and reducing unauthorised use.

4.3 SOCIETAL CONSIDERATIONS

The social component focuses more on the effects on society than it does on the danger presented by an individual. Numerous studies have provided evidence of the increased technological impact on societal structure. This is a result of an increase in the number of

users. As predicted by plenty of authors and the pioneers of technology, the professional sector will be greatly impacted by the implications of these technological advancements.

Due to the widespread social adoption of technology, both beneficial and bad effects have been mixed onto society. The initiative will have an influence on society in both positive and negative ways. The ability for many users to assess the robustness and correctness of their firewall rules/chains is one of its good effects. Additionally, it offers a study environment devoid of infrastructure, making firewall implementation easier for the user. The society is seriously threatened by several policy mistakes in firewall setting. The society may suffer as a result of these policy mistakes, which may be utilised to forcibly restrict some communications.

Additionally, a firewall policy mistake may result in a security breach by leaving openings that online threats might use to their advantage. Additionally, it gives hackers a chance to take advantage of their weaknesses. To reduce the firewall's negative societal impact, a careful approach should be adopted while creating the firewall regulations. Like this, in the project, the user is needed to develop the rules following a critical assessment of the rule and its implications for the firewall policies in order to lessen the adverse social impact.

5. DESIGN AND IMPLEMENTATION

5.1 DESIGN

The main aim of the project is to create a switch as a firewall using a SDN controller. The switch is designed to operate as a firewall system to examine the operability of the switch as a firewall. The design is implemented keeping the research question in mind. Moreover, the main objective of the switch is to work as a centralized firewall which means that each switch in the topology is designed to operate as a firewall providing centralized and efficient security. Another factor considered while designing the SDN system is to ensure the addition of the scalability feature. Therefore, the system should be capable to add more workstations and nodes to aid the scalability. The project design is aimed to answer the below mentioned initial questions:

- What network topology to be considered whole designing the experiment?
- Which SDN controller should be used in the project considering the system requirements?
- How can the performance of the network be measured?

The design and implement section of the reports highlights the design created to address the questions mentioned above. The section also calls out the challenges encountered while designing the proposed system. Additionally, it encapsulates the technologies used to overcome the design challenges efficiently. This part of the report also covers the different features of firewall accomplished using available technology.

5.2 REQUIREMENTS

In order to create an efficient switch as a firewall, the system demands for the following:

1. A network topology created virtually
2. Virtual environment to push the configurations and create a network

3. Python script consisting of the network configurations
4. Mininet environment to virtually implement the network
5. Wireshark to measure the performance of the network in case of different scenarios

The project was accomplished in a phased approach where the objectives of the project was further classed as sub objectives or checkpoints. These checkpoints were also used to define the progress of the project. With the completion of every checkpoint one target part of the project was completed. The project plan was created in the beginning while describing the detailed project plan. A specific time frame was allocated to each task depending upon the length, complexity and research required to accomplish the tasks. The project lifecycle witnessed plenty of challenges. Some challenges were resolved through the progress in the plan. However, other challenges caused hindrance in making the system better and competitive as compared to the modern technology. The project plan also listed the technologies required for the successful accomplishment of the project goals. The tools required are listed below:

5.3 ESSENTIAL TOOLS USED

1. Virtualization platform: Recently, virtualization has changed the entire outlook of the technology. Virtualization refers to the ability of the system to split and share the resources among OS. The concept of virtualization was introduced by IBM in 1964 and was called CP/CMS system (Rodríguez-Haro et al., 2012). Virtualization allows the user to create networks and systems within the system. It is extremely advantageous for developers and testers. There are a variety of available platforms providing virtualization. The analysts use these platforms to test different conditions and rule sets majorly because the virtual systems and platform provide them with the flexibility to make errors to enhance the performance and operability of the system. If the errors cases damage the system, the users can simply start again without getting to fix the main operating system. Different virtualization software is available in the market depending upon the type of processor chip and the base operating system, for example, VMware, parallels, Virtual box, UTM etc. The purpose of all these platforms is to ensure that a user can create and launch another operating system without disrupting the configuration and settings of the current base operating system. For the purpose of this project, UTM is used. UTM is a virtualization software which can be used in latest MacBooks running on M1 M2 chip processors. Post the installation of the virtualization software, an ISO file containing the intended operating system needs to be downloaded and launched on the virtual platform.
2. Ubuntu ISO file: As mentioned earlier, the virtualization software needs an ISO file including the operating system to provide the virtual platform to the user. For this project, Mininet is required to create a virtual network inside the system. Mininet is an emulator which can deploy an entire network on a single computer (Kaur et al., 2022). Main purpose behind development of Mininet emulator is to provide the researchers and testers with a platform to carry out the research in SDN and OpenFlow. The major advantage of Mininet is the ability to use it in the absence of the heightened infrastructure and low cost. Furthermore, Mininet allows the processing of the unmodified code in the system. All these advantages make Mininet an easy-to-use emulator to design a virtual network. For the purpose of this project, Mininet is used in the virtual Ubuntu machine to implement the desired topology and eventually create the switch in the topology as the firewall

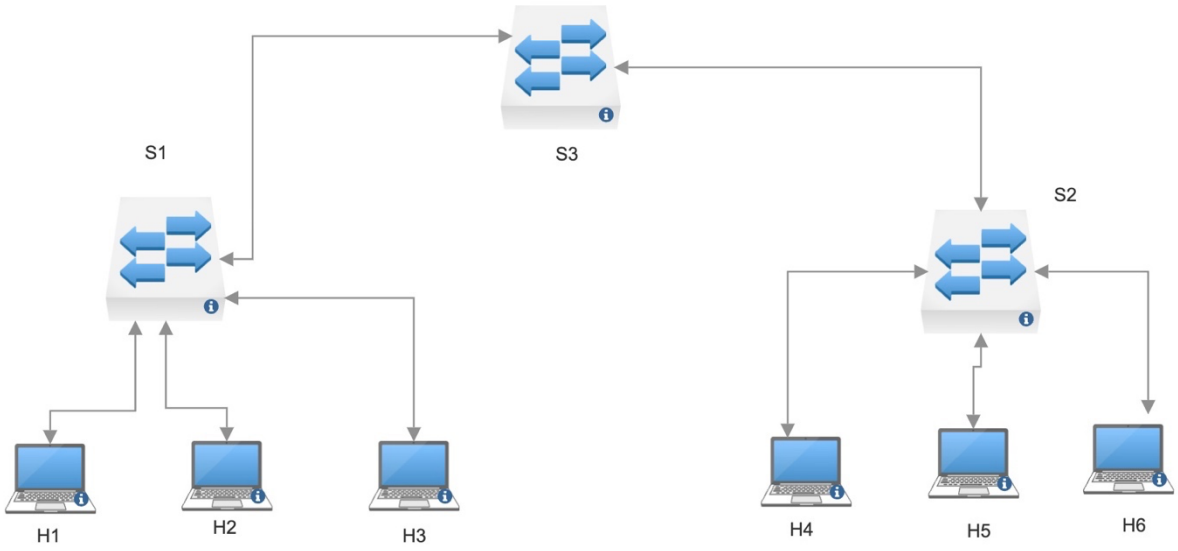
3. Python file: A python file is required to create the desired topology. The python file created includes the code for the topology which includes the network devices, links between the devices and finally, the Ip addresses assigned to all the network devices. The created topology is tested by using switch as a firewall with and without the controller for testing and analysis.
4. Wireshark: Wireshark is a packet analyser for the network (Wireshark, 2012). It captures and analyses the data packets travelling through a network. The role of a Wireshark tool is to gather as much information about the data packets as possible. The common uses of a Wireshark packet analyser are examining and troubleshooting network problems, verify network applications and finally, learn the operability of the network communication protocols. Wireshark is widely used because of its enormous features and usability. For this project, Wireshark is used to analyse the network packets travelling through the designed topology. The data rate of packets is analysed and calculated to understand the performance of the designed network in different scenarios.

6. DESIGN IMPLEMENTATION & RESULTS

Post the successful installation of the virtual machine tool (UTM), Ubuntu ISO and Mininet the python code is created for the desired topology. (Appendix 1) The topology consists of 6 hosts (h1, h2, h3, h4, h5, h6) and three switches (s1, s2, s3). The connections between the hosts and switches are as follows:

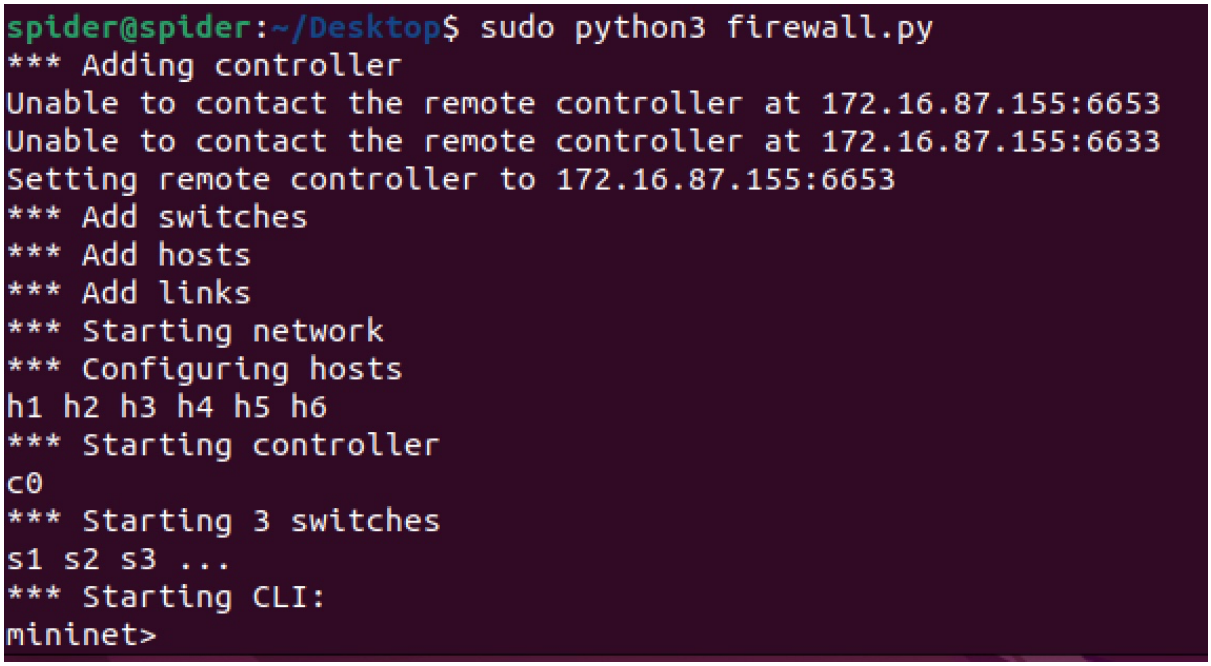
- h1 <-> s1
- h2 <-> s1
- h3 <-> s1
- s1 <-> s3
- s2 <-> s3
- h4 <-> s2
- h5 <-> s2
- h6 <-> s2

The created topology is demonstrated by the following network diagram:

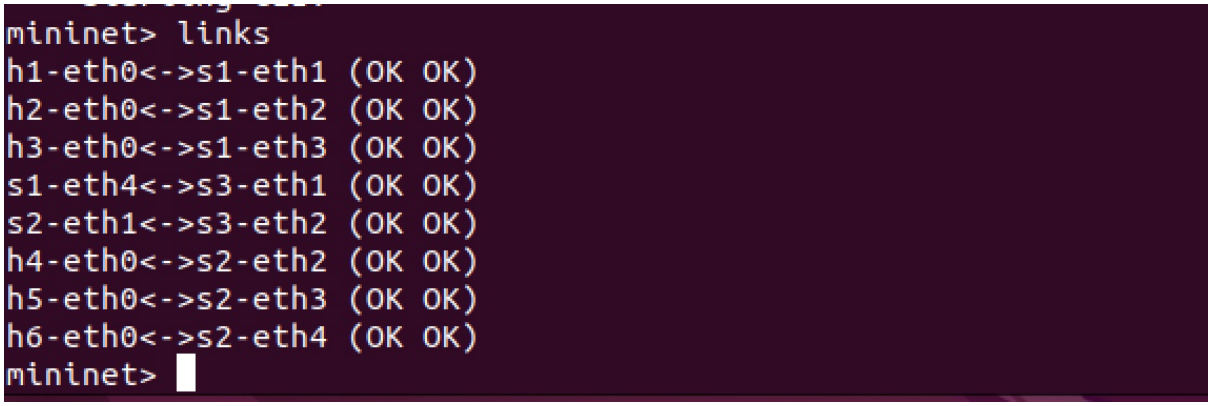


The network topology has been kept simple to understand the working of the switches as the firewall. All the switches are designed to behave as firewalls which means that when the right set of rules are defined, the network is provided with utmost security owing to the use of centralized firewall system. The major advantages of using all the three switches as firewall systems is scalability which means that if one of the switches fail, it does not leave the entire system vulnerable, but the other switch keeps providing the firewall functionality and eventually provides persistent protection to the network.

The python file is created in the Ubuntu environment using an emulator software and the network is formulated inside the virtual machine. The network post running the file is embodied by the image below.



Post the successful implementation of the python file, the created network is tested to understand if the system is operational and ensure connectivity. To check the connections a basic Mininet command called “links” is used.



The output presented above confirms the links and the topology as depicted in the network diagram.

Finally, the connectivity is tested using the pingall command. This command is used to test the connectivity of all the hosts and switches connected in the network. The results demonstrated that the hosts could connect with each other. The ICMP stream is initiated automatically between all the hosts which means that all the 6 hosts try to communicate with each other and the results evidence that all the 6 hosts cannot communicate with each other.

```
*** Ping: testing ping reachability
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X X
h4 -> X X X X X
h5 -> X X X X X
h6 -> X X X X X
*** Results: 100% dropped (0/30 received)
```

As displayed above, if no entry is specified in the flow table, the switch does not let the packets pass through. Therefore, the further steps revolve around specifying the entry in the flow tables individually for the switches which provides a criterion to the switch to decide whether to accept the packets or not.

After the successful implementation of the basic virtual network, the next step is to test the firewall functionality by launching the firewall without the controller first. The command used to initiate that is as follows:

```
mininet>sh ovs-ofctl add-flow s1 action=normal
mininet>sh ova-ofctl add-flow s2 action=normal
```

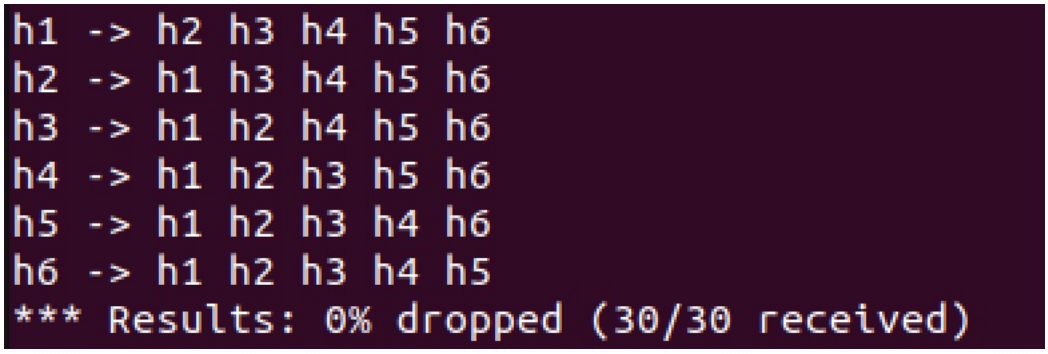
These commands make an entry to the flow table allowing the packets to flow between all the hosts connected to switch 1 and switch 2. Without making the entry to the flow table to behave normally with the packets sent from one host to the other, the network by default drops all the packets traversing through the network. This implies that without defining the flow entry, the packets communicating between all the hosts will be dropped as displayed below:

```
h1 -> X X X X X
h2 -> X X X X X
h3 -> X X X X X
h4 -> X X X X X
h5 -> X X X X X
h6 -> X X X X X
```

Post implementing the firewall using the commands mentioned above the hosts connected to s1 can communicate with each other. Similarly, the hosts connected to s2 can communicate between each other. However, as no rules are defined for s3 therefore, the hosts connected to

s1 are not able to communicate with the hosts connected to s2 as the only link between s1 and s2 is through s3.

The results demonstrate that the hosts of s1 are still not able to communicate with the hosts of s2. Finally, to initiate the communication between the hosts of s1 and hosts of s2 the flow table entry is made on s3. The results comnfirm that after making the flow entry to the third all the hosts can connect with each other.

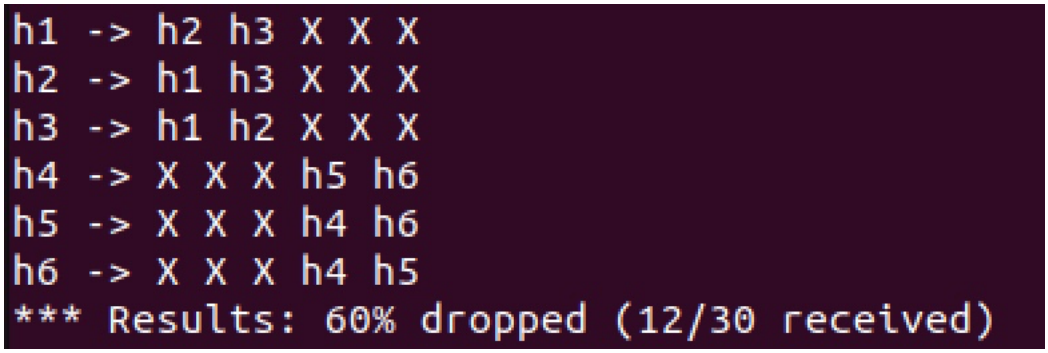


Therefore, it can be confirmed that a basic packet filtering ability can be initiated in the system without using any controller. However, it confirms that only the basic functionality can be initiated between the hosts by creating a firewall. Furthermore, this packet filtering feature can only benefit the hosts in different subnets with plenty of restrictions. The restrictions specify that if the system is not suitable if it is desired to allow packets from only one host to the other from different subnets. The functionality either blocks all the packets or allow all of them. This indicates that implementing the firewall functionality on the switches in a SDN is only feasible for small scale simple networks. The further results section of the report highlights the performance of the network while using a simple firewall network. Despite of good performance, the functionality does not suffice the need of implementing a firewall for the packets initiated from a specific host.

To overcome the drawbacks of the switch firewall without the controller, two approaches are used and discussed. In the first approach, the SDN OpenFlow controller is used to define the instructions to accept or drop the packets on layer 2 and layer 3. Both the approaches provide specific firewall functionality to the switches.

6.1 PACKET FILTERING ON LAYER 2 AND LAYER 3

Filtering the packets on layer 2 and layer 3, provides flexibility in deciding the fate of the packets coming from a specific source directed towards a particular destination. The rules on layer 2 and 3 requires more parameters to be passed on to the system. These parameters include the source of the packet, packet's destination and the actions to be taken on the packet. Therefore, in this scenario, the switch matches the condition of the packets travelling through the system with the flow entry made in individual switches. If the packet matches a condition, the switch checks for the corresponding action from the list and decides the fate of the packet. The previous section allowed the transmission of all the packets by entering the flow rules to all the switches to behave normally i.e., to accept the packets. Post the successful implementation of the rules on layer 2, the pingall command is initiated again to understand the operations performed by the switch as a firewall when the layer 2 rules are implemented blocking specific packets.



The results reflect that when the flows are added to s1, s2 and s3 all the packets are transmitted in the network between the hosts through all the switches. However, upon adding the rules to layer 2 and layer 3, the packets travelling from hosts connected to switch 1 to hosts connected to switch 2 or vice-versa are dropped at switch 3. The rule set used for the project is aimed to filter the communication packets between switch 1 and switch 2 leading to a private communication channel. The results confirm that the hosts are still able to communicate with each other if connected to the same switch but when they must pass through switch 3 the packets are dropped. The rules configured on layer 2 and layer 3 are depicted below.

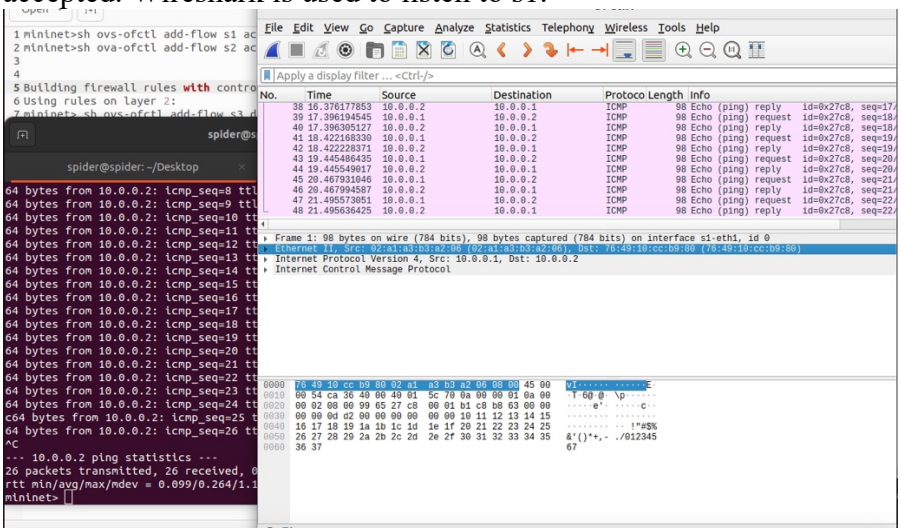
```
Using rules on layer 2:
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:04,actions=drop
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:05,actions=drop
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:01,dl_dst=00:00:00:00:00:06,actions=drop
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:04,actions=drop
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:05,actions=drop
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:02,dl_dst=00:00:00:00:00:06,actions=drop
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:04,actions=drop
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:05,actions=drop
mininet> sh ovs-ofctl add-flow s3 dl_src=00:00:00:00:00:03,dl_dst=00:00:00:00:00:06,actions=drop

Using rules on layer 3

mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.1,nw_dst=10.0.0.4,actions=drop
mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.1,nw_dst=10.0.0.5,actions=drop
mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.1,nw_dst=10.0.0.6,actions=drop
mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.2,nw_dst=10.0.0.4,actions=drop
mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.2,nw_dst=10.0.0.5,actions=drop
mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.2,nw_dst=10.0.0.6,actions=drop
mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.3,nw_dst=10.0.0.4,actions=drop
mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.3,nw_dst=10.0.0.5,actions=drop
mininet> sh ovs-ofctl add-flow s3 priority=500,ip,nw_src=10.0.0.3,nw_dst=10.0.0.6,actions=drop
```

6.2 PERFORMANCE ANALYSIS

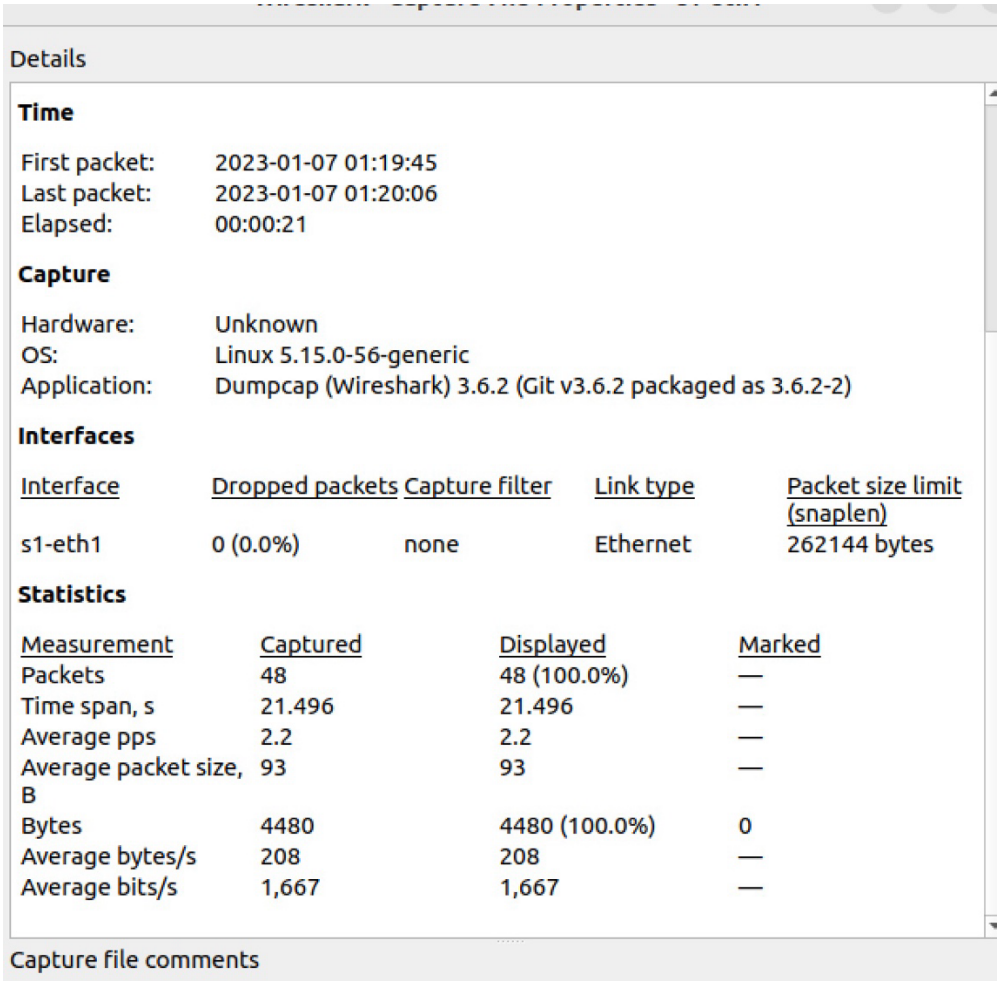
To understand the performance of the network with firewall, Wireshark is used to capture the packets and analyse the packet transfer rate to measure the performance of the network. Wireshark analyses all the packets passing through the system and gather information like source, destination, protocol used, packet type etc. Additionally, Wireshark also analyses the network performance by checking the number of packets being transferred per second. This can be achieved by measuring the length of the communication. The following steps discuss the performance of the network when using the switch as the firewall.

1. Initiating an ICMP stream between the two hosts on the same switch. In the first experiment, an ICMP (ping) stream is initiated between h1 and h2. Both h1 and h2 are connected through the same switch and the firewall rules are allowing the communication between the two to be approved. Which means that the packets

travelling through s1 from one host to another should behave normally and should be accepted. Wireshark is used to listen to s1.

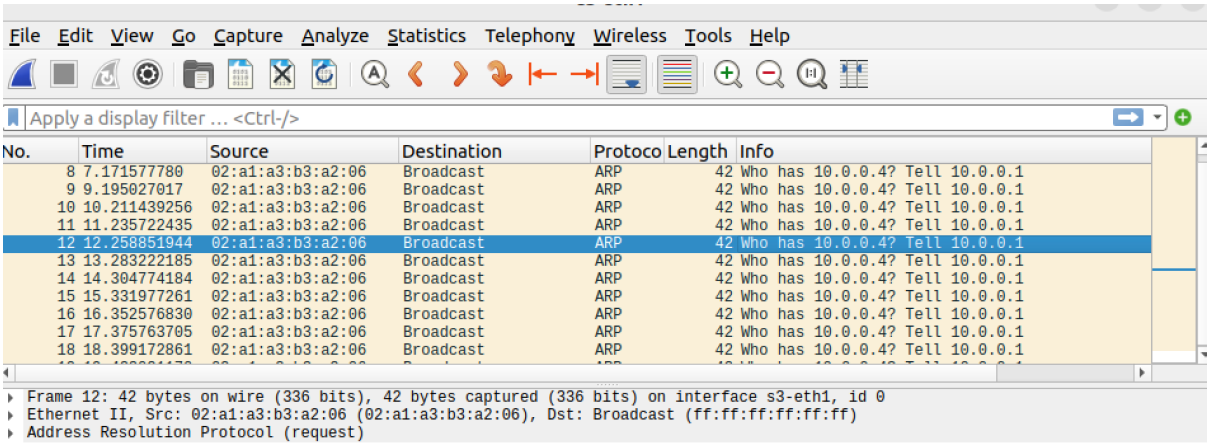


Additionally, the capture statistics are gathered



The results indicate that 48 packets are transferred in 21.4 s. This implies that an average of 2.2 packets are received every second at the destination.

2. Initiating an ICMP stream between hosts from different subnets. H1 is connected to switch 1 and host 5 is connected to switch 2. The tests are conducted after implementing the firewall rules on layer 2 and 3. These rules define the packets to be dropped if the hosts of s1 are trying to communicate with the hosts of s2



As evidenced by the Wireshark capture file, the switch is only initiating the ARP packets which are the address resolution packets, used by the access points (routers or switches) to identify the IP address of the hosts. The switch sends the ARP packets to the network to identify the host. Post identification of the host, the switch analyses the packets transmitted by comparing it with the condition mentioned in the flow table. As the condition is met which was defined while configuring the ruleset, the packets are dropped and the communication between the 2 is not successful. Thus, it confirms the effective implementation of the firewall ruleset.

Similar tests are conducted on all the rules defined for layer 2 and layer 3 and the results evidence that the packets follow the rules efficiently and provide an infrastructure less, state of the art firewall system using Switch as a firewall.

7. PRESENT USE AND FUTURE WORK

The implementation of switch as a firewall can be used by students, researchers and security analysts for a variety of uses. As the model runs on a very little infrastructure, therefore it can be used by students to work on various projects and conduct analysis of the network. It is also used by many researchers to understand the behaviour of different hosts and switches under a confined environment to analyse the efficiency of the security system. The beneficial impact of the model where the user does not have to invest resources and capital in designing the system opens doors to a lot of potential future work to be carried out in the field of technology or cyber security.

SDN has provided opportunities to the modern cyber security to design extensive and secure intrusion detection and prevention systems. Similarly, this concept can also be used with some modifications to work as an efficient IDS and IPS. As the designed firewall can match the condition of the packets with or without the controller therefore, an addition of using a log file to capture and save the data in the system can provide the functionality of the intrusion detection system. The purpose of the intrusion detection system is to alert the user when the attack is initiating. When the packets are transmitted from the source to the destination the packet checker validates the packets against the set of rules defined by the user. When the packet matches the condition a log entry can be made which can alert the user of the potential attack. A similar concept can be used in intrusion prevention system where when a warning or alert is issued, the system can take necessary actions (pre-defined) and produce an effective prevention system.

Furthermore, some more additions can be made on the proposed system to enhance the firewall functionality. Currently the system is designed to work as a basic packet filtering firewall using the generic information like IP address, protocols etc. The future work can focus on implementing the stateful inspection feature. Instead on just relying on the generic information, the system can be designed to inspect the state of the packets instead of just the IP address. This enables the firewall system to track the packets basis the parameters like length of the packets, number of packets, etc. producing a more reliable firewall system. Adding to the list of potential research opportunity is improving the security policies and configurations. The firewall rules and policies misconfiguration can lead to disastrous result of system being unstable or leading the attacker to gather information which can cause harm to the user. Therefore, improving the firewall rule configurations for the system can help improve the accuracy of the system and minimize false alarms.

The study can also be used to develop intelligent systems capable to learn from the patterns and frequency of data packets. Deep learning concepts can be integrated to train the system to achieve efficiency, reliability and least user involvement. The system can continuously learn from testing different set of rules and aim towards making a fail proof security system.

8. PROJECT PLAN AND REFLECTION

The project is wrapped up in the chapter below, which focuses on the tool's assessment. Project management includes reflection as a key component. Schon proposed that academic project management reflection practise helps students transfer the expertise to the workplace. (Schön, 1983) Numerous ideas and studies have shown that reflection is frequently left out of projects due to a lack of time (Anbari, Carayannis and Voetsch, 2008). However, writers have carried out research studies and surveys to emphasise the significance of reflection in the actual environment. The students are prepared to handle the real-world issues they face at work via reflection on their studies and projects (Flynn and Levie, 2021).

The initial step in the project life cycle was the background research. Creating a practical project requires extensive research throughout, which is evident by the increasing reference list. Most of the background research carried out until now is around the research in SDN to understand the working of the software defined network. The research also enabled to figure out the use of an open-source controller to push the python script to the Mininet environment.

Through the research the project was exposed to the opportunity to use different controllers like OpenFlow, onos, etc. to identify the one best suitable for the project.

The second step was to design the python script to be able to implement a Mininet network using SDN. The initial python file is aimed at creating a virtual network in Mininet with the default configurations including 6 hosts and 2 switches. The presented topology aims to demonstrate a small-scale network. The future work in the project will aim to extend the python file to configure one switch as the packet filtering firewall.

The result analysis is the next step of the project plan. This section will analyse the behaviour of the switch when the hosts are trying to communicate with each other. The configured switch can work as a barrier between the hosts and the switch will make the decision whether to forward the packet or drop it basis the rule set. The section will also aim to analyse the performance of the network to understand the difference in overall network operability and Quality of service

The final step is to write a report collaborating all the results from the project. The result is an amalgamation of the security efficiency of the programmable switch and the overall performance of the network. To analyse the overall performance of the system it is imperative to use the right set of tools to analyse the rate of the packet transfer. With the introduction of additional configurations, the network has more chances to deliver a negatively impacted Quality of Service. Therefore, to test the stability and operability of the network, a packet capture tool is used, and the reports generated are then studied.

9. CONCLUSION

The research project concludes in stating that a switch can be used as a firewall system through SDN. SDN can provide a centralised firewall functionality to all the switches to provide more security than a conventional firewall system. It also highlights that using SDN to implement a firewall does not impact the performance of the system drastically. The project faced a variety of challenges during the life cycle. One of the major challenges was achieving the results while using combinations of different rule set in which some of the rules accept and others drop the packets. These combinations can cause confliction in deciding the fate of the packets. Many challenges faced during the project can be resolved by using a hardware firewall. Although, virtual environment can provide a real-life testing scenario. However, the absence of hardware systems can cause some sophisticated features to respond against the desirable result. The report also answers the research questions presented. The result confirms that SDN can be used to enhance the security of a network. SDN also provides solution to modern day networking challenges like flexibility and reliability. This means that the security system can continue to perform efficiently even if the new hosts are added to the network. Also, the security system designed using SDN is completely flexible as it gives the user an opportunity to program different set of rules and conditions for the firewall system. SDN is the future of network security due to its programmable features and flexibility. A vast majority of organisations and security analysts are already using the system to improve the strength of the security and providing efficient protection against the emerging security threats.