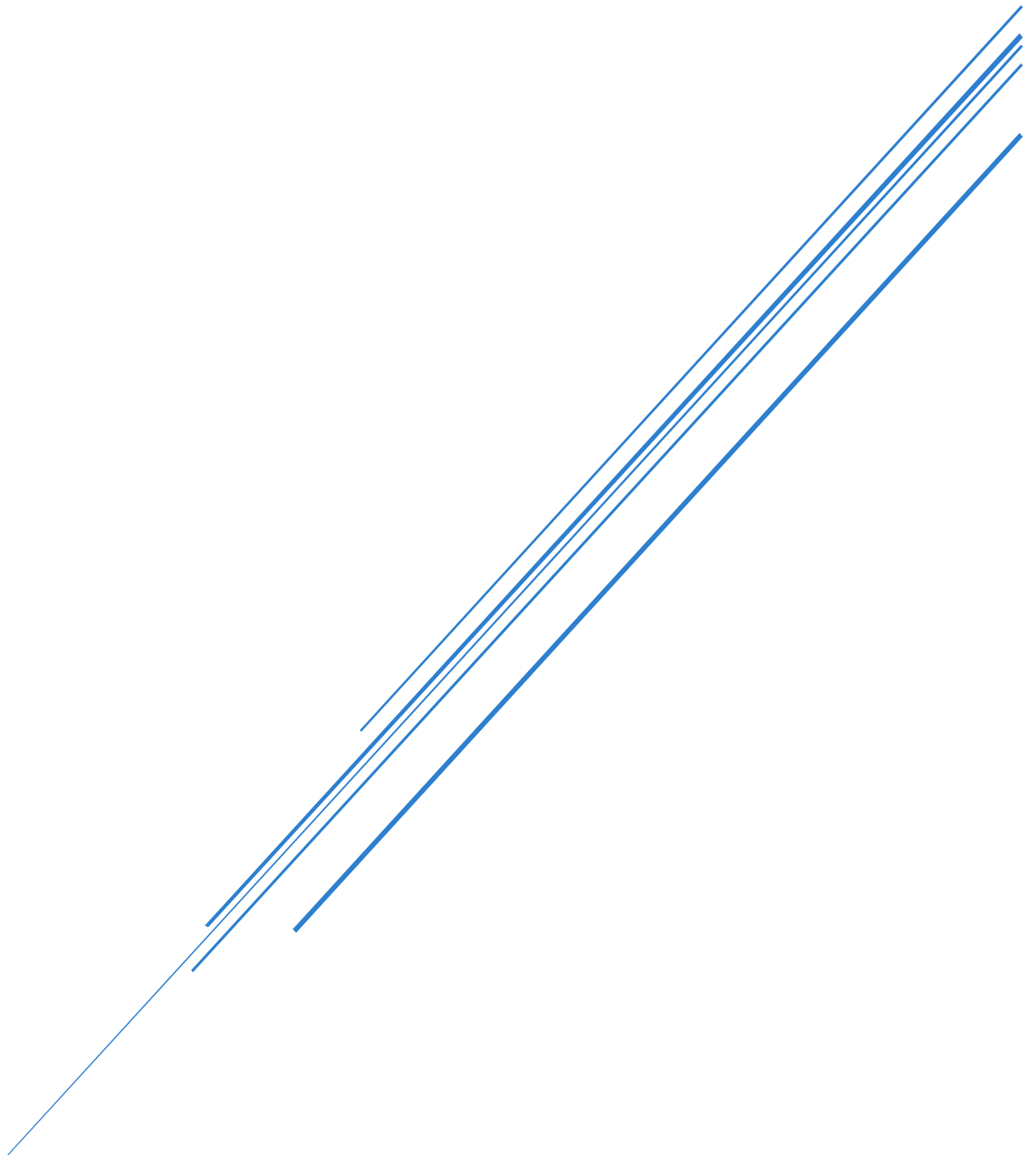


ACCA6020 - PROJECT PROPOSAL

Network Security and Risk Mitigation in Software Defined
Networks



Che Morris - 2203458
Cybersecurity & Networks

Contents

Introduction	2
Literature Review	3
Overview.....	3
Key Developments & Trends within SDN	3
Security Concerns within SDN.....	4
Research Gap	4
Research Problem Statement	5
Research Methodology & Materials	5
Project Plan	6
Month 1: Planning, Extended Literature Review & Research Design	6
Month 2: Testbed Setup & Initial Vulnerability Testing	6
Month 3: Development and Implementation of Security Solutions	6
Month 4: Final Testing, Analysis & Documenting	7
References	8

Introduction

The introduction and rapid adoption of Software Defined Networks (also referred to as SDNs) has made a substantial difference to how industries run and control their infrastructure. Since SDNs decouple the control and data planes, they are often used these days as it allows for a reduction in running and maintenance costs, central control of the network and improved flexibility [1]. The benefits seem great, but they do bring their own problems, these being mainly distributed denial-of-service attacks (DDoS), issues with it being centralized meaning if one of the controllers becomes compromised the entire network can and malicious applications being introduced to the controllers [2].

This project, titled “Advanced Network Security and Risk Mitigation in Software Defined Networks”, aims to investigate and address the vulnerabilities within SDNs and propose solutions to mitigate the risks and vulnerabilities posed. The motivation behind completing this research is due to the growing reliance upon SDNs throughout industries; including but not limited to, cloud computing, data centres and telecommunications where network security is extremely important. The importance is backed up by the significant rise in both data breaches and DDoS attacks targeting critical services, the latter being a significant weakness of SDNs [3].

Throughout this investigation, I hope to suggest and plan out potential security solutions that can help better protect SDNs from their weaknesses and better protect them from potential threats. By attempting to propose these mitigation strategies, hopefully the project will help contribute to the development of better SDN security and enhance the security of network infrastructure.

Literature Review

Overview

Software Defined Networking (SDN) are a relatively new approach (first began to come to light in 2007) to networks that separates the control plane from the data plane, allowing for centralized control of the network [2]. The decoupling of the control and data planes allow for network administrators to dynamically manage traffic, optimize the network and its resources, implement changes on the fly and they provide better flexibility when compared to traditional networks that rely solely on hardware. SDN has been adopted across multiple industries in recent years especially within big hitters like cloud computing and data centres due to the scalability and efficiency. SDN can also support other means of networking such as network automation and is closely related with network function virtualization (NFV) [4].

Key Developments & Trends within SDN

As mentioned previously, the separation of the control and data planes are a key part of how SDN works. Compared to traditional networks where control is distributed throughout multiple devices and where each device makes its own decisions, SDN changes this as control is left to one SDN controller which can control and instruct each part of the network (the routers and the switches) on how to handle data and process it simultaneously [5].

The programmability and centralization are one of the core benefits of SDN. It allows for network administrators to send instructions out to the network simultaneously and can allow for them to easily adjust network behaviour depending on demand and adapt the network if needed in real time. This is due to the centralization of SDN, allowing for network administrators to easily change the network as needed through one piece of software, allowing for improved control and efficiency over all [6].

OpenFlow is a protocol that is widely used throughout networks relying on SDN, allowing for communication between the central SDN controller and the network devices connected to it, it allows for the controller to make decisions on the data of the switches and routers and decide how packets get routed throughout the network [5]. There are alternatives to OpenFlow, one of these being Open Network Operating System (ONOS) which is open source. Compared to OpenFlow, it isn't just a protocol that is used but is a whole operating system and is mainly designed for more large-scale networks [7].

Another development within SDN has been Network Function Virtualization (NFV) which has become closely related and almost complementary to SDN. As SDN decouples the control and data planes, NFV virtualizes network services such as firewalls and intrusion detection systems which can suit the flexibility required within cloud computing for example [4].

Security Concerns within SDN

The advantages of SDN are extremely significant and can be extremely useful for network administrators who decide to use it; however, due to the centralization of SDN it can be quite an attractive target for attackers. An attack on the central controller can be devastating, allowing attackers complete control over the network which can lead to outages or data breaches [3]. We already know a distributed denial-of-service attack on the central controller can have a huge effect whilst utilizing SDN and can bring the entire network to its knees but the programmability side of SDN can also pose risks as attackers could attempt to exploit vulnerabilities in the central controller or the applications running on top of it as demonstrated by S. Shin and G. Gu [8].

Research Gap

The rise of SDN across the last decade or so has led to an increase in write ups and research based on it, but there is a significant gap in research addressing the security issues and risks that come with it. As go for most networks throughout history, a lot of the security measures developed for SDN are static (things like encryption and access controls) or are reactive (responding after an attack has already happened). As S. Shin and G. Gu mention in their paper, the traditional security methods used by networks are whilst still useful, not as sufficient in the ever-evolving day where attacks can happen remotely [8].

In the current day, new technologies like machine learning and AI have come along which could become useful in protecting SDN. Machine learning could be used within SDN to attempt to detect anomalies within network traffic and gain experience based on past threats to help predict and mitigate potential attacks based on said data [9]. There has also been discussion of utilizing blockchain techniques by Sharma et al [10] which involves applying blockchain technology to SDN to allow real time detection of potential attacks and security threats which could be a big help in protecting SDN networks.

Research Problem Statement

This project aims to address the security vulnerabilities that exist within Software Defined Networking (SDN), mostly focusing on the risk posed by how SDN has a centralized controller. As cyber attacks become more advanced and more specialised for this type of networking, current security techniques can become insufficient as they often don't deal with a threat in real time. This project will investigate these vulnerabilities and the existing security measures and attempt to develop more adaptive real time security measures to enhance SDN security.

Research Methodology & Materials

The project will be using a mixed-methods approach to combine quantitative and qualitative methodologies. The quantitative approach will be used to test and experiment with the already existing security measures and to test the ones that I propose to use. The qualitative approach will be used to understand and analyse SDN vulnerabilities through case studies and research.

My research will mostly be experimental using testbeds like Mininet to simulate attacks and test existing and the security measures I propose be used. This strategy I think is the best moving forward as it is a decent way of evaluating real world scenarios but within a closed and contained environment.

I'll be collecting and analysing data taken from these controlled environments which will allow me to see how the network behaves under different conditions. I will analyse this by comparing things like packet loss and the latency to see how the solutions perform.

The research data will come mainly from those testbeds and public SDN datasets (if available) which will provide real information for testing the vulnerabilities and solutions. Collecting said data would be done through logging tools within the SDN environment to capture patterns and other information.

To remain ethical, all testing will be done on my own environments and no real-world systems will be tested upon.

Software that I will be using will be Python for scripting, Wireshark for analysing packets and Ryu/OpenDaylight for implementing the SDN controller. I think I may need access to server hardware but I might be able to just run it on my PC to experiment instead.

Project Plan

I've decided the best course of action is to span this project across 4 months and it will be broken down as follows:

Month 1: Planning, Extended Literature Review & Research Design

1. Finalize the scope of the project, the objectives & detailed plan
2. Conduct an in-depth literature review on SDN vulnerabilities and security solutions
3. Define metrics for evaluating security solutions

Milestone: To complete the literature review and a well-defined research design.

Month 2: Testbed Setup & Initial Vulnerability Testing

1. Setup the SDN Testbed (Mininet with OpenFlow) and configure the network environment
2. Simulate common SDN attacks like DDoS and control plane hijacking on the test bed to assess the vulnerabilities
3. Analyse the results of the attacks and develop threat models based on the testbed findings

Milestone: To have the testbed fully working with documented vulnerabilities and threat models.

Month 3: Development and Implementation of Security Solutions

1. Design and implement security protocols to address the previously tested vulnerabilities
2. Integrate the security solutions to the test bed and see if they work
3. Begin testing the solutions against the attack scenarios again

Milestone: To have the security solutions fully integrated and first stage of testing completed

Month 4: Final Testing, Analysis & Documenting

1. Complete in-depth testing of the solutions and evaluate their performance
2. Analyse the collected data and compare the performance of the network before and after my solutions
3. Write the final report detailing my findings, analysis and proposed solutions

Milestone: To have all the testing, data analysis and report completed.

References

- [1] B. A. A. Nunes, et al., "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 3, pp.1617–1634, 2014. doi: 10.1109/SURV.2014.012214.00180.
- [2] D. Kreutz, et al., "Software-Defined Networking: A Comprehensive Survey," *ArXiv e-prints*, vol. 103, 2014. doi: 10.1109/JPROC.2014.2371999.
- [3] S. Scott-Hayward, S. Natarajan and S. Sezer, "A Survey of Security in Software Defined Networks," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp.623–654, 2016. doi: 10.1109/COMST.2015.2453114.
- [4] R. Mijumbi, et al., "Network Function Virtualization: State-of-the-Art and Research Challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, 2015. doi: 10.1109/COMST.2015.2477041.
- [5] N. McKeown, et al., "OpenFlow: enabling innovation in campus networks," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp.69–74, mar 2008. doi: 10.1145/1355734.1355746. Available: <https://doi.org/10.1145/1355734.1355746>
- [6] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE Communications Magazine*, vol. 51, no. 2, pp.114–119, 2013. doi: 10.1109/MCOM.2013.6461195.
- [7] "Open Network Operating System (ONOS) SDN Controller for SDN/NFV Solutions." <https://opennetworking.org/onos/> (accessed Oct 10,).
- [8] S. Shin and G. Gu, "Attacking software-defined networks: a first feasibility study," *Proceedings of the Second ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, Hong Kong, China, 2013. doi: 10.1145/2491185.2491220. [Online]. Available: <https://doi.org/10.1145/2491185.2491220>
- [9] S. Wang, et al., "Machine Learning in Network Anomaly Detection: A Survey," *IEEE Access*, vol. 9, pp.152379–152396, 2021. doi: 10.1109/ACCESS.2021.3126834.
- [10] J. H. Ryu, et al., "A blockchain-based decentralized efficient investigation framework for IoT digital forensics," *J.Supercomput.*, vol. 75, no. 8, pp.4372–4387, aug 2019. doi: 10.1007/s11227-019-02779-9. Available: <https://doi.org/10.1007/s11227-019-02779-9>