SDN South-bound Interface Attack Mitigation

Lucio Jankok, Student ID: 8832641

January 13, 2017
Version: 1.0

University of Leiden

Universiteit Leiden

The Hague Security Delta
HSD Campus
Cyber Security Academy

Mitigating the SBSTE-Attack against the SDN South-bound Interface

SDN South-bound Interface Attack Mitigation

Lucio Jankok, Student ID: 8832641

1. Reviewer Dr. Ir. Pieter Burghouwt
HSD Campus
The Hague University of Applied Science

2. Reviewer Dr. Ir. Jan van der Lubbe
HSD Campus
TU Delft

Supervisors Dr. Ir. Pieter Burghouwt and Dr. Ir. Jan van der Lubbe

January 13, 2017

Lucio Jankok, Student ID: 8832641

SDN South-bound Interface Attack Mitigation

Mitigating the SBSTE-Attack against the SDN South-bound Interface, January 13, 2017

Reviewers: Dr. Ir. Pieter Burghouwt and Dr. Ir. Jan van der Lubbe

Supervisors: Dr. Ir. Pieter Burghouwt and Dr. Ir. Jan van der Lubbe

University of Leiden

Cyber Security Academy

HSD Campus

The Hague Security Delta

Wilhelmina van Pruisenweg 104

2595 AN The Hague

Abstract

Both IDC [1] and Gartner [2] expect a significant growth in the implementation of Software Defined Networking (SDN) in the subsequent years. According to IDC, SDN, which is primarily meant for Data Centres, will be incorporated into other network domains such as Wide Area and Campus networks. As this movement continues to spread, SDN will move to more dispersed network environments for which it was not originally designed. SDN has no built-in mechanism to protect the South-bound interface of the SDN controller against Denial of Service attacks. Denial of Service attacks against the SDN controller's South-bound interface have been demonstrated to be possible with only a moderate set of hardware requirements. Without a security solution to detect and mitigate Denial of Service attacks against the SDN controller on its South-bound interface, SDN will introduce this security liability into the Wide Area and Campus networks. It is therefore necessary to design a solution for mitigating DoS attacks against the SDN controller on its South-bound interface.

By means of a conceptual model, this study proposes a security solution to mitigate the DoS attacks against the SDN controller on its South-bound interface. The focus is on a specific type of DoS attack where the trust between the infrastructure layer and the control layer is misused to channel a DoS attack from the infrastructure layer against the control layer. This DoS attack is composed of a massive number of information requests from the infrastructure layer to the control layer. This type of DoS attack is from here on named the *South-bound Timing and Exhaustion attack (SBSTE-attack)*. After researching the current solutions for mitigating DoS attacks against the SDN controller on its South-bound interface, the conclusion was made that none of these solutions provides an adequate mitigation for the SBSE-attack. The security solution proposal provides a multi-staged mitigation mechanism against the SBSTE-attack by providing real-time traffic monitoring and analysis in combination with dynamic programmability of the infrastructure layer. The security solution ignores how the SBSTE-attack is originated and instead focuses solely on the mitigation of the result of the SBSTE-attack on the South-bound interface of the control layer.

[1] https://www.idc.com/getdoc.jsp?containerId=prUS41005016
[2] http://blogs.gartner.com/andrew-lerner/2014/12/08/predicting-sdn-adoption/

The logic of the conceptual model is depicted by means of a flow-diagram. The conceptual model makes use of four timers which are semantically explained as well as formulated by means of simple mathematical equations. The theoretical effectiveness of the conceptual model is showed by means of an attack-tree in which the correct application of the conceptual model is assumed. From these premises, it is shown that the security solution proposal provides a mitigation against the SBSTE-attack. Therefore, it can be concluded that the security solution proposal provides a contribution to the existing set of security solutions on mitigating DoS attacks against the SDN controller because none of the reviewed security solutions provides an adequate mitigation against the SBSTE-attack.

Contents

1 Introduction

> *You can't do better design with a computer, but you can speed up your work enormously.*
>
> **— Wim Grouwel**

Software Defined Networking (SDN) marks a paradigm shift from decentralisation to the centralisation of network control. Contrary to the hop-by-hop data transportation concept inherent to decentralised networking [1], Software Defined Networking provides both a holistic and an abstract management view of the network infrastructure. Both International Data Corporation (IDC) [2] and Gartner [3] expect a significant growth in the implementation of SDN in the subsequent years. According to IDC, SDN, which is primarily meant for Data Centres, will be incorporated into other network domains, including Wide Area and Campus networks. As this movement continues to take place, SDN will move to more dispersed network environments for which it was not originally designed. From a Cyber Security standpoint, SDN's expansion makes it necessary to attest whether the original security controls pertaining to SDN are sufficient for these new network domains. If not properly secured, centralisation could increase the impact of an attack because of the direct impact on all of the dependent satellite components. Adding to this, the probability of attacks against SDN increases as SDN spreads into potentially more dispersed network domains with more complex levels of manageability and a greater number of network participants. As Gartner explains, when both probability and impact increase, the risk also increases exponentially[1]. Studies on how to design proper security solutions to make SDN fit for entering less manageable and dispersed network domains are therefore necessary as SDN continues to penetrate the global market. The focus of this research is to find a solution for the early detection and mitigation of Denial of Service attacks against the centralised SDN controller conducted from the infrastructure layer in which the network devices (SDN switches) reside.

[1] http://www.infocellar.com/networks/ip/hop-count.htm
[2] https://www.idc.com/getdoc.jsp?containerId=prUS41005016
[3] http://blogs.gartner.com/andrew-lerner/2014/12/08/predicting-sdn-adoption/

1.1 SDN Characteristics

Beyond centralisation, the recent approach of Software Defined Networking also aims to provide a decoupled, dynamic, and flow-based network traffic transportation mechanism [2]. The SDN architecture, thus, consists of three different layers or planes, as presented in figure 1.1.

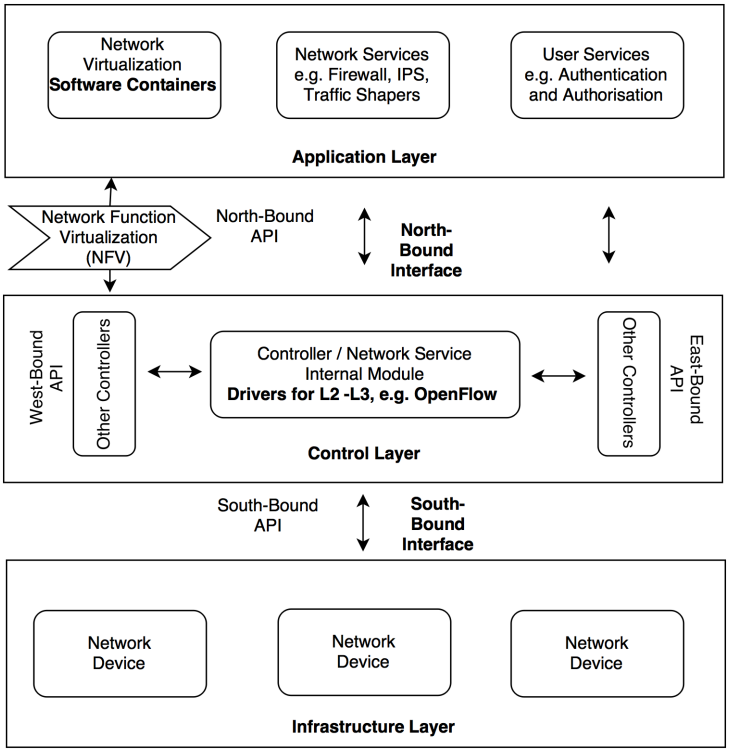

Fig. 1.1: SDN Three-Layer Architecture

The network devices (SDN Switches) reside in the infrastructure layer, which is the bottom layer in figure 1.1. This layer is responsible for the transportation of data packets. The infrastructure layer initially lacks the information necessary to transport a certain packet from point A to B because that type of information is decoupled from the infrastructure layer and is placed in the control layer where it is centralised into the SDN controller.

The control layer (the middle layer) is where the SDN controller resides. The SDN controller is a centralised software component which provides the infrastructure layer with packet switching/flow information. The packet switching/flow information can be pushed from the control layer to the infrastructure layer or requested from the control layer by the infrastructure layer.

The network functions (e.g. network/user services) are placed in the application layer (the topmost layer). These network functions can instruct the SDN controller

in the control layer to send instructions to the infrastructure layer where the network devices reside. This particular relationship between the application layer and the control layer denotes the programmability of the SDN infrastructure. The application layer provides SDN-aware applications with an abstract view of the network and provides a framework for applications to dynamically request network resources [3].

The communication path between the application layer and the control layer is called the North-bound interface, whereas the communication path between the infrastructure layer and the control layer is called the South-bound interface [4]. In case of multiple SDN controllers working together, there are also the East and West-bound interface for intra-controller communication [4].

From a design architecture point of view, SDN consists of external and internal (controller core) modules [2]. The internal module resides at the control layer and consists of a network manager, network operating system and APIs, as well as internal services and drivers. The external module consists of the application layer (North-bound), the control layer (West and East-bound), and the infrastructure layer (South-bound).

1.2 OpenFlow

OpenFlow is a flow-based South-bound protocol which utilises flow-tables to orchestrate the path that a network packet takes through the infrastructure layer. From the perspective of SDN, the OpenFlow protocol is considered an external module [2]. The network programmer can create flow-tables statically or leave the flow-tables to be on the fly created based on requests of trusted SDN-aware applications in the application layer. Flow-tables are therefore not necessarily static and are pushed to the infrastructure layer or are to be requested from the control layer as needed. According to Li, OpenFlow is currently the most deployed SDN concept [5]. The OpenFlow South-bound protocol is also completely open sourced which makes it easier to adapt. Thus, the focus of this research will be on OpenFlow in order to provide a better representation of the SDN South-bound protocol.

An SDN infrastructure can be based on SDN switches in the infrastructure layer or can be a mix of SDN switches mixed with SDN OpenFlow-hybrid switches. SDN OpenFlow-hybrid switches supports the OpenFlow South-bound protocol and also have the capability to switch packets in the infrastructure layer completely independent of the SDN controller. A SDN OpenFlow-hybrid switch has therefore the ability to revert to the current decentral paradigm of packet switching in which

each switch is statically programmed with the knowledge to transport packets in the infrastructure layer independently of the SDN controller. SDN OpenFlow-hybrid switches are considered as a solution for the specific scenario in which SDN has to gradually replace a network infrastructure consisting of network devices based on the current decentral paradigm of packet switching. The focus of this research will not be on this transition scenario and therefore will be based on a SDN infrastructure without SDN OpenFlow-hybrid switches.

1.3 SDN versus Contemporary Security Implementation

When implementing security middle-boxes (e.g., Firewalls, IDS) in the current conventional network paradigm, the placement is fixed in the network infrastructure. The limitation of this type of implementation is the lack of flexibility due to both its fixed placement in the network infrastructure and due to the static network configuration for transporting traffic through the middle-boxes. SDN together with Network (Function) Virtualisation at the application layer, however, could provide a security automation framework that solves the limitations of static placement of security middle-boxes [6]. This level of automation could essentially result in a network infrastructure that acts more like an organism which can generate mitigation against attacks on demand and decommission the mitigations when no longer needed [7]. This characteristic of SDN could potentially result in SDN dealing in a more efficient and effective manner with contemporary cyber-attacks than can be accomplished within the current decentral network paradigm.

SDN may create a new realm to the core of cyber-space by offering opportunities for new services and new methods for transporting packets and mitigating cyber-attacks [7]. From this perspective, the relevance of SDN to the cyber-security landscape is significant because it could provide a solution to the static placement of security middle-boxes and the static configuration of the traffic transport mechanism through these middle-boxes. By introducing network programmability to network security, SDN may provide fully automatic security solutions which are not possible within the current decentral network paradigm. Despite that SDN has the potential of introducing new efficient and effective ways of mitigating cyber-attacks, however, SDN could also be prone to its own set of vulnerabilities. Therefore, instead of focusing only on the benefits that SDN brings, it is necessary to simultaneously focus on SDN's potential vulnerabilities.

1.4 Problem Statement

While decoupling the control and infrastructure layer, SDN places the control layer in a centralized position. From the infrastructure layer, each time there is an incoming packet with no matching flow-table, a query is sent to the control layer to obtain flow information for that specific packet. The OpenFlow Switch Specification [8] depicts this mechanism as follows:

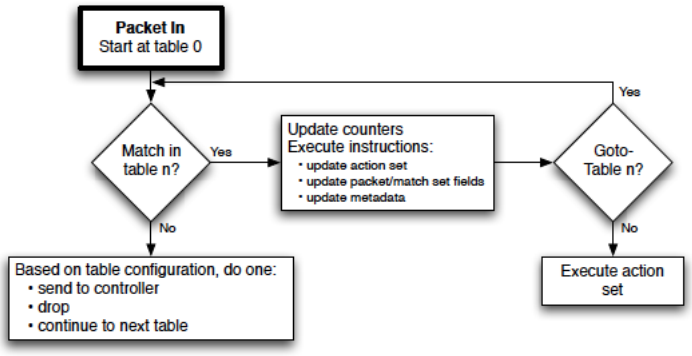

Fig. 1.2: OpenFlow Packet Forwarding according to the OpenFlow Switch Specification [8]

An explanation of figure 1.2 follows next. A packet arrives at the SDN switch in the infrastructure layer. The switch's task is to switch the packet from the input port from which the packet arrived to an output port to which the packet must leave the switch to go further down its destination path. In order to put the packet on the correct output port, the switch needs to know specific information about the packet. The switch finds this information by looking at the packet specifics like the source and destination addresses. Based on the packet specifics, the SDN switch next tries to find a matching flow-table in its current set of saved flow-tables. The set of stored flow-table runs from "0" to "N" in which "N" is the total number of flow-tables the SDN switch has locally saved. Starting from "table 0" to the last available table "N", a match is iteratively sought for a flow-table which matches the packet specifics. If no match is found, then the switch can either ignore (drop) the packet or request the SDN controller for flow information for that particular packet. If the packet is matched by one of the available flow-tables, then the network device can either move to a specific flow-table entry or execute a specific switching action.

The point of focus is on the specific packet which has no matching flow-table on the switch. Assuming that the network programmer has not opted for the switch to ignore packets with unknown flow-tables, the switch will request flow-table information from the SDN controller. As such, each time that a packet for which there is no matching flow-table enters the infrastructure layer, a flow-table information request is sent from the infrastructure layer to the control layer.

According to Shin, this key SDN property can be exploited first by fingerprinting (by means of a timing attack) whether a given network uses OpenFlow switches to next deduct which packet flows are unknown to the infrastructure layer [9]. The attacker uses this information to generate specifically crafted packets which will cause the infrastructure layer to request large amounts of flow information from the control layer. According to Shin this will result in two resource starvation scenarios [9]:

1. Control layer resource consumption (DoS) attack
2. Infrastructure layer resource consumption (DoS) attack.

This kind of cyber-attack is not directly aimed at South-bound interface of the control layer, but instead misuses the trusted relationship between the control layer and the infrastructure layer in order to perform a resource starvation (Denial of Service) attack against the South-bound interface of the control layer / SDN controller. Therefore, conventional security methods like encryption or public key infrastructure systems would be ineffective in stopping or mitigating this type of attack. This type of attack will be named from here on as *South-bound Timing and Exhaustion attack (SBSTE-attack)*. The effect of resource starvation on the control layer could result in the control layer's inability to process requests from the application and/or infrastructure layer and would therefore result in a severely disrupted SDN infrastructure.

The SBSTE-attack can be used as the base for extensive attacks by hindering the SDN controller to process requests from the application layer, and therefore making the entire infrastructure more susceptible to staged attacks. On a more basic level, this kind of attack can be used to disrupt the basic dynamic character of the SDN infrastructure, and by this disrupting all network traffic in the infrastructure layer for which there is no available flow information.

According to Scott, data Leakage (timing-attack) issues in SDN security is the least researched aspect in the area of SDN security [4]. Since the fundamental concept of SDN is to consult the central controller for unknown traffic flows, without wanting to propose changes to this SDN concept, the specific issue at stake is the saturation of the SDN South-bound control layer by means of a SBSTE-attack. As noted by Shin, this type of attack can be conducted remotely with only moderate hardware requirements, and therefore poses a formidable threat against the complete SDN infrastructure [9].

Scott has identified 6 attack vectors against the SDN infrastructure [4]:

1. Unauthorised Access (All Layers/Interfaces)
2. Data Leakage (Infrastructure Layer)

3. Data Modification (e.g. man-in-the-middle) (Control - Infrastructure Layer)
4. Malicious/Compromised Applications (Application - Control Layer)
5. Denial of Service (Control - Infrastructure Layer)
6. Configuration Issues (All Layers/Interfaces).

The SBSTE-attack is a combination of items 2 and 5 from Scott's list of potential attack vectors against the SDN infrastructure. If the SBSTE-attack type is not addressed properly, the SDN infrastructure would remain susceptible to be remotely disrupted by an attacker using only moderate hardware resources. This essentially means that a potentially promising technology with dynamic automated mechanisms for providing new ways to mitigate cyber-attacks, in fact itself amplifies the impact of DoS attacks to the level of efficiency that a remote attacker could potentially disrupt an entire SDN network infrastructure. This attack type has been successfully demonstrated by Shin by means of a new SDN network scanning prototype tool (SDN Scanner) and by applying it on an SDN test environment using real world experimental data [9].

The issue here is that with the prognosed adaptation of SDN, this new networking paradigm will replace the current dominant decentral network paradigm. In order for SDN to replace the dominant networking paradigm without having to become a (cyber) security liability in the networking landscape, inherent security issues of SDN itself must be addressed.

1.5 Objective

The objective of this study to create a security solution in order to mitigate the SBSTE-attack on the South-bound interface of the control layer and more specifically on the South-bound interface of the SDN controller. A second objective is to evaluate the proposed security solution. From this evaluation, knowledge will be inferred as to whether the proposed security solution does provide a mitigation to the SBSTE-attack and as to what combination of security controls provide this level of mitigation. The knowledge acquired through this evaluation is a contribution to the list of current security solutions for mitigating DoS attacks against the South-bound interface of the control layer, more specifically the South-bound interface of the SDN controller.

1.6 Research Questions

How does one effectively mitigate the SBSTE-attack against the SDN South-bound interface?

1. What are the existing security solutions or approaches regarding the mitigation of DoS attacks against the SDN South-bound interface?
2. Which of the existing security solutions or approaches can mitigate the SBSTE-attack against the SDN South-bound interface?
3. What new security solution can contribute to the mitigation of the SBSTE-attack against the SDN South-bound interface?

1.7 Outline

- Chapter 2 presents the research approach.
- Chapter 3 is a survey on the current security solutions regarding the mitigation of DoS attacks against the SDN South-bound interface.
- Chapter 4 presents a security solution for mitigating the SBSTE-attack.
- Chapter 5 discusses the evaluation of the new security solution and provides a discussion on the research results and on potential future research work.
- Chapter 6 ends with a conclusion.

2 Research Approach

> *Imagination is more important than knowledge. Knowledge is limited. Imagination encircles the world.*
>
> **— Albert Einstein**

The objective of this research is to produce a security solution to mitigate the SBSTE-attack. The process to develop such a mechanism falls into the domain of design science. A selection from Hevner's guidelines for design science provides a scope for this research. In this chapter, I provide an integrated research approach by mapping Akhunzada's security solution field classification [10] and Wieringa's engineering cycle [11] onto a selection of Hevner's design science guidelines [12].

Hevner provides a high-level approach to design science and describes the framework of a design science research by presenting the following guidelines and their objectives:

1. Design as an artefact
2. Problem Relevance
3. Design Evaluation
4. Research Contributions
5. Research Rigour
6. Design as a Search
7. Communication of Research

What is missing in Hevner's description of design science, however, are the engineering steps needed in the design science research. Wierenga fills this gap by describing the engineering steps, which are later on discussed, needed to conduct a design science research. What is always needed is specific domain knowledge regarding the specific domain in which the security solution (artefact) must apply. Akhunzada provides a security solution classification in the SDN domain which can be used to categorise security solutions. Similarly, Akhunzada's security solution classification serves as a framework to describe the proposed security solution to mitigate against the SBSTE-attack. But first, a literature study in the specific research domain provides clarity about the research gaps present in the research domain

and is therefore a necessary step in the research approach. This integrated research approach is depicted in Figure 2.1 where literature is at the centre embodying the three angles of the triangle.

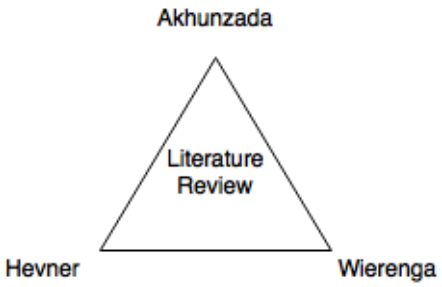

Fig. 2.1: Research Approach

2.1 Research Approach Application

The application of these four elements in this research will next be presented.

A literature study provides the body of knowledge upon which this study is conducted and is therefore the first step in this research approach. The literature study is next denoted as Element 1.
Element 1 - Literature Study.
From a high-level generic approach, the aim of the literature study is threefold: one it is to serve as the base of the research approach; second, it provides the background on which the relevance of the research questions can be validated; and third, it presents an overview of the current body of knowledge regarding the specific research subject.

After determining the body of knowledge upon which this study is conducted, the next step is to determine the scope. A scope is needed in order to specify the outcome of this study. Scoping is next denoted as Element 2.
Element 2 - Hevner's Design Guidelines.
The scope of this research is to design a security solution for the SBSTE-attack and to evaluate this security solution. Because of this limited scope, only the first three items from Hevner's design guidelines apply:

1. Problem relevance
2. Design as an artefact
3. Design evaluation

Since the scope of this study is to design an artefact and to evaluate the artefact, the context for which the artefact is build must be made clear in order for the artefact to

be properly evaluated. Contextualizition is next denoted as Element 3.
Element 3 - Akhunzada's Classification.
The scope of this research is to provide a security solution for mitigating the SBSTE-attack in an OpenFlow SDN environment. Akhunzada provides a classification of SDN security solutions. Akhunzada's classification of security solutions works to describe this study's proposed security solution. The following classification items are provided by Akhunzada:

1. SDN layer/interfaces
2. Security measures
3. Implementation (simulation/emulation) environment
4. Security objectives

A central item in the design of the security solution is to adhere to a secure design implementation. The reason for this approach is the desire to minimise the possible attack surface the proposed security solution could introduce. Another reason for this approach is the desire to make the proposed security solution a part of the overall SDN design and as such to be a required feature. Without the proposed security solution, the original SDN South-bound interface design would remain a security liability. Making the proposed security solution non-optional addresses this. Therefore, this study uses the following selection of the extending list of Akhunzada's classification of existing SDN security solution:

1. Secure design of SDN

After the scope and the context have been determined, the last element, the engineering cycle used to build the artefact is next defined.
Element 4 - Wieringa's Engineering Cycle.
The fourth and last element is Wierenga's engineering cycle. Wierenga's engineering cycle usefully translates the selected design guidelines of Hevner into engineering tasks: Design as an artefact, Problem Relevance, and Design evaluation. The following selection of Wierenga's engineering cycle maps onto Hevner's problem relevance guideline:
Problem investigation. A problem investigation is needed in order to clarify the context and to clarify why the problem is a problem in the first place and what type of solution is required. Wierenga suggests to address the following items during the problem investigation process:

- Stakeholders
  - Addressed in Chapter 4, Design Approach section
- Goals
  - Addressed in Chapter 1, Objective section

- Phenomena
  - Addressed in Chapter 1, Problem Statement section
- Causes
  - Addressed in Chapter 1, Problem Statement section
- Effects
  - Addressed in Chapter 1, Problem Statement section

After the relevance of the problem and the necessity of a solution have been addressed, the next step is to address the steps needed to build a solution that addresses the identified problem. The following selection of Wierenga's engineering cycle maps onto Hevner's design as an artefact guideline:
Treatment design. A treatment is the interaction between an artefact and a certain context. An artefact is designed in order to create a treatment. Wierenga states to address the following items during the treatment design:

- Available treatments
  - Addressed in Chapter 3, Related Work. It is necessary to know which treatments are already available.
- Specify requirements
  - Addressed in Chapter 4, Security Solution Proposal. It is necessary to know to which specific requirements the artefact must adhere.
- Contribution to goals
  - Addressed in Chapter 4, Section Requirements. It is necessary to know how the requirements contribute to the specific goals.
- Design a solution proposal
  - Addressed in Chapter 4, Security Solution Proposal. The security solution proposal can be built only after the problem investigation and as the last step in the treatment design. This also concludes the mapping to Hevner's design as an artefact guideline.

After building the artefact the third and last step is the validation process. The following selection of Wierenga's engineering cycle maps onto Hevner's design evaluation design guideline:
Design Evaluation. Wierenga suggests to address the following items during the design evaluation:

- Context & Solution proposal
  - Addressed in Chapter 5, Security Solution Evaluation.
- Effects satisfy requirements
  - Addressed in Chapter 5, Security Solution Proposal.
- Sensitivity for different contexts
  - Addressed in Chapter 5, Discussion.

2.2 Research Approach Objectives

The research approach addresses the following research questions:

1. What are the existing security solutions or approaches regarding the mitigation of DoS attacks against the SDN South-bound interface?
2. Which of the existing security solutions or approaches can mitigate the SBSTE-attack against the SDN South-bound interface?
3. What new security solution can contribute to the mitigation of the SBSTE-attack against the SDN South-bound interface?

By building upon the work of Hevner, Akhunzada, and Wierenga, the aim of this study it to provide a security solution to mitigate the SBSTE-attack against the SDN South-bound interface. A second objective is to evaluate the proposed security solution and to contribute with the knowledge originating from this evaluation to the existing academic body of knowledge in the SDN security domain.

3 Related Work

> *A picture is worth a thousand words. An interface is worth a thousand pictures.*
>
> **— Ben Shneiderman**

This section provides a review on the current security solutions against DoS attack on the SDN South-bound interface by addressing the following research questions:

1. What are the existing security solutions or approaches regarding the mitigation of DoS attacks against the SDN South-bound interface?
2. Which of the existing security solutions or approaches can mitigate the SBSTE-attack against the SDN South-bound interface?

This chapter begins by providing an introduction that highlights the necessity of conducting research on SDN security. The chapter then presents a critical consideration of the different security solutions, and concludes with a classification of the security solutions based on a set of security controls that will offer insight on the methods of approach and areas of focus for each of the security solutions.

3.1 SDN Security

As SDN brings new and exciting functionality to the networking landscape, it also introduces new security issues to the cyber-security landscape. According to Akhunzada, security was not considered as part of the initial design of SDN and the security and dependability of SDN has largely been a neglected topic and remains an open issue [10]. According to Yoon, security is the one area of SDN that requires further development [13].

Scott-Hayward, for example, makes a distinction between the Data Leakage (timing) attack and the Denial of Service attack which becomes apparent when she describes the current solutions to these security issues [4]. She describes the definition of a Denial of Service attack as a remote attack directly against the South-bound interface of the infrastructure or control layer. Although the result of the Data Leakage attack

can contribute to a successful Denial of Service attack against the Control layer, the attack is actually being conducted from the infrastructure layer and is indirectly being triggered by a remote system. Therefore, this type of attack does not fit within Scott-Hayward's definition of DoS attack. According to Scott-Hayward, there are no proposed solutions to date for such Data Leakage issues. However, the type of Data Leakage described by Scott-Hayward is inherent to the design philosophy of the SDN South-bound protocol. As such, finding a solution for the Data Leakage would be the same as making a redesign of that specific aspect of the South-bound protocol. Another approach is to find a mitigation solution for the Denial of Service attack subsequent to the Data leakage attack.

3.1.1 Security Solutions for South-bound Interface DoS Attacks

This section addresses the following research question:

1. What are the existing security solutions or approaches regarding the mitigation of DoS attacks against the SDN South-bound interface?

By reviewing the pertinent literature of the field, I have identified nine security solutions against DoS attacks in the SDN infrastructure. The nine solutions are as follows:

AVANT-GUARD, Shin [9], is an SDN security solution against DoS attacks in the control-infrastructure layer where the attacker uses a spoofed IP address. However, this security solution fails to provide a mitigation for the SBSTE-attack because the SBSTE-attack doesn't necessarily depend on spoofing IP addresses. The SBSTE-attack is aimed at the South-bound interface and only uses the infrastructure layer as a stepping stone for the subsequent attack. Hence the SBSTE-attack is also denoted as an indirect (via the infrastructure layer) attack against the South-bound interface. Yet, the AVANT-GUARD security solution does limit the attacker's freedom to create specific traffic types.

CPRecovery, Fonseca [14], is an SDN security solution against DoS attacks in the control-infrastructure layer by means of providing control layer redundancy through failover mechanisms. While this security solution does provide redundancy in the control layer, it does not provide a control mechanism to mitigate the source of the attack. CPRecovery provides a mechanism to increase the resilience of the control layer, but it does not address the DoS attack directly. This security solution can be used as a control mechanism to mitigate the result of the attack on the control layer. However, it does not mitigate the result of the attack on the infrastructure layer

because the redundancy provided is only for the control layer. Mitigating resource starvation attack on the control layer without considering the accompanying resource starvation on the infrastructure layer could eventually result in the same network disruption effect if the infrastructure layer gets oversubscribed. In such a scenario, the control layer would still be able to process the request of the infrastructure layer, but the infrastructure layer would find it challenging to communicate through its oversubscribed line with the control layer.

Yao's source address validation scheme, Yao [15], is the third SDN security solution against DoS attacks in the control-infrastructure layer by providing pre-emptive protection against IP spoofing which could lead to DoS attacks. Since this security solution is aimed at spoofed IP address used by the attacker, like AVANT- GUARD, it does not provide a mitigation against the SBSTE-attack. It does however limit the attacker's freedom to create specific traffic types.

The fourth solution is Naous' ident++ protocol [16], which provides an SDN security solution against DoS attacks in the control-infrastructure layer. This security solution provides an elaborate delegation of control and trust such that users and end-hosts can participate in network security enforcement. Although this security scheme could be used as a security control against the SBSTE-attack, it does this at a price of introducing higher complexity in the SDN infrastructure. However, higher complexity in the SDN infrastructure could be a source for new attack vectors and is therefore not a preferred security solution approach. The security control relies on a trust relationship between the control layer and the infrastructure layer which might be highly infeasible in many situations (e.g. large datacentre with many virtual machines).

FortNox [17] focuses on rule conflicts and authorisation and as such does not provide a mitigation for the SBSTE-attack since the SBSTE-attack is misusing an already trusted connection between the infrastructure and the control layer.

Alsmadi addresses two types of security controls against data leaking in SDN [2]:

1. Install all possible flow rules in the infrastructure layer
2. Implement a system for fake response time from control layer to the infrastructure layer

Alsmadi analyses these security controls to emphasise that there is currently no adequate security solution to mitigate the SBSTE-attack. Although Alsmadi asserts his own reasons for not considering these security controls as a viable solution against the SBSTE-attack, the conclusions of Alsmadi regarding these two types of security controls are not need to be analysed further. Instead of taking the conclusions of

Alsmadi as is in this study, these security controls are being considered again on their level of mitigation against the SBSTE-attack.

The first security control is to take a fully proactive approach by installing all flow rules beforehand. The second security control is to make fake response time from control layer to the infrastructure layer. However, installing all flow rules beforehand would result in a high administrative burden and will also have a negative impact on the flexible nature of the SDN infrastructure. Implementing a system for fake response time between the control and infrastructure layer deviates too much from the core concept of the South-bound protocol, in which no manipulation of flow request response time exists, and is more in line with security through obscurity. This security control has a direct negative impact on the SDN performance because the fake response times can only be larger than the real response times. In the end the result of this consideration is in line with Alsmadi's conclusions and denotes once more that these sets of security controls fail to provide a viable solution against the SBSTE-attack.

Hussein proposes a technique for DoS prevention against the control layer by malicious hosts and switches by introducing an SDN software agent which communicates with a new security layer between the control and infrastructure layer [18]. On the switch level, Hussein proposes to apply preconfigured MAC-IP-Switch-Port binding to mitigate against spoofing. This solution relies on specific malicious devices in the infrastructure layer to be detected based on IP's which fail the MAC-IP-Switch-Port binding. However, pre-configuring MAC-IP-Switch-Port binding on the switch level is a legacy security method which only works effectively in small to medium sized networks. Therefore, this solution is not a feasible solution in the SDN infrastructure with the potential of thousands of virtual hosts. Furthermore, by introducing new interfaces and new layers, this introduces possible new attack vectors and more complexity to the SDN South-bound interface. The security solution by Hussein therefore does not provide a mitigation against the SBSTE-attack since the issue is about misusing a trusted connection between the infrastructure and the control layer where IP spoofing must not necessarily be part of the attack composition.

According to Li [5], trust models and validation mechanisms should be added to protect the SDN controller from threats such as DoS attacks. Yet, this approach does not account for the SBSTE-attack where the attackers misuse the trusted and validated connection between the infrastructure and control layer.

Finally, Nayak proposes Resonance as a security solution to deal with malicious hosts inside an SDN network infrastructure [19]. While Resonance provides an enhanced Network Admission Control solution which is more advanced than the current legacy solutions, it assumes that a DHCP server is used in the network

infrastructure and base its security mechanism on this. However, the dependency on a DHCP server makes this solution not applicable in mitigating against the SBSTE-attack since in the SBSTE-attack scenario no dependency exists on a DHCP server because the infrastructure layer could exist on only statically addressed hosts and network devices.

3.2 Summary

Table 3.1 provides a summary of the security solutions on DoS attacks against the South-bound interface analysed throughout this chapter. Each solution is mapped against Akhunzada's classification list of SDN security controls:

1. Secure Design of SDN - SD
2. Security Audit - AUD
3. Security Enforcement Policy - POL
4. Security Augmentation - AUG
5. Security Monitoring and Analysis - MON
6. Fault Tolerance - RED

Since the table consists of security controls and the SBSTE-attack, a distinction is made to denote with an "x" if a security solution utilises a security control. The effect of the security solution on the SBSTE-attack is denoted with a "+" for satisfactory, a "-" for insufficient and a "+/-" for weak.

Author	SD	AUD	POL	AUG	MON	RED	SBSTE
Shin [9]	-	-	-	-	x	-	-
Fonseca [14]	-	-	-	-	-	x	-
Yao [15]	-	-	-	-	x	-	-
Naous [16]	-	-	x	-	-	-	-
Porras [17]	-	-	x	-	-	-	-
Alsmadi [2]	-	-	x	-	x	-	+/-
Hussein [18]	-	-	-	-	x	-	-
Li [5]	-	-	x	-	-	-	-
Nayak [19]	-	-	x	-	-	-	-

Tab. 3.1: Existing security solutions against SDN DoS attacks, identified by the author of the scientific publication and classified by the underlying basic security controls by Akhunzada [10] with the resulting applicability on the SBSTE-attack.

In this study, Secure by Design of an SDN system is defined as:

1. A security solution which is part of the internal SDN architecture
2. A security solution which is enabled by default
3. A security solution which is built with the aim of minimising the attack surface

The following research question will now be answered:

1. Which of the existing security solutions or approaches can mitigate the SBSTE-attack against the SDN South-bound interface?

From table 3.1 the following is evident:

1. Only the security controls addressed by Alsmadi present an attempt to directly mitigate against the SBSTE-attack. However, in accordance with Alsmadi, these security controls are not found to be effective security controls for mitigating against the SBSTE-attack. Installing all flow rules beforehand would result in a high administrative burden and will also have a negative impact on the flexible nature of the SDN infrastructure. Implementing a system for fake response time from control layer to the infrastructure layer is more in line with security through obscurity and would not be an effective solution to the SBSTE-attack. This security control has a direct negative impact on the SDN performance because the fake response times can only be larger than the real response times.
2. None of the presented security controls falls into the category of secure design. Although the security control by Hussein could be considered as a secure design approach, it is not because the control depends on extensive manual intervention in the infrastructure layer. According to Akhunzada [10], efforts in the secure design of SDN are limited as can be inferred from table 3.1.
3. None of the nine current security solutions uses Augmentation as a security control and as such does not make use of the security control mechanism provided by SDN.

The security controls addressed by Alsmadi presumes that the attacker has no other ways of knowing that SDN is being used in a particular network infrastructure. As already noted, the first security control addressed by Alsmadi requires a fully proactive approach by installing all flow rules beforehand. The second addressed security control makes fake response time between the control and infrastructure layer based on the nature of the network attack. Installing all flow rules beforehand would result in a high administrative burden, but will on the other hand not degrade the dynamic nature of the network since the application layer will still be able to reprogram the infrastructure layer as needed. Such an approach also deviates

significantly from the core idea of the South-bound protocol behaviour in which no manipulation of flow request response time exists. The consequence of such approach is the negative effect on SDN performance. By manipulating the response time, the latency can only be increased which then would result in network performance degradation.

The first security control is too labour intensive to be of practical use in large network infrastructures where thousands of flow tables may exist. The second security control is more in line with security through obscurity and would become less effective once the attacker finds other easier ways of knowing if a specific network has SDN deployed. The addressed security controls by Alsmadi are, therefore, limited security controls against the SBSTE-attack on the South-bound interface. For these reasons the security controls as addressed by Alsmadi are still ineffective security controls against the SBSTE-attack.

Table 3.1 illustrates that there is still no security solution with all of the following characteristics:

1. Uses new SDN security methods - Security Augmentation
2. Uses the Secure Design of SDN approach - SD
3. Provide a solution against the SBSTE-attack - SBSTE

Thus, none of the presented security solutions or approaches presents a viable solution for mitigating the SBSTE-attack against the SDN South-bound interface.

Chapter 4 presents a new security solution that uses both Security Augmentation and the Secure Design of SDN in order to provide a security solution against the SBSTE-attack.

4 Security Solution Proposal

> *Innovation distinguishes between a leader and a follower.*
>
> — **Steve Jobs**

In this chapter, I present a security solution for the SBTSE-attack on the South-bound interface in accordance with the research approach described in chapter 2. I begin by addressing the following research question:

1. What new security solution can contribute to the mitigation of the SBSTE-attack against the SDN South-bound interface?

4.1 Assumptions

The following assumptions are at the base of this research:

1. OpenFlow is used as the South-bound protocol.
2. The SDN infrastructure contains no OpenFlow-Hybrid network devices.
3. The network programmer has not configured the infrastructure layer to drop/ignore packets for which there is no matching flow-table.
4. The SBSTE-attack is a feasible attack requiring only moderate hardware and can be automated such that even a script kiddie can perform.

Regarding assumption one, OpenFlow is the most deployed SDN concept and therefore provides a better representation of the SDN South-bound protocol. Without the second assumption, the SBSTE-attack could be mitigated by reverting back to conventional packet forwarding. The second assumption also implies that the context of this research is based on a full non-hybrid SDN infrastructure. Without the third assumption, the SBSTE-attack could be mitigated by dropping/ignoring all packets for which there are no existent flow-tables on the infrastructure layer. The third assumption necessarily negates the opposite assumption, which could be that the infrastructure layer has knowledge of all relevant flows and can therefore safely drops/ignore packets for which there are no existing flow tables. Although this oppo-

site assumption could make sense in some controlled cases, it would not withstand in the case where flexibility is required between the infrastructure and control layer. The fourth assumption relies on the fact that once a vulnerability is known, the probability of an exploit increases. For the security solution to be incorporated in the design of the SDN controller, it must also follow the same distribution model as the OpenFlow source-code. Making the logic of the security solution open-sourced implies that any vulnerability in the logic is readily visible to any one with the proper knowledge. It is, therefore, paramount that the security solution proposal must take care not to increase the attack surface and not to allow bypasses.

4.2 Design Approach

Based on the specific selection of Wieringa's engineering cycle in section 2.1, which maps onto the Hevner's design as an artefact design guideline, this chapter addresses the following components of Wierenga's engineering cycle:

1. Stakeholders.
2. Specify requirements.
3. Contribution to goals.
4. Design a solution proposal.

4.2.1 Stakeholders

The following stakeholders are identified:

1. Malicious hacker
2. Network users
3. National authorities
4. Network providers

A large SDN network infrastructure is an attractive target for DoS-attacks. This creates the threat. SBSTE-protection mitigates the risk that results from this threat and the intrinsically vulnerable SDN South-bound interface construction. Network users expect a robust and high-available network infrastructure. Network users do not value DoS-attack protection which drops or significantly limits their legitimate traffic. The SBSTE-protection is contributing to a robust network infrastructure and to mitigating attacks against the network infrastructure's availability. National authorities require that the national critical infrastructures are "attack-resilient". Network providers, under pressure of the above-mentioned stakeholders, are motivated to implement SBSTE-protection (amongst other protections). An SBSTE-solution

that can be built in an existing SDN-environment is for the network providers an attractive solution because of cost efficiency.

The design of the new security solution against the SBSTE-attack must take all of these potential socio-technical aspects into account. This is the part in which the work of Akhunzada is merged with the work of Wieringa to comply to Hevner's problem relevance guideline.

In the requirements specification, Akhunzada's list for classification of security solutions classifies the new security solution. This is the part in which the work of Akhunzada is merged with the work of Wieringa to comply to Hevner's design as an artefact guideline.

4.2.2 Requirements

The requirements for the security solution are the necessary conditions to which the proposal must adhere in order to account for the stakeholders' objectives. Thus, the requirements must contribute to the following goals:

1. Applicable in existing SDN-implementations
    - The aim is to present a security solution proposal which does not require fundamental changes in the functioning of the South-bound protocol or on the interaction between the control layer and the infrastructure layer. As such, the aim is to propose a security control mechanism which does not disrupt the fundamental working of the OpenFlow South-bound protocol.
2. Minimise the resulting attack surface
    - A second aim is that the security control proposal must itself not introduce new attack vectors from the South, West or North-bound interfaces.

As mentioned in section 3.2, the new security solution must adhere to the following characteristics based on the categories provided by Akhunzada [11] for classifying SDN security solutions:

1. The new security solution must use new SDN security methods (Security Augmentation).
2. The new security solution must use the Secure Design of SDN approach.
3. The new security solution must be aimed at mitigating the SBSTE-attack.

Based on this study's stated objectives, I have compiled the following list of requirements:

1. The security solution must operate at the control layer in order to satisfy the goal of minimising the attack surface.
2. The security solution must use existing traffic identifiers to detect anomalies. This is to satisfy the goal of enhanced implementation feasibility.
3. The security solution must follow the secure design methodology.
4. The security solution must not be optional.
5. The security solution must not require fundamental adaptations to the South-bound protocol or the SDN controller. This satisfies the goal of enhanced implementation feasibility.

The main goal for the security solution proposal is both the feasibility of its implementation and the requirement to only introduce a minimal set of changes in the original SDN design architecture. The approach is, therefore, not to change the fundamental working of the internal or external SDN modules but to introduce a new module which, in a non-disruptive way, works together with the existing modules.

4.3 Conceptual Overview Security Solution

The following illustration, figure 4.1, provides a conceptual overview of the security solution.

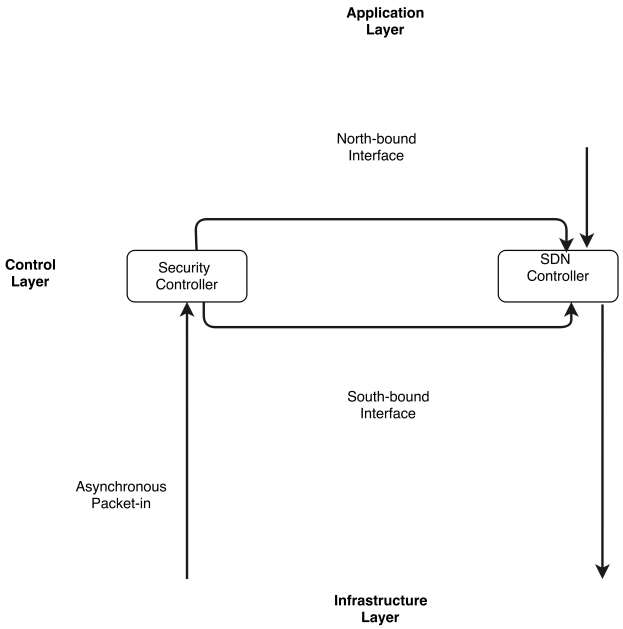

Fig. 4.1: Conceptual Overview Security Controller

From a high-level perspective, the concept is to introduce a security controller and to change the information flow between the infrastructure layer and the control

layer. A second controller is introduced to function as the security controller. The communication from the infrastructure layer to the control layer flows through the security controller whereas the communication from the control layer to the infrastructure layer flows through the SDN controller to the infrastructure layer. The security controller communicates with the SDN controller on its South and North-bound interface and does not require any modification on the existing South-bound protocol because there is no need to modify the packet format or content.

The security solution proposal does not mitigate the data-leaking timing attack in which the attacker obtains the information on the particular packet characteristics that a specific infrastructure layer cannot process directly. Instead, the security control proposal provides mitigation for the second stage of the attack in which the attacker sends massive numbers of packages to the infrastructure layer for which no existing flow-table exist.

4.4 A Closer Look at the Security Controller

In order to mitigate the second stage of the attack, a modified data-bus is required on the control layer. This data-bus controls the information flow from the application layer to the control layer and between the control layer and the infrastructure layer. The security controller can then communicate directly with the SDN controller on its South-bound and North-bound interface.

In figure 4.2, the working of this modified data-bus is presented.

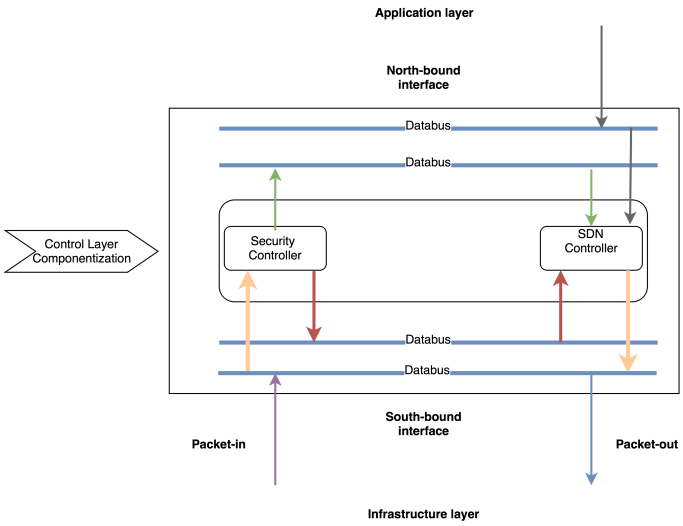

Fig. 4.2: New data-bus design

The role of the modified data-bus is as follows:

1. To provide the security controller with access to the South-bound and North-bound interfaces.
2. On the South-bound interface the security controller is allowed to receive packets (packet-in) from the infrastructure layer but is not allowed to send packets (packet-out) directly to the infrastructure layer.
3. The SDN controller, on the other hand, can send packets to the infrastructure layer (packet-out) but is not allowed to receive packets (packet-in) directly from the infrastructure layer.
4. The security controller is permitted to send packets to the SDN controller on its South and North-bound interface following the standard rules of the South-bound protocol.
5. Packet flow from the infrastructure layer to the control layer (packet-in) can only flow through the security controller to the SDN controller.

The resulting communication scenario is denoted as a half-duplex design. In this study, the term half-duplex is used to denote a design concept where the path from source to destination is not the same as the path from destination to source.

Once an anomaly has been found, the security controller proceeds as follows:

- Instruct the SDN controller through its North-bound interface to send instructions to the infrastructure layer to ignore (block) a certain source of flow requests.
- In case NFV is part of the infrastructure, the security controller can instruct the SDN controller through its North-bound interface to send instructions to the infrastructure layer to reroute specific traffic through a traffic shaper.

Altogether, the security solution consists of the modified data-bus, the security controller, and the interaction between the security controller, as well as the SDN controller and the modified data-bus. The task of the security controller is to detect anomalies and to react on them by applying specific mitigation actions. The question then becomes how this security controller detects the anomalies and on what logic the specific mitigation actions are applied, this question is addressed next. The idea is to use specific match fields from the OpenFlow Switch Specification [8] in order to avoid any changes at the South-bound protocol level.

According to the OpenFlow Switch specification, each flow table entry contains the following:

1. Match (e.g. Header) fields to match against packets.

2. Counters to update for matching packets.
3. Actions to apply to matching packets.

The following match fields from the OpenFlow Switch Specification [8] are used to determine packet anomalies. The input switch port is mandatory and must be combined with the Ethernet Source Address and the IP Source Address.

- Input Switch Port - in_port
- Ethernet Source Address - ether_src
- IP Source Address - ip_src

After the match fields have been determined, the next step is to address the internal logic of the security controller. The focus is on mitigating the effect of massive numbers of asynchronous messages from the infrastructure layer to the control layer. Asynchronous messages are messages sent from the infrastructure layer to the control layer without the controller soliciting them from the infrastructure layer. The initiator of a packet-in type packet is the SDN switch. Every packet which does not have a packet-in header is passed directly and unmatched from the security controller to the SDN controller. A traffic anomaly is defined as a certain traffic rate of asynchronous messages in a certain time interval.

The following flowchart, figure 4.3, depicts the logic of determining and acting on traffic anomalies.

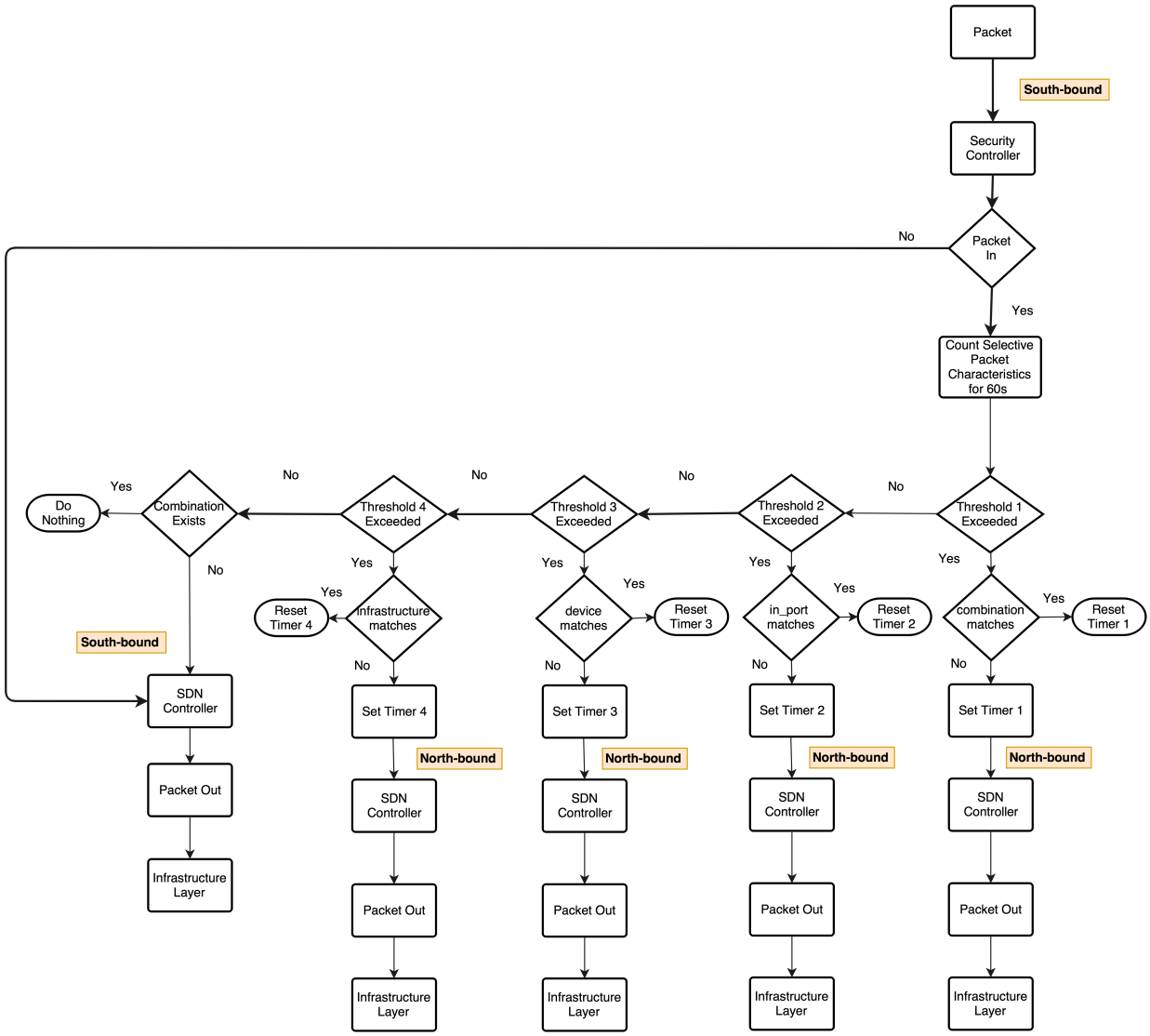

Fig. 4.3: Security Solution Logic

For a time-interval of 60 seconds the security controller counts all combinations of ether_src plus in_port and ip_src plus in_port. The security controller determines the following:

1. For a specific device, the combination of ether_src plus in_port or ip_src plus in_port is greater than the threshold. This will be noted as scenario 1.
2. For a specific device, the amount of asynchronous packet-in packets for a specific port is greater than the threshold. This will be noted as scenario 2.
3. For a specific device, the total number of asynchronous packet-in packets across all in_ports is greater than the threshold. This will be noted as scenario 3.

4. For a specific set of devices under control of the SDN controller, the total number of asynchronous packet-in packets is greater than the threshold. This will be noted as scenario 4.
5. The asynchronous packet-in packet is not part of scenario's 1 through 4.
6. The packet is not a packet-in type packet.

The security controller, therefore, recognises five different scenarios' when an asynchronous packet-in packet is received. A separate set of counters, thresholds and timers exists for each of the first four scenarios'. Each individual timer is used as a safety measure to minimise the possibility that the security controller sends redundant instructions to the SDN controller. As such, the attack surface where the security controller itself could be misused to send vast amounts of redundant instructions to the SDN controller is minimised. When a packet received is not a packet-in type packet, the packet is passed directly to the SDN controller on its South-bound interface.

The following is a deep dive into the security solution logic.

1. Each timer is set for a specific scenario and is denoted as $T_{1\text{-}4}$.
2. Each threshold is set for a specific scenario and is denoted as $Trh_{1\text{-}4}$
3. A specific threshold is based on packets coming from a $device_i$ on $port_j$.
4. The specific packet is from $source_s$ and $port_j$.
5. A packet is evaluated as $source_{sij}$ which is the $source_s$ of the traffic in respect to the $port_j$ of a specific $device_i$.

The thresholds are set by the network programmer depending on the capacity of the given SDN network infrastructure. The formulas, 4.1 through 4.4, used to attest if a given threshold has been reached, are kept basically the same for all thresholds to ensure consistency through the entire security solution.

1 The calculation of the current counter value for threshold 1 is calculated as the sum of $source_{si}$ for one specific $source_s$ on one specific $device_i$.

$$Trh_1 = \sum source_{si} \tag{4.1}$$

2 The calculation of the current counter value for threshold 2 is computed as the sum of $source_{sij}$ for one specific $device_i$ on one specific $port_j$

$$Trh_2 = \sum source_{sij} \tag{4.2}$$

where $_{ij}$ are static.

3 The calculation of the current counter value for threshold 3 is computed as the sum of $source_{sij}$ for one specific $device_i$.

$$Trh_3 = \sum source_{sij} \tag{4.3}$$

where $_i$ is static.

4 The calculation of the current counter value for threshold 4 is computed as the sum of $source_{sij}$ for a set of $device_i$ where i has a range from 1 to N and N is the number of SDN switches making a particular SDN infrastructure.

$$Trh_4 = \sum source_{sij} \tag{4.4}$$

where $_i$ is static.

A timer is set when one of the Thresholds is exceeded in a 60-second interval. Since we have four thresholds, four timers are needed $T_{1\text{-}4}$. There are four timers types. For each timer type, however, there could be multiple instances depending on how many exceeded thresholds per threshold type are recorded in a particular timeframe. Therefore, there are also multiple instances per threshold type. The thresholds, counters, and timers consist of an array of thresholds, counters, and timers per type.

Since the security solution does not block any packet-in packets originating from the infrastructure layer, a DoS attack is not stopped but only mitigated. Since the DoS attack is not stopped, the same packet or combination of packets which caused a specific threshold and timer to be set, can pass multiple times through the security controller in the case of a sustained DoS attack. In such a case, the security controller is wasting resources to reapply the specific mitigation action as long as the threshold is being exceeded. This inefficient action can be prevented by means of a timer reset mechanism. The logic of the timer reset mechanism is explained using the following simplified use-case.

Suppose that a certain threshold is set to 600 packets-in packet type for an interval of 60 seconds. This means that on average 10 packets-in packet type are allowed per second. Packet-in packet type bursts are allowed as long as the threshold of 600 packets-in type packet per 60-second interval is not exceeded. Next, suppose that the packet-in packet type exceeds the 600 packets-in type packet per 60-second interval. The next step is to reprogram the output port for the specific packet-in type packets such that the packet-shaper is logically placed between the infrastructure layer and the control layer for the specific packet-in packet type traffic. Take note that this can be applied for any of the four scenarios discussed earlier. The traffic shaper is set such that the specific packet-in packet type threshold + 1 is enforced for

the 60-second interval. Just as with the scenario without a traffic shaper, packet-in type packet bursts are allowed if the limit of 601 packets per 60-second interval period is not exceeded. If this limit is exceeded, the traffic shaper will buffer the exceeded packet-in type packets for the next 60-second interval. When the traffic shaper limit of 601 packet-in type packet per 60-second interval is exceeded, the 600 packet-in type packet on the security controller is also exceeded by exactly 1 packet. This informs the security controller that the condition for mitigating the specific DoS attack still applies. Depending on the specific matched $Trh_{1\text{-}4}$, the corresponding $T_{1\text{-}4}$ will be reset. The mechanism results in a continuous real-time monitoring of the DoS attack in 60-second intervals even after the mitigation has been enforced.

In the case of scenario 5, no anomaly has been detected. In this case the security controller passes the unmodified flow-request (packet-in) to the SDN controller via the South-bound interface of the SDN controller.

To determine to what extent the proposed security control compares to the existent security controls on DoS attacks on the SDN South-bound interface, the following table is presented. Since the table consists of security controls and the SBSTE-attack, a distinction is made to denote with an "x" if a security solution utilises a security control. The effect of the security solution on the SBSTE-attack is denoted with a "+" for satisfactory, a "-" for insufficient and a "+/-" for weak.

Author	SD	AUD	POL	AUG	MON	RED	SBSTE
Shin [9]	-	-	-	-	x	-	-
Fonseca [14]	-	-	-	-	-	x	-
Yao [15]	-	-	-	-	x	-	-
Naous [16]	-	-	x	-	-	-	-
Porras [17]	-	-	x	-	-	-	-
Alsmadi [2]	-	-	x	-	x	-	+/-
Hussein [18]	-	-	-	-	x	-	-
Li [5]	-	-	x	-	-	-	-
Nayak [19]	-	-	x	-	-	-	-
Jankok	x	-	x	x	x	-	+

Tab. 4.1: Proposed security solution versus existing security solutions against SDN DoS attacks, identified by the author of the scientific publication and classified by the underlying basic security controls by Akhunzada [10] with the resulting applicability on the SBSTE-attack.

4.5 Summary

The security solution proposal presents a new way of thinking about the control layer by advocating that the control layer must consist of a SDN controller and a security controller. Contrary to the security controls addressed by Alsmadi, no effort is placed on the mitigation of data leaking (timing-attack) in which the attacker can deduct if SDN is being used in a given network infrastructure. Such a mitigation solution not only requires an adjustment in the fundamentals of SDN behaviour but is also of little use when the attacker has other ways to find out if SDN is being used in a particular network infrastructure.

The presented security solution relies on a half-duplex construction with the infrastructure layer together with a set of thresholds, counters and timers. In order to increase the implementation feasibility of the security solution, the security solution does not require any change in the South-bound protocol or in the fundamental working of the SDN controller The South-bound protocol behaviour and the SDN controller behaviour does not need to be adjusted in order to implement the security solution. Since the security controller does not change the behaviour of the OpenFlow South-bound protocol, it is completely transparent to the SDN controller. This security solution, therefore, presents a non-disruptive proposal in mitigating the SBSTE-attack on the SDN South-bound interface.

5 Security Solution Evaluation

> *Users do not care about what is inside the box, as long as the box does what they need done.*
>
> — **Jef Raskin**
> about Human Computer Interfaces

Applying Hevner's design research method, this chapter evaluates the proposed security solution in chapter 4. The evaluation is conducted in accordance with section 2.1 by addressing the following items:

1. Context & Solution proposal
2. Effects satisfy requirements
3. Sensitivity for different context

5.1 Context & Solution Proposal

The context is a network infrastructure in which only OpenFlow enabled switches are used in the infrastructure layer. The network programmer has not programmed the OpenFlow switches to discard packages for which no flow-table exists in the infrastructure layer. There is no mechanism to filter packet-in type packets from the infrastructure layer to the control layer. There is also no effort to mitigate the data-leaking timing attack. In this particular context, an attacker can perform a resource starvation attack (SBSTE-attack) on the South-bound interface without any mitigation mechanism available in the SDN infrastructure to detect and mitigate this attack. As mentioned before, the aim is not to mitigate the data-leaking timing attack in which the attacker deducts if a particular network infrastructure uses SDN, and which flow requests will result in packet-in type packets from the infrastructure layer to the control layer. Nor is the focus to stop the packet-in type packets. Rather, the overall approach is to mitigate the effect such packets have on the South-bound interface of the SDN controller. The security solution proposal focuses on mitigating the effect of the packet-in type packets on the South-bound interface of the control layer. The security solution proposal presents a mitigation solution on a conceptual level and does not include evaluation within an actual SDN infrastructure.

Through a data-leaking timing attack, the attacker knows which packets requires control-layer consultation. The attacker next computes a vast number of packets with similar characteristics to the packets that the infrastructure layer does not know how to handle directly and therefore needs to consult the control layer. The attacker sends all these packets to the infrastructure layer. The result of this is that the infrastructure layer will send vast numbers of asynchronous packets to the control layer requesting flow information. This attack can result in the saturation of the South-bound interface of the control layer and also to the South-bound interface of the specific network device(s) in the infrastructure layer.

In this specific type of attack, the attacker uses one or more devices in the infrastructure layer to direct the attack to the South-bound interface of the control layer. The specific mitigation is to reduce the resource consumption of the asynchronous packets on the South-bound interface of the control layer.

5.1.1 Attack Tree

In addition to the context and solution proposal, the following attack tree in figure 5.1 presents a staged attack in which the attacker progressively tries to oversubscribe the South-bound interface of the control layer.

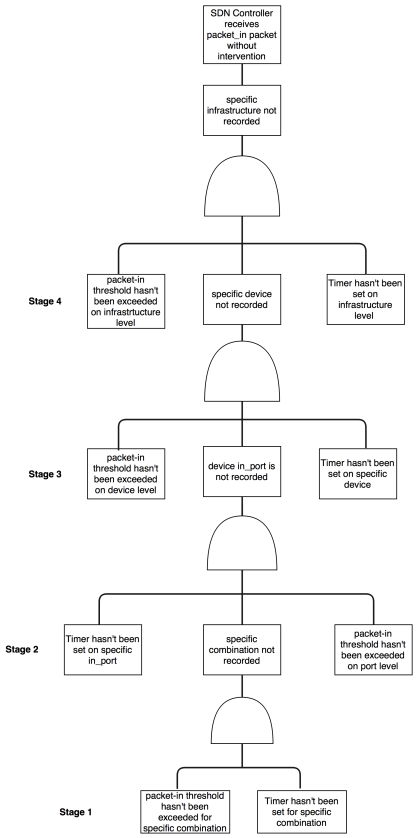

Fig. 5.1: Attack Tree

In the first stage, the attacker is not faking his source MAC or IP addresses and tries to overwhelm the control layer via the infrastructure layer by extrapolating various destinations for which the infrastructure layer does not have a matching flow-table. The attack gets mitigated by the first stage mitigation due to the specific combination of source MAC and IP addresses exceeding the specific threshold. The mitigation will only be applied for the packet-in packets for the specific combination of the source MAC or IP address.

The attacker moves to the second level attack in which he does not want his source MAC or IP in combination with a specific device input port to be recorded and mitigated. The attacker therefore spoofs many source MAC and IP addresses with the sole purpose of getting passed through the first phase of mitigation. Having passed through the first phase of mitigation, the attacker can eventually reach the device in_port threshold. At that point, the attack is mitigated at this second stage. The mitigation will only be applied for the packet-in packets for the specific in_port pertaining to the specific network device.

The attacker recognizes the mitigation and moves to the third level of attack in which he still spoofs many source MAC and IP addresses, but now instead of using one specific input port, he uses several input ports and also assures that the attack per input port is beneath the input port threshold. Now the attack passes through the first two stages. Eventually, the attacker will reach the device threshold. At that point the attack is mitigated at this third stage. The mitigation will be applied for all packet-in packets for the specific device.

The attacker recognizes the mitigation and moves to the fourth level of attack in which he combines the attack of the third level and distributes this among different network devices in the infrastructure layer. The attacker further assures that the device threshold is not reached. Now the attack passes the first three stages. Eventually, the attacker reaches the infrastructure threshold for packet-in packets. At that point the attack is mitigated by applying the mitigation to all packet-in packets for a random set of devices. The devices are randomly selected to reduce the chance of the attacker deducting which devices (Switches) will be mitigated and by this using this information for other malicious types of actions.

5.2 Discussion

Without the security solution in place, the attacker could cause DoS attacks against the South-bound interface by causing network devices in the infrastructure layer to generate massive numbers of asynchronous packages to the control layer. By

oversubscribing the control layer, the controller will eventually be unable to process packet-in packets or to send out packet-out packets. The result would be that the network infrastructure dependent on the particular SDN controller would become demoted to a legacy network infrastructure with even less decision capability, finally resulting in a disrupted SDN infrastructure.

5.2.1 Effects Satisfy Requirements

As figure 5.1 depicts, the security solution proposal increases the number of steps the attacker must undertake to get a packet-in type packet to the SDN controller without this attack being mitigated. These steps are additional effort that the attacker must put into orchestrating his attack and can be seen as a significant cost increase of the attack. Therefore, the security solution proposal increases the cost of performing the SBSTE-attack successfully. As table 4.1 and section 4.4 outline, the security solution proposal satisfies to the requirements stated in section 4.2.2:

1. Applicable in existing SDN-implementations
    - The aim is to present a security solution proposal which does not require fundamental changes in the functioning of the South-bound protocol or on the interaction between the control layer and the infrastructure layer. As such, the aim is to propose a security control mechanism which does not disrupt the fundamental working of the OpenFlow South-bound protocol.
2. Minimise the resulting attack surface
    - A second aim is that the security control proposal must itself not introduce new attack vectors from the South, West or North-bound interfaces.

Therefore, it can be concluded that the security solution proposal satisfies to the given requirements and that the effects satisfy the requirements.

5.2.2 Sensitivity for Different Context

The security solution proposal is only effective in the particular context where the SDN infrastructure does not consists of hybrid SDN devices which can revert to classical decentral switching. The security solution would be redundant in the particular context where the network programmer has configured the SDN switches to discard all packet-in type packets with unknown flow-tables in the infrastructure layer. The security solution proposal depends on augmentation to be effective. In particular, the security solution depends on a traffic shaper (NFV) to comply to

the requirement of not dropping packet-in packets on the infrastructure layer. This dependency is inherent to the inclusion of SDN security augmentation.

Thus, this chapter address a third research question:

1. What new security solution can contribute to the mitigation of the SBSTE-attack against the SDN South-bound interface?

The proposed security solution introduces a security controller in the control layer and introduced a half-duplex solution alongside a modified data-bus in the control layer. An important aspect of the solution is the use of timers and the use of Network Function Virtualisation. The proposed security control is a contribution to the mitigation of SBSTE-attacks against the SDN South-bound interface.

5.3 Future Work

The proposed mechanism presents a security solution for mitigating the SBSTE-attack against the South-bound interface of the control layer. In this research a simplified count mechanism has been used together with thresholds set by the network programmer. A more elaborate mechanism could be possible by for instance implementing a self-learning system based on machine learning. Thus, the mechanism for detecting traffic anomalies in asynchronous packet-flows could be an interesting area for further research. The proposed control mechanism also does not provide a mitigation at the North-bound interface of the control layer. Thus, research on how to expand the security control proposal to include mitigation for compromised applications in the application layer could be an important area for future research.

The proposed security solution presents a solution for the SBSE-attack on a conceptual level. Further research is also needed to attest (in an experimental environment) if the proposed security solution proves to be as effective as table 5.1 suggests. In order to implement the proposed security solution in practice, chapter 4 of this study presents the logic and parameters that the proposed security solution needs to implement.

6 Conclusion

> *Nature shows us only the tail of the lion. But there is no doubt in my mind that the lion belongs with it even if he cannot reveal himself to the eye all at once because of his huge dimension.*
>
> **— Albert Einstein**

The SBSTE-attack which is conducted against the South-bound interface of the control layer, poses a realistic use-case for resource starvation attacks against the SDN controller. The effect of such attacks is the disruption of packet forwarding functionality in the infrastructure layer which depends on the respective SDN controller under attack. The SBSTE-attack has been proven to be effective with the attacker using moderate hardware configuration. Without a solution for this type of DoS attack against the SDN controller, the SDN controller is considered to be a liability to network security.

From the evaluation of the related work on security solutions in chapter 3 for mitigating the SBSTE-attack, it becomes apparent that none of the reviewed security solutions provide an adequate method for mitigating the SBSTE-attack on the SDN South-bound interface. None of the security solutions provides a combination of the following security controls:

1. Secure design of SDN
2. Security Enforcement Policy
3. SDN Security Augmentation
4. Security Monitoring and Analysis

Therefore, this study proposes a new security solution based on the following principles:

1. The security solution must provide a solution for mitigating the SBSTE-attack by using a combination of the following security controls: Secure design of SDN, Security Enforcement Policy, SDN Security Augmentation, and Security Monitoring and Analysis.

2. The security solution must adhere as closely as possible to current SDN technology by not changing the fundamental working of the South-bound protocol, and by utilising existing traffic parameters of the South-bound protocol.

Table 6.1 consists of security controls and the SBSTE-attack, therefore, a distinction is made to denote with an "x" if a security solution utilises a security control. The effect of the security solution on the SBSTE-attack is denoted with a "+" for satisfactory, a "-" for insufficient and a "+/-" for weak. Table 6.1 highlights how the proposed security solution relates to the current reviewed security solutions.

Author	SD	AUD	POL	AUG	MON	RED	SBSTE
Shin [9]	-	-	-	-	x	-	-
Fonseca [14]	-	-	-	-	-	x	-
Yao [15]	-	-	-	-	x	-	-
Naous [16]	-	-	x	-	-	-	-
Porras [17]	-	-	x	-	-	-	-
Alsmadi [2]	-	-	x	-	x	-	+/-
Hussein [18]	-	-	-	-	x	-	-
Li [5]	-	-	x	-	-	-	-
Nayak [19]	-	-	x	-	-	-	-
Jankok	x	-	x	x	x	-	+

Tab. 6.1: Proposed security solution versus existing security solutions against SDN DoS attacks, identified by the author of the scientific publication and classified by the underlying basic security controls by Akhunzada [10] with the resulting applicability on the SBSTE-attack.

The proposed security solution provides mitigation against the SBSTE-attack as depicted into the following attack-tree.

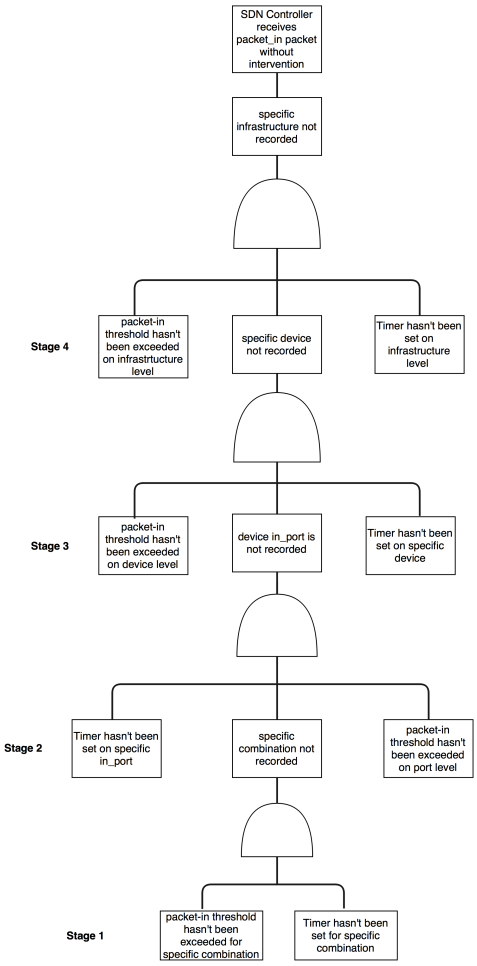

Fig. 6.1: Attack Tree

Each stage presents a set of conditions to which the attack must comply in order to move to the next stage. Each stage can be interpreted as a cost increase from the attacker's point of view and therefore functions as mitigation against the SBSTE-attack. As figure 6.1 outlines, the security solution provides a staged mechanism for mitigating the SBSTE-attack on the South-bound interface of the control layer.

The security solution proposal is dependent on a traffic shaper based on Network Function Virtualisation (NFV) principles. This dependency is inherent to SDN security augmentation. The security solution is only effective in the context of a native SDN infrastructure without hybrid OpenFlow SDN switches. The security solution proposal is not needed in the case where the OpenFlow SDN switches have been programmed to discard packets for which there are no matching flow-table in the infrastructure layer. However, discarding packets for which there are no matching flow-table in the infrastructure layer, would be limiting the dynamic

functionality of SDN and by this, limiting the full capabilities of SDN. The proposed security solution is designed not to limit the capabilities of SDN, and not to require a redesign of the South-bound protocol. Instead, the proposed security solution incorporates existing SDN security features, uses existing packet parameters from the OpenFlow South-bound protocol, and is placed in the control layer where it transparently communicates with the SDN controller on its South-bound and North-bound interface.

Bibliography

[1]L. Stevens and F. C. Byrnes, „Assessing risks using gartner risk assessment methodology“, *Gartner*, 2008 (cit. on p. 1).

[2]I. Alsmadi and D. Xu, „Security of software defined networks: A survey“, *Computers & Security*, vol. 53, pp. 79–108, 2015 (cit. on pp. 2, 3, 17, 19, 33, 42).

[3]W. Braun and M. Menth, „Software-defined networking using openflow: Protocols, applications and architectural design choices“, *Future Internet*, vol. 6, no. 2, pp. 302–336, 2014 (cit. on p. 3).

[4]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 (cit. on pp. 3, 6, 15).

[5]W. Li, W. Meng, and L. F. Kwok, „A survey on openflow-based software defined networks: Security challenges and countermeasures“, *Journal of Network and Computer Applications*, vol. 68, pp. 126–139, 2016 (cit. on pp. 3, 18, 19, 33, 42).

[6]M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, „Software defined networking: State of the art and research challenges“, *Computer Networks*, vol. 72, pp. 74–98, 2014 (cit. on p. 4).

[7]A. O. Oluwasogo and O. Oluwaseyitan, „A security architecture for software defined networks (sdn)“, *International Journal of Computer Science and Information Security*, vol. 13, no. 7, 2015 (cit. on p. 4).

[8]O. N. Foundation, „Openflow switch specification“, 2011 (cit. on pp. 5, 28, 29).

[9]S. Shin, V. Yegneswaran, P. Porras, and G. Gu, „Avant-guard“, pp. 413–424, 2013 (cit. on pp. 6, 7, 16, 19, 33, 42).

[10]A. Akhunzada, A. Gani, N. B. Anuar, *et al.*, „Secure and dependable software defined networks“, *Journal of Network and Computer Applications*, vol. 61, pp. 199–221, 2016 (cit. on pp. 9, 15, 19, 20, 33, 42).

[11]R. Wieringa, „Introduction to design science methodology“, 2013 (cit. on pp. 9, 25).

[12]A. R. Hevner, S. Ram, S. T. March, and J. Park, „Design science in information systems research1“, *MIS Quarterly*, vol. 28, no. 1, pp. 75–105, 2004 (cit. on p. 9).

[13]C. Yoon, T. Parka, S. Leea, *et al.*, „Enabling security functions with sdn: A feasibility study“, *Computer Networks*, vol. 85, pp. 19–35, 2015 (cit. on p. 15).

[14]P. Fonseca, R. Bennesby, E. Mota, and A. Passito, „A replication component for resilient openflow-based networking“, *IEEE Network Operations and Management Symposium*, 2012 (cit. on pp. 16, 19, 33, 42).

[15]G. Yao, J. Bi, and P. Xiao, „Source address validation solution with openflow/nox architecture“, *IEEE International Conference on Network Protocols*, vol. 19, 2011 (cit. on pp. 17, 19, 33, 42).

[16]J. Naous, R. Stutsman, D. Maziéres, N. McKeown, and N. Zeldovich, „Delegating network security with more information“, *WREN*, 2009 (cit. on pp. 17, 19, 33, 42).

[17]P. Porras, S. Shin, V. Yegneswaran, *et al.*, „A security enforcement kernel for openflow networks“, *HotSDN*, 2012 (cit. on pp. 17, 19, 33, 42).

[18]A. Hussein, I. H. Elhajj, A. Chehab, and A. Kayssi, „Sdn security plane: An architecture for resilient security services“, pp. 54–59, 2016 (cit. on pp. 18, 19, 33, 42).

[19]A. Nayak, A. Reimers, N. Feamster, and R. Clark, „Resonance: Dynamic access control for enterprise networks“, *WREN*, 2009 (cit. on pp. 18, 19, 33, 42).

List of Figures

List of Tables