UNIVERSITY OF TWENTE.

MASTER THESIS

A Signature-Based Approach to DDoS Attack Mitigation Using BGP Flowspec Rules

Author
Joeri KOCK

Examination board
dr. J.J. Cardoso De SANTANNA
dr. A. PETER
Vincent DE JAGER

November 20, 2019

Abstract

Distributed Denial-of-Service (DDoS) attacks aim to to prevent the legitimate use of a service. Since the magnitude and frequency of these attacks are increasing, DDoS attacks are becoming an increasingly bigger problem for the Internet. BGP Flowspec is an extension to the Border Gateway Protocol (BGP), designed to provide a granular approach to DDoS mitigation. BGP Flowspec defines a network flow based on e.g. the source, destination and other packet specific information. This flow can be matched dynamically to incoming traffic in order to either drop the traffic, place it into a different forwarding instance for further examination, or police to a desired rate. Related research shows its potential for DDoS attack mitigation. However, BGP Flowspec lacks in precision, potentially resulting in the filtering of legitimate traffic. This could have a negative impact on the underlying network. Therefore, a minimization and maximization problem arises: on one hand, it is desirable to maximize the amount of DDoS traffic blocked. On the other hand, the negative impact on the network needs to be minimized. The goal of this research is to address this problem by investigating how DDoS attack mitigation can be improved by using BGP Flowspec. This research presents a methodology for generating BGP Flowspec rules using a signature-based approach, as well as an evaluation of these rules. This evaluation showed that many DDoS attacks can be effectively mitigated using this approach. However, some DDoS attacks are too generic to be mitigated using BGP Flowspec. Since DDoS attacks are very different from each other, a generic solution is very challenging to design.

Keywords – DDoS, Mitigation, BGP Flowspec, Flow Specification, Mitigation Impact, Self-adaptive, Network Operator

Contents

Chapter 1

Introduction

Over the years, Distributed Denial-of-Service (DDoS) attacks are becoming an increasingly bigger threat for the Internet. An attacker uses thousands of arbitrary hosts (usually with little to no security) and installs software on these hosts in order to utilize them for attacking a target system. DDoS attacks are evolving quickly and becoming more complex. The largest DDoS attack to date targeted GitHub with a traffic rate of 1.3 terabytes per second (Tbps), sending packets at a rate of 126.9 million per second [20, 2]. This increment in capacity and complexity makes it increasingly difficult to mitigate these attacks. Almost half (45%) of companies experience DDoS attacks nowadays, with more than 90% of those companies having experienced one in the past 12 months [45]. Defending against DDoS attacks is particularly challenging, since they do not exploit a specific vulnerability in a system. Instead, they exploit the very fact that the system is connected to the Internet, as well as the fact that the capacity of systems is always finite and expensive. Though the victim may have proper security methods installed, there is often little that can be done.

A simplified figure illustrating a DDoS attack can be seen in Figure 1.1. Note that an attacker usually has a set of machines commonly referred to as "command & control". These machines are responsible for directing the bots, which are the machines on which the malware is installed. The set of command & control machines and bots combined are called a botnet. This botnet is used to generate a large quantity of Internet traffic, which is then routed to the victim.

A successful DDoS attack negatively impacts an organization's reputation, in addition to damaging existing client relationships. Significant financial losses can amount to $40,000 per hour for major enterprises [45]. With 50% of DDoS attacks lasting between 6-24 hours, the average DDoS cost can be assessed at about $500,000 — with some running significantly higher. Costs are not limited to the IT department however; they also have a large impact on units such as security and risk management, customer service, and sales. Examples of other consequences are (1) severe impact on eCommerce, resulting in substantial revenue loss, (2) inflated IT costs from Internet Service Providers (ISPs) and Infrastructure as a Service (IaaS) providers for bandwidth overages, or computing power, (3) short or long term damage to online reputation for critical services (governments, trading platforms, financial services, health care, etc.) and (4) forcing the IT staff to focus on the DDoS attack acting as a "smokescreen", while the bad actors are exfiltrating data from target systems [21].

Several systems and methods have been proposed over the years in order to mitigate DDoS attacks. The mitigation of incoming DDoS traffic can be done on multiple stages during the traffic's route. There are 7 of these stages where DDoS attacks could be prevented or mitigated. These

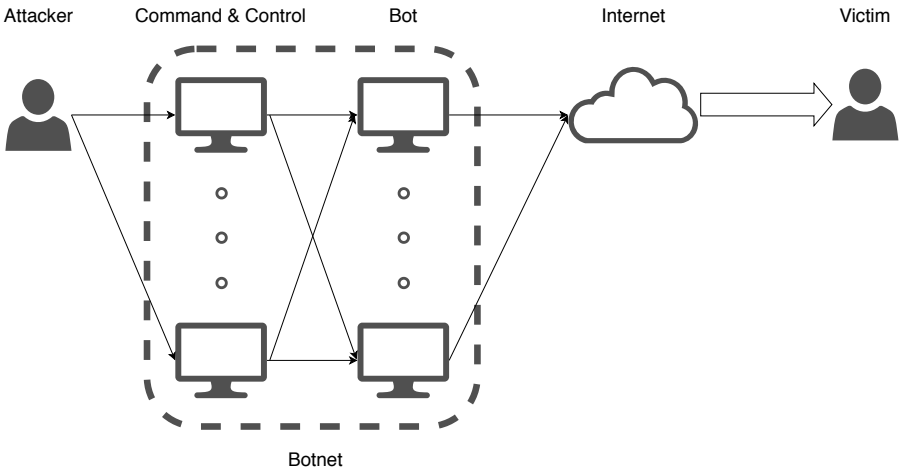

Figure 1.1: An illustration of a DDoS attack.

stages are the attacker, the botnet, the reflector, the Internet Exchange Point (IXP), the Internet Service Provider (ISP), the organization and the target machine itself. Examples of mitigation methods are firewalls, Intrusion Detection Systems (IDS) such as Snort [63], Suricata [54], Bro [57] and IBM QRadar [33] or by using a Web Application Firewall (WAF) [81]. In addition, rather than scanning incoming network traffic, some tools can inspect the the internals of the host's machine. Examples of such systems are OSSEC [56], AIDE [41] and Samhain [65]. These methods all apply mitigation at different stages of the packet's route.

At the IXP and ISP stage, DDoS traffic can be mitigated as well. During these stages, the Border Gateway Protocol (BGP) is used for packet routing between large networks. BGP Flowspec [44] is an extension for BGP that can filter incoming network traffic at this level. BGP Flowspec supports 12 fields from the Network and Transport layer of the Internet Protocol (IP). These fields are used to define a "Flow Specification" (i.e. rules for incoming traffic) and an action for this traffic (e.g. discard or redirect the traffic). BGP Flowspec rules can be generated at the consumer level (where the DDoS attack is detected) and sent to the ISP, where the rules are installed on the network's edge routers. When incoming traffic satisfies the flow specification, it is discarded, redirected or policed at an established rate.

The advantages of BGP Flowspec are that it can be much more precise in blocking DDoS traffic compared to methods such as black holing [76]. Furthermore, routers that use BGP are responsible for handling large amounts of network traffic. This results in BGP Flowspec having a large throughput capability. However, since there are only 12 fields that can be used for traffic filtering, BGP Flowspec is probably not as accurate in mitigating DDoS traffic as systems that filter traffic on e.g. the Application Level. This inaccuracy can lead to BGP Flowspec blocking legitimate traffic as well as DDoS traffic, which will have a negative impact on the network. Henceforth, the negative impact on the underlying network caused by BGP Flowspec traffic filtering will be referred to as "mitigation impact". One of the core challenges of this research is to find a way to quantify this impact. As a result of the mitigation impact, a minimization and maximization problem arises. On one hand, it is desirable to block as much DDoS traffic as possible. Contrarily, the mitigation impact on the network needs to be minimized.

The goal of this research is to address this minimization and maximization problem by evaluating how effective BGP Flowspec rules are for DDoS attack mitigation. In order to accomplish this goal, we split our investigation in the following Research Questions:

- **RQ1:** How does BGP Flowspec theoretically compare to existing DDoS mitigation solutions?
- **RQ2:** How can we write BGP Flowspec rules based on known DDoS attacks?
- **RQ3:** How effective are our BGP Flowspec rules for DDoS attack mitigation?

BGP Flowspec was originally designed for DDoS mitigation, aimed at providing a granular approach at filtering traffic at this level. Literature shows that BGP Flowspec has a high potential in effectively filtering DDoS traffic at the ISP and IXP level [32]. However, it is currently not used by ISPs for mitigating DDoS attacks [64]. The **contribution of this research** is two-fold: firstly, this research will provide a methodology for generating BGP Flowspec rules using a signature-based approach, i.e. generate rules based on existing DDoS attack data. To the best of our knowledge, no tools exist that use existing DDoS attacks in order to generate BGP Flowspec rules. Secondly, no research exists on evaluating the effectiveness of BGP Flowspec for DDoS attack mitigation. In this research, we will evaluate our generated rules as well as quantify the mitigation impact on the underlying network.

The rest of this document is structured as follows: the topic of DDoS attacks will be described in chapter 2. Here, we will elaborate on DDoS attacks, the methods of mitigating DDoS attacks, introduce the topic of BGP Flowspec and answer RQ1. Next, in chapter 3, we will describe our approach to generating BGP Flowspec rules using known DDoS attack data and answer RQ2. Subsequently, in chapter 4, we will describe the methodology for evaluating our BGP Flowspec rules, as well as present the results of this evaluation. In chapter 5, we will provide a conclusion by answering RQ3, explain the contribution of this research and elaborate on future work.

Chapter 2

State-of-the-art on DDoS Attacks

Recall research question 1:

How does BGP Flowspec theoretically compare to existing DDoS mitigation solutions?

In order to properly answer this research question, it is necessary to have an elaborate knowledge of the DDoS field. This knowledge is essential to understanding the various mitigation techniques, which will be explained in chapter 2.2. Therefore, the goal of this chapter is to provide that knowledge. We will achieve this goal by dividing this chapter into the following parts. We will first discuss the background of DDoS attacks in 2.1. Next, we will elaborate on classifying all DDoS attacks into 2 different types, as well as describe the 10 most commonly used DDoS attacks in 2.1.1. After that, in 2.1.2 we will mention the causes for DDoS attacks existing and in 2.1.3 the motivation that attackers have when performing a DDoS attack. We will close this chapter with some concluding remarks.

2.1 Background

A denial-of-service attack (DoS) is characterized by an explicit attempt to prevent the legitimate use of a service [47]. In addition, a distributed DoS attack (DDoS) makes use of multiple attack hosts in order to attain this goal. As DDoS is a far bigger threat to the Internet at the moment of writing this document, the focus will be on DDoS attacks.

Usually, when a target system experiences a DDoS attack, an attacker uses an Internet connection to flood the target with e.g. TCP or UDP packets. In order to prevent the target from using their legitimate services, these packets are sent in large quantities in a short period of time, overloading e.g. the target's bandwidth. As a result, the target system is inaccessible, making it impossible to host other services. The attacker often uses thousands of computers distributed around the world to further amplify the magnitude of the attack. The set of hosts used to perform the attack is known as a Botnet [1]. Devices in a botnet are usually computers with little to no security, where DDoS malware can be installed relatively easily.

2.1.1 Types of DDoS attacks

Here, we will explain the 2 types in which DDoS attacks can be classified. Additionally, the most commonly used DDoS attacks will be listed, as well as the attack types they belong to.

Broadly speaking, DoS and DDoS attacks can be divided into two types [35]:

- **Volumetric Attacks** - Commonly referred to as brute-force attacks, a volumetric attack sends a high amount of traffic, or request packets, to a targeted network in an effort to overwhelm its bandwidth capabilities [71]. Opposed to semantic attacks, volumetric attacks are usually much more difficult to mitigate, since they abuse legitimate services. This means filtering would also affect legitimate traffic, resulting in a mitigation impact. Furthermore, in many cases, the target's resources are limited, rendering it impossible for the target to do anything [47].
- **Semantic Attacks** - Rather than aiming to exhaust the target's bandwidth, semantic attacks exploit a specific feature or bug at the victim's machine. These attacks can usually be somewhat mitigated by the victim by modifying the abused protocols or deploying network traffic filtering. Semantic attacks don't need to generate as much traffic as volumetric attacks in order to inflict damage.

As Mirkovic and Reiher [47] state: "Countering semantic attacks by modifying the deployed protocol or application pushes the corresponding attack mechanism into the brute-force category. For example, if the victim deploys TCP SYN cookies to combat TCP SYN attacks, it will still be vulnerable to TCP SYN attacks that generate more requests than its network can accommodate." For this reason, many DDoS attacks that are relatively devastating to a victim are examples of volumetric attacks.

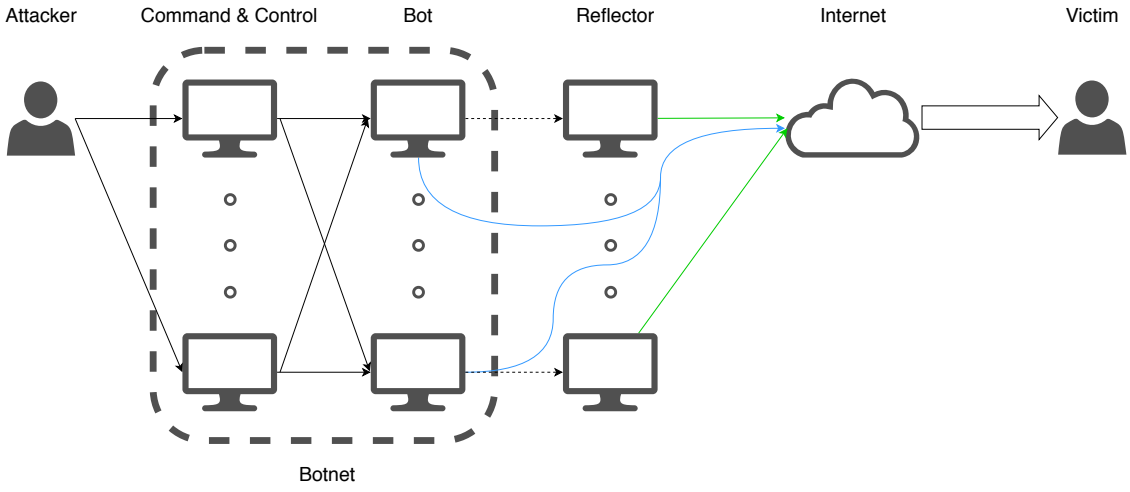

Figure 2.1: The difference between the 2 types of DDoS attacks emphasized.

Recall the overview of a DDoS attack from Figure 1.1 from the previous chapter. In Figure 2.1, we have expanded that picture in order to illustrate the difference between a volumetric and a semantic DDoS attack. The blue lines illustrate a volumetric attack, with the traffic coming directly from the bots. The green lines illustrate a semantic attack, where a set of reflectors is used.

In their report "2019 State of the Internet / Security: DDoS and Application Attacks", Akamai [2] makes the distinction between DDoS attacks that are launched by a botnet or with reflection methods. A reflection attack is when the reply is sent back to the claimed origin of the request. With a spoofed source IP, the attacker can make the reflecting server send the reply to the selected victim. This distinction can be compared to the aforementioned distinction between volumetric and semantic attacks, with them corresponding to botnet and reflection attacks respectively.

In the remaining part of this chapter, we will elaborate on the 10 most commonly used DDoS attacks [6], as well as classify each one as either volumetric or semantic.

- **Memcached DDoS Attack**
  A memcached attack is a volumetric DDoS attack. A memcached server is a server with a caching system for databases in order to speed up websites and networks. If these servers are vulnerable, attackers can abuse them by sending spoofed requests with the target's IP in the header, to which the memcached server will respond. The server will send a much larger amount of data back to the target [15].
- **NTP Amplification Attack**
  An NTP amplification attack is a volumetric DDoS attack. In a Network Time Protocol (NTP) amplification attack, an attacker uses the functionality of an NTP server in order to send traffic to a target. More specifically, the attacker sends a request (with a spoofed IP address of the target) to the NTP server, in which he requests a list. The server will respond by sending the list to the spoofed IP address. This way, the size of the response from the server is much larger than the original request [16].
- **DNS Amplification Attack**
  A DNS amplification attack is a volumetric DDoS attack. A DNS amplification attack makes use of a Domain Name System (DNS) server. It is another reflection attack. In a DNS amplification attack, an attacker sends a spoofed request to a DNS resolver. In order to create a large amount of traffic, the attacker structures the request in a way that generates as large a response from the DNS resolvers as possible. As a result, the target receives an amplification of the attacker's initial traffic, and their network becomes clogged with the large amount of traffic [9].
- **SSDP Attack**
  An SSDP attack is a volumetric DDoS attack. An SSDP attack uses Universal Plug and Play (UPnP) devices in order to execute the attack. Whenever a UPnP device wants connect to a network, after receiving an IP address, the device will send a message to a certain multicast IP address. Next, this address will tell everyone in the network information about the new device. When other devices in the network receive this information, they will send a request to the new device asking for a full list of its features and services. An SSDP attack exploits this last step, since the response of the new device in this last step generates a large amount of traffic. An attacker sends spoofed UDP packets to available UPnP devices, which will all respond by sending a complete list of everything the device has to offer to the victim [17].
- **DNS Flood**
  A DNS flood is a semantic DDoS attack. In a DNS flood attack, the goal is to disrupt the services of DNS resolvers. If a domain has no DNS resolution, a website running in that domain will be compromised. DNS flood attacks use many IoT devices such as IP cameras to send requests to the DNS resolver. This results in the DNS server being overwhelmed by the traffic, rendering the target offline. A DNS flood attack is especially difficult to mitigate, since the traffic often comes from a multitude of unique locations. Furthermore, the requests are queries for real records on the domain. Therefore, it is difficult for the DNS resolver to

distinguish the malicious traffic from legitimate traffic [10].

- **HTTP Flood**
  A HTTP flood is a semantic DDoS attack. A HTTP flood is a rather basic kind of DDoS attack. In a HTTP flood attack, the attacker (usually through a botnet) sends a large amount of HTTP packets (e.g. GET, POST, HEAD etc.) to a server, overwhelming it with the amount of traffic [12].
- **SYN Flood Attack**
  A SYN flood is a semantic DDoS attack. A SYN flood attack makes use of the handshake in setting up a TCP connection. Normally, when a user wants to set up a connection, he sends a SYN packet to the server, asking to set up a connection. The server will respond with a SYN-ACK packet and leave a port open while waiting for an ACK packet, which will never come. The attacker sends many of these requests in a short period of time. At some point, the server will have all available ports utilized for this, making legitimate TCP connections unavailable [18].
- **UDP Flood Attack**
  A UDP flood is a volumetric DDoS attack. A UDP flood, by definition, is any DDoS attack that floods a target with User Datagram Protocol (UDP) packets [19]. The attack sends a large number of UDP packets to a targeted server with the aim of overwhelming that device's ability to process and respond. When a server receives a UDP packet at a particular port, it will check if any programs are running and listening for requests at that port. If this is not the case, it will send an ICMP packet back stating the destination was unreachable. A UDP Flood attack abuses this by making the target server send many of these responses back in a short period of time. This way, the target's resources can become exhausted. A UDP Flood attack usually has spoofed source addresses, so that the identity of the attacker remains hidden.
- **Ping (ICMP) Flood Attack**
  A ping flood attack is a semantic DDoS attack. The Internet Control Message Protocol (ICMP) is an Internet layer protocol used in several tools, e.g. traceroute and ping. It is mainly used to determine the health and connectivity of the device to which the request is sent to. By having many devices in a botnet send ICMP requests to a target, the target is forced to use a lot of resources to send an ICMP response to each of these requests [13].
- **Low and Slow Attack**
  A low and slow attack is a semantic DDoS attack. A low and slow attack targets thread-based web servers. The goal is to occupy every thread with a slow request, which results in a denial of service for genuine users. An example of a tool using this type of attack is Slowloris. Using this tool, the attacker slowly sends partial HTTP requests. The target server will keep the connection open, waiting for the rest of the header. If this is done slowly on every thread, the server will be occupied waiting, obstructing the thread [14]. Another example of a low and slow attack is the tool R.U.D.Y. [50]. This tool generates HTTP post requests to fill out a form. In these requests, it tells the target server how much data it can expect (which is usually a large amount). The data is sent in very slowly, but just fast enough to prevent the server from timing out. Since the server is expecting more data to arrive, it will keep the connection open. This can again result in clogging up the thread.

2.1.2 Causes for the existence of DDoS attacks

After the background and types of DDoS attacks, it is important to reflect on why DDoS attack are possible. We will now present 3 causes why DDoS attacks still exist. These causes are (1) the limited resources of devices, (2) the "end-to-end paradigm" design of the Internet, and (3) the interdependance on devices. Subsequently, the motivation that attackers have for executing these attacks will be explained in 2.1.3.

- **Limited resources**
  Every router, network and other system in the Internet has limited resources. DDoS attacks exploit this fact, and make use of the limited bandwidth, processing power and storage capacities. This problem will persist, as target devices will always have limited resources. With today's tools, even the most well-protected resources are vulnerable to DDoS attacks. This doesn't mean that there are no solutions, however; it means that it is unlikely that the DDoS problem will dissolve completely.

- **Design of the Internet**
  DDoS attacks are not only still possible, but easy to execute. One of the reasons for this is that the Internet is designed according to the "end-to-end paradigm". The fundamental notion behind this paradigm is that when two processes communicate with each other over the Internet, the reliability of that communication can be expected from the end hosts rather than the intermediate hosts. "To a large extent, the core of the network provides a very general data transfer service, which is used by all the different applications running over it. The individual applications have been designed in different ways, but mostly in ways that are sensitive to the advantages of the end to end design approach." [5]. The end-to-end paradigm shifts the complexity to end hosts, leaving the network between these hosts only responsible for packet forwarding. A good example of this paradigm in practise is the responsibility of the Transmission Control Protocol (TCP). This protocol is located one layer above IP, and is responsible for the delivery guarantee of packets that are sent from sender to receiver. On one hand, this design choice allows for relatively easy implementation of complex features in the Internet since these features can be built on top of this implementation. However, when one end user sends malicious traffic, the intermediate network will do nothing to stop it from arriving at its destination, since the hosts in the network are not designed to police traffic.

  The aforementioned paradigm allows for some negative consequences for the Internet. For example, it makes the network vulnerable to IP spoofing, where a fake source IP is inserted into a packet's header so the real sender's identity is hidden. Furthermore, it creates opportunities to perform DDoS attacks, since the network will forward all malicious traffic to the victim without question. There have been proposals to rethink the design of the Internet [5]. However, this design is not intended for preventing DDoS attacks, but rather to have a better implementation for the Internet in general. Therefore, this design does not necessarily solve the DDoS problem. It is also uncertain whether all actors in the current Internet will adopt this new design.

- **Interdependence on devices**
  In their research, Long & Thomas state: "Regardless of how well defended your assets may be, your susceptibility to many types of attacks, particularly DoS attacks, depends on the state of security on the rest of the global Internet." [43]. Many security exploits make use of a vulnerability or security flaw in a system. If the designer misses or wrongly implements a security measure, an attacker can take advantage of this and compromise the target system. However, DDoS attacks are generally executed from hosts in the Internet that are located

outside of the target's network. These are usually devices that have poor security measures installed. This makes it difficult to prevent DDoS attacks from happening, since the Internet will always have devices that can be easily subverted by an attacker. In addition, due to the rise of the Internet of Things (IoT), the number of devices connected to the Internet will only increase in the future. This will allow for devices for an attacker to use, further increasing the potential capacity of DDoS attacks.

2.1.3 Motivations behind DDoS attacks

After understanding how DDoS attacks are possible in the present day, it is relevant to know why attackers feel the need to execute these attacks. As stated before, the goal of a DDoS attack is to prevent the legitimate use of a service. This part will elaborate on why someone would want to prevent the legitimate use of a service. We will list 6 motivations [61]: financial, political, rivalry, cyber warfare, a smoke screen and boredom.

- **Financial** - financial motivations for DDoS attacks often involve the extortion of the target system, i.e. a ransom that the target has to pay in order to get their services back online. If the target is a company, the attack generally is performed during e.g. a critical sales period, when the company has more reason to pay the ransom. Attacks motivated by financial reasons often result in revenue loss, service loss and potential public embarrassment for the target. This way of performing a DDoS attack has recently largely been overshadowed by the increasing presence of ransomware such as WannaCry [69]. When targeted with ransomware, it is always advised to never pay the ransom, despite how important the captured data might be. There are also mitigation initiatives for this such as The No More Ransom Project [26], educating users about how ransomware works and offering known decryption tools.
- **Political** - political motivations revolve around hacktivism. Hacktivism means the motivation is not necessarily money, but rather political motives the attacker wants to express through the attack. An example of this is the 2015 GitHub attack [11], which lasted several days and adapted itself around implemented DDoS mitigation strategies. The DDoS traffic originated in China and it is strongly suspected that the Chinese Government oversaw the attack. This DDoS attack specifically targeted the URLs of two GitHub projects aimed at circumventing Chinese state censorship. It is speculated that the intent of the attack was to try and pressure GitHub into eliminating those projects.
- **Rivalry** - another reason might be to execute a DDoS attack on a competing company or government. If this is done during a critical sales period or other notable event, the result will be damage to the target's reputation. Therefore, there is a commercial benefit for the attacker. An example of this is an attack in June of 2018, which saw cyber criminals bring down the Bitfinex cryptocurrency exchange [34]. The system crash during this attack was followed by a wave of garbage traffic, pointing to a multistage attack that was likely intended to undermine credibility in the site. It was probably competitive rivalry that caused the renowned online poker site, Americas Cardroom, to suffer a DDoS attack that forced first the interruption and then cancellation of a tournament.
- **Cyber warfare** - a DDoS attack can be seen as a weapon in order to disrupt the enemy's services during a cyber war [53]. An example of a DDoS attack motivated by cyber warfare is the 2007 attack on Estonia [83], targeted at government services as well as financial institutions and media outlets. The attack was a result of the political conflict between Russia and Estonia, and has directly led to the creation of international laws for cyber warfare [11]. This

had a large effect on the Estonian government, since Estonia was one of the early adopters of online government.

- **Smoke screen** - a DDoS attack can be part of a larger plan, where the DDoS attack itself is not the main goal. The attack serves as a distraction for a business, while hackers execute the real attack while the IT department is occupied on mitigating the result of the DDoS attack. A survey by Kaspersky Lab [38] showed that over half of businesses questioned (56%) are confident that DDoS has been used as a smokescreen for other kinds of cyber crime, and of those business respondents, a large majority (87%) reported that they had also been the victim of a targeted attack.
- **Boredom** - there are examples of people executing a DDoS attack out of boredom. Since it is relatively easy to DDoS a machine with the use of Booters [67], people that lack a technical background are able to execute an attack. An example of this is kids performing a DDoS attack on their own school, simply because they could [24].

2.1.4 Concluding remarks

The goal of this chapter was to provide an elaborate knowledge of the DDoS field. In order to achieve this goal, we have given an extensive description of DDoS attacks by dividing this chapter into 4 parts. Firstly, we provided some background information on DDoS attacks. Next, we discussed the 2 types in which DDoS attacks can be classified: volumetric and semantic. Volumetric attacks are attacks where the reply is larger than the request. By using amplification, an attacker can use few resources to attack a large target. Semantic attacks generally don't generate a response as large as a volumetric attack, but exploit a specific feature or bug in a protocol used by the victim. We also listed the 10 most commonly used DDoS attack nowadays, and classified each one as either volumetric or semantic. Out of all DDoS attacks, volumetric attacks are the most devastating, since they are able to generate a very large amount of traffic, overwhelming the target.

Subsequently, we elaborated on the causes of the existence of DDoS attacks. Mainly due to the way the Internet is designed (the end-to-end paradigm), DDoS attacks are and will remain a problem in the future. Furthermore, since all machines have a limited amount of resources, they are always vulnerable to being a target of a DDoS attack. We also examined the motivations behind DDoS attacks. These motivations include financial, political, rivalry, cyber warfare, a smoke screen and even boredom.

Combining all information gained from this chapter, it is clear DDoS attacks are still a big problem and challenge nowadays (and will continue to be), and it is important to understand them fully before tackling mitigation. In the next chapter we will discuss DDoS mitigation, the various levels where mitigation can be performed, as well as different mitigation methods and tools.

2.2 Mitigation of DDoS attacks

In this chapter, we will elaborate on the various ways DDoS attacks can be mitigated. When a DDoS attack is initiated from a certain source, the attack traffic can be blocked at various stages during the packet's route. "Ideally, DDoS attacks are mitigated close to the attacker, and mitigation only affects malicious traffic" [32]. Henceforth, we will call these stages "mitigation levels". The goal of this chapter is to provide a clear overview of the different methods and tools of mitigation. In this chapter, we will first observe the mitigation levels (2.2.1), and list the applicable Internet layers and see which mitigation methods can be applied. These mitigation methods will be explained afterwards in 2.2.2. For each method, we will explain how DDoS traffic is mitigated, as well as list a set of tools with which these methods can be applied in practice. We will close the chapter with some concluding remarks.

2.2.1 Mitigation levels

First, we will discuss the different levels at which DDoS traffic can be mitigated. Figure 2.2 is an expanded version of the overview picture we have seen, where the route that the packet travels is emphasized. The levels in the route at which DDoS traffic can be detected and mitigated are indicated with letters A through G. These represent the attacker, botnet, reflector, IXP, ISP, organization and target machine level respectively. For the remainder of this part, each level of mitigation is listed. For each level, we will list the Internet layers that are used, as well as the mitigation methods that can be applied. After that, each mitigation method will be explained in more detail.

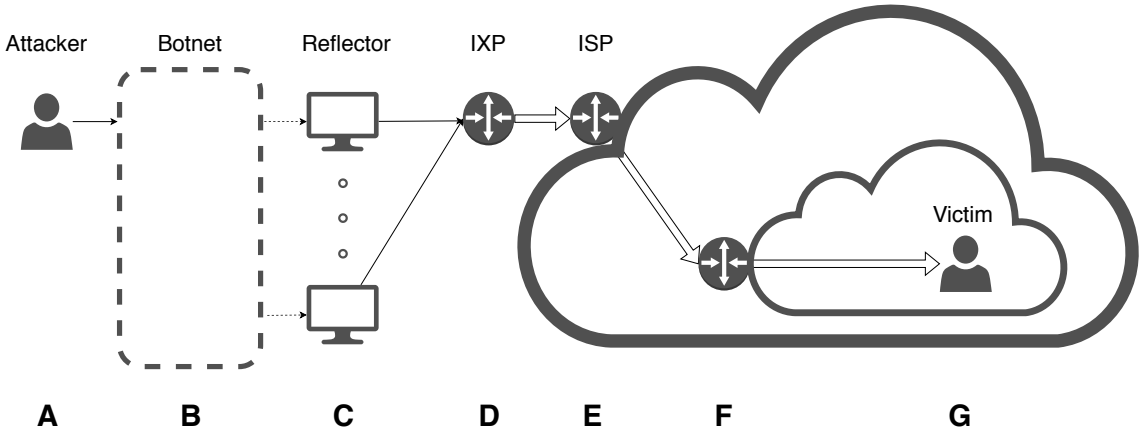

Figure 2.2: A simplified overview of a DDoS attack packet's route, with the different mitigation points marked with letters A, B, C and D.

- **A: Attacker level**

  Mitigating DDoS attacks at the attacker level is possible. However, this would be a job for e.g. the police, and lies out of the scope of this research.

- **B: Botnet level**

  Mitigating at this level involves the prevention of a botnet being used. This is out of the scope of this research.

- **C: Reflector level**

  Mitigating at this level requires preventing reflectors from being used in a DDoS attack. This lies out of the scope of this research.

- **D: IXP level**

  Before the packet will enter an autonomous network, it usually will pass through an Internet Exchange Point (IXP). At this level, the options on mitigating incoming traffic are relatively limited, partly due to the recent discussion on Net Neutrality [58], where it is questioned whether an IXP should have the ability to filter traffic based on its content. While outside the scope of this research, it is important that this topic is kept into account when equipping IXPs with this ability.

Internet layers	Mitigation methods
Data link	Blackholing
Network	

- **E: ISP level**

  At the ISP level, every mitigation method (as listed in 2.2.2) could be used. However, this is generally not done, since this would result in high costs for the end user. When an ISP puts DDoS mitigation measures in place that could be done by the end user as well, the ISP has to bill the end user for this. Furthermore, there is still the issue of net neutrality; it may not be desirable for an ISP to see the details of internet traffic, and filter it according to this information.

Internet layers	Mitigation methods
Data link	Blackholing
Network	Intrusion Detection Systems
Transport	Network firewall
Application	Web Application Firewall

- **F: Organization level**

  Deeper in the network, in the organization where the target machine is located (e.g. someone's home network), there are many options to mitigate DDoS traffic. Mitigating at this stage could be less effective, since the hardware cannot handle a throughput as large as an ISP's edge router. However, mitigation tools in the organization level allow for more granularity compared to the IXP and ISP level.

Internet layers	Mitigation methods
Network	Intrusion Detection Systems
Transport	Network firewall

- **G: Target machine**

On the machine of the target of the DDoS attack itself, it is also possible to block DDoS traffic.

Internet layers	**Mitigation methods**
Network	Intrusion Detection Systems
Transport	Web Application Firewall
Application	

In order to further illustrate the difference between the levels, Table 2.1 shows the aforementioned levels and their corresponding Internet layers and mitigation methods. Similarly, Table 2.2 shows the mitigation methods that can be used for each level.

Internet layer	**Levels**			
	IXP	ISP	Organization	Victim
7 - Application		✓		✓
6 - Presentation				
5 - Session				
4 - Transport		✓	✓	✓
3 - Network	✓	✓	✓	✓
2 - Data link	✓	✓		
1 - Physical				

Table 2.1: Each mitigation level and the Internet layer(s) where it can be applied.

Mitigation methods	**Levels**			
	IXP	ISP	Organization	Victim
Blackholing	✓	✓		
Intrusion Detection Systems		✓	✓	✓
Network firewall		✓	✓	
Web Application Firewall		✓		✓

Table 2.2: Each mitigation level and the mitigation methods that can be used.

2.2.2 Mitigation methods and tools

For each mitigation level in Figure 2.2, we have listed the Internet layers and methods of mitigating traffic at that level. In the following part, we will elaborate on each method and list various tools that can be used.

DNS Redirection

In DNS redirection [80], a mitigation provider masks the target's IP address as one of the mitigation provider itself. All traffic is sent to the mitigation provider, which can then first filter out any malicious traffic before sending it back to the target. However, since this method uses an alteration in a DNS resolver, it only works on the application layer. Therefore, when a direct-to-origin attack occurs, the target IP can still be discovered and thus targeted with DDoS traffic. For this reason, DNS Redirection is not really used anymore.

Blackholing

Blackholing [22] is a mitigation method against DDoS attacks. The network traffic is routed elsewhere, to a "black hole". This means that all traffic that is routed to the black hole will be dropped. This can have different consequences depending on the protocol used. When TCP is used, a notification will be returned to the source notifying that the traffic has been dropped. In connectionless protocols such as UDP, this is not the case. In DDoS mitigation, there are two types of black holes: a Destination Remotely-Triggered Black Hole (D/RTBH) and a Source Remotely-Triggered Black hole (S/RTBH). Both lead to essentially a null route that drops the traffic, but the difference between the two is the traffic that is being filtered.

In a D/RTBH, traffic that is headed towards a given destination is blocked. If a device is experiencing a DDoS attack, a D/RTBH can be set up that filters all traffic with the target's IP as destination. However, this also blocks all benign traffic that is meant for the target. This means that the DDoS attack partly succeeded, since only outgoing traffic from the target's machine is now possible.

A more favorable way of black holing could be a S/RTBH. In this case, traffic that originates from a given source IP address is dropped. This is useful in a non-distributed DoS attack, since the legitimate traffic coming from other sources will not be blocked. However, in most modern DDoS attacks, the traffic comes from many different sources. A typical DDoS attack can have around 10.000 source IPs, making it infeasible to blackhole each one.

It can be quickly noticed that blackholing is not the most optimal solution for mitigating DDoS attacks, since in many cases, legitimate traffic is blocked. Nevertheless, it is still a widely available option for organizations or individuals that don't have access to modern DDoS mitigation tools. It can also be useful when the target of an attack is a smaller machine or site that is part of a larger network. In this situation, blackholing the target can prevent the other machines in the network from being affected by the attack.

Tools:

- BGP Flowspec [44]

Intrusion Detection Systems

A commonly used way of mitigating DDoS traffic is the use of Intrusion Detection Systems (IDS). There are two types of IDSs. As Paxson states: "We can divide these systems into two types, those that rely on audit information gathered by the hosts in the network they are trying to protect, and those that operate 'stand-alone' by observing network traffic directly, and passively, using a packet filter" [57]. To simplify, and IDS either focuses on incoming network traffic (Network-based IDS (NIDS)) or on the host's machine itself (Host-based IDS (HIDS)). An NIDS is typically

installed at strategic points in the network where they are able to effectively detect and monitor traffic going to all devices in the network. An NIDS reads all inbound packets and searches for any suspicious patterns. When threats are discovered, based on its severity, the system can take action such as notifying administrators, or barring the source IP address from accessing the network.

The other type of IDS, as mentioned before, is called a Host-based IDS. As the name suggests, an HIDS is installed on the system itself, capable of monitoring and analyzing the internals of that system. An HIDS monitors the state of the machine constantly, enabling it to detect modifications to this state. When an intruder attempts to gain control of the machine, he will generally leave a trace (e.g. installing a key logger, installing malware for a botnet, etc.). If such an intruder would pass an NIDS that is installed in the network, there is a chance for the HIDS to detect the modified machine state. HIDS generally work with a database of system objects that it should monitor. It generates a checksum of this data, allowing it to easily check if the system state has been modified.

NIDS tools:

- Snort [63]
- Suricata [54]
- Bro [57]
- IBM QRadar [33]

HIDS tools:

- OSSEC [56]
- AIDE [41]

There are also hybrid systems that combine the two approaches to provide one comprehensive solution [59, 70, 72].

In addition to IDSs being installed on either the network or the host itself, there is a different way of classifying IDSs. In this case, they vary in the way they are implemented and detect traffic. García-Teodoro et al. explain the between these two techniques: "Signature and anomaly-based systems are similar in terms of conceptual operation and composition. The main differences between these methodologies are inherent in the concepts of 'attack' and 'anomaly'. An attack can be defined as 'a sequence of operations that puts the security of a system at risk'. An anomaly is just 'an event that is suspicious from the perspective of security'. Based on this distinction, the main advantages and disadvantages of each IDS type can be pointed out." [28]

A signature-based IDS looks for specific patterns such as byte sequences in files or network traffic. These detection methods can be applied to HIDS as well as NIDS. In a HIDS, the tool will scan for log and config files searching for modifications. In an NIDS, the tool will scan for checksums of network packets. In addition, a signature-based NIDS generally has a database of signatures that represent malicious packets. Since a lot of hackers use the same tools to accomplish their goal (e.g. crack a password), these tools will generate the same traffic signatures every time. This makes a signature-based IDS a suitable way of detecting malicious actions. The drawback of signature-based IDSs is that it is impossible to detect new attacks, since no signatures of it exist yet.

This drawback that signature-based IDSs experience can possibly be solved by using an anomaly-based IDS. Here, the approach is to classify traffic behaviour. This can e.g. be done using machine learning techniques in order to construct a model that represents 'normal' traffic behaviour, and compare any new traffic to this model. This way, an 'anomaly' can be detected, as this traffic will be out of the range of the regular traffic pattern and thus classified as malicious behaviour. Similar to signature-based IDSs, anomaly-based IDSs can be applied to HIDS as well as NIDS. In a HIDS,

an anomaly might be a large number of repeated failed login attempts, suggesting a hacker is trying to crack a password. In an NIDS, an anomaly can be any network traffic pattern that doesn't match the model of 'good' behaviour. The drawback of anomaly-based IDSs is that there is a probability of suffering from false positives; legitimate traffic that was previously unknown, might be classified as an anomaly due to that traffic not matching the machine learning model.

Network Firewall

A network firewall, also commonly referred to as a packet filter, is used to filter incoming network traffic. A rule set is established, upon which the firewall either allows or denies incoming packets based on that rule set. A network firewall has default rules, but a system administrator can define rules in order to filter traffic. "A firewall typically creates a barrier between a trusted internal network and an untrusted external network, such as the Internet." [55]

Tools:

- IPtables [51]
- Berkeley Packet Filter [46]

Web Application Firewall

A Web Application Firewall (WAF) is an application security measure deployed between a web client and a web server. When a WAF is in place, it performs an inspection of incoming and outgoing HTTP traffic. It differs from a regular firewall in that a WAF is able to filter the content of specific web applications, while a network firewall serves as a security barricade between servers. A WAF is generally successful in preventing attacks that originate from web application security flaws (e.g. SQL injection and cross-site scripting (XSS)).

Tools:

- ModSecurity [62]

2.2.3 Concluding remarks

The goal of this chapter was to provide a clear overview of the different methods and tools of mitigation. In order to explain this clearly, we have divided the DDoS mitigation field into 7 different levels where DDoS mitigation can take place, as well as the different methods of mitigation that can be applied at these levels. These levels are (A) the attacker, (B) the botnet, (C) the reflectors, (D) IXP, (E) ISP, (F) organization and (G) the victim's machine. At each level, some form of mitigation or prevention is possible. Since levels A, B and C are outside of the scope of this research, we elaborated further on the latter four.

Many methods of mitigation exist, some of which we elaborated on in this chapter. These methods are blackholing, Intrusion Detection Systems (IDS), a network firewall and a Web Application Firewall (WAF). We analyzed each method, and gave an overview of which mitigation method can be used on which level, as well as some example tools that can be used to apply the method. Furthermore, we listed the Internet layers that are applicable for each level.

Using this information, it is made clear why certain mitigation methods are more effective than others, and why some methods work at a given stage where others don't. Some methods allow

for a larger throughput capability (e.g. BGP Flowspec), while others show potential in offering more granularity (e.g. a network firewall).

When comparing the aforementioned methods and tools, the most interesting tool to acknowledge is BGP Flowspec. The reason for this is that it can be applied at the IXP and ISP level, where a lot of traffic passes through (thus having a large throughput capability). Furthermore, BGP Flowspec allows for more granularity than regular blackholing, which is the way mitigation is currently being done at these levels. Moreover, as we will make clear in the next chapter, very little research has been done on the effectiveness of this tool in mitigating DDoS traffic. The next chapter will describe BGP Flowspec in detail, as well as why it has potential to be extremely effective for DDoS mitigation.

2.3 BGP Flowspec

After gaining the knowledge on DDoS attacks, mitigation stages and methods, we will move our focus to the main topic of this document: BGP Flowspec. This is a tool that can be used for DDoS mitigation at the IXP and ISP levels as described in the previous chapter. While BGP Flowspec can be used for applying blackholing at these levels (see 2.2.2), BGP Flowspec also has the potential to block traffic more granularly. In this chapter, the goal is to gain knowledge on BGP Flowspec and understand how and why it can be an effective tool for mitigating DDoS traffic. We will first provide background information in 2.3.1. Next, we will elaborate on BGP Flowspec's limitations (2.3.2) Subsequently, we will describe the mitigation impact this tool can have on the underlying network, since it is possible to block legitimate traffic unintentionally. This will be discussed in 2.3.3. Lastly, we will review related research (2.3.4). We will finish the chapter with some concluding remarks.

2.3.1 Background

The Border Gateway Protocol (BGP) [60] is the most important routing protocol in the Internet, since it provides communication between autonomous networks. Examples of autonomous networks communicating at this level are ISPs. An ISP can choose an internal routing protocol itself (such as OSPF [49] or Routing Information Protocol [31]), but communication between autonomous networks is always done using BGP. In BGP, two routers can become each others 'peers' upon starting a communication session. This session is set up with TCP and is configured manually. Every 60 seconds, a keep-alive message is sent to sustain the connection.

BGP Flowspec is an extension to the BGP routing protocol [44]. Its feature is to allow for filtering of network traffic among a large number of BGP peer routers. This way, it could be an effective method against DDoS attacks over networks. In contradiction to e.g. blackholing (where all traffic from or to a certain host is dropped), BGP Flowspec allows for a much more granular approach. It allows for construction of rules that match a defined network flow by offering 12 parameters. Examples of these are, among others, source/destination IP, packet length and flags. All 12 parameters are described in Table 2.3. Routers at the edge of an ISP's network can apply Flowspec rules at any time. Furthermore, when traffic that satisfies the Flowspec rules arrives, the router can perform either of the following 3 actions [44]:

- Drop the traffic entirely
- Redirect the traffic elsewhere for analysis

- Allow the traffic at a reduced rate

The aforementioned 12 parameters are defined as Network Layer Reachability Information (NLRI). Any set of NLRI (as defined in Table 2.3) defines a network flow, i.e. a group of traffic packets that can be grouped together and labeled accordingly. An incoming network packet is considered to match the flow specification when it matches all components that are in the specification. When this occurs, one of the aforementioned actions will take place.

NLRI type	**QoS match fields**	**Description**	**Example value**
Type 1	Destination address	Defines the destination prefix to match.	130.89.161.0/24
Type 2	Source address	Defines the source prefix to match.	130.89.161.0/24
Type 3	IP Protocol	Contains a set of {operator, value} pairs that are used to match the IP protocol value byte in IP packets.	1, 3, 5, 17-19
Type 4	Source or destination port	Defines whether TCP, UDP or both will be packets will be influenced	1-80, 443
Type 5	Destination port	Defines the destination port that will be influenced by Flowspec	1-80, 443
Type 6	Source port	Defines the source port that will be influenced by Flowspec	1-80, 443
Type 7	ICMP type	Any (range of) ICMP types	0, 3-5
Type 8	ICMP code	Any (range of) ICMP codes	3, 6-15
Type 9	TCP flags	Any amount of TCP flags	ACK, FIN, PUSH, SYN
Type 10	Packet length	Match on the total IP packet length (excluding Layer 2 but including IP header)	40, 255-1518
Type 11	DSCP	Match on the Class Of Service flag	40, 255-1518
Type 12	Fragmentation bits	Any amount of IP fragmentation flags	dont-fragment, is-fragment

Table 2.3: Flowspec tuple definition possibilities [8]

BGP Flowspec was proposed as a standard specified in RFC 5575 [44] in August 2009. To be able to utilize BGP Flowspec, an ISP's routers must use BGP's Capability Advertisement facility to exchange the Multiprotocol Extension Capability Code [3]. In 2015, Steinberger et al. conducted a survey in order to assess the technical ability to use BGP Flowspec among ISPs [74]. Their research shows that 52% do not currently support BGP Flowspec and 69% do not even plan to use it in 3 years. However, since this survey was conducted in 2015, the percentage of ISPs that support BGP Flowspec could have increased.

2.3.2 Limitations

Despite BGP Flowspec having limitations in both its support and other limitations such as hardware, Hinze et al. [32] argue that BGP Flowspec has potential to improve current DDoS mitigation practices. They claim that current blackholing drops a significant amount of valid traffic, whereas

the use of Flowspec could improve this while requiring very little additional information. However, there are some practical limitations to BGP Flowspec that are important to address. These limitations are a result of the implementation of the BGP Flowspec standard, as well as the hardware that it uses.

When examining various existing DDoS attack data sets, it can be seen that a DDoS attack often originated from multiple sources. For example, the data retrieved from DDoSDB [68] shows that many attacks originate from as much as 10.000 source IPs. According to the BGP Flowspec standard, the type 2 NLRI can be used to define a flow according to a source prefix. However, a given BGP Flowspec rule can only define one source prefix. If we want to generate a rule set that successfully defines this DDoS attack based on only this NLRI field, we would have to generate 10.000 BGP Flowspec rules. This does not scale for larger attacks.

Additionally, there are limitations on the hardware that vendors provide. For example, Cisco, a major network hardware vendor, has a hardware lineup that is designed for IXP-level routing. These routers use BGP for communication with other networks and are part of the Series Aggregation Services platform [8]. The operating system that is used on these routers (IOS XR) involves a limit of 3000 BGP Flowspec rules. Recalling the earlier example of a DDoS attack that originates from 10.000 source IPs, it is not feasible to apply BGP Flowspec rules to define this attack based only on the source IPs. Furthermore, IOS XR limits the number of multi-value ranges within a BGP Flowspec rule to 5 [42]. Loibl and Bacher [42] conducted research to produce a working set of configuration suitable for BGP Flowspec for multiple hardware vendors. They address various limitations in BGP Flowspec, such as bugs on certain hardware, as well as missing features in the standard. At the time of publishing, there were ongoing efforts to update the RFC 5575 standard, where BGP Flowspec is specified. It is unclear whether the bugs and missing features have been fixed in this update.

Finally, implementing a BGP Flowspec configuration is challenging, since all hardware vendors that support it (e.g. Cisco, Juniper, Huawei) have a different configuration language for implementing the rules. This makes it difficult to generate rules automatically and exchange them. Loibl and Bacher [42] also address that though there are use cases for exchanging BGP Flowspec rules among ISPs, the carriers seem to hesitate introducing this concept.

2.3.3 Mitigation impact

For an ISP to apply BGP Flowspec rules, it is of high importance to assess the impact these rules can have on the network. As previously mentioned, "mitigation impact" implies the negative impact a BGP Flowspec rule has on the ISP's network. This applies to all hosts in the network, including the target host of the DDoS traffic on which the Flowspec rule is targeted. BGP Flowspec allows for the filtering of 12 fields in the IP header, resulting in relatively low granularity compared to e.g. a firewall. For this reason, it is important for a Flowspec rule to have the smallest possible mitigation impact on the network. The impact is defined by a set of factors. One of these factors is the number of false positives in the filtered packets, e.g. benign traffic that is nonetheless filtered by the Flowspec rule. This has a negative impact on the network, since non-DDoS traffic will be dropped at the ISP's edge routers.

In order to illustrate the mitigation impact, note the following example: a company with IP-address 130.89.10.1 is experiencing DDoS attack traffic coming from more than 10.000 IPs. Consider that this company has a system in place that analyzes the incoming traffic and constructs BGP Flowspec rules based on the characteristics of the incoming traffic, in this case the source IP. However, since this particular attack has many source addresses, there could be a need for

more than 10.000 Flowspec rules in order to properly define a network flow that corresponds to the DDoS traffic. As discussed in the limitations (2.3.2), routers can only handle a maximum of 3000 installed BGP Flowspec rules. For this reason, defining this network flow with source IPs is infeasible. A solution for this can be to use prefixes (note that type 2 of the NLRI fields is defined by a source prefix to match) in order to 'bundle' multiple source IPs together. For example, if 100 of the source IPs start with 50.62.24.xx, we can group these together by creating a Flowspec rule with type 2 field "50.62.24.0/24". However, in this situation, all traffic originating from this prefix is blocked. If the company receives a legitimate data packet from IP address 50.62.24.11, this data packet satisfies the condition of being in the 50.62.24.0/24 prefix. As a result, the packet is classified as a part of the DDoS traffic flow, and thus is dropped upon entering the ISP's network. This situation results in a negative impact on the network, and should be avoided.

One of the core challenges that is relevant for this research is minimizing the aforementioned mitigation impact on the network. In order to solve this challenge, it is necessary to measure this impact in some way, in order to quantify its magnitude. When quantification of the mitigation impact is possible, a system can be built that self-adaptively deploys Flowspec rules, quantifies their impact and adjusts the rules in place if necessary. For example, if a set of rules measures an impact that is above a certain threshold, one or more rules can be omitted from the set in order to reduce the impact.

2.3.4 Related work

In order to fulfill the goal of this document, it is necessary to review existing literature related to the topic of this research. Here, we describe the research related to DDoS mitigation and BGP Flowspec. In addition, we analyze all relevant papers in a table (Table 2.4) in order to obtain an overview of the research that has and hasn't been done at the time of writing this document. These two elements combined will provide knowledge on where this research lies in the DDoS field, and what gaps it attempts to fill. The papers discussed in this section have been retrieved using Google Scholar. After attempting several search queries, it was found that the term "bgp flowspec denial of service" was the most effective, resulting in 205 results. Out of these 205 results, irrelevant papers, books and presentations were subtracted. After this subtraction, 9 papers remained that focus specifically on either DDoS mitigation, BGP Flowspec or mitigation impact (or a combination of these elements). These 9 papers will be discussed and reviewed in this section.

Mirkovic and Reiher [47] provide a detailed classification of the state-of-the-art of DDoS attacks. Although published in 2004, it still allows for a comprehensive taxonomy of DDoS attack as well as defense mechanisms. However, it lacks in providing a comparison of the effectiveness of the mentioned defense mechanisms. It also doesn't propose how these different mechanisms can be combined into one, more complete solution to DDoS attacks.

Rather than looking to mitigate DDoS traffic close to the destination, Mirkovic et al. [48] propose to detect this traffic closer to the source. They developed a system called D-WARD that is able to stop incoming DDoS traffic at the edge routers of a network. This prevents a large amount of useless traffic entering the network. They do address several shortcomings of this method, however. They is a possibility of false positives occurring when legitimate traffic flows start during the DDoS attack. It also lacks in detecting more subtle UDP attacks and repeated attacks, since it retains no memory of previously detected DDoS traffic.

Some solutions focus specifically on the application of BGP Flowspec features for disseminating the rule set [44]. This can only work for large-scale networks, as smaller networks do not use

BGP as a routing protocol. Jamous et al. [36] proposed a system that is capable of near real-time detection and diversion of malicious network traffic. However, this research does not focus on the impact on the network, which is relevant to take into account. Furthermore, it does not propose BGP Flowspec as a self-adaptive solution, which could mean that Flowspec rules negatively impacting a network will not be solved.

To illustrate how extensively BGP Flowspec is used, Steinberger et al. [74] conducted a survey among ISPs in order to assess their ability to use BGP Flowspec. They show that 52% do not currently support BGP Flowspec and 69% do not even plan to use it in 3 years.

In their research, Somani et al. [73] show the importance of assessing the mitigation impact (described as 'collateral damage'). However, this research focuses on DDoS mitigation in general, rather than using BGP Flowspec specifically.

Dietzel et al. [23] proposed an advanced blackholing system with more granularity. They mention Flowspec as a tool for this as well, stating that, in the inter-domain environment, it requires ISPs to trust and cooperate with each other. "Flowspec, a popular intra-domain attack mitigation technique, relies on trust, cooperation, and sharing of resources among different networks when deployed in the inter-domain environment. Unfortunately, these requirements are hard to satisfy when networks with diverse resources as well as different or even conflicting policies and business strategies form the Internet." Essentially, the problem stated here is ISPs providing their resources to solve another ISP's problem. This can raise doubt about liability, since the Internet is a highly competitive environment.

Gev et al. [29] present Backward Traffic Throttle (BTT), a mechanism to mitigate network flooding attacks. This system uses traffic statistics prior to the attack in order to control the bandwidth of incoming traffic, which can be compared to using fingerprint data as a basis for defining traffic flows. Furthermore, they take into account the mitigation impact by measuring the degradation of legitimate traffic rates during an attack (mentioned as the "goodput ratio"). In their research, they compare their BTT mechanism to BGP Flowspec among others on several fields. They argue that their method does not require non-standard router features (BGP Flowspec does require this) and that BTT does not require the identification of attack flows, while BGP Flowspec does. Furthermore, they argue that BGP Flowspec requires many rules commonly based on source and destination aggregates, which does not scale well.

Hinze et al. [32] examined the potential of BGP Flowspec for DDoS mitigation at the IXP and ISP level. They show that BGP Flowspec can effectively be used for DDoS mitigation with very little extra information on the DDoS traffic flow. However, the generation of rules is still a manual operation. Also, mitigation impact is not included. They conclude that DDoS mitigation at IXP level using BGP Flowspec has a high potential.

In order to provide a further overview of where this research lies in the existing field, Table 2.4 shows all aforementioned papers and the topics they focus on. It can be clearly seen that very few papers focus on the mitigation impact. The papers that do talk about mitigation impact do not focus on BGP or BGP Flowspec specifically. Furthermore, no existing research proposed a self-adaptive approach, where rules are generated, tested on effectiveness and mitigation impact, and re-evaluated.

	Features					
Paper	DDoS detection	DDoS Mitigation	BGP Flowspec	Fingerprint data	Self-adaptive	Mitigation impact
Mirkovic and Reiher [47]	✓	✓				
Mirkovic et al. [48]	✓	✓				
Marques et al. [44]			✓			
Jamous et al. [36]	✓	✓	✓			
Steinberger et al. [74]			✓			
Somani et al. [73]	✓					✓
Dietzel et al. [23]	✓	✓	✓			✓
Gev et al. [29]	✓	✓		✓		✓
Hinze et al. [32]	✓	✓	✓			

Table 2.4: Literature review with several related topics marked for each paper.

2.3.5 Concluding remarks

The goal of this chapter was to gain knowledge on BGP Flowspec and understand how and why it can be an effective tool for mitigating DDoS traffic. In order to achieve this goal, we have divided this chapter into 4 parts. We first looked into background information on BGP Flowspec, in order to see which criteria are possible to define a network flow, as well as the actions that can be performed. BGP Flowspec allows for the filtering of traffic with the use of 12 fields. The traffic that matches the flow can either be dropped, redirected elsewhere for analysis or allowed at a reduced rate.

Moreover, we discussed the limitations of BGP Flowspec. When defining a traffic flow with a BGP Flowspec rule, only one instance of each field can be used. Therefore, rule generation does not generally scale well for attacks with a large number of source IPs. Additionally, there are limitations in the hardware that vendors provide, restricting the number of BGP Flowspec rules that can be installed at a given time. Lastly, since all hardware vendors have a different implementation of their configuration language, it is challenging to generate rules automatically and exchange them.

Next, we elaborated on the mitigation impact. If a traffic flow is defined using BGP Flowspec rules, it is possible that legitimate traffic is blocked unintentionally. This results in a negative impact on the underlying network. One of the core challenges of this research is to find a way to measure this impact in order to quantify its magnitude. If this quantification is possible, we will be able to design a self-adaptive system where BGP Flowspec rule generation is based on the mitigation impact.

Additionally, we have given an extensive review of related work. The related literature shows that BGP Flowspec has a high potential in blocking DDoS traffic in a manner that is more granular than blackholing practices. BGP Flowspec has received some adoption in intra-domain environments and has been shown to have good reaction time and performance [7, 64]. However, existing research is lacking in addressing and quantifying the negative impact on the network.

Ideally, ISPs and IXPs have a system in place that is self-adaptive, i.e. a system that measures the performance as well as the mitigation impact of a set of Flowspec rules and adjusts the rule set accordingly. This way, it is possible to generate effective DDoS traffic filtering while minimizing the negative impact on other users in the network.

2.4 Conclusions

The goal of this document was to provide the answer to Research Question 1. The question was defined as follows:

How does BGP Flowspec theoretically compare to existing DDoS mitigation solutions?

To answer this question, we divided this chapter into 3 sections: DDoS attacks (chapter 2), the mitigation of DDoS attacks (chapter 2.2), and BGP Flowspec (chapter 2.3). Firstly, we discussed the DDoS field. We provided some background information on DDoS attacks, and discussed the 2 types in which DDoS attacks can be classified: volumetric and semantic. We also listed the 10 most commonly used DDoS attack nowadays, and classified each one as either volumetric or semantic. Out of all DDoS attacks, volumetric attacks are the most devastating, since they are able to generate a very large amount of traffic, overwhelming the target. Subsequently, we elaborated on the causes of the existence of DDoS attacks. Mainly due to the way the Internet is designed (the end-to-end paradigm), DDoS attacks are and will remain a problem in the future. Furthermore, since all machines have a limited amount of resources, they are always vulnerable to being a target of a DDoS attack. We also examined the motivations behind DDoS attacks. These motivations include financial, political, rivalry, cyber warfare, a smoke screen and even boredom. Combining all information gained from this chapter, it is clear DDoS attacks are still a big problem and challenge nowadays (and will continue to be), and it is important to understand them fully before tackling mitigation.

In chapter 2.2, we provided a clear overview of the different methods and tools of mitigation. In order to explain this clearly, we have divided the DDoS mitigation field into 7 different levels where DDoS mitigation can take place, as well as the different methods of mitigation that can be applied at these levels. These levels are (A) the attacker, (B) the botnet, (C) the reflectors, (D) IXP, (E) ISP, (F) organization and (G) the victim's machine. At each level, some form of mitigation or prevention is possible. Since levels A, B and C are outside of the scope of this research, we elaborated further on the latter four. Many methods of mitigation exist, some of which we elaborated on in this chapter. These methods are blackholing, Intrusion Detection Systems (IDS), a network firewall and a Web Application Firewall (WAF). We analyzed each method, and gave an overview of which mitigation method can be used on which level, as well as some example tools that can be used to apply the method. Furthermore, we listed the Internet layers that are applicable for each level. Using this information, it is made clear why certain mitigation methods are more effective than others, and why some methods work at a given stage where others don't. Some methods allow for a larger throughput capability (e.g. BGP Flowspec), while others show potential in offering more granularity (e.g. a network firewall). When comparing the aforementioned methods and tools, the most interesting tool to acknowledge is BGP Flowspec. The reason for this is that it can be applied at the IXP and ISP level, where a lot of traffic passes through (thus having a large throughput capability). Furthermore, BGP Flowspec allows for more granularity than regular blackholing, which is the way mitigation is currently being done at these levels.

In chapter 2.3, the goal was to gain knowledge on BGP Flowspec and understand how and why it can be an effective tool for mitigating DDoS traffic. We first looked into background information on BGP Flowspec, in order to see which criteria are possible to define a network flow, as well

as the actions that can be performed. BGP Flowspec allows for the filtering of traffic with the use of 12 fields. The traffic that matches the flow can either be dropped, redirected elsewhere for analysis or allowed at a reduced rate. Moreover, we discussed the limitations of BGP Flowspec. When defining a traffic flow with a BGP Flowspec rule, only one instance of each field can be used. Therefore, rule generation does not generally scale well for attacks with a large number of source IPs. Additionally, there are limitations in the hardware that vendors provide, restricting the number of BGP Flowspec rules that can be installed at a given time. Lastly, since all hardware vendors have a different implementation of their configuration language, it is challenging to generate rules automatically and exchange them. Next, we elaborated on the mitigation impact. If a traffic flow is defined using BGP Flowspec rules, it is possible that legitimate traffic is blocked unintentionally. This results in a negative impact on the underlying network. One of the core challenges of this research is to find a way to measure this impact in order to quantify its magnitude. If this quantification is possible, we will be able to design a self-adaptive system where BGP Flowspec rule generation is based on the mitigation impact. Lastly, we have given an extensive review of related work. The related literature shows that BGP Flowspec has a high potential in blocking DDoS traffic in a manner that is more granular than blackholing practices. BGP Flowspec has received some adoption in intra-domain environments and has been shown to have good reaction time and performance [7, 64]. However, existing research is lacking in addressing and quantifying the negative impact on the network.

In short, to answer research question 1: **When comparing to other DDoS mitigation solutions, BGP Flowspec has one of the largest throughput capabilities, but lacks in granularity.**

In the follow-up research, we will fine tune this lack of granularity. In order to achieve this, we propose a self-adapting mechanism, in which both the effectiveness and mitigation impact of each rule set is quantified. This quantification can be evaluated, in order to continuously improve the rule set and thus the mitigation of the incoming traffic. This way, we can effectively mitigate DDoS traffic while minimizing the negative impact on other users in the network.

Chapter 3

BGP Flowspec Rule Generation

The previous chapter presented an in-depth look into DDoS field and the challenges it poses. We discussed how DDoS attacks can be mitigated, and introduced BGP Flowspec and it's relevant literature. We concluded that, when comparing BGP Flowspec to other DDoS mitigation solutions, it has one of the largest throughput capabilities, but lacks in granularity.

The goal of this chapter is to answer Research Question 2:

How can we write BGP Flowspec rules based on known DDoS attacks?

This chapter will focus on the generation of rules. Using existing DDoS attack data, we will present a methodology to generate BGP Flowspec rules using this data. This rule generation process is divided into 3 parts: dissecting and summarizing the network characteristics of an existing DDoS attack, generating the rule set for BGP Flowspec, and parsing the rule set to the language of the router's Operating System.

A simplified overview of the rule generation process is displayed in Figure 3.1. It can be seen that the rule generation process is divided into 3 main components: the DDoS dissector, the rule set generator and the parser. In this chapter, we will focus on each component of the overview below and explain the data used as input, the process that takes place and the output data that is generated.

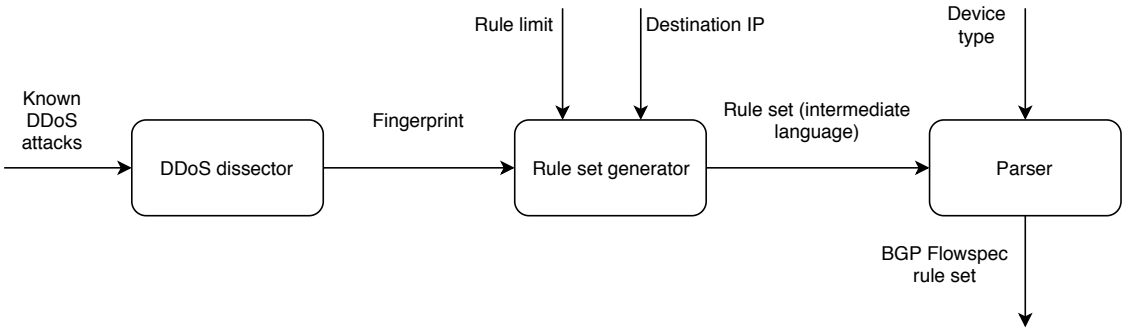

Figure 3.1: An illustration of the rule generation process.

3.1 DDoS dissector

The first component of the rule generation process is the DDoS dissector. This is a function that is responsible of generating a summary of the network characteristics out of known DDoS attack traffic. This summary is called a "fingerprint". It should be noted that the dissector is a separate module and part of DDoSDB [68]. DDoSDB is a platform designed to help DDoS attack victims and the academic community to get access to information about DDoS attacks. It collects its data from collaborators (often DDoS attack victims) that gathered data from DDoS attacks they experienced.

For this research, the dissector is **not** developed by the authors and is thus taken for granted. There exists a close collaboration between the authors of this work and the developer of the dissector on DDoSDB. Therefore, the authors were able to invoke changes to the dissector depending on the requirements for the rule generator. For this reason, the dissector on DDoSDB was chosen. We will look into the dissector in more detail.

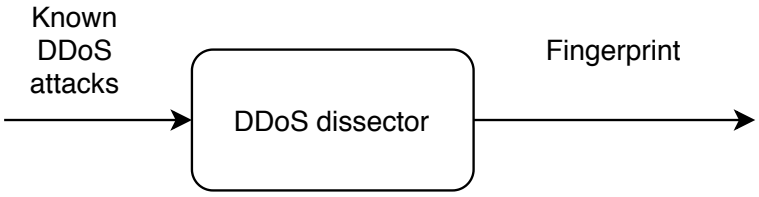

Figure 3.2: Overview of the dissector

I should be noted that a dissector, as described in this section, is not necessary for our rule generation algorithm. Our algorithm for generating the BGP Flowspec rules requires the network

characteristics of the attack, which can also be extracted manually from the attack traffic. Instead, our rule generation algorithm relies on the dissector from DDoSDB. To the best of our knowledge, no other tools exist that extract the network characteristics of known DDoS attacks. Therefore, it is used for our rule generation algorithm.

3.1.1 Input

The input of the DDoS dissector is a collection of known DDoS attacks. For this research, the DDoS traffic required will be retrieved from DDoSDB [68]. The network traffic used as a source to our experiment originates from DDoSDB [68] as well. The data on DDoSDB is published on the platform in the form of an attack trace. The trace can be read from various formats, including pcap, pcapng, netflow v5, v9, IPFIX, and sflow. To the best of our knowledge, DDoSDB is the largest public database with DDoS attacks, containing over 900 different attacks and corresponding fingerprints. Other DDoS attack data sets are known within the academic community [77, 82], but none of these are as extensive as DDoSDB.

Since a data set is used within the network of the authors of this research, this data set is biased. Several data sets have been used to test signature-based IDSs or firewalls in the last decade, with KDDCUP'99 being mostly widely used. Previous work includes statistical analysis of these data sets, and has found issues that result in a poor evaluation of anomaly detection approaches. This type of analysis has not been performed on the data set used in this research, and should be included in future work.

3.1.2 Process

As mentioned before, the dissector is responsible for generating a fingerprint of an attack vector. A fingerprint is defined as a summary of the network characteristics of that attack vector. For each attack vector on the platform, a fingerprint is generated. The dissector also anonymizes the fingerprint data in order to protect the identities of the victims, which means the destination IP address is omitted from the fingerprint. Furthermore, it should be noted that if there are multiple attack vectors present in a DDoS attack trace, each attack vector is presented separately on the platform with its own corresponding fingerprint. The source code of DDoSDB is publicly available on GitHub [66].

3.1.3 Output

The output of the dissector is the fingerprint in a JSON file format. This makes the fields easily readable and easy to convert to a BGP Flowspec rule. Below is a simplified example fingerprint retrieved from DDoSDB, with the unique ID "39b684d5e448abcece02cdf29dde6cda".

```
{
    file_type: "pcap"
    protocol: "TCP"
    additional: {
        tcp_flag: ".......A...."
    }
    src_ips: [
        {
            as: "10421"
            ip: "129.118.222.16"
            cc: "US"
        }
        {
            as: "9158"
            ip: "129.142.140.175"
            cc: "DK"
        }
        ...
    ]
    total_src_ips: 66610
    src_ports: [
        49298
        58509
        33571
        ...
    ]
    total_src_ports: 41757
    dst_ports: [
        80
    ]
    total_dst_ports: 1
    key: "39b684d5e448abcece02cdf29dde6cda"
    start_time: "2015-04-13 00:59:52"
    duration_sec: 33.30585813522339
    avg_pps: 2005.5630973026116
    avg_bps: 108300.40725434101
    multivector_key: "39b684d5e448abcece02cdf29dde6cda"
    src_ips_size: 66610
}
```

This fingerprint shows fields from several different Internet Layers in its corresponding attack. It can be seen that TCP is used as the protocol, as well as the TCP flags that are set. Furthermore, it contains the entire set of source IPs, source ports and destination ports. For these sets, the fingerprint also has a field that tells the total number of instances from these fields. Lastly, the bottom fields of the fingerprint contain extra information such as the key field, start time and duration, packets per second (pps) and bytes per second (bps) and the multivector key. If an attack is a multivector attack, each vector will have it's own fingerprint. However, the vectors will have the key of their corresponding multivector fingerprint in this field. Different fingerprints include additional fields, such as the ICMP type and ICMP code.

3.2 Rule set generator

The next component for generating the rule set is the actual generator of the rule set based on the fingerprint. This component generates all BGP Flowspec rules necessary to specify the attack flow as described in the fingerprint. An expanded overview of this component can be seen below.

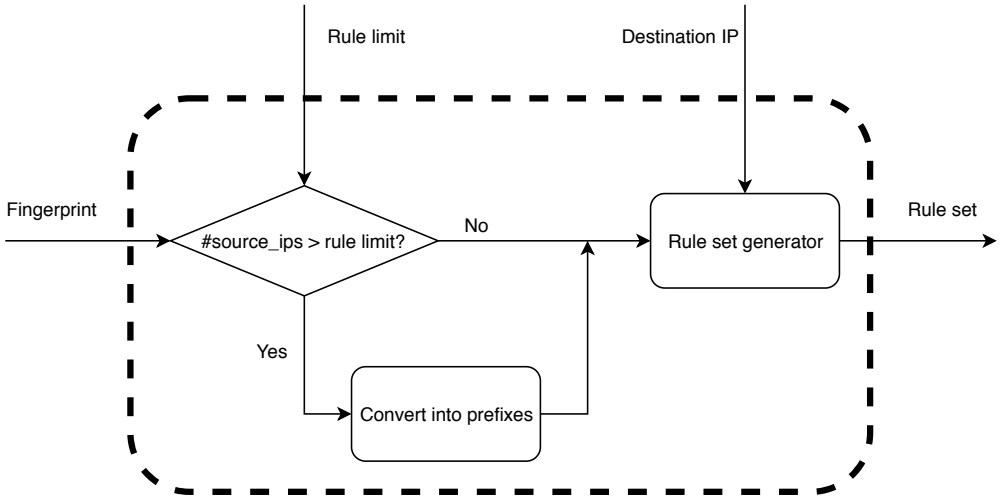

Figure 3.3: An expanded overview of the rule set generator.

3.2.1 Input

The rule generation module has multiple input values: the fingerprint that is generated by the dissector (in JSON format), the maximum number of rules that can be installed on the router at once, and the destination IP for each rule (since this is not present in the fingerprint).

The rule limit of the router is used as an input to this component, since different hardware vendors have different rule limits. For example, Cisco, a major network hardware vendor, has a hardware lineup that is designed for IXP-level routing. These routers use BGP for communication with other networks and are part of the Series Aggregation Services platform [8]. The operating system that is used on these routers (IOS XR) involves a limit of 3.000 BGP Flowspec rules that can be installed at a given time. When examining various existing DDoS attack data sets, it can be seen that a DDoS attack often originated from many sources. For example, the data retrieved from DDoSDB [68] shows that many attacks originate from as much as 10.000 source IPs. According to the BGP Flowspec standard [60], the type 2 NLRI can be used to define a flow according to a source prefix. However, a given BGP Flowspec rule can only define one source prefix. This means that an attack vector with n source IP addresses requires a rule set of size n. Therefore, filtering 10.000 source IP addresses is infeasible on the aforementioned Cisco routers. Further documentation exists on Junos OS [37], the operating systems used by Juniper routers. Here, a limitations of 6000 BGP Flowspec rules is specified. In order to ensure reproducibility of this research, the rule limit of the router is used as a generic input. It should be stated that reducing the number of rules is possible by combining IP addresses into larger prefixes. Technically, using this method, any set of IP addresses can be reduced to any size. However, this increases the mitigation impact, as larger prefixes have a higher probability of including legitimate IP addresses as well.

3.2.2 Process

Before the rule set can be generated, the limitation on the number of rules needs to be addressed. As mentioned previously, only one source prefix can be specified in each rule. This means that, when the set of source IPs is larger than the maximum number of rules that can be installed, this source IP set needs to be reduced. An algorithm was created to realize this reduction. The algorithm takes a set of IP addresses and a maximum rule amount, and outputs a new set of IP prefixes with a size smaller than the maximum rule amount. Pseudocode for this algorithm can be seen below:

Algorithm 1 Reduce source IP prefix list

INPUT: Set of IP addresses ip_list
INPUT: Maximum rule amount max_rules
  $current\_prefix = 32$
  Sort ip_list
  $result\_list = ip\_list$
  while len($result\_list$) $>$ max_rules **do**
    $current\_prefix = current\_prefix - 1$
    Convert ip_list to $/current\_prefix$
    if duplicates occur in this list **then**
      Group duplicates together and add to $result\_list$
    end if
  end while
OUTPUT: Set of IP prefixes that does not exceed max_rules

The algorithm does the following: it starts with a list of IP addresses and a maximum rule amount. Firstly, it sorts the list of IP addresses in order to loop through it. Then, each iteration, it check whether the size of the resulting list is higher than the maximum rule amount. If this is true, the IP set needs to be reduced further. The $current\_prefix$ starts at /32 and counts down. The first iteration, at /31, each IP address is converted to a /31 prefix. The algorithm then checks this list for duplicates. If these occur, these IP addresses can be grouped into a single /31 prefix. The duplicates of this iteration are added to the resulting list. The algorithm keeps doing this process (reducing the current prefix size and grouping addresses) until the size of the resulting list is smaller than the maximum rule amount. The code for this algorithm is available publicly on GitHub [40].

It is important to note that executing the above algorithm potentially has a result on the mitigation impact. In order to illustrate this, recall the following example: a company with IP-address 130.89.10.1 is experiencing DDoS attack traffic coming from more than 10.000 IPs. Consider that this company has a system in place that analyzes the incoming traffic and constructs BGP Flowspec rules based on the characteristics of the incoming traffic, in this case the source IP. However, since this particular attack has many source addresses, there could be a need for more than 10.000 Flowspec rules in order to properly define a network flow that corresponds to the DDoS traffic. As discussed in the limitations (2.3.2), some routers can only handle a maximum of 3000 installed BGP Flowspec rules. For this reason, defining this network flow with source IPs is infeasible. We can use prefixes in our rules in order to 'bundle' multiple source IPs together. For example, if 100 of the source IPs start with 50.62.24.xx, we can group these together by creating a Flowspec rule with type 2 field "50.62.24.0/24". In this situation, all traffic originating from this prefix is blocked. If the company receives a legitimate data packet from IP address 50.62.24.11, this data packet satisfies the condition of being in the 50.62.24.0/24 prefix. As a result, the packet

is classified as a part of the DDoS traffic flow, and thus is dropped upon entering the ISP's network. While this algorithm allows for a smaller rule set, it may result in a negative impact on the network.

Next, the rule set is generated. Recall the 12 NLRI fields that BGP Flowspec can use for filtering traffic. These fields are listed in Table 3.1.

Type 1	Destination prefix	Type 7	ICMP type
Type 2	Source prefix	Type 8	ICMP code
Type 3	IP protocol	Type 9	TCP flags
Type 4	Source or destination port	Type 10	Packet length
Type 5	Destination port	Type 11	DSCP
Type 6	Source port	Type 12	Fragmentation bits

Table 3.1: The 12 BGP Flowspec NLRI fields.

A BGP Flowspec rule is defined as any subset of the 12 NLRI fields that are presented in this table. As discussed before, one instance of each field can be used within a rule. In order to generate effective BGP Flowspec rules for mitigating DDoS attacks, it is necessary that the rule set resembles the DDoS attack traffic flow as closely as possible. This minimizes the risk of benign traffic being blocked. We will address each NLRI field and observe whether it can be used to generate a rule using the fingerprint data from DDoSDB.

1. **Destination prefix**
   As mentioned before, the data on DDoSDB is anonymized in order to protect the identities of the victims. This means the destination IP of the attack is not included in the fingerprint. However, in a real-world scenario, and ISP would use a BGP Flowspec rule set to block incoming traffic, which means the destination of that traffic is located somewhere within the ISP's network. Therefore, it is reasonable to assume the destination IP is known at the time of the application of the rule set, and can thus be included in every rule. Though this field is not available in our source data, we will proceed under the assumption that this data is available, since that would be the case in a real-world environment. Since it is not included in the fingerprint, we use the destination prefix as an input to our rule generation algorithm.

2. **Source prefix**
   As mentioned previously, only one instance of each field can be included in a BGP Flowspec rule. DDoS attacks often originate from many sources, resulting in a requirement for one rule for each source IP. However, since this field specifies a source **prefix**, multiple IP addresses can be clustered into one prefix. Doing so will result in a smaller rule set, but increases the risk of benign IP addresses being included in the filtered traffic flow.

3. **IP Protocol**
   The IP protocol of the attack is always constant for one attack vector. This means it can always be added to a BGP Flowspec rule with no extra cost. If multiple rules are necessary for defining one attack vector, the IP protocol can be added to each rule.

4. **Source or destination port**
   The distribution of ports used depends on the type of the attack. For example, in a DNS reflection attack, the source port of a packet is always 53 and the destination port is random. The dissector analyzes this, and generates a list of source ports and a list of destination ports. This results in four possible cases for the distribution of ports:

- One-to-one, where there is only one source port and one destination port;
- One-to-many, where there is only one source port and more than one destination port;
- Many-to-one, where there is more than one source port and one destination port;
- Many-to-many, where there is more than one source port and more than one destination port.

When analysis was performed on each fingerprint to observe the port distribution, it was observed that either the source or destination port list has a length of 1. Furthermore, whenever there is more than one port in either list, the list contains many ports that are distributed randomly. For example, the fingerprint listed in the previous section has 41.757 source ports and 1 destination port. Since including a list of n ports would require a rule set of n rules, it is not feasible to include a port list of that size. For this reason, the source or destination port list will only be included if it has a length of 1.

The BGP Flowspec standard allows for port filtering with NLRI types 4-6, which are either the source or destination port, specifically the destination port, and specifically the source port respectively. Since the latter two overrule the type 4 field, only types 5 and 6 will be used.

5. **Destination port**
   As mentioned previously, this field will be used if the DDoS attack is targeted towards 1 destination port.
6. **Source port**
   As mentioned previously, this field will be used if the DDoS attack originates from 1 source port.
7. **ICMP type**
   The ICMP type field is available in the fingerprint data. Therefore, if the IP protocol used is ICMP, this field will be included in each rule in our rule set.
8. **ICMP code**
   Similar to the ICMP type, the algorithm will include this field if applicable.
9. **TCP flags**
   The dissector is able to read this field from the attack trace. If the IP protocol is TCP, the TCP flags will be included in each rule in our rule set.
10. **Packet length**
    Similar to the port distribution, the fingerprint generated by the DDoSDB dissector generates a list of all packet lengths that are in the attack vector. When this list has a length of 1, the packet length will be included in each rule in our rule set.
11. **DSCP**
    The DSCP field is not available in our source data. Therefore, it will not be included.
12. **Fragmentation bits**
    The fragmentation bits field is not available in our source data. Therefore, it will not be included.

Recall that the dissector generates one fingerprint for every attack vector. Observing each NLRI field in the BGP Flowspec standard, it can be seen that, except for NLRI type 2, all fields are constant for one attack vector. This means that these fields can be included in each rule with no extra cost, regardless of the number of rules. Furthermore, hardware limitations only address the

number of rules that can be installed on a router at a given time. To the best of our knowledge, there exists no limitation regarding the number of NLRI fields within a rule. As a result, the type 2 field is what determines the size of the rule set. For example, when the set of source IP addresses is reduced to 100 prefixes in our algorithm (by clustering IP addresses into prefixes), the size of the BGP Flowspec rule set will be 100. Each rule will have a different source prefix, whereas all other fields will remain constant.

Executing the above steps will result in a rule set that does not exceed the maximum number of rules that is allowed, while using as much information as the BGP Flowspec standard allows.

3.2.3 Output

The output of our rule set generator will be in the format of an intermediate language. Each field in our rule corresponds to the types in line with RFC5575 [44]. This is a temporary way of storing the information of the rule set. As a result, the rule set can be easily parsed to any router vendor's OS language. This makes our rule generation algorithm reproducible, allowing for parsing the rule set to different router vendors' Operating Systems.

Recall the example fingerprint presented earlier in this chapter:

```
{
    file_type: "pcap"
    protocol: "TCP"
    additional: {
        tcp_flag: ".......A...."
    }
    src_ips: [
        {
            as: "10421"
            ip: "129.118.222.16"
            cc: "US"
        }
        {
            as: "9158"
            ip: "129.142.140.175"
            cc: "DK"
        }
        ...
    ]
    total_src_ips: 66610
    src_ports: [
        49298
        58509
        33571
        ...
    ]
    total_src_ports: 41757
    dst_ports: [
        80
    ]
    total_dst_ports: 1
```

```
    key: "39b684d5e448abcece02cdf29dde6cda"
    start_time: "2015-04-13 00:59:52"
    duration_sec: 33.30585813522339
    avg_pps: 2005.5630973026116
    avg_bps: 108300.40725434101
    multivector_key: "39b684d5e448abcece02cdf29dde6cda"
    src_ips_size: 66610
}
```

Our algorithm uses every field possible to generate our rule set. Before the rule set is generated, the source IP set is evaluated. In total, there are 66.610 source IP addresses in this fingerprint. If a network operator wants to allocate fewer rules to blocking this DDoS attack, the prefix algorithm (as described above) will be executed. This results in a set of prefixes that will be used for our rule set. For source IP prefix, one rule is generated. An example of one rule generated from this fingerprint can be seen here:

```
{
    'type1': '172.168.0.2/32',      // Destination address
    'type2': '129.255.253.138/32',  // Source prefix
    'type3': [6],                   // Protocol, in this case TCP
    'type5': [80],                  // Destination port
    'type9': ['ack']                // Set of TCP flags
}
```

There are 5 fields in this fingerprint that can be used by BGP Flowspec: the Destination IP (type 1), the source IP (type 2), the protocol (type 3), the destination port (type 5) and the TCP flags (type 9). Each of these fields is included in the resulting rule. Note that the fingerprint has multiple source IP addresses. In this case, one instance of the above rule will be generated for each source IP prefix. In each rule, the source IP field will be different, whereas all other fields remain the same for each rule.

3.3 Parser

The last component of the rule generation process is the parser. The rule has been generated and saved in the aforementioned format. For different hardware vendors, parser functions can be implemented that translate the rule set into the correct syntax.

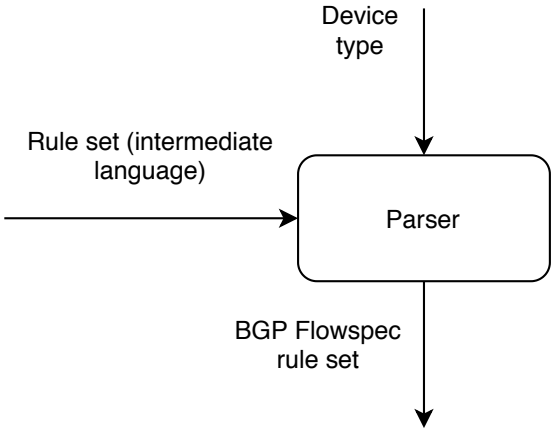

Figure 3.4: Overview of the parser.

3.3.1 Input

The input to this function is the output of the rule set generator, i.e. the rule set in dictionary format.

3.3.2 Process

In the evaluation presented later in this research, the BGP Flowspec rules are parsed to JunOS, the Operating System of Juniper Routers [37]. The function iterates through all rules, and retrieves the individual fields in each rule. This is then converted to a config file that can be installed on a Juniper router that supports BGP Flowspec. The source code for this component is publicly available on GitHub [40].

3.3.3 Output

The output for a Juniper OS configuration is defined as a "flow route", which corresponds to one BGP Flowspec rule. Each flow route has 2 components: a "match" field and an "action". The match field is a list of all NLRI fields that are used for traffic filtering in that rule and the corresponding values. The action describes what the router should do with the traffic that satisfies the match conditions.

Recall the example rule that was mentioned previously. The resulting rule filtered traffic with the following fields: the Destination IP (type 1), the source IP (type 2), the protocol (type 3), the destination port (type 5) and the TCP flags (type 9). The output of the parsed rule from this example can be seen below.

```
flow {
        term-order standard;      // Use standard order of NLRI fields
        route 258021 {
                match {
                    destination 172.168.0.2/32;
                    source 223.66.110.60/32;
                    protocol tcp;
                    destination-port 80;
                    tcp-flag ack;

                }
                then discard;
        }
}
```

For simplicity, the above rule only has one /32 source IP prefix. If the fingerprint describes multiple source IP prefixes, one route is created for each prefix. The rule above has label "258021". This label is used to reference the rule by the JunOS software and is generated randomly. The rule has match conditions on the destination IP address, source IP address, protocol, destination port and a TCP flag. When traffic satisfying these match conditions enters this router's network, it will be discarded.

3.4 Conclusions

In this chapter, we discussed the rule generation process. The goal was to use a known DDoS attack and convert this to BGP Flowspec rules. It is desirable to do this as specific and unambiguous as possible, in order to minimize the mitigation impact.

The rule generation process was divided into 3 parts: dissecting a fingerprint (summary of network characteristics) from the DDoS attack traffic, generating the rule set using this fingerprint, and parsing the rule set to the appropriate language. It should be noted that the dissector is not required for generating the rule set. However, since there existed a close collaboration between the authors of this work and the creators of DDoSDB, the dissector from DDoSDB was utilized to generate the fingerprints. The methodology and thus the results presented in this research rely completely on the validity of this dissector and we assume the output of this component is always correct. This research does not validate the correctness of the dissector. This implies that, if the DDoS dissector does not output a correct fingerprint based on an attack, the rules generated from that fingerprint will be incorrect. Future work should include validating this dissector and comparing results with other data sources.

We addressed the fact that a BGP router has a limit on the number of BGP Flowspec rules that can be installed at once. Furthermore, in a real-world scenario, the network operator may not want to allocate the router's full capacity to a single DDoS attack. For this reason, we presented an algorithm to reduce the number of rules required by converting the source IP addresses to larger prefixes. Doing this will increase the probability for a higher mitigation impact. Recall Research Question 2:

How can we write BGP Flowspec rules based on known DDoS attacks?

This chapter presented the methodology for generating BGP Flowspec rules based on the DDoS attack data from DDoSDB. Using the aforementioned prefix algorithm, the size of the rule set can be reduced if desired. In addition, our method used every available field possible for the rule. In the next chapter, we will attempt to evaluate our generated BGP Flowspec rules.

Chapter 4

Evaluation of BGP Flowspec Rules

The previous chapter presented the methodology used for generating our BGP Flowspec rules. Using existing DDoS attacks and a parser that summarizes these attacks into a fingerprint [68], we have developed a method for generating BGP Flowspec rules that filter network traffic based on this fingerprint. Furthermore, we presented an algorithm to reduce the number of source prefixes to use in a rule set. This way, fewer rules can be allocated to a single DDoS attack. However, doing so will likely result in a higher mitigation impact, since legitimate IPs are more likely to be included in these prefixes. It is up to the network operator to use this algorithm to tweak the size of the rule set.

In this chapter, we will provide a methodology for evaluating our generated BGP Flowspec rules. We will present an experiment that includes a network infrastructure with two configured BGP routers. In this infrastructure, DDoS attacks from DDoSDB will be routed through these BGP routers with our generated rules installed. Alongside this attack traffic, legitimate network traffic will be routed in order to evaluate whether our BGP Flowspec rules can discard the DDoS traffic while accepting the legitimate traffic. This chapter is divided into 2 parts: the methodology (where we will discuss the generation of legitimate network traffic, setup of the experiment, and the evaluation metrics) and the results.

4.1 Methodology

In this section, we will explain our methodology for evaluating the BGP Flowspec rules. The goal of this methodology is to provide a reliable method of evaluating the performance of our rules. The performance of our rules can be measured by two different components: (1) whether the rule set can effectively block DDoS traffic and (2) whether the rule set is successful in not blocking legitimate traffic that is passing the router at the same time.

4.1.1 Methodology for traffic generation

Since the aim is to test the DDoS traffic alongside legitimate traffic, this legitimate traffic needs to be generated. In the experiment presented in this thesis, each DDoS attack from DDoSDB is considered an individual test case. In this section, we will discuss what source files are used, how the traffic is generated and the output that results from this method.

Input

The data from DDoSDB is used as the first input to this process. As mentioned previously, DDoSDB has a database of approximately 900 attacks. For each attack, it contains the source file (in PCAP format) and the fingerprint (in JSON format). Both of these files are required for the traffic generation.

In order to achieve an accurate representation of "legitimate network traffic", a sample packet capture file is used. As a source trace file, we use `bigFlows.pcap` provided by TCPReplay [39]. According to their website, it captures "real network traffic on a busy private network's access point to the Internet" and contains 40686 flows and 132 network protocols. It sends 359.457 KB of data in 791.615 packets in 5 minutes. This file is commonly used in research for testing network traffic when a "realistic" network capture is required. It has a large variety in flows and protocols used, making it a suitable sample for our legitimate traffic. Therefore, we use this file as a source for the experiment.

However, this source file is a recording of a busy private network. Therefore, the source IP addresses from the fingerprint of the DDoS attack we are testing are not present in this particular network, i.e. the sets of IPs in the fingerprint and the `bigFlows` file are completely distinct. If one would generate BGP Flowspec rules based on the fingerprint data, the Flowspec filter would filter all traffic based on the source IP field, generating a perfect result. Therefore, in order to better simulate the capability of BGP Flowspec in distinguishing the legitimate traffic from the DDoS traffic, the decision has been made to let the two source IP sets overlap.

Overlap

As mentioned in the previous paragraph, testing the `bigFlows` file against BGP Flowspec rules from our fingerprint will always work perfectly since these two capture files are recorded in two distinct networks. The goal of this experiment is to test whether BGP Flowspec is able to correctly distinguish the DDoS traffic from the legitimate traffic. In a real-world scenario, this means that the BGP Flowspec rules need to correctly classify DDoS traffic from legitimate traffic, even if that legitimate traffic is coming from the same source as the DDoS traffic. For this reason, we introduced the **overlap**. For our test cases, we define the legitimate traffic to have an overlap with

the DDoS traffic. This means that some machines that are sending DDoS traffic are also sending legitimate traffic to our destination. Our goal is to test if BGP Flowspec is still able to distinguish the two sets considering this overlap.

The overlap is one of the key components of our research. By forcing this value, the test cases will show results for multiple scenarios per fingerprint. We also consider the worst-case scenario, i.e. when the overlap is 100%. This analysis has an expected result; our hypothesis is that in most cases, the other fields available in BGP Flowspec are sufficient in order to identify the attack traffic and classify it correctly. What we contribute by proposing in this research is to analyse the mitigation impact of the incoming DDoS attack, and to adjust the rule set accordingly.

The overlap is defined as a percentage. 50% overlap means that 50% of the packets in the `bigFlows` file originate from an IP address that is also in the source IP set of the fingerprint. An overlap of 100% means that legitimate traffic is only being sent by machines that are also sending DDoS traffic. Before each test case, the `bigFlows` file is updated accordingly. In order to properly test the performance of each fingerprint, it is necessary to test different values of overlap. For our experiment, we will test an overlap value O where $O \in \{0, 25, 50, 75, 100\}$.

Process for preparing `bigFlows`

There are two source files that are used for generating the resulting PCAP file that is used for testing. These are the aforementioned `bigFlows` file and the PCAP file of the DDoS attack retrieved from DDoSDB. A couple of preprocessing steps are required before these capture files can be tested:

- The destination IP of each packet (both `bigFlows` and the DDoS file) needs to be changed to "172.168.0.2". This is the IP of the receiving machine in our experiment setup.
- At the receiving machine, both the DDoS traffic and legitimate traffic will arrive. In order to properly analyze the results, the packets need to be separated at the receiving machine. In order to achieve this, the Differentiated Services Code Point (DSCP) value is set to 255 for each packet in the legitimate traffic. The DSCP value is used for classifying and managing Quality of Service (QoS) for IP traffic. This field is not present in our fingerprints, and thus cannot be included in our BGP Flowspec rules, making it suitable for this objective. DSCP has commonly used values defined in RFC 2475 [4]. DSCP value 255 has no defined class or goal. The DSCP value of every packet in `bigFlows` is updated to 255.
- Depending on the overlap, the source IP addresses of `bigFlows` need to be changed. For overlap percentage O, O% of the packets' source IP field is changed to a source IP from the fingerprint. The packets that are changed are selected randomly.
- Lastly, the two resulting PCAP files are combined to generate the final PCAP file that can be used for testing. When the resulting PCAP file is generated, the MAC addresses have to be changed according to the machines in the virtual environment in order to be able to forward the traffic.

The algorithm below shows how the legitimate traffic is generated.

Algorithm 2 Traffic generation algorithm

INPUT: bigFlows.pcap
INPUT: DDoS.pcap and corresponding fingerprint.json
  $overlap\_set \leftarrow [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100]$
  for *overlap* in *overlap_set* **do**
    for *packet* in *bigFlows.pcap* **do**
      $DSCP\_field \leftarrow 255$ ▷ Used for analysis later
      $destination\_IP \leftarrow 172.168.0.2$ ▷ Destination IP in the experiment
      if $random\_integer\{0, 100\} < overlap$ **then**
        $source\_IP \leftarrow$ source IP from $fingerprint$
      end if
    end for
  end for
  Change each destination IP in $DDoS.pcap$ to 172.168.0.2
  Combine resulting $bigFlows.pcap$ and $DDoS.pcap$
OUTPUT: Resulting PCAP file

Output

As mentioned before, the output of this algorithm is the final PCAP file that can be forwarded through the BGP Flowspec router. For each fingerprint and overlap percentage, this code is executed. The generation of the traffic for each fingerprint takes approximately 10-15 seconds for $O \in \{0, 25, 50, 75, 100\}$.

4.1.2 Setup for the evaluation

In the previous section, we explained that for the evaluation of our rules, the DDoS traffic will be tested alongside legitimate traffic. We showed how the DDoS traffic and fingerprint are used for the generation of the legitimate traffic. The source file for the legitimate traffic is `bigFlows`, a network capture file from a real infrastructure. We introduced the **overlap**, a method of forcing an overlap between the source IP addresses of the DDoS and legitimate traffic. In this section, we will show how the experiment is set up. We will show how the BGP Flowspec rules are injected into the BGP router, what the topology looks like and how the experiment is executed. Finally, we will discuss the evaluation metrics used for evaluatin the results in order to draw meaningful conclusions.

Failing with SURFnet and succeeding with GNS3

In order to properly perform the experiment as described in this thesis, the goal was to test our data on a real BGP Flowspec router. A collaboration was set up with SURFnet [75], the ISP for universities and research institutes in The Netherlands. SURFnet has a close relationship with the research community, which made possible that they were willing to aid this research and provide access to install BGP Flowspec rules on a router. The router was a Juniper MX240 (Figure 4.1). However, due to an unfortunate mistake from our side, this collaboration was cancelled. At a point in the execution of the experiment, BGP Flowspec rules were installed on this router that

prevented other users in the network from using their services. Lessons learned: when other users are using services in a network, be very careful and think twice about installing firewall rules in a setup like this. Nevertheless, our gratefulness goes out to SURFnet, since they intended to help us achieve our goal for this research.

In order to be able to still execute the experiment, the decision was made to simulate the same topology in Graphical Network Simulator-3 (GNS3) [30, 52]. GNS3 is a network software emulator that allows for installation of virtual machines. These machines can be used to simulate a network and test infrastructures. The Operating System that was used by the router from SURFnet (Figure 4.1) was emulated in this environment. Similar tools exist to emulate network infrastructures and test them, but GNS3 has the largest user base (with over 11 million downloads [27]) and an elaborate documentation. It is also used by large companies including Exxon, Walmart, AT&T and NASA [84].

The usage of GNS3 rather than a real machine has some implications for this research. First of all, the throughput capacity is smaller in a simulated environment. The Juniper MX240 router had a bandwidth of 100Gb/s, making it very quick to route all capture files through. The `iperf` tool [25] was used to test the virtual connection between the machines in GNS3. This tool returned a maximum bandwidth of 5 Mb/s. This means automation of all 900 fingerprints is infeasible, since it takes 2-3 weeks to route all traffic through the routers. Secondly, since a simulated environment is used, there is no guarantee that a real-world scenario would produce the same results. Therefore, it is advised that future research will conduct a similar experiment to test the effectiveness of BGP Flowspec in a real-world environment. We will elaborate on this at the end of this document.

Figure 4.1: The Juniper MX240 router

Configuration for BGP

This section will describe how the routers were configured for BGP. For a print of the full configuration, please refer to the appendix (section A.2). Recall that our experiment essentially requires 3 types of devices:

1. A machine that generates the traffic required as specified in the previous section;
2. A BGP Flowspec supported router to send the traffic through;
3. A receiving machine to capture and analyze the incoming traffic.

In order for a router to use BGP Flowspec, it needs to be configured for BGP. This means a BGP session needs to exist between two BGP-enabled routers. There are two ways routers can be

configured for BGP: as internal or external peers. When two BGP-enabled routers are located in the same Autonomous System (AS), their BGP session is called an internal BGP session. When the routers are located in separate ASs that are each others peers, a BGP session is configured on each interface of a point-to-point link. Generally, such sessions are made at network exit points with neighboring hosts outside the AS. In our experiment, it is irrelevant which type of BGP session is used. Both allow for traffic filtering using BGP Flowspec. Therefore, for simplicity, this experiment will contain an internal BGP session between two routers.

The loopback interface (lo0) is used to establish connections between internal BGP (IBGP) peers. The reason for this is that the loopback interface is always up as long as the device is operating. If a physical interface address is used instead and that interface goes up and down, the IBGP peering session also goes up and down. Since the loopback interface is always up, it provides a fallback option for the IBGP session. We will use two Juniper routers, with loopback interfaces 192.168.0.1 and 172.168.0.1. Both are configured for BGP AS 17. This number is randomly chosen and does not have any meaning.

BGP Flowspec has a way of validating a rule set. This is done as follows: the first criterion ensures that the filter is being advertised by the next-hop used by unicast forwarding for the destination address embedded in the flow route. For example, if a flow route is given as 10.1.1.1, proto=6, port=80, the receiving BGP-enabled device selects the more specific unicast route in the unicast routing table that matches the destination prefix 10.1.1.1/32. On a unicast routing table containing 10.1/16 and 10.1.1/24, the latter is chosen as the unicast route to compare against. Only the active unicast route entry is considered. This follows the concept that a flow route is valid if advertised by the originator of the best unicast route.

However, this validation can be disabled, so it allows for direct installation of BGP Flowspec routes. This is required for our experiment, in order to automatically install rules on the router. In order to achieve this, a `policy-statement` is defined. When advertising the BGP Flowspec route, this `policy-statement` is used as validation rather than the default way. We configured this policy statement to always accept the BGP Flowspec route, which allows for easy installation of the rules. The configuration for this looks like this:

```
[edit protocols]
protocols {
    bgp {
        group FLOWSPEC {
            type internal;
            local-address 10.1.1.3;
            family inet {
                unicast;
                flow {
                    no-validate NO-VAIDATION;
                }
            }
            neighbor 10.1.1.2 {
                description FS-SERVER;
            }
        }
    }
}

[edit policy-options]
```

```
policy-statement NO-VAIDATION {
    term 1 {
        then accept;
    }
}
```

There are two things that need to be configured. The first is to configure BGP and it's FLOWSPEC group. This will enable BGP Flowspec filtering on the router. The most important field here is the `no-validate` field. This points to a policy-option called "NO-VALIDATION". This means that, when a new rule is committed to the router, it will refer to this policy statement for validating the rule. If this is accepted, the rule will be installed.

The policy statement itself is configured under `policy-options`. In this configuration, we have set the statement to always accept the rule under no condition.

Rule injection

Since both routers are configured to accept all BGP Flowspec routes, any route can be specified and installed at any time. For this experiment, we will configure a static flow route. In a static flow route, the match conditions and action are set manually. An example of a static flow route looks like this:

```
route static-flow1 {
    match {
        source 10.1.1.1/32;
        destination 172.168.0.2/32;
        protocol udp;
        destination-port 53;
    }
    then discard;
}
```

The above route blocks all UDP traffic with destination port 53 that originates from a machine with IP address 10.1.1.1. For each rule, the destination IP is set to the receiving machine. Our rule generation algorithm (as explained previously) generates a rule set with as many fields filled in as possible. For example, a fingerprint with 200 source IP addresses generates 200 BGP Flowspec routes (each with a different source IP) that are injected into the router for testing.

GNS3 Topology

The topology used for the experiment can be seen in Figure 4.2. It consists of:

- **PC-1** IP 192.168.0.2
  This is a basic Linux machine that is connected to R1. This machine is responsible for the traffic generation as described previously. After the traffic is generated and the rules are injected in one of the routers, this machine forwards the traffic to PC-2. This is done using `iperf` [25].
- **R1** IP 192.168.0.1
  This is one of the two Juniper routers. It has a configured IBGP session with R2 and an

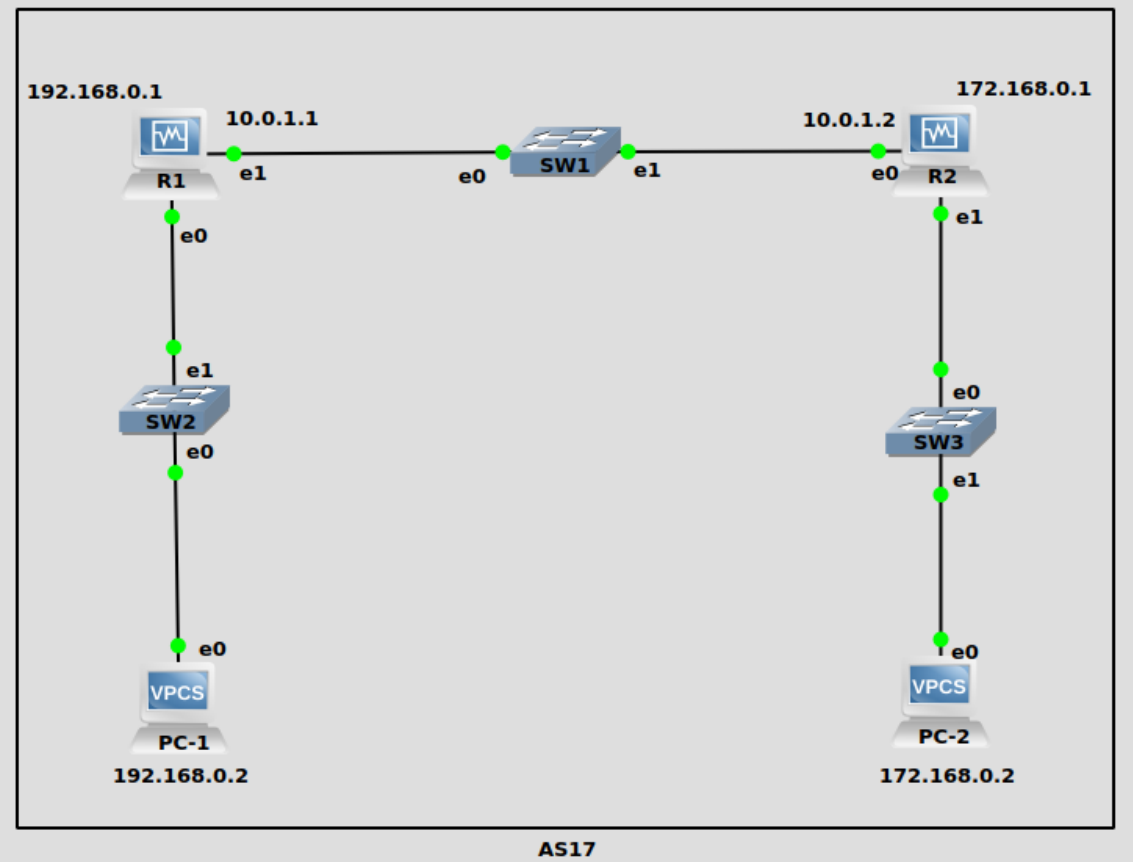

Figure 4.2: The topology in GNS3

ethernet connection to PC-1. On this router, the BGP Flowspec rule set that is generated from the fingerprint will be injected. If it receives network traffic that satisfies the traffic flow specified, this router is configured to drop that traffic. This means that in theory, the DDoS attack traffic is dropped and the legitimate traffic is routed through.

- **R2** IP 172.168.0.1
  This is the second Juniper router. It has a configured IBGP session with R1 and an ethernet connection to PC-2. Since the BGP Flowspec route only needs to be installed on one of the two routers, this router is only used for configuring the IBGP session and forwarding the traffic that it receives from R1 to PC-2.
- **PC-2** IP 172.168.0.2
  This is a basic Linux machine that is connected to R2. This machine will receive the traffic that PC-1 generates. It captures all incoming network traffic, and analyzes it immediately.

Execution process

The execution of the experiment is illustrated in Figure 4.3. Note that this sequence describes the testing process for one fingerprint. Firstly, the preprocessing steps are performed. These are

retrieving the DDoS traffic and its corresponding fingerprint from DDoSDB and generating the legitimate traffic using the fingerprint and the `bigFlows` file. This is all done on PC-1. Then, based on the fingerprint that is going to be tested, the rule set is generated, converted to the JunOS format and installed on the router R1.

Next, the test can begin. After the preprocessing is done and the rule set is installed, the receiving machine PC-2 will start to record incoming network traffic using `TCPDump` [78]. TCPDump is a well known command line packet analyzer tool. TCPDump can capture incoming network traffic and save it to a file. This makes analysis of the incoming traffic possible and allows it to be automated (since we need to loop through many fingerprints).

Next, using `TCPReplay` [79], the capture file is send over the network to PC-2 with the rule set installed on R1. R1 will discard incoming traffic that satisfies the BGP Flowspec traffic flow and accept all other traffic. At PC-2, the captured traffic is outputted to a file.

When all traffic has been sent, the C&C machine communicates to PC-2 that the recording can stop. At this moment, the output file contains all traffic that was not discarded by R1. Since the file size is very large (up to 400 MB), it is infeasible to store all recorded captures and analyze them later. Therefore, the resulting capture is analyzed, after which the capture file discarded.

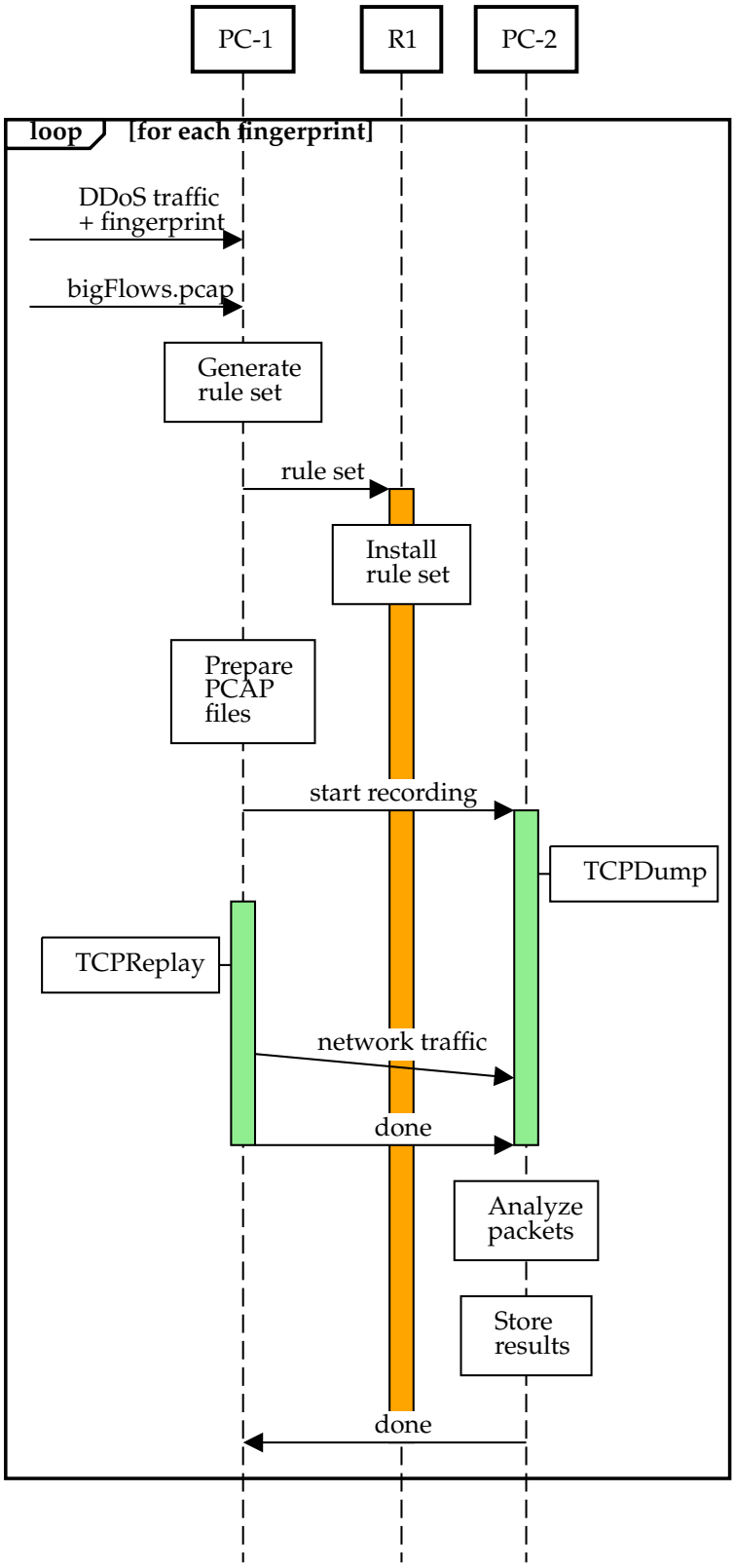

Figure 4.3: A sequence diagram of the experiment setup.

4.1.3 Evaluation metrics

The previous sections have shown how the legitimate traffic for our experiment is generated, and the setup of the experiment itself. We showed how the generated rules are injected into the BGP router, the configuration for BGP on the two routers, the topology in GNS3, and the execution of the experiment. In this section, we will elaborate on which evaluation metrics can best be used in the context of this research.

The destination machine PC-2 will receive two types of traffic, the resulting packets from the DDoS capture and the resulting packets from the `bigFlows` file. As mentioned previously, we use the DSCP value field in order to distinguish the two types at this point, since BGP Flowspec is unable to read and thus filter traffic based on this value. The incoming capture file is analyzed, and each packet is counted. There are 4 options to classify all the traffic:

- True positive: DDoS packets that are discarded by the BGP Flowspec rules;
- True negative: Legitimate packets that are received at the destination;
- False positive: Legitimate packets that are discarded by the BGP Flowspec rules;
- False negative: DDoS packets that are received at the destination.

Since we are evaluating at the receiving machine, the number of true negatives and false negatives can be counted. From these values, the number of true positives and false positives can be derived (by subtracting the true/false negatives from the total number of packets).

Using the above classification, the results can be evaluated. For each fingerprint, four different evaluation metrics are calculated. These are the False Positive Rate (FPR), True Positive Rate (TPR), Accuracy (ACC) and Precision (or Positive Predicted Value (PPV)). These are calculated as follows:

$$FPR = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

In classification studies, the choice of which evaluation metric (FPR, TPR, ACC, PPV) to use depends on the context of the research. For this research, the Precision (or Positive Predicted Value) is the most useful. By definition, Precision is the fraction of relevant instances among the retrieved instances. In the case of this experiment, the Precision is the fraction of DDoS packets among the total amount of packets dropped by the router (i.e. the total amount of packets classified as DDoS traffic).

In practice, Precision is used in situations where false negatives are more acceptable than false positives. In the context of this research, this means the case of DDoS packets that reach their destination is more acceptable than the case of legitimate packets that are discarded by the router. This is applicable in the context of this research; legitimate packets being dropped by the router is worse than a fraction of DDoS packets arriving in the network. The former case results in a negative mitigation impact and should therefore be avoided. For this reason, the Precision value will be used for evaluating the performance of a fingerprint.

4.2 Results

In the previous sections, we have presented the methodology used for evaluating our generated BGP Flowspec rules. We discussed the generation of legitimate network traffic, the setup for the evaluation procedure (BGP configuration, rule injection, the topology in GNS3 and the execution process) and the metrics that are used for evaluation.

In this section, we will present the results of the evaluation. Recall the testing procedure for a given fingerprint: first, the fingerprint and its corresponding attack trace are retrieved from DDoSDB. Using this fingerprint, the rule set for BGP Flowspec is generated (using our rule generation algorithm as presented in the previous chapter) and injected into the router. Next, the legitimate traffic is generated based on the percentage of overlap. After this, the two combined traffic flows are forwarded through the router to the receiving machine. At the receiving machine, the number of DDoS and legitimate packets are isolated (using the DSCP field) and counted.

As mentioned in the previous chapter, the usage of the GNS3 environment rather than a real machine implies a significant reduction in throughput. Using the `iperf` tool, the maximum throughput of the virtual connection was assessed. This tool returned a bandwidth of 5 Mb/s. This means automation of all 900 fingerprints is infeasible, since it takes 2-4 weeks to route all traffic through the routers. For this reason, the decision was made to test a sample of 100 fingerprints from the DDoSDB database. This set of fingerprints was selected at random, and includes fingerprints of all protocols that are present in the source database.

The rest of this section will elaborate on findings that emerged from the evaluation.

4.2.1 Evaluation of DDoS attack traffic

The evaluation of the DDoS traffic considers the case of true positives and false negatives. **For each evaluated fingerprint, no false negatives occurred**. Recall that, in our methodology, false negatives are DDoS attack packets that are received at the destination machine. As a result, every fingerprint had a true positive rate of 100%, since every DDoS attack packet was dropped by the BGP router. This is one of the main findings derived from the evaluation of our rules. It means the BGP Flowspec rule set was always successful in identifying the DDoS traffic and blocking it accordingly. The reason for this finding is because of our algorithm for generating rules: the rule generation algorithm considers every available field in BGP Flowspec to use in the rule. Since the fingerprint contains the summary of all network characteristics of the DDoS attack, the rule set generated by this fingerprint will also contain this information. Therefore, when the rules are generated and applied to the DDoS traffic, it always gets blocked.

As an example, consider a UDP Flood DDoS attack and corresponding fingerprint. A UDP Flood attack sends a large number of UDP packets to a specific port of the targeted system. When the target system receives a UDP packet at a particular port, it will check if any programs are running and listening for requests at that port. This means that all traffic from this UDP Flood attack is targeted at a specific port. The fingerprint corresponding to this attack will contain the protocol field with value UDP, and have a specific destination port. This is sufficient information for the rule set to classify this traffic (because the port is very specific). For this reason, no false negatives occur when this fingerprint is evaluated. For other fingerprints, the rule is either relatively generic (i.e. not having many fields) or very specific (having a lot of fields). In both cases, the DDoS attack traffic will satisfy the rule, and will thus be blocked at the BGP router. This observation (the fact that no false negatives occur) happens with every fingerprint out of the 100 fingerprints that were evaluated.

4.2.2 Evaluation of legitimate network traffic and overlap

As mentioned previously, the rule set generated by each fingerprint was successful in classifying the DDoS attack traffic and blocking it at the router. This is to be expected, since our rule generation algorithm considers every available field in BGP Flowspec to use in our rule. However, with some fingerprints, packets of the legitimate traffic were dropped by the BGP Flowspec rules at the router, meaning there were occurrences of **false positives**. Recall that, in our methodology, false positives are legitimate network traffic packets that are discarded by the BGP Flowspec rules.

Before the false positives are presented, Table 4.1 shows the distribution in protocol of `bigFlows`.

Item		Count	Percentage
IPv4		791179	99,94%
	UDP	152664	19,30%
	TCP	634795	80,23%
	ICMP	3720	0,47%
IPv6		436	0,06%

Table 4.1: Distribution of protocols in `bigFlows`

Observe that 0,06% of the packets are IPv6 packets. On the BGP router, either IPv4 or IPv6 rules can be installed. For this reason, the decision was made to omit the IPv6 packets from this experiment. Since the `bigFlows` file is not distributed equally in regards to the protocol, calculating the absolute number of false positives does not give a good representation of the evaluation of our rules. A TCP attack will naturally result in more false positives than an ICMP attack. For this reason, when plotting the false positives, we will calculate the False Positive Rate (FPR) in relation to the total number of packets from that protocol. This way, fingerprints of different protocols can be compared.

Before the results regarding the false positives are presented, we will recall the **overlap**. As mentioned previously, testing the legitimate network traffic file against our BGP Flowspec rules will always work perfectly. The reason for this is that the DDoS and legitimate traffic both originate from completely different networks. In other words, the source IP addresses from both network traffic sets are completely distinct. If we test the legitimate traffic against the BGP Flowspec rules, it will never be blocked because it originates from a different network. In order better evaluate our rule set, we decided to let the source IP addresses of the DDoS and legitimate traffic overlap partially. With the overlap active, our generated rule set will have to distinguish the two types of traffic, even when they (partly) originate from the same machines.

The overlap is defined as a percentage. An overlap of 50% means that 50% of the packets in the `bigFlows` file (legitimate traffic) originate from an IP address that is also in the source IP set of the fingerprint. An overlap of 100% means that legitimate traffic is only being sent by machines that are also sending DDoS traffic. It should be noted that the packets in `bigFlows` whose source IP address are chosen randomly.

In Figure 4.4, the FPR is shown as a percentage in relation to the overlap. This graph shows only four fingerprints, so the increase in FPR in relation to the overlap can be seen better. Each fingerprint was tested with 0%, 25%, 50%, 75% and 100% overlap.

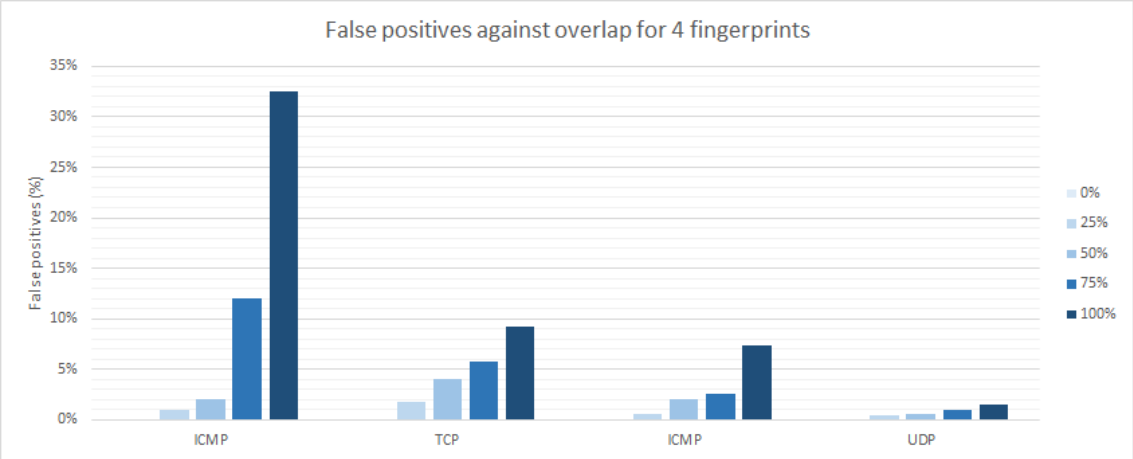

Figure 4.4: FPR plotted against the overlap for 4 different fingerprints.

Before this graph is analyzed, it is important to address the false positive percentage that is calculated. It should be noted that the Y-axis value of this graph is the percentage of false positive packets in relation to the total number of packets from that protocol. This means that, when this value is calculated, the number of false positive packets are divided by the total number of packets from that protocol (rather than divided by the total number of packets in general). Since the distribution of protocols in `bigFlows` is uneven, calculating the Y-axis value in this way will give a better overview to compare the FPR of different protocols.

Next, we will address the FPR in relation to the overlap. From Figure 4.4, we can observe the FPR increases along with the overlap, but not at the same rate for every fingerprint. The reason for this is that the packets in `bigFlows` whose source IP is changed are selected randomly. If the overlap is 50%, half of the packets are chosen at random and those packets' source IP is updated. However, since over 80% of packets are TCP (as described in Table 4.1), the probability of a TCP packet being changed is higher than any other protocol. For this reason, we observe that the false positive percentage for the second fingerprint in the graph (TCP) increases much more linearly than the other three fingerprints. The two ICMP fingerprints deviate much more from this linear increase, since only 0,47% of packets in `bigFlows` are ICMP. This also means that, if this experiment is executed again, different values will be generated for overlap percentages 25%, 50% and 75% (since the packets are selected randomly each time). This is a limitation of this method of evaluation; an improved method is to test each fingerprint many times and aggregate the results gained from the evaluation. When the overlap is 100%, the number of false positives will always be the same, since every packet in `bigFlows` is changed.

Another observation from this graph is that there is a substantial difference in FPR between attacks. This can be explained by the difference in fingerprints between attacks. Some fingerprints are very precise (i.e. the BGP Flowspec rule has a relatively large number of fields), while other fingerprints are very generic (e.g. containing only the source IP, protocol, and an ICMP code). Most UDP fingerprints resulted in a relatively low number of false positives, since the corresponding attack was directed at a specific port. On the other hand, some ICMP attacks (such as the first attack in the graph above) resulted in a high FPR. In this case, the high FPR was caused by the fingerprint only containing the "protocol" and "ICMP code" field.

4.2.3 Evaluating the worst-case scenario

As we showed in the previous section, the FPR increases along with the overlap. However, the rate of this increase depends on the protocol. Since TCP accounts for the largest number of packets in our legitimate traffic, a TCP fingerprint has a much more linear increase in FPR than a fingerprint using other protocols. We explained that, when our fingerprints would be evaluated again, the results will be different each time. However, this is not the case when the overlap is 100%. Therefore, in order to be able to compare all of the 100 evaluated fingerprints, the remainder of this section will show the evaluation considering the worst-case scenario, i.e. an overlap of 100%.

Figure 4.5 shows the FPR for all 100 evaluated fingerprints. The fingerprints in the graph are coloured according to the protocol, and sorted by FPR.

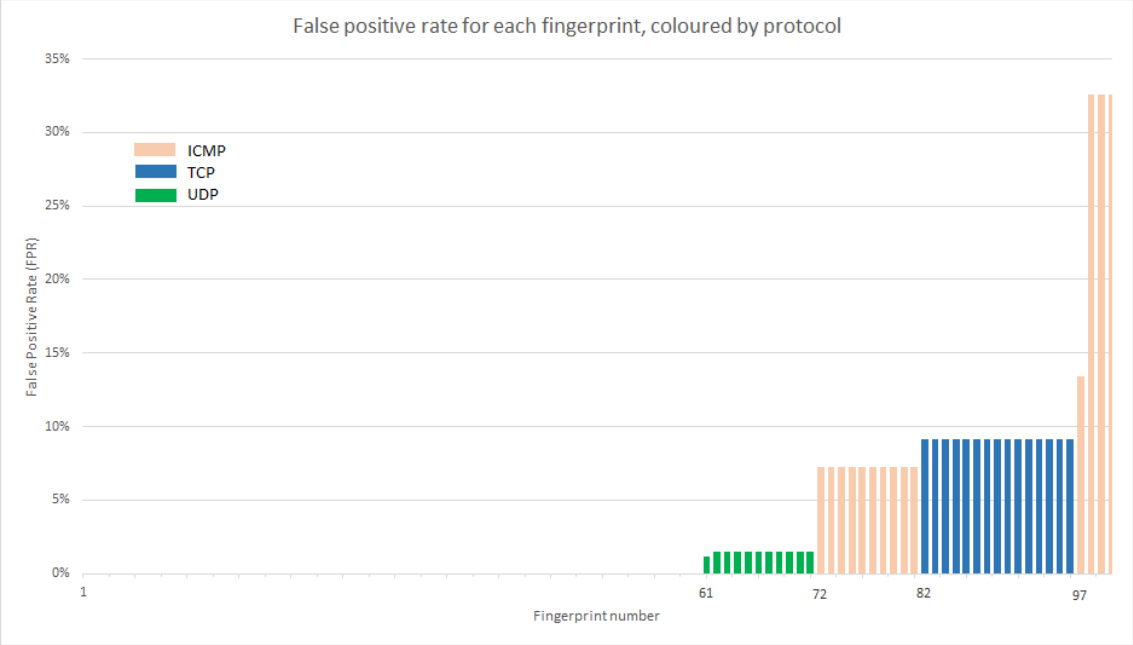

Figure 4.5: The number of false positive packets plotted against the overlap percentage.

Out of the 100 tested fingerprints, 60 did not generate any false positive occurrences. As mentioned before, none of the fingerprints resulted in false negatives. For these 60 fingerprints, this means that the BGP Flowspec rule set was always able to distinguish the DDoS traffic from the legitimate traffic, even when their source IP sets overlapped completely. This result is to be expected, since our rule generation algorithm always uses all available fields in BGP Flowspec to use in the rule. When examining these 60 fingerprints more closely, it showed that they all have a specific source or destination port. This port is not present in our legitimate traffic, resulting in 0 false positives when evaluated.

Furthermore, it can be observed that there is a clear distinction between protocols regarding the FPR. Many fingerprints result in an equal number of false positives. The reason for this is because this graph shows the worst-case scenario, i.e. the FPR for when the overlap was 100%. This means

that this graph shows the ability of our BGP Flowspec rules to correctly classify the two types of traffic (DDoS and legitimate) when the source IP sets are equal. The reason for many fingerprints generating the same number of false positives is because in these cases, the fingerprints had the same characteristics. Fingerprints 61-72 are fingerprints with UDP as protocol and 53 as source port. For each of these cases, the same number of packets in our legitimate traffic satisfies this rule. Therefore, these fingerprints have the same number of false positives.

Fingerprints 72-82 are ICMP fingerprints. In these cases, the rule only consisted of ICMP as protocol and a specific ICMP code (code 3, Destination Unreachable), which is also a relatively generic rule. Fingerprints 82-97 are TCP fingerprints. These are all HTTP DDoS attacks, meaning their fingerprints have TCP as protocol and destination port 80. All other fields were empty in our BGP Flowspec rule.

Three attacks result in a significantly high FPR, all of them ICMP attacks. These are the rightmost three bars in the graph. The reason for this is likely the fact that there are very few ICMP packets in `bigFlows` (0,47%), which can lead to inaccurate results. Furthermore, the 3 ICMP fingerprints with the highest FPR were lacking in fields to use for BGP Flowspec filtering. In these cases, the BGP Flowspec rule only contained the source/destination IP, protocol (ICMP) and ICMP type (code 11, Time Exceeded). There were many source and destination ports and no other fields available. Because of the inaccuracy of these cases, these particular fingerprints are not suitable for generating BGP Flowspec rules.

4.2.4 Confusion matrix

The table below shows the confusion matrix for the fingerprints that were tested. Since we defined the Precision score as the best evaluation metric in the context of this research, the table is sorted by this value. In order to preserve readability, the table only shows the first 10 fingerprints, which are the 10 fingerprints that scored the lowest Precision value. The full table can be found in section A.1.

ID	Protocol	Classification Performance							
		TP	FN	TN	FP	FPR	TPR	Accuracy	Precision
1	TCP	12665	0	733117	58062	7,339%	100%	92,78%	17,91%
2	TCP	12665	0	733117	58062	7,339%	100%	92,78%	17,91%
3	TCP	14918	0	733117	58062	7,339%	100%	92,80%	20,44%
4	TCP	129450	0	733117	58062	7,339%	100%	93,69%	69,04%
5	UDP	9818	0	788905	2274	0,287%	100%	99,72%	81,19%
6	UDP	19701	0	788905	2274	0,287%	100%	99,72%	89,65%
7	UDP	20004	0	788905	2274	0,287%	100%	99,72%	89,79%
8	ICMP	11192	0	789968	1211	0,153%	100%	99,85%	90,24%
9	TCP	909760	0	733117	58062	7,339%	100%	96,59%	94,00%
10	ICMP	19134	0	789968	1211	0,153%	100%	99,85%	94,05%

Table 4.2: The protocol, confusion matrix, FPR, TPR, Accuracy and Precision for 10 fingerprints. This table is sorted by the Precision score.

This table shows the 10 fingerprints that generated the lowest Precision value when evaluated

using our methodology. The first 3 fingerprints are anomalies in regards to their Precision value. After evaluating the corresponding fingerprints, all 3 contained the fields below:

```
route 194486 {
    match {
        destination 172.168.0.2/32;
        source x.x.x.x/32;
        protocol tcp;
        destination-port 80;
    }
    then discard;
}
```

As can be seen, this fingerprint belongs to a HTTP DDoS attack. It only has the protocol and destination port fields. Since we consider the worst-case scenario in this case, the source IP field is not useful for classifying the DDoS traffic correctly. These fingerprints generated a relatively high number of false positives, since all HTTP traffic from the legitimate traffic file was blocked by the BGP Flowspec rules. They all generated the same number of false positives of 58.062, since the fingerprints of these attacks were the same. Other fingerprints also have this number of false positives. However, since the first 3 fingerprints in the table have a much lower number of true positives (i.e. DDoS attack packets), the Precision value is not as low. These attacks result in a high number of legitimate traffic being blocked, which means that there is a mitigation impact on the underlying network. For this reason it can be concluded that, for this type of attack, BGP Flowspec is not accurate enough to effectively block the attack without causing mitigation impact on the network.

Similar to Figure 4.5, the number of false positives is similar for many fingerprints in this table. The reason for this is because this table shows the results for 100% overlap, meaning the source IP field is not usable for classifying the DDoS traffic. The difference in the Precision value is caused by the number of True Positives; fingerprints that have a relatively low number of DDoS packets (TP) and a high number of False Positives score a low precision, whereas fingerprints with a high number of True Positives score a high precision. Since the overlap is 100% in this table, the sets of source IP addresses of both the DDoS traffic and the legitimate traffic contained the same values. The evaluation of these cases is whether the BGP Flowspec rule set was able to correctly classify the two types of traffic, when they both originated from the same source. It can be seen that, apart from the first 4 fingerprints, the rule set scores relatively well in doing so. However, the anomalies in this table should be considered. Not every fingerprint that we evaluated is considered suitable for a BGP Flowspec rule set.

Figure 4.6 shows the Precision value for all 100 evaluated fingerprints. The graph is sorted by this Precision value.

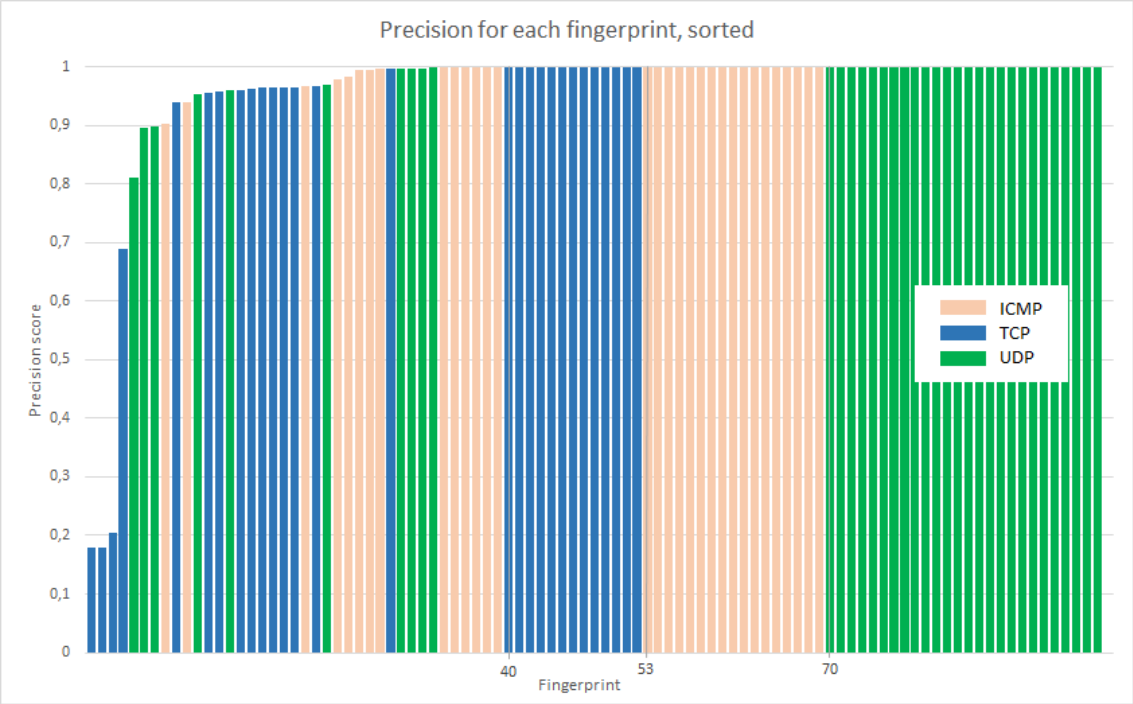

Figure 4.6: The precision for each fingerprint.

It can be observed that a small number of fingerprints score a low precision. However, these fingerprints had an Accuracy score over 92%. In the context of this research, it means that in these cases, there is a high probability that the DDoS traffic gets discarded upon entry to the network, but there is also a high probability that legitimate traffic gets classified as DDoS traffic. The latter effect results in a high mitigation impact on the network, which is undesired. Since no False Negatives occurred, the fingerprints that resulted in a high number of False Positives scored a low Precision value. These were only a few cases; 90 of the 100 evaluated fingerprints scored a Precision value over 95%. This can be clearly observed in the graph.

The difference in precision for each protocol can also clearly be seen. The leftmost four fingerprints are all TCP. Observe that TCP fingerprints generally score the lowest precision, while UDP fingerprints generally score the highest precision. From fingerprint 40 onward, every fingerprint scored a precision of 100%. Therefore, the rightmost 60 bars in this graph are sorted by protocol. This way, the difference in protocols can be observed better.

Apart from the leftmost 4 out of the 100 fingerprints, no fingerprint results in a precision below 80%. This graph shows that most fingerprints work very well in classifying DDoS traffic and distinguishing it from legitimate network traffic. Since we defined the Precision score as the most useful evaluation metric for the context of this research, it can be concluded that BGP Flowspec works well in classifying DDoS traffic based on this fingerprint data.

However, exceptions must be considered. A small number of our evaluated fingerprints scored very low, meaning that if applied in a real network, the mitigation impact would be above an acceptable threshold. These fingerprints, as they are used in this research, should not be used for generating a BGP Flowspec rule set. Furthermore, there is a substantial difference between pro-

tocols when evaluating the fingerprints. UDP fingerprints generally score a significantly higher Precision than TCP attacks. DDoS attacks are very different from each other. A generic solution that works for all DDoS attacks is very challenging (if not impossible) to design, and DDoS attacks should be considered and evaluated individually before being used for the generation of BGP Flowspec rules.

4.2.5 Impact evaluation for network operator

This section has presented the results of the evaluation of our generated BGP Flowspec rules based on DDoS attack fingerprints. We showed that many DDoS attack fingerprints are successful in distinguishing DDoS attack data from legitimate network traffic. When these fingerprints are used for the generation of BGP Flowspec rules, these rules correctly discard incoming DDoS attack traffic while accepting legitimate network traffic. However, some fingerprints are too generic to be utilized in this way, and will result in a high mitigation impact on the underlying network when used in BGP Flowspec rules.

A real-world scenario introduces new challenges to this problem. In a real network, the edge routers of this network use BGP for outside communication and thus are able to filter traffic using BGP Flowspec. The operator of this network is responsible for the application of rules for BGP Flowspec. Furthermore, each BGP router has a limit on the number of rules that can be installed. Since the network can experience multiple DDoS attacks at once, the network operator should decide how many rules should be allocated to mitigating a single DDoS attack. In order to decide this, the overlap as proposed in this research can be used. We propose the following methodology:

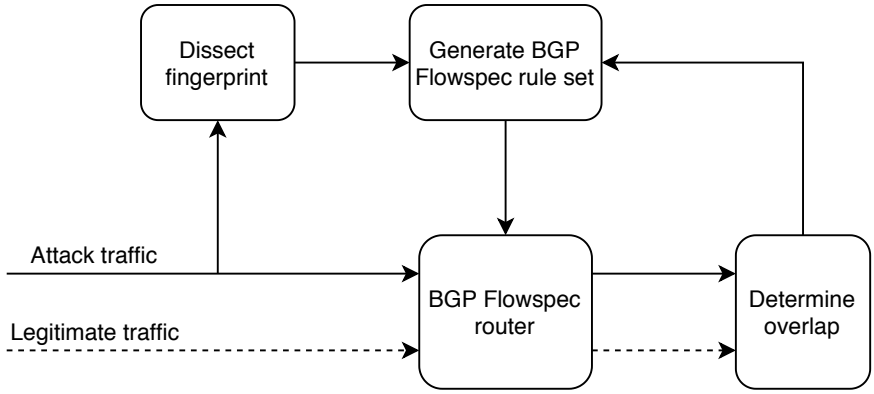

Figure 4.7: An illustration of the evaluation process for the network operator.

When a host in the network experiences a DDoS attack, the network operator is notified. Using the incoming DDoS attack traffic, the fingerprint for this traffic as well as the rule set for BGP Flowspec is generated. This process is done using the methodology as presented in this work. Next, the network operator can assess the overlap. By using the fingerprint of this DDoS attack, the operator can separate the DDoS attack traffic from all other legitimate network traffic that is entering the network. If an overlap exists, it means that some machines in this legitimate traffic are also sending DDoS traffic. Using the fingerprint, this set of machines can be determined. The next step is up to the operator to decide; blocking the machines that are in the overlap will result in complete mitigation of the DDoS attack, but might also result in mitigation impact (i.e. legitimate

traffic being blocked). If no blocking of legitimate network traffic is desired, the operator can decide to accept incoming traffic from machines in the overlap.

In the evaluation presented in this research, the overlap was forced upon the legitimate network traffic. We showed that for some fingerprints, using BGP Flowspec will result in a high mitigation impact if an overlap occurs. Therefore, the network operator should individually consider each DDoS attack and evaluate whether the usage of BGP Flowspec is desired for the situation. It should be noted that the system as proposed in Figure 4.7 was not developed by the authors of this work. Future research should implement such a system, using a real network with a BGP router and a network operator to evaluate the results. Performing this analysis might lead to BGP Flowspec being used more by network operators in their pursuit for mitigating the DDoS problem in their network.

Chapter 5

Conclusions

In the introduction of this thesis, we defined the goal of this research. The goal of this research was to investigate how DDoS attack mitigation can be improved by using BGP Flowspec. In order to achieve this goal, three research questions were defined. This chapter will conclude this research by answering the three research questions. The research questions are:

- **RQ1:** How does BGP Flowspec theoretically compare to existing DDoS mitigation solutions?
- **RQ2:** How can we write BGP Flowspec rules based on known DDoS attacks?
- **RQ3:** How effective are our BGP Flowspec rules for DDoS attack mitigation?

In order to answer the first research question, we have looked at the DDoS field in general. Combining all information gained from performing a literature study in this field, it is clear that DDoS attacks are still a big problem and challenge nowadays (and will continue to be), and it is important to understand them fully before tackling mitigation. In order to describe the mitigation of DDoS attacks, we have divided the DDoS mitigation field into 7 different levels where DDoS mitigation can take place, as well as the different methods of mitigation that can be applied at these levels. These levels are (A) the attacker, (B) the botnet, (C) the reflectors, (D) IXP, (E) ISP, (F) organization and (G) the victim's machine. At each level, some form of mitigation or prevention is possible. BGP Flowspec is a mitigation tool that is implemented at the IXP or ISP level. The related literature shows that BGP Flowspec has a high potential in blocking DDoS traffic in a manner that is more granular than blackholing practices. BGP Flowspec has received some adoption in intra-domain environments and has been shown to have good reaction time and performance. However, existing research is lacking in addressing and quantifying the mitigation impact on the network, i.e. legitimate traffic that could potentially be blocked by the BGP Flowspec rule set. In short, when comparing to other DDoS mitigation solutions, BGP Flowspec has one of the largest throughput capabilities, but lacks in granularity.

The aim of the second research question was to be able to write effective BGP Flowspec rules based on existing DDoS attack data. The goal was to use a known DDoS attack and convert this to BGP Flowspec rules as specific as possible. This known DDoS attack is retrieved from DDoSDB, on which it also has a corresponding fingerprint (a summary of all network characteristics of that attack). We have developed a method of converting this fingerprint data into BGP Flowspec rules. This algorithm uses as much information as possible that the BGP Flowspec standard supports. A fingerprint of a known DDoS attack is processed, and the rules are generated in an intermediate language. This makes reproducability for different routers possible. Furthermore,

we have developed an algorithm for the network operator that can reduce the number of rules required for defining a single DDoS attack. This algorithm bundles source IP addresses together into larger prefixes, and inserts these prefixes into the rule set. This way, the network operator can choose to allocate fewer or more rules to blocking a DDoS attack, making the rule set more lenient or strict. Reducing the number of rules will result in a higher mitigation impact on the network.

The third research question was about evaluating our BGP Flowspec rules. In order to test the effectiveness of the generated BGP Flowspec rules, an experiment was executed. In a simulated environment, an infrastructure was created to test the performance of the rule set. This infrastructure consists of an attacking machine, two BGP Flowspec routers, and a receiving machine. We evaluated 100 fingerprints by measuring the performance of the BGP Flowspec rules on the DDoS attack traffic and a sample of legitimate network traffic. Furthermore, we introduced a variable "overlap" in the source IP addresses for both types of traffic. No False Negatives occurred in the experiment, i.e. no DDoS traffic was received at the destination. However, some fingerprints caused legitimate traffic to be discarded by our BGP Flowspec rules. We calculated the Precision score for all 100 fingerprints. It was observed that most fingerprints score a relatively high Precision (90 out of 100 scored a Precision score higher than 95%). According to our evaluation, BGP Flowspec generally works better on UDP DDoS attacks than attacks of other protocols. TCP attacks score significantly lower, and in some cases, exceptionally low. If these fingerprints are used for BGP Flowspec rules and are applied in a real network, the mitigation impact would be above an acceptable threshold. These fingerprints are too generic for the generation of BGP Flowspec rules and should therefore not be used for this purpose. DDoS attacks are very different from each other; a generic solution that works for all DDoS attacks is very challenging (if not impossible) to design. In practice, DDoS attacks should be considered and evaluated individually before being used for the generation of BGP Flowspec rules.

5.1 Contribution

We will address the contribution of this work in the DDoS mitigation research field. The main contribution of this work is the rule generation. To the best of our knowledge, no tools exist that use known DDoS attack data to generate rules for BGP Flowspec. Research shows BGP Flowspec has a high potential in blocking DDoS traffic. This research proposed a methodology for generating BGP Flowspec rules out of known DDoS attacks. This rule generation method is publicly available (`https://github.com/joerikock/Master-Thesis-2019`) and can be used to generate BGP Flowspec rules based on an attack fingerprint. Furthermore, we have developed an algorithm to tweak the size of the rule set for characterizing a single DDoS attack. A network operator can use this algorithm to reduce the size of the rule set, but likely at the expense of a higher negative impact on the underlying network.

We have provided a methodology for evaluating our generated BGP Flowspec rules. By evaluating a fingerprint with its DDoS attack traffic as well as legitimate network traffic, we presented a method of evaluating the Precision score for this fingerprint. However, it should be taking into account that our evaluation is executed on a simulated environment, and that our source data set is biased. The rule generation algorithm presented in this work should be evaluated on a real infrastructure in future research.

5.2 Future work

This research has shown a BGP Flowspec rule generation methodology, as well as a methodology of evaluating these rules. This evaluation showed that BGP Flowspec rules are generally very effective in classifying DDoS traffic using a signature-based approach. However, the methodology presented in this work has its limitations. The data set used for the DDoS attack can be biased, since it is developed partly by the authors of this work. Future work should include evaluating different DDoS attack data sets using the methodology presented in order to compare the results of different data sets. Furthermore, a sample of legitimate network traffic was used to test the BGP Flowspec rules against this traffic. Future work should include evaluating other sources of legitimate network traffic and compare the results to this work.

Our rule generation algorithm is designed to interpret a DDoS fingerprint and generate a BGP Flowspec rule set. This rule set is converted into an intermediate language, allowing for parsing the rule into different vendor's OS languages. Currently, an implementation of this parser exists for converting the rule set to JunOS, the OS used by Juniper routers. Future work should consider extending this parser to allow for conversion to other OS languages, such as IOS (used by Cisco routers).

Though the work presented in this research is based on real DDoS attack data, a virtual simulated environment (GNS3) was used to execute the experiment. For this reason, the results from our evaluation are not guaranteed to be similar in a real-world environment. Therefore, future research should include a similar methodology that is executed on a real infrastructure in order to compare the outcome to the results presented in this research.

Though this work has shown that BGP Flowspec is generally an effective tool in DDoS attack mitigation using a signature-based approach, application of this work in a real-world scenario introduces new challenges. We have proposed a methodology that a network operator can use to calculate the impact of applied BGP Flowspec rules. This network operator can continuously evaluate the installed rule set and adjust it if necessary. Future work will have to implement such a system and evaluate within a real network infrastructure and a network operator. If such as system were to be implemented and tested, operators of large networks using BGP routers could be persuaded more into using BGP Flowspec in their pursuit for mitigating the DDoS problem.

Appendix A

Appendix

A.1 Detection performance table

ID	Protocol	Classification Performance							
		TP	FN	TN	FP	FPR	TPR	Accuracy	Precision
(...) d5147796c9157bc50	ICMP	104410	0	790679	500	0,0632%	100%	99,94%	99,52%
(...) 9618265f72b69ff2a	TCP	8316	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) f8ba12f6175291411	ICMP	18144	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 3552dd45b4dd3ba54	TCP	129450	0	733117	58062	7,3387%	100%	93,69%	69,04%
(...) 4310458f8e4f170cf	TCP	20481	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 7922ee510b1a7c7a5	UDP	720318	0	789469	1710	0,2161%	100%	99,89%	99,76%
(...) b3bf9e7e7d0823370	ICMP	720318	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 7b4fe5901ffabfe1c	ICMP	456707	0	790908	271	0,0343%	100%	99,98%	99,94%
(...) 4d5677ad5d12b5ccf	TCP	1632452	0	733117	58062	7,3387%	100%	97,60%	96,57%
(...) 80db8f2ffca43d794	UDP	160	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) f3d982f72fbb61218	UDP	825647	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) d375b16bbe0852386	TCP	18662	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) e8f7adbbead019c09	UDP	348	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 9ed9ce06eeb0045fa	TCP	909760	0	733117	58062	7,3387%	100%	96,59%	94,00%
(...) bc57907d78a02f1d6	TCP	13348	0	791140	39	0,0049%	100%	100,00%	99,71%
(...) 38732f0892fbbff27	UDP	122	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 2b8187cb5e90a31fb	ICMP	12685	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 57ed689c39fcd6159	UDP	1314799	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 916cc6cffb4b0e5da	TCP	5860	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 4db7ca8bcd753e846	UDP	1334362	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) d72a28ebf43546f88	TCP	1710803	0	733117	58062	7,3387%	100%	97,68%	96,72%
(...) 205e7a7d29f50e3da	ICMP	11847	0	790908	271	0,0343%	100%	99,97%	97,76%
(...) 14b162ea93923fa14	UDP	2956185	0	791173	6	0,0008%	100%	100,00%	100,00%
(...) 297074ce4accd5a23	TCP	1649665	0	733117	58062	7,3387%	100%	97,62%	96,60%
(...) 54195d80db579dc44	TCP	1236080	0	733117	58062	7,3387%	100%	97,14%	95,51%
(...) 1c17a0679befa348d	TCP	1417111	0	733117	58062	7,3387%	100%	97,37%	96,06%
(...) 64c5796844050591e	UDP	181	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) e42eb86ce015d7b0a	ICMP	975926	0	790908	271	0,0343%	100%	99,98%	99,97%
(...) b5a60d353087f88b6	UDP	36193	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 0020dee459a32a5d5	UDP	20004	0	788905	2274	0,2874%	100%	99,72%	89,79%

ID	Protocol	Classification Performance							
		TP	FN	TN	FP	FPR	TPR	Accuracy	Precision
(...) e8a301e772884d1df	UDP	99979	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 29c15191d8021eddf	UDP	410913	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 866da7e0621430fbd	UDP	1119028	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 4d59747bf32a1f967	ICMP	3123	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 6f1617a9bdfe54c06	ICMP	481735	0	790908	271	0,0343%	100%	99,98%	99,94%
(...) 7645e538efbc1ea11	UDP	2	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) f812de431d699bba7	ICMP	72451	0	790908	271	0,0343%	100%	99,97%	99,63%
(...) c541eb28115b7b968	TCP	13070	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) f7e85e8db571ed5cd	TCP	7121	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 68d8d5ed6e2e5dac5	UDP	4395918	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 85803900c88a5e271	ICMP	34843	0	789968	1211	0,1531%	100%	99,85%	96,64%
(...) b2114cec0d6362597	UDP	2497	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) a7dc3695830b084af	UDP	325	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 5ca1506b5c086c75c	TCP	1541791	0	733117	58062	7,3387%	100%	97,51%	96,37%
(...) fa10d56b7a1a70a9f	TCP	7569	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 0394f94d8230a26fd	ICMP	7391	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 9f4266a91efd0298b	ICMP	184468	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 7038dcf6f99266aa2	UDP	804769	0	788905	2274	0,2874%	100%	99,86%	99,72%
(...) f290fa07cd5f00a36	TCP	12665	0	733117	58062	7,3387%	100%	92,78%	17,91%
(...) f857118f740a0bbf1	TCP	23	0	733117	58062	7,3387%	100%	92,66%	0,04%
(...) 0f5576c1a23c07ed7	UDP	1743	0	791175	4	0,0005%	100%	100,00%	99,77%
(...) f0049abff4bfe3f11	ICMP	97635	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 71f6ea70c08af489e	ICMP	8036	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 981b8ab1b27c9853a	ICMP	14968	0	790908	271	0,0343%	100%	99,97%	98,22%
(...) b76c6e997ea31b9ed	ICMP	8961	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) cfbab9092ff6beea1	UDP	9818	0	788905	2274	0,2874%	100%	99,72%	81,19%
(...) 2fc85c49a13a09b20	TCP	8028	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 4ad9879b07826bc45	TCP	10993	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 69993431075783e86	UDP	153	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 39329988d6979810f	UDP	74980	0	788905	2274	0,2874%	100%	99,74%	97,06%
(...) 606e9770040538f80	UDP	0	0	788905	2274	0,2874%	100%	99,71%	0,00%
(...) 8600789a11f4a8123	ICMP	124333	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 80840041c9d22d5ed	ICMP	19134	0	789968	1211	0,1531%	100%	99,85%	94,05%
(...) 3984235fa92c6ddc8	ICMP	5877	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 6e890bbe922fab1c4	UDP	46987	0	788905	2274	0,2874%	100%	99,73%	95,38%
(...) d07e2257943dcebf5	ICMP	11192	0	789968	1211	0,1531%	100%	99,85%	90,24%
(...) 19add8c943c77d007	UDP	0	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 90970062b19a69553	ICMP	1505	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) a3657b5971fd952fa	UDP	295	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) eac273531f60f042a	ICMP	18330	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 9f9cc719ea7ebab11	UDP	19701	0	788905	2274	0,2874%	100%	99,72%	89,65%
(...) 71bd7c3c0733cd36b	TCP	6952	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 27940a499be26083e	ICMP	52132	0	790908	271	0,0343%	100%	99,97%	99,48%
(...) 999696c9de0d26b7b	UDP	3756314	0	791175	4	0,0005%	100%	100,00%	100,00%
(...) 43f2a69ca98ac95cc	UDP	179	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 672cab158b258432d	UDP	15340	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 0cf35aeb477914bce	TCP	1307667	0	733117	58062	7,3387%	100%	97,23%	95,75%
(...) 9fb7f8a2271611df5	TCP	14918	0	733117	58062	7,3387%	100%	92,80%	20,44%
(...) 1e4c3bce637831c20	ICMP	4939	0	791179	0	0,0000%	100%	100,00%	100,00%

ID	Protocol	Classification Performance							
		TP	FN	TN	FP	FPR	TPR	Accuracy	Precision
(...) a86b24bfc4ebda2cd	UDP	0	0	788905	2274	0,2874%	100%	99,71%	0,00%
(...) fdecab6e910af2512	TCP	12899	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 6aced25c267e1d784	ICMP	210097	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 562ae8db8f075aa7f	ICMP	649183	0	790908	271	0,0343%	100%	99,98%	99,96%
(...) 68ebf51a10953c842	UDP	0	0	788905	2274	0,2874%	100%	99,71%	0,00%
(...) 8a3729d54382034a3	TCP	13172	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 11323013f1afe775b	TCP	12665	0	733117	58062	7,3387%	100%	92,78%	17,91%
(...) 4c21fdd5a52fd827b	ICMP	5224	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 135ae8c9f98f84cbf	TCP	1632452	0	733117	58062	7,3387%	100%	97,60%	96,57%
(...) 5f5100081ce48cf00	ICMP	429002	0	790908	271	0,0343%	100%	99,98%	99,94%
(...) 5cdd12603753fecb4	TCP	13596	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) a6b36014dcc6730ab	ICMP	4825	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) cdba94672810cde76	UDP	142	0	791173	6	0,0008%	100%	100,00%	95,95%
(...) 09729031bbcbfaddc	ICMP	567381	0	790908	271	0,0343%	100%	99,98%	99,95%
(...) 3ccaff90e5cffb40f	UDP	204	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) a0635dd4a5bcd46e6	UDP	188	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) ee619db0c95efeadd	UDP	91	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 09264df216157a473	TCP	1649665	0	733117	58062	7,3387%	100%	97,62%	96,60%
(...) 7a069a25b54cc2a22	UDP	1125000	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 0c1eee178329cbed3	UDP	145	0	791179	0	0,0000%	100%	100,00%	100,00%
(...) 908d71b2c4b814335	UDP	2120446	0	788905	2274	0,2874%	100%	99,92%	99,89%

A.2 BGP configuration

The configuration below is the full configuration for router 1. Router 2 is similarly configured, as BGP neighbor with router 1. The BGP Flowspec rules are only installed on router 1.

```
root@flowspec-router> show configuration
## Last commit: 2019-11-02 08:28:28 UTC by root
version 12.1R1.9;
system {
    host-name flowspec-router;
    root-authentication {
        encrypted-password "$1$yaBz944f$A6dGAY4R.IxmKl5raDf3P.";
    }
    syslog {
        user * {
            any emergency;
        }
        file messages {
            any notice;
            authorization info;
        }
        file interactive-commands {
            interactive-commands any;
        }
    }
}
interfaces {
    em0 {
        unit 0 {
            description to-PC1;
            family inet {
                address 192.168.0.1/24;
            }
        }
    }
    em1 {
        unit 0 {
            description to-R2;
            family inet {
                address 10.0.1.1/24;
            }
        }
    }
    lo0 {
        unit 0 {
            family inet {
                address 192.168.0.11/32;
            }
        }
    }
```

```
}
routing-options {
    static {
        route 172.168.0.0/24 next-hop 10.0.1.2;
    }
    autonomous-system 17;
    flow {
        term-order standard;
        route 258021 {
            match {
                destination 172.168.0.2/32;
                source 223.66.110.60/32;
                protocol tcp;
                destination-port 80;
                tcp-flag ack;

            }
            then discard;
        }
    }
}
protocols {
    bgp {
        group internal-peers {
            type internal;
            description "Connection to R2";
            local-address 192.168.0.11;
            family inet {
                unicast;
                flow {
                    no-validate ACCEPT_FLOW;
                }
            }
            export ACCEPT_FLOW;
            peer-as 17;
            neighbor 172.168.0.11;
        }
    }
}
policy-options {
    policy-statement ACCEPT_FLOW {
        term 1 {
            then accept;
        }
    }
}
```

Bibliography

[1] M. Abu Rajab, J. Zarfoss, F. Monrose, and A. Terzis. A Multifaceted Approach to Understanding the Botnet Phenomenon. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*, IMC '06, pages 41–52, New York, NY, USA, 2006. ACM. ISBN 1-59593-561-4. doi: 10.1145/1177080.1177086. URL http://doi.acm.org/10.1145/1177080.1177086.

[2] Akamai. DDoS Attacks — Akamai. Online; accessed on 2019-03-14. URL https://www.akamai.com/us/en/resources/ddos-attacks.jsp.

[3] T. Bates, R. Chandra, D. Katz, and Y. Rekhter. Multiprotocol Extensions for BGP-4. Technical report, RFC Editor, 1 2007. URL https://www.rfc-editor.org/info/rfc4760.

[4] e. a. Blake. RFC 2475. Online; accessed on 23-10-2019. URL https://tools.ietf.org/html/rfc2475.

[5] M. S. Blumenthal and D. D. Clark. Rethinking the design of the Internet: the end-to-end arguments vs. the brave new world. *ACM Transactions on Internet Technology (TOIT)*, 1(1): 70–109, 2001.

[6] D. Breslaw and D. Bekerman. Security Glossary: Top 12 DDoS Attack Types. Online; accessed on 2019-03-14. URL https://www.incapsula.com/blog/security-glossary-top-12-ddos-attack-types-need-know.html.

[7] K. Carriello. Arm Yourself Against DDoS Attacks: Using BGP Flow Specification for Advanced Mitigation Architectures. Technical report. URL https://forum.ix.br/files/apresentacao/arquivo/131/04120930kleber.pdf.

[8] Cisco. Implementing BGP Flowspec. Technical report. URL https://www.cisco.com/c/en/us/td/docs/routers/crs/software/crs-r6-2/routing/configuration/guide/b-routing-cg-crs-62x/b-routing-cg-crs-62x_chapter_011.pdf.

[9] Cloudflare. DNS Amplification DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/dns-amplification-ddos-attack/.

[10] Cloudflare. DNS Flood DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/dns-flood-ddos-attack/.

[11] Cloudflare. Famous DDoS Attacks. Online; accessed on 2019-03-11, . URL https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/.

[12] Cloudflare. HTTP Flood DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/http-flood-ddos-attack/.

[13] Cloudflare. Ping (ICMP) Flood DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/ping-icmp-flood-ddos-attack/.

[14] Cloudflare. Low And Slow DDoS Attack Definition. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/ddos-low-and-slow-attack/.

[15] Cloudflare. Memcached DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/memcached-ddos-attack/.

[16] Cloudflare. NTP Amplification DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/ntp-amplification-ddos-attack/.

[17] Cloudflare. SSDP DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/ssdp-ddos-attack/.

[18] Cloudflare. SYN Flood DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/syn-flood-ddos-attack/.

[19] Cloudflare. UDP Flood DDoS Attack. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/udp-flood-ddos-attack/.

[20] Cloudflare. Famous DDoS Attacks — The Largest DDoS Attacks Of All Time. Online; accessed on 2019-02-14, . URL https://www.cloudflare.com/learning/ddos/famous-ddos-attacks/.

[21] cWatch. DDoS Attack — What are the consequences of a DDoS attack? Online; accessed on 2019-02-19. URL https://cwatch.comodo.com/what-is-a-ddos-attack/.

[22] C. Dietzel, A. Feldmann, and T. King. Blackholing at IXPs: On the Effectiveness of DDoS Mitigation in the Wild. In *eXploring Xfinity A First Look at Provider-Enabled Community Networks*, pages 319–332, 2016. doi: 10.1007/978-3-319-30505-9{_}24. URL http://link.springer.com/10.1007/978-3-319-30505-9_24.

[23] C. Dietzel, M. Wichtlhuber, G. Smaragdakis, and A. Feldmann. Stellar : Network Attack Mitigation using Advanced Blackholing. *CoNEXT*, 2018.

[24] DOSarrest. Schools become DDoS attack targets. Online; accessed on 2019-02-14. URL https://www.dosarrest.com/news-and-events/schools-become-ddos-attack-targets/.

[25] J. Dugan, S. Elliott, B. A. Mah, J. Poskanzer, and K. Prabhu. iPerf - The ultimate speed test tool for TCP, UDP and SCTP. Online; accessed on 23-10-2019. URL https://iperf.fr/.

[26] Europol, Dutch Police, and McAfee. The No More Ransom Project. Online; accessed on 2019-03-11. URL https://www.nomoreransom.org/en/index.html.

[27] S. Fogarty. GNS3 Network Simulator raises its game. Online; accessed on 13-11-2019. URL http://www.networkcomputing.com/networking/gns3-network-simulator-raises-its-game/d/d-id/1319279.

[28] P. García-Teodoro, J. Díaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers and Security*, 28 (1-2):18–28, 2009. ISSN 01674048. doi: 10.1016/j.cose.2008.08.003.

[29] Y. Gev, M. Geva, and A. Herzberg. Backward Traffic Throttling to Mitigate Bandwidth Floods. *2012 IEEE Global Communications Conference (GLOBECOM)*, pages 904–910, 2012. doi: 10.1109/GLOCOM.2012.6503228.

[30] J. Grossmann, D. Ziajka, and P. Pękala. GNS3. Online; accessed on 23-10-2019. URL https://www.gns3.com/.

[31] C. L. Hedrick. Routing information protocol, 1988.

[32] N. Hinze, M. Nawrocki, M. Jonker, A. Dainotti, T. C. Schmidt, and M. Wählisch. On the Potential of BGP Flowspec for DDoS Mitigation at Two Sources: ISP and IXP. *Proceedings of the ACM SIGCOMM 2018 Conference on Posters and Demos - SIGCOMM '18*, pages 57–59, 2018. doi: 10.1145/3234200.3234209. URL http://dl.acm.org/citation.cfm?doid=3234200.3234209.

[33] IBM. IBM QRadar Security Intelligence. Online; accessed on 2019-02-22. URL https://www.ibm.com/nl-nl/security/security-intelligence/qradar.

[34] T. Ibragimov, O. Kupreev, E. Badovskaya, and A. Gutnikov. DDoS attacks in Q2 2018. Online; accessed on 2019-03-11, 2018. URL https://securelist.com/ddos-report-in-q2-2018/86537/.

[35] Incapsula. Distributed Denial of Service Attack (DDoS) Definition. Online; accessed on 2019-03-11. URL https://www.incapsula.com/ddos/ddos-attacks.html.

[36] Z. E. Jamous, S. Soltani, Y. Sagduyu, and J. Li. RADAR: An automated system for near real-Time detection and diversion of malicious network traffic. *2016 IEEE Symposium on Technologies for Homeland Security, HST 2016*, pages 1–6, 2016. doi: 10.1109/THS.2016.7568889.

[37] Juniper Networks. Junos OS. Online; accessed on 11-7-2019, 2016. URL https://www.juniper.net/us/en/products-services/nos/junos/.

[38] Kaspersky Lab. Research Reveals Hacker Tactics: Cybercriminals Use DDoS as Smokescreen for Other Attacks on Business. Online; accessed on 2019-04-11. URL https://www.kaspersky.com/about/press-releases/2016_research-reveals-hacker-tactics-cybercriminals-use-ddos-as-smokescreen-for-other-attacks-on-business.

[39] F. Klassen and AppNeta. TCPReplay - Sample Captures. Online; accessed on 18-6-2019. URL http://tcpreplay.appneta.com/wiki/captures.html.

[40] J. Kock. Rule Generation - GitHub. Online; accessed on 15-11-2019. URL https://github.com/joerikock/Master-Thesis-2019/tree/master/ruleGenerator.

[41] R. Lehti, P. Virolainen, R. van den Berg, and H. von Haugwitz. AIDE - Advanced Intrusion Detection Environment. Online; accessed on 2019-02-22. URL http://aide.sourceforge.net/.

[42] C. Loibl and M. Bacher. BGP Flow Specification Multi Vendor and Inter AS Interoperability. Technical report, 2017. URL https://www.nextlayer.at/wp-content/uploads/2018/06/loibl-bacher-bgp-flowspec-interop-012017.pdf.

[43] N. Long and R. Thomas. Trends in denial of service attack technology. *CERT Coordination Center*, pages 648–651, 2001.

[44] P. Marques, N. Sheth, R. Raszuk, B. Greene, J. Mauch, and D. McPherson. Dissemination of flow specification rules. Technical report, RFC Editor, 2009.

[45] T. Matthews. Imperva Incapsula Survey: What DDoS Attacks Really Cost Businesses. Online; accessed on 2019-02-22. URL https://lp.incapsula.com/ddos-impact-report.html?_ga=2.241157382.42049987.1544434310-1793849327.1544434310#.

[46] S. McCanne and V. Jacobson. The BSD Packet Filter: A New Architecture for User-level Packet Capture. URL https://www.usenix.org/legacy/publications/library/proceedings/sd93/mccanne.pdf.

[47] J. Mirkovic and P. Reiher. A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39, 2004. ISSN 01464833. doi: 10.1145/997150.997156. URL http://portal.acm.org/citation.cfm?doid=997150.997156.

[48] J. Mirkovic, G. Prier, and P. Reiher. Attacking DDoS at the source. *Proceedings - International Conference on Network Protocols, ICNP*, pages 312–321, 2008. ISSN 10921648. doi: 10.1109/ICNP.2002.1181418.

[49] J. Moy. OSPF version 2, 1997.

[50] M. M. Najafabadi, T. M. Khoshgoftaar, A. Napolitano, and C. Wheelus. RUDY Attack : Detection at the Network Level and Its Important Features. *Association for the Advancement of Artificial Intelligence*, pages 282–287, 2015.

[51] Netfilter. iptables - iptables tree. Online; accessed on 2019-04-01. URL https://git.netfilter.org/iptables/.

[52] J. C. Neumann. *The book of GNS3: build virtual network labs using Cisco, Juniper, and more*. No Starch Press, 2015.

[53] L. H. Newman. Triton Malware Details Show the Dangers of Industrial System Sabotage — WIRED. Online; accessed on 2019-03-11, 2018. URL https://www.wired.com/story/triton-malware-dangers-industrial-system-sabotage/?CNDID=50121752.

[54] Open Information Security Foundation. Suricata — Open Source IDS / IPS / NSM engine. Online; accessed on 2019-02-22. URL https://suricata-ids.org/.

[55] R. Oppliger. *Internet security: firewalls and beyond*. 1997.

[56] OSSEC Foundation. OSSEC — Open Source HIDS Security. Online; accessed on 2019-02-22. URL https://www.ossec.net/.

[57] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks Volume 31, Issues 23–24*, 31:2435–2463, 1999.

[58] J. Pil Choi and B.-C. Kim. Net neutrality and investment incentives. *The RAND Journal of Economics*, 41(3):446–471, 8 2010. ISSN 07416261. doi: 10.1111/j.1756-2171.2010.00107.x. URL http://doi.wiley.com/10.1111/j.1756-2171.2010.00107.x.

[59] Quadrant Information Security. Sagan Main Wiki. Online; accessed on 2019-02-22. URL https://wiki.quadrantsec.com/bin/view/Main/SaganMain.

[60] Y. Rekhter, S. Hares, and T. Li. A Border Gateway Protocol 4 (BGP-4). RFC 4271, 2006. URL https://rfc-editor.org/rfc/rfc4271.txt.

[61] J. Reo. What Motivates DDoS Attackers? Online; accessed on 2019-02-22. URL https://www.corero.com/blog/690-what-motivates-ddos-attackers.html.

[62] I. Ristic. ModSecurity: Open Source Web Application Firewall. Online; accessed on 2019-04-01. URL https://modsecurity.org/.

[63] M. Roesch and others. Snort: Lightweight intrusion detection for networks. In *Lisa*, volume 99, pages 229–238, 1999.

[64] J. Ryburn. DDoS Mitigation Using BGP Flowspec. Technical report, 2014. URL http://www.linkedin.com/in/justinryburn.

[65] Samhain Labs. Samhain HIDS. Online; accessed on 2019-02-22. URL https://la-samhna.de/samhain/.

[66] J. J. Santanna. DDoSDB - GitHub. Online; accessed on 15-11-2019, . URL https://github.com/ddos-clearing-house/ddosdb.

[67] J. J. Santanna, R. van Rijswijk-Deij, R. Hofstede, A. Sperotto, M. Wierbosch, L. Z. Granville, and A. Pras. Booters—An analysis of DDoS-as-a-service attacks. In *Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on*, pages 243–251. IEEE, 2015.

[68] J. J. C. Santanna. DDoSDB: Collecting and Sharing the most important information of DDoS attacks. Online; accessed on 2019-02-22, . URL https://ddosdb.org/.

[69] H. Security. WannaCry, Petya et al: Protecting your organisation from ransomware. Online; accessed on 2019-02-22. URL https://www.huntsmansecurity.com/blog/wannacry-petya-et-al-protecting-your-organisation-from-ransomware/.

[70] Security Onion Solutions. Security Onion. Online; accessed on 2019-02-22. URL https://securityonion.net/.

[71] SiteLock. Ask a Security Professional: DDoS Attacks — Part Four: Volumetric Attacks. Online; accessed on 2019-02-22. URL https://www.sitelock.com/blog/2017/01/ddos-attacks-part4-volumetric/.

[72] SolarWinds. Log and Event Manager. Online; accessed on 2019-02-22. URL https://www.solarwinds.com/log-event-manager-software.

[73] G. Somani, M. S. Gaur, D. Sanghi, and M. Conti. DDoS attacks in cloud computing: Collateral damage to non-targets. *Computer Networks*, 109(March 2015):157–171, 2016. ISSN 13891286. doi: 10.1016/j.comnet.2016.03.022.

[74] J. Steinberger, A. Sperotto, H. Baier, and A. Pras. Collaborative attack mitigation and response: A survey. *Proceedings of the 2015 IFIP/IEEE International Symposium on Integrated Network Management, IM 2015*, pages 910–913, 2015. doi: 10.1109/INM.2015.7140407.

[75] SURFnet. SURFnet. Online; accessed on 23-10-2019. URL https://www.surf.nl/.

[76] C. Systems. Remotely Triggered Black Hole Filtering - Destination based and Source based. Online; accessed on 2019-02-22. URL https://www.cisco.com/c/dam/en_us/about/security/intelligence/blackhole.pdf.

[77] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani. A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pages 1–6. IEEE, 2009.

[78] Tcpdump. TCPDump. Online; accessed on 23-10-2019. URL https://www.tcpdump.org/manpages/tcpdump.1.html.

[79] TCPReplay. Sample Captures. Online; accessed on 17-10-2019. URL https://ieeexplore.ieee.org/abstract/document/7919486.

[80] TechTarget. What is DNS redirection? Online; accessed on 2019-02-19, . URL https://whatis.techtarget.com/definition/DNS-redirection.

[81] TechTarget. Web application firewall (WAF). Online; accessed on 2019-02-22, . URL https://searchsecurity.techtarget.com/definition/Web-application-firewall-WAF.

[82] C. Thomas, V. Sharma, and N. Balakrishnan. Usefulness of DARPA dataset for intrusion detection system evaluation. In *Data Mining, Intrusion Detection, Information Assurance, and Data Networks Security 2008*, volume 6973, page 69730G. International Society for Optics and Photonics, 2008.

[83] I. Traynor. Russia accused of unleashing cyberwar to disable Estonia — The Guardian. Online; accessed on 2019-03-11. URL https://www.theguardian.com/world/2007/may/17/topstories3.russia.

[84] Wikipedia. Graphical Network Simulator-3. Online; accessed on 13-11-2019. URL https://en.wikipedia.org/wiki/Graphical_Network_Simulator-3.