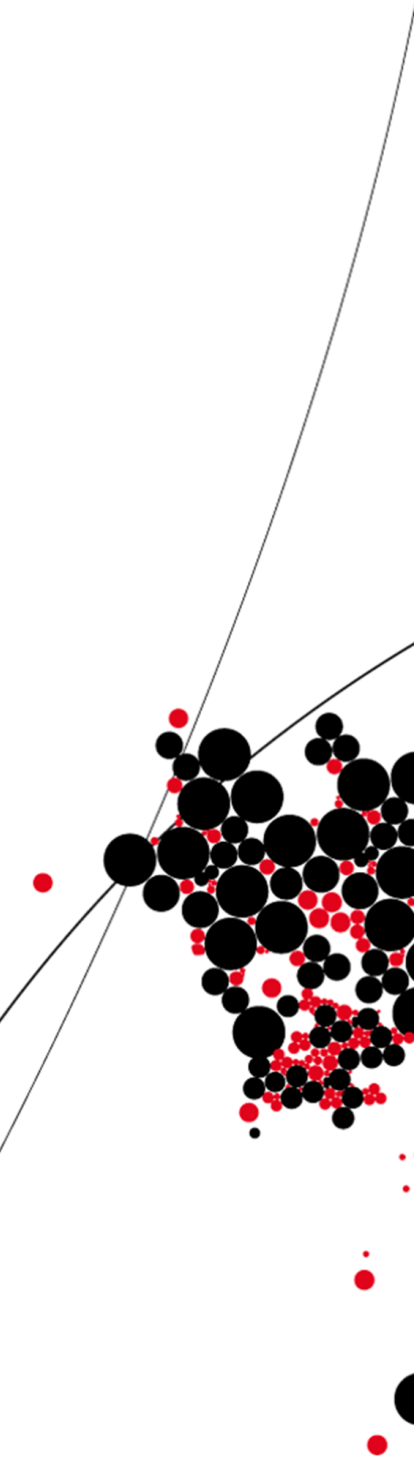
UNIVERSITY OF TWENTE.

**Faculty of Electrical Engineering,
Mathematics & Computer Science**



DDoS Attack Fingerprint Extraction Tool: Making a Flow-based Approach as Precise as a Packet-based

Jessica G. Conrads

M.Sc. Thesis
August 2019

Supervisors:
Dr. J.J. Cardoso de Santanna
Dr. C.E.W. Hesselman
Dr. J.L. Moreira

Faculty of Electrical Engineering,
Mathematics and Computer Science
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

Abstract

Twenty years after one of the first Distributed Denial of Service (DDoS) attacks happened, this type of attack is still increasing in power and frequency. For example, in the third quartile of 2018, the number of attacks increased by 71% compared to the quartile before.

There are mainly two ways of recording internet traffic to get information about attacks, packet-based and flow-based network measurements. While flow-based contains summarized information of packets and is more suitable to high-speed networks, packet-based contains more complete information for further mitigation purpose, specially attacks that are based on payload (e.g. application layer DDoS). Although usually more information leads to more precision in the defence against DDoS attacks, network operators usually prefer flow-based measurements as it requires less hardware (memory, CPU, storage). Characteristics taken from these measurements about DDoS attacks can be called DDoS fingerprints. A recent research project developed a tool, the DDoS dissector, to extract fingerprints/characteristics from network measurements. The current DDoS dissector gets fingerprints of DDoS attacks based on the destination IP address, the application protocol, the ports and application information. But it can only be used with packet-based measurement data. Therefore, network operators using flow-based measurements cannot profit of getting fingerprints by using the DDoS dissector. In this thesis, the main contribution is to make usage of flow-based measurements as precise as packet-based on the task of extracting key characteristics of DDoS attacks (fingerprint).

For the analysis, more than 250 attack traces are used for validating the methodology. The comparison is based on three requirements: (1) the number of attack vectors extracted from both network trace types (packet-based and flow-based) should be the similar, (2) the types of attack vectors extracted from both trace types should be the similar, and (3) the set of source IP addresses within the DDoS fingerprints extracted from both traces should also be similar. The methodology of the packet-based DDoS dissector is adapted and three corrections were made: (1) information about the protocols, (2) information about ICMP and (3) classifying a one port to one port attack. These changes were not enough to fulfil the requirements, so that also the packet-based DDoS dissector is changed to use the port for determining

the service instead of the protocol field. After changing this, the three requirements were fulfilled.

This thesis has four contributions. First, a better documentation of the current DDoS dissector is given, which is used by organisations such as the Dutch National High-Tech Crime Unit police and several Dutch ISPs. Second the current DDoS dissector is improved in this thesis. Third, a novel flow-based DDoS dissector is proposed. Fourth and the main contribution, the new flow-based DDoS dissector is improved to be comparable to the packet-based approach.

The results show that, in a worst case, 88% of source IP addresses in a fingerprint extracted from a flow-based measurement are the same as in a packet-based. The remaining 12% is false negative, which means that **no** potentially legitimate traffic would be blocked in case such a fingerprint would be used for blocking traffic.

Contents

Abstract iii

List of Acronyms vii

List of Figures xi

List of Tables xi

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Research Questions And Overall Methodology . . . 2
- 1.3 Framework . . . 3
- 1.4 Thesis Organization . . . 3
- 1.5 Contribution . . . 4

2 Background & Related Work 5
- 2.1 Comparing Network Measurement Types . . . 5
- 2.2 DDoS Attacks . . . 12
- 2.3 Related Work On DDoS Attack Fingerprinting . . . 16
- 2.4 Concluding Remarks . . . 19

3 Requirements and Dataset 21
- 3.1 Requirements . . . 21
- 3.2 An Existing Packet-Based DDoS Fingerprint . . . 22
- 3.3 Stopping Threshold . . . 26
- 3.4 Dataset . . . 26
- 3.5 Concluding Remarks . . . 27

4 Flow-Based DDoS Attack Fingerprints 29
- 4.1 Applying The Structure Of The Packet-Based DDoS Dissector . . . 29
- 4.2 Adapting The Code Of The Packet-Based DDoS Dissector . . . 36
- 4.3 Adjusting The Number Of Source IP Addresses . . . 40

4.4 Concluding Remarks 43

5 Conclusions And Future Work **45**
5.1 Conclusions 45
5.2 Future Work 47

Bibliography **49**

List of Acronyms

CLI	Command-line Interface
DACS	Design and Analysis of Communication Systems
DNS	Domain Name System
DDoS	Distributed Denial of Service
Gb/s	Gigabytes per second
GUI	Graphical User Interface
HTTP	Hypertext Transfer Protocol
ICMP	Internet Control Message Protocol
IP	Internet Protocol
NTP	Network Time Protocol
TCP	Transmission Control Protocol
Tbps	Terabytes per second
UDP	User Datagram Protocol
WSN	Wireless Sensor Network

List of Figures

2.1 Architecture Netflow [1] 7
2.2 Evolution of DDoS Attacks [2] 13
2.3 Elements of a DDoS attack 13

3.1 Steps of the analysis from the DDoS dissector 24
3.2 Fingerprint of a pcap file 25
3.3 Number of source IP addresses considering 520 attack vectors 27

4.1 Threshold for a 1 to 1 attack 33
4.2 First comparison between packet-based and flow-based approaches to extract DDoS fingerprints. Depicting the number of traces containing one or multiple attack vectors (and its types of attacks) 34
4.3 Second comparison packet-based and flow-based approaches to extract DDoS fingerprints. Depicting the number of traces containing one or multiple attack vectors (and its types of attacks). 37
4.4 Number of source IP addresses from flows compared to packets . . . 39
4.5 Analysing the *Certainty Threshold* for NTP attacks 42
4.6 Analysing the *Certainty Threshold* for DNS attacks 43

List of Tables

2.1 Converting of measurement types 10
2.2 Measurement types 11
2.3 DDoS attack types 14
2.4 Related work of DDoS attack fingerprints 18

4.1 ICMP types 31
4.2 First example of a port distribution 32
4.3 Second example of a port distribution 32

Chapter 1

Introduction

1.1 Motivation

In a Distributed Denial of Service (DDoS) attack, an attacker misuses a large number of devices for making a target service or device unreachable to intended users [3]. One of the first DDoS attacks was reported in 1999 [4]. Twenty years later, these types of attacks are still an increasing problem. In 2018, Netscout reported a peak of 1.7 Terabytes per second (Tbps) in size and Akamai reported that, in the third quartile of 2018, the number of attacks increased by 71% compared to the quartile before [5].

For detection and mitigation purposes, the network traffic containing a DDoS attack can be measured in several ways (e.g. packet, flow, log and sflow). The most common ways are packet-based and flow-based measurements. Flow-based measurement summarizes packets with the same characteristics between two devices (e.g. source and destination IP addresses, source and destination port and protocol value). This summarizing feature facilitates measurements and attack detection in high speed networks. Packet-based measurements contains the entire information exchanged between devices. Therefore, this type of measurement seems unique for detecting *specific* types of attacks, especially those that require payload information (e.g. application-layer-based DDoS attacks).

It is unquestionable by network operators that the detail level of packet-based measurements leads to the observation of a more precise set of attack characteristics. This set of characteristics is fundamental for the precision of follow-up mitigation strategies (e.g., firewalls, packet diversion, and scrubbing centres). The main contribution of this thesis is to make the usage of flow-based measurements as precise as using packet-based on the task of extracting key characteristics of DDoS attacks. This contribution is essential to network operators who have only flow-based measurement capabilities on determining, for example, a precise list of source Internet Protocol (IP) addresses involved in a DDoS attack (with close to zero false positive).

In this thesis, the key characteristics of a DDoS attack is called DDoS fingerprint. In literature [6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16], there is a misunderstanding on the words 'fingerprint', 'pattern', 'characterisation' and 'signature', when related to DDoS attack. For example, Lee and Shieh [8] consider a fingerprint as the path that a packet takes between the source and the destination; Osanaiye [11] considers a fingerprint as the Operating System of the device sending the attack; Shimoni and Barhom [7] consider fingerprint as the difference of packet inter-arrival time for inferring malware communication. Sanmorino and Yazid [17] have a similar understanding of fingerprints by defining this as set of key characteristics of an attack. This thesis is intended to be more generic for enabling the fingerprint to be applied to any signature or rule-based solution in future. An example for a fingerprint in this thesis is a DNS attack from source port 53 to many destination ports, from 60 source IP addresses against one destination address with the DNS query 'example.com'. This example could be extracted from network measurement that contains potentially legitimate and attack traffic.

Although using packet-based network measurement leads to more information about the attack, network operators usually use the flow-based network measurement. A recent research project about DDoS attack fingerprints developed a tool generating fingerprints based on packet-based measurement [18]. As it extracts only fingerprints from packets, network operators using flow-based measurement cannot use the DDoS dissector. By implementing a flow-based approach the network operators can get fingerprints from the DDoS dissector, but only with the information available in a flow. The goal of this thesis is to make a flow-based fingerprint as precise as a packet-based one, although lesser information about the attack can be taken from the traces than available for the packet-based approach. To achieve this goal three research questions were formulated.

1.2 Research Questions And Overall Methodology

The following section gives an overview of the methodology to answer the three research questions.

- **RQ1:** What is the state of the art on DDoS attack fingerprinting and its relation to different types of network measurements?

By answering this research question an overview on the topics of (1) the different measurement types, of (2) DDoS attacks and of (3) the fingerprinting of DDoS attacks will be given. This literature review is done to understand the background of the research. Papers with the following keywords are analysed: 'DDoS attack', 'fin-

gerprint', 'characteristics', 'pattern', 'flow-based', 'packet-based' and 'network measurements'.

- **RQ2:** How to generate a DDoS attack fingerprint based on flows?

Network operators usually use the flow-based approach to measure their internet traffic. To create the fingerprint the code of the DDoS Dissector of Santanna [18] must be adapted. The research project of the DDoS dissector will be adapted to create fingerprints based on flows. For this adaptation, a strict set of requirements are proposed. Then the entire DDoS dissector code is re-written for being compatible with flows and meet the requirements.

- **RQ3:** How comparable are DDoS attack fingerprints generated from flows and packets?

After answering RQ2 and producing a new source code for the current DDoS Dissector comes the validation (RQ3). For this validation a threefold set of requirements is defined: (1) similar number of attack vectors, (2) similar number of attack types and (3) similar list of source IP addresses. For this thesis, more than 250 attack traces are used which are available in packet and flow-based format.

1.3 Framework

The master thesis is carried out in the Design and Analysis of Communication Systems (DACS) group of the University of Twente. For evaluating the precision of the packet-based and the flow-based approach, more than 250 DDoS attack traces are considered. Those traces were downloaded from a Dutch initiative for sharing attack data and fingerprints (`http://ddosdb.org`). Three metrics are considered: (1) the number and (2) the types of attack extracted from those measurements and (3) the list of source IP addresses within fingerprints. Ideally, the aim for the flow-based approach is to achieve the same precision as using the packet-based traces. The source code is publicly available at `https://github.com/ddos-clearing-house/ddos_dissector/tree/research`.

1.4 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2 the different network measurement types are explained and several details about DDoS attacks are

reviewed that are important to understand the existing DDoS fingerprint tool. Also, the related work highlighting the misunderstandings about DDoS fingerprinting is described. In Chapter 3 the dataset and requirements including the existing DDoS fingerprint tool are described. After that, in Chapter 4, a DDoS attack fingerprinting tool based on flow measurements is proposed and the results are compared with the findings obtained by the packet based. Finally, in Chapter 5, conclusions and recommendations are given.

1.5 Contribution

In this thesis, several contributions are done. First, a better documentation of the current DDoS Dissector is given, which is currently used by organisations such as the Dutch National High Tech Crime Unit police and several Dutch ISPs. Second, the current version of the DDoS dissector is improved. Third, in this thesis a novel flow-based DDoS Dissector is proposed, so that also operators using flow-based network measurements can profit from the DDoS Dissector. Fourth, the new flow-based DDoS Dissector is improved to be comparable to the packet-based DDoS Dissector.

Overall, the results show that, in the worst case, 88% of source IP addresses in a fingerprint extracted from a flow-based measurement is the same as in a packet-based. The remaining 12% is false negative, which means that **no** potentially legitimate traffic will be blocked in case such a fingerprint would be used for blocking traffic.

Chapter 2

Background & Related Work

The goal of this chapter is to answer Research Question 1 ('What is the state of the art on DDoS attack fingerprinting and its relation to different types of network measurements?'). To understand the background of DDoS attack fingerprints from different measurement types, in the first section information about the network measurement types is given. It follows an overview about DDoS attacks. With the information from the previous sections, DDoS attack fingerprints can be explained and the misunderstandings about DDoS attack fingerprinting can be highlighted. The chapter ends with the main take aways to highlight the contribution of this work.

2.1 Comparing Network Measurement Types

There are several types of network measurement, e.g. packet, flow, log and sflow. The most used ones are the packet-based and the flow-based approach. The packet-based approach captures all packets that passed the network towards the target machine. A packet can be divided into two parts: the headers and the payload. In the headers information about the source and the destination is given, while the payload contains the actual message of the data. Therefore, the entire information about the network traffic is available. However, for the data included in the packets a high amount of storage is needed and the throughput and time needed to capture the data is very high [19]. Examples for packet-based measurement are pcap and sflow.

In the flow-based network measurement, a flow summarizes all packets with the same source and destination IP address, source and destination port and protocol value that are shared between two computers. As the information included in a flow, is lesser than in a packet-based record, the measurement is faster and the needed storage is lesser than for the packet-based approach [19]. By doing a flow-based measurement a sampling of the data can be used to reduce the network traffic that is

captured and the necessary storage. In sampling of 1:1, all packets are used in the record. When using a sampling of 1:10, only every tenth packet is captured in the flow. So, 9 out of 10 packets are not considered for the statistic of the flow. Examples of flow-based measurements are Netflow and IPFIX. In summary, the following main examples of both packet-based and flow-based formats of measurements are found, which will be detailed in the following sub-sections. After that a comparison between these main formats is performed.

- packet-based: pcap, pcapng and sflow
- flow-based: Netflow (v5 and v9) and IPFIX

2.1.1 Packet Capture (pcap)

Pcap (Packet Capture) is used to capture network traffic and to analyse / compute traffic statistics and reports including network protocols being used, communication problems, network security, and bandwidth usage. In Unix-Systems, pcap is implemented by the library libpcap. With libpcap traffic from various network media such as ethernet, serial lines and virtual interfaces can be captured and it has the same interface on every platform [20]. To overcome some limitations (e.g. less capture related information) of the library of libpcap, pcap Next Generation Dump File Format (pcapng) was developed [21].

A common example for capturing pcap and pcapng data is the tool Wireshark. All information is displayed in a Graphical User Interface (GUI) and can be filtered. More information can be found at Wireshark [22]. Tcpdump is another tool for capturing pcaps. Unlike Wireshark, Tcpdump does not have a GUI and has to be used directly in the Command-line Interface (CLI). The packets can be filtered with several parameters (i.e., IP protocol, source port and destination IP address). More information and the different parameters can be found on [23].

The format pcap is the most used packet-based measurement type used from operators, which gives all information about packets that are available. As not always all information is needed, many operators prefer the flow-based measurement. The most used tool to measure flows is Netflow.

2.1.2 Netflow

Netflow is a network protocol developed by Cisco. It collects IP traffic information and monitors the network traffic in form of flows. The first version of Netflow was developed in 1990. The most used versions are v5 and v9 [24]. In Netflow v5 there are seven key fields: (1) source interface, (2) type of service, (3) source IP address,

(4) destination IP address, (5) source layer 4 port, (6) destination layer 4 port and (7) IP protocol [25]. This version (v5) is limited to IPv4 and has a fixed format. In Netflow v9, IPv6 and a dynamic packet format are included. A Netflow v9 template has to be sent periodically [26]. Netflow flows also give the information about the number of packets summarized in one flow and the statistics of it.

The Netflow architecture consists of three parts: the exporter, the collector and the analyser as it is shown in Figure 2.1. The exporter receives the traffic from the router and exports the traffic in the Netflow format to the collector. Then the collector captures this data and stores it in the database and forwards it to the analyser. The analyser then analyses the records.

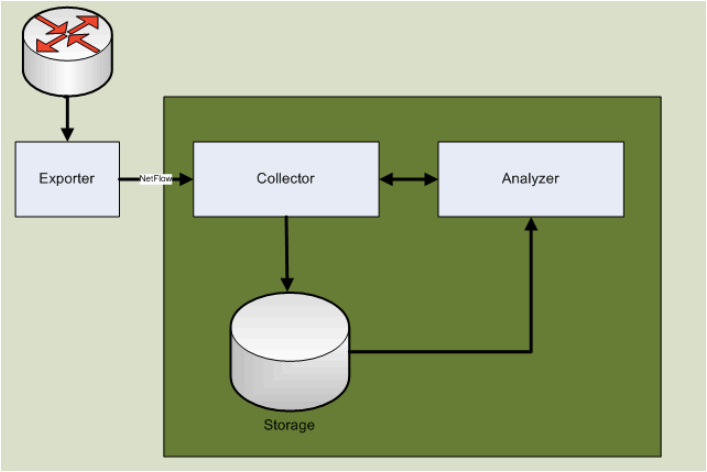


Figure 2.1: Architecture Netflow [1]

Netflow files can be read by using the tool nfdump [27]. Nfdump is similar to tcpdump (similar syntax) and is also used with the CLI. It displays Netflow files, can filter them and save the filtered versions. More information and the different options can be found on `https://manpages.ubuntu.com/manpages/disco/man1/nfdump.1.html`. The package of nfdump also includes nfcapd, which is a collector for the traffic. Following, for understanding the syntax of nfcapd, an example is provided.

```
nfcapd -b 127.0.0.1 -p 9995 -l <storage-path>
```

The *-b* specifies the bindhost and *-p* indicates the appropriate port from which the data should be collected. *-l* specifies the memory location. Another flow-based approach is IPFIX.

2.1.3 IPFIX

IPFIX was created in 2013 as a standard by the Internet Engineering Task Force (IETF) to expand the information collected by measuring internet traffic. It is a com-

mon and universal protocol for exporting IP flow information from network devices and is described in RFC7011 [28]. IPFIX collects flow information from switches, routers and other network devices that support the protocol and analyse the traffic flow information. It has the ability to integrate information which is normally sent to Syslog or SNMP in the IPFIX packet. The default port number of IPFIX is 4739. IPFIX is based on Netflow v9. Therefore, is mostly similar to Netflow (e.g. structure, output), but its field length is not fixed [29]. IPFIX can be used with nfdump [27]. There is also a packet-based measurement type, which has a surprising similar structure as Netflow and IPFIX: sFlow.

2.1.4 sFlow

sFlow is a mechanism for capturing traffic data in switches or routers with packet-based measurement. It collects its data from the device with sampling technology and therefore is suitable for high speed networks [30]. It consists of the sFlow agent (implementation of the sampling mechanism on hardware) and the sFlow Collector (central server, that collects the data from all agents). The architecture of sFlow can be compared to Netflow (Figure 2.1), although it captures packets. There are two ways to sample the sFlow: statistical packet-based sampling of switched flows and time-based sampling of network interface statistics [31] [32]. It captures '1 in n' packets from the traffic data. It copies the first bytes (v5: 128 bytes) and exports it in User Datagram Protocol (UDP). These first bytes contain the packet headers, which is necessary to construct traffic information. The focus of sFlow is on the packets and not on flows. As sFlow uses sampling technology, some IP conversions might be missing in the sFlow packets [33].

sFlows can be captured with nfdump, if nfdump is configured with enabling sFlow [34]. In that case the command sfcapd, which is comparable to nfcapd, can be used as a collector for sFlow data. Another tool to capture sFlow data is the sFlowTool [35]. Furthermore, Wireshark can be used to open sFlow files. By having one measurement format, it can be converted to other ones. This is explained in the next subsection.

2.1.5 Tools for converting from/to network measurements

The goal of the thesis is to achieve the same results for flow-based fingerprinting as for packet-based fingerprinting (RQ3). To make a comparison possible, for both fingerprintings the same data has to be used. To obtain the same data for flows and packets, one measurement type has to be chosen before converting the file to the other types. In the following different tools are explained.

Softflowd The Netflow traffic analyser Softflowd is used as an exporter for the network data. It can read the data from a pcap-file and replay it as Netflow version 1, 5 or 9. The following command can be used to export the data:

```
softflowd -r file.pcap -n 127.0.0.1:9995 -d
```

With *-r* the pcap-file is selected that should be read by the exporter. The host and the port to which the data should be exported are included with the *-n*. The *-d* notes that Softflowd should not fork and daemonise itself. A sampling of the data can be done with the addition *-s*. An example for a sampling rate of 1:10 is:

```
softflowd -r file.pcap -n 127.0.0.1:9995 -s 10
```

In case of using nfcapd as a collector, for the sampling, the command to capture the records would be:

```
nfcapd -b 127.0.0.1 -p 9995 -l <storage-path> -s -1/10
```

The *-s* provides the sampling rate. If it is negative, it will hard overwrite any device specific announced sampling rates [36]. By testing this command, it was shown that in both commands the sampling rate must be given and the sampling rate of the nfcapd needs to be negative to obtain the right outcome.

Nfpcapd is an extension of nfcapd. It can be used by configuring nfdump with the extension *–enable-readpcap*. It can directly read pcap files and export them to Netflow with the following command [27]:

```
nfpcapd -r file.pcap -l <storage-path>
```

The *-r* selects the file which should be read and *-l* provides the path to the folder where the converted file should be saved. This command splits one pcap-file into several flow-files. The time of the packets stays the original time and the flows are stored in 5-minute slots. To get one file, the flows have to be read with nfdump and saved as one flow-file. It is striking that the number of packets from the pcap-file does not match with the number of packets summarized in the flows. As it lacks documentation about the command nfpcapd, the reason for this mismatch cannot be found. Also, it is not said to which Netflow version the file is converted.

YAF consists of two tools: YAF and yafscii. YAF can be used to read a pcap file and convert it to a IPFIX-based file format [37]. The following command is used for it:

```
yaf --in <input-file> --out <output-file>
```

With $--in$ the input file is named and with $--out$ a name for the output file is given. The output file is not in a human-readable language. To see the content, it

has to be converted to a txt-file. For this, YAF uses the function yafscii. To convert a file with yafscii to a text-file, the filename is named after $--in$:

```
yafscii --in <file>
```

The Table 2.1 summarizes which tools can be used to convert one file type to another. A pcap file can be converted to a Netflow by using the tools nfdump (nfcapd) and softflowd. Besides these two tools, in literature some other tools were named (e.g. nProbe, FlowTraq) [38] [39]. From a Netflow file it cannot be converted to a pcap-file as then a lot of information is missing as a pcap-file consists of much more data than a Netflow-file. If nfdump is mentioned in the table, it includes the whole functions of the packet nfdump (e.g. nfcapd). By having an sFlow as input, it can be converted to a pcap file or to a Neflow v5. Indirectly it can also be converted to a Netflow v9 and IPFIX by first converting to Netflow v5 and then converting from Netflow v5 to v9 or IPFIX. For converting another file format to sFlow no specific tool was found. By converting one measurement type to another, the measurement types can be compared to each other. This is done in the next subsection.

Table 2.1: Converting of measurement types

input \ output	pcap file	NetFlow v5	NetFlow v9	IPFIX	sFlow
pcap	-	nfdump [27] softflowd [27] nProbe [38] FlowTraq [39]	nfdump [27] softflowd [27] nProbe [38] FlowTraq [39]	YAF [37] nProbe [38] FlowTraq [39]	
NetFlow v5		-	nfdump [27] nProbe [38]	nfdump [27] nProbe [38]	
NetFlow v9		nfdump [27] nProbe [38]	-	nfdump [27] nProbe [38]	
IPFIX		nfdump [27] nProbe [38]	nfdump [27] nProbe [38]	-	
sFlow	sFlowTool [35]	sFlowTool[35]	(indirectly)	(indirectly)	-

2.1.6 Comparing Network Measurement formats

Overall, in this section four different types of measurements are explained. In Table 2.2 information (field, tools) about the different network measurement types is summarized. For each type, the fields and the tools are given. For example, for the netflow the seven fields (source interface, type of service, source IP address, destination IP address, source layer 4 port, destination layer 4 port, IP protocol) are named. In the column tools, nfdump and softflowd are named. In the last two columns, a comparison of record size and file size is done where all records are based on the same traffic data. To achieve this the data was converted from a pcapng to other measurement types. In this example, a pcap-file contains 1264 records and has a file size of 87KB. A pcap(ng)-file has the same amount of records for the same traffic, but a file size of 109K as it contains more information for each

packet. The flow-based types (Netflow v5, v9 and IPFIX) contain only 2 records as the packets are summarized and also the file size is smaller. The Netflow v5 has a file size of 456B, while Netflow v9 has a file size of 464B and IPFIX a file size of 1.1K, as they contain more information. For sFlow no tool was found to convert from any measurement type to sflow. Therefore, no comparison can be made for this measurement type.

Table 2.2: Measurement types

			example data	
type	**fields**	**tools**	**records**	**file size**
pcap [40]	transport header (e.g. source Port, destination Port) internet layer (e.g. source IP, destination IP, protocol) link layer (e.g source address, destination address) application payload	tcpdump [23] tshark [41] wireshark [22]	1264	87KB
pcap(ng) [42] [40] [21]	same as pcap extended time stamp precision capture interface information capture statistics mixed link layer types name resolution information user comments	tcpdump [23] tshark [41] wireshark [22]	1264	109K
netflow v5 [25]	source interface type of service source IP address destination IP address source layer 4 port destination layer 4 port IP protocol	nfdump [27] softflowd [27]	2	456B
netflow v9 [43]	same as Netflow v5 source and destination MAC addresses IPv6 support improved details on VLANs & MPLS connections flow sampling, which is kind of like sFlow interface name and description (usually requires SNMP)	nfdump [27] softflowd [44]	2	464B
IPFIX [45]	source IP (v4, v6) destination IP (v4, v6) source port destination port next hop address source mac address packet count	YAF [37]	2	1.1KB
sFlow [46]	source MAC address destination MAC address type of service source and destination IP source and destination port ipv6 source and destination source and destination VLAN next hop	sFlowTool [35] nfdump (sfcapd) [27]		

Overall, in this section the packet-based (pcap, sFlow) and flow-based (Neflow, IPFIX) measurement types are explained and the different attributes are shown. Packets include more information than the others as they have payload information, but these ones need more powerful hardware. The flows have less information, but they need less hardware. This can be best seen in the example shown in the last

subsection. While a pcap(ng)-file has 1264 records, a Netflow- and IPFIX-file have only 2 records. In context of recognizing DDoS attacks the different types also have advantages and disadvantages. The header information of a packet-based file is mainly used to recognize attacks, which aim at vulnerabilities in the network stack implementation or scanning the operating system. The payload is mostly used to recognize attacks that are against vulnerable applications. As the payload is not included in the flow record, there is not much information available to detect DDoS attacks. But it can be used to see the information pattern of attacks. For many attacks, this is sufficient information. The next section explains DDoS attacks and their types.

2.2 DDoS Attacks

In a Distributed Denial of Service (DDoS) attack, an attacker uses many machines to carry out such an attack. These machines are called bots and are chosen because they are vulnerable. The attacker installs software on the bots to carry out a DDoS attack on the target with the aim that the target becomes unavailable [3].

Figure 2.2 illustrates the evolution of DDoS attacks [2]. On the one hand the attack occurrences from 2013 to 2017, while on the other hand the attack peak records in Gigabytes per second (Gb/s) from 2011 to 2017 are shown. From 2013 to 2015 in each quarter the number of attacks was below 1,000 until it rose in the 4th quarter of 2015 to almost 1,500 attacks. In the last quarter of 2016 the number of attacks was above 5,000 and in the 4th quarter of 2017, there was an increase of 14% of DDoS attacks compared to the 4th quarter of 2016 [47]. In 2018, there were 6,263 DDoS attacks until September [48]. Also, the attack peaks are rising. In 2012 the attack peak record was nearly 100Gb/s. In 2014, it was already four times higher and in 2016 the attack peak record rose above 1000Gb/s.

A DDoS attack consists of three main elements as shown in Figure 2.3: (1) the attacker, (2) the attack infrastructure, and (3) the victim/target. Note that the infrastructure used to perform an attack can be composed of three types of machines: (a) the command and control (C&C) machines, (b) the infected machines (bots), and (c) public services. Although an attacker can use only C&C machines to send an attack, usually these machines are used to access the infected machines for performing attacks. The combination of C&C and bot machines is known as botnet [49]. Lately, the usage of public services (e.g., Domain Name System (DNS), Network Time Protocol (NTP), Memcache) for reflecting and amplifying the attack traffic became very common as then the DDoS attacks become even more powerful. Between the attack infrastructure and victim there is the network measurement. This is based on getting information about the attacks and will be explained more later in this section.

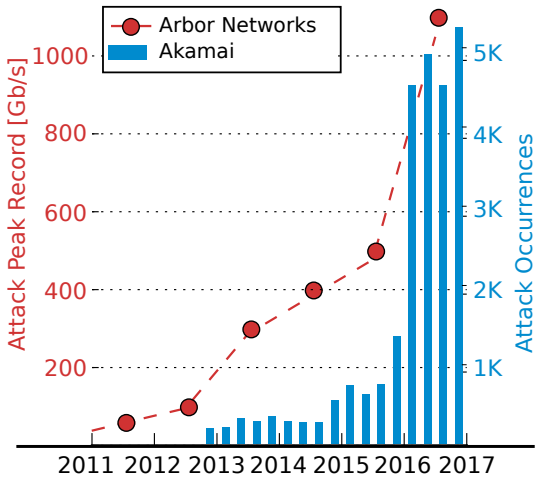


Figure 2.2: Evolution of DDoS Attacks [2]

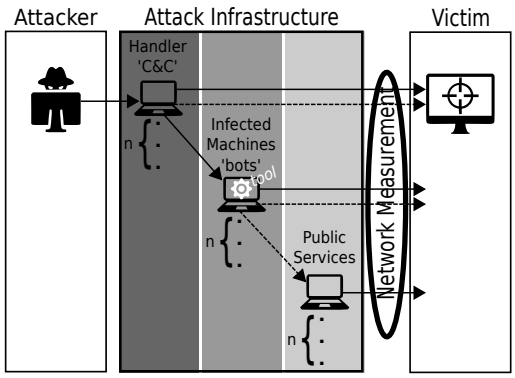


Figure 2.3: Elements of a DDoS attack

Regarding DDoS attacks, three observations must be taken into consideration. First, attackers have 'their own' attack infrastructure. Researchers have observed that there is almost **no** overlap of source IP addresses in attack infrastructures from different groups of attackers [2]. Second, the tool running in the infected machines produces attacks with specific characteristics. Third, although the traffic measured can be spoofed, the tool that generates the spoofed request has a specific algorithm that decides which IP addresses to spoof. These three observations are described to highlight that the key characteristics of a DDoS attack (fingerprint) could be used for legal attribution purposes.

There are several taxonomies for DDoS attacks to distinguish between different types of DDoS attacks. One of them is from Akamai [50]. They differ between infrastructure DDoS attacks (as UDP fragment, DNS, SYN, ACK) and application DDoS attacks (POST, GET, PUSH). Mirkovic and Reiher [51] use another taxonomy. They distinguish between several characteristics of the attacks (e.g. between the automation of an attack). It is also distinguished between semantic and brute-force.

An example for semantic is the SYN attack, where the attacker initiates multiple connections which will never be completed. Brute-Force uses a higher volume of attack packets than a semantic attack. It uses higher amount of seemingly legit transactions than the victim can handle and based on this the victim becomes out of service. Although both taxonomies have a good approach, the taxonomy by Akamai is used in the remaining of this document.

Table 2.3 shows the top 10 most common attacks and from where the attack happens:

- A means that the handler directly attacks the vicitm.
- B means that the bots attack the victim.
- C means that the attacker uses the public service.

The third column shows the amplification factor. It says how much the attack is amplified by using this service, as this is only given for attacks using services. The last column gives specific characteristics about each attack type.

Table 2.3: DDoS attack types

Attack Type	Attack from	Amplification Factor [52]	Attack Characteristics
UDP Fragment Flood	B	-	protocol IPv4
DNS	C	28 to 54	source port 53 specific DNS queries
SYN	B	-	flag TCP SYN
CLDAP	C	56 to 70	source port 389
NTP	C	556.9	source port 123 ntp reqcode
UDP	B	-	-
CHARGEN	C	358.8	source port 19
SSDP	C	358.8	source port 1900
ACK	B	high	flag TCP ACK
SNMP	C	358.8	source port 161

The most used type, according to Akamai [47], is the UDP Fragment Flood. It is one kind of the UDP floods, which means that the victim is flooded by a huge amount of UDP packets. In case of a UDP Fragment Flood, the bots send fragmented packets with the maximum size. So, the channel is flooded with regularly a few packets [53]. The attacks happen from the agent and therefore they do not have an amplification factor. As specific information, the protocol IPv4 can be named.

Another type of attack which is used often is the DNS amplification attack. The DNS amplification attack is a reflection attack and therefore it also includes the public service. The source IP addresses are forged (as the victim's IP address) and therefore the DNS server responds to the victim. If a lot of requests are sent to DNS servers, the victim is flooded by responses [54]. A DNS attack has an amplification factor between 28 and 54. As attack characteristics the port 53 can be given. Also, a specific DNS query can be named.

The SYN attack uses the weakness of the Transmission Control Protocol (TCP) for its attack. A SYN packet is sent to a port with the status listening. Usually, in the SYN packet the source IP address is given, but in a DDoS, attack the source IP addresses are usually spoofed. When the victim receives the SYN packet, it answers with the SYN/ACK packet to the given (spoofed) IP address. The victim waits until the timeout for the ACK to complete the connection process, but will never receive the ACK as the IP address is spoofed [17]. The SYN attack can be recognized by the TCP SYN flag.

Regarding the network measurement depicted in Figure 2.3, all traffic (packets) sent from the attack infrastructure:

1. goes to a single destination IP address,
2. and comes from a set of source IP addresses (without making distinction to which part of the attack infrastructure they belong),
3. which send packets with a specific IP protocol (e.g. ICMP, TCP and UDP).
4. If the IP protocol is TCP or UDP, the packets will contain source(s) and destination(s) port numbers,
5. and each port number usually has a service associated, which contains payload information (e.g. port 53: DNS; port 123: NTP; port 80: HTTP)

Therefore, a fingerprint of a DDoS attack must contain *at least* these five sets of information. It is said 'at least' because some attacks can have some specific characteristics that can be further investigated after these five characteristics are filtered from the traffic.

At the same moment in time, different parts of the attack infrastructure can be used to send different types of DDoS attacks against different services. For example, while some bots could be sending a spoofed SYN attack against a Hypertext Transfer Protocol (HTTP) server, another set of bots could be sending spoofed requests to DNS servers, which consequently will answer the request to the target machine (as a reflection and amplification attack) to random port numbers ($>$1024). In this case the target machine is suffering a multi-vector attack composed of two attack vectors: TCP SYN flood and DNS amplification attack. From these two examples can be generalized that a single attack vector will have all source IP addresses to send to a specific combination of ports. This information is crucial for the analysis done in Chapter 4. One of the four following combinations of port numbers is possible for each attack vector:

1. from many source ports to one specific destination port. For example the TCP SYN flood described.

2. from one specific source port to many destination ports. For example the DNS amplification attack described.

3. from one specific source port to one specific destination port.

4. from many source ports to many destination ports.

In this section, the general aspects and the increase of DDoS attacks are explained. It is shown that attacks can happen from bots directly or public services are involved and each kind of attacks has different characteristics. There are at least five sets of information that can be taken from attacks and the attack can happen from/to different combinations of ports. For doing e.g. a mitigation of DDoS attacks a characterisation/pattern/fingerprint of an attack is necessary. This is the topic of the next chapter.

2.3 Related Work On DDoS Attack Fingerprinting

There are several academic papers about 'fingerprints of DDoS attacks', but there is a misunderstanding on the definition and the usage. Some related words to fingerprint are characterisation, pattern, profile and signature.

Lakhdari [6] considers fingerprinting as the set of flow duration, direction, inter-arrival time, number of exchanged packets and packets size, for detecting malicious network flows and the attribution to malware families. This fingerprint is used for detection, mitigation and attribution and is meant for general malicious IP traffic. As measurement data flows are used. The fingerprint is generated by building a classification model which classifies the network traces with a machine learning algorithm.

Shimoni and Barhom [7] consider fingerprint as the inter-arrival time difference of packets for determining malware communication. They use it for identifying malware traffic and use packets as their input. The classification is also based on a learning algorithm.

Lee and Shieh [8], Yaar et al. [9], and Saurabh and Sairam [10] consider fingerprinting of DDoS attacks as the path of the packet. In this case the fingerprint is used for determining spoofed IP packets. They save their fingerprint in the IP packet header. All packets that use the same path, are marked with the same identifier. Only one packet needs to be classified as malicious. All other packets with the same identifier are then automatically classified as malicious.

Osanaiye [11] considers fingerprint as the operating system of the IP addresses involved in an attack. For different operating systems, the header field differ. The operating system fingerprinting is done active by sending probes to the true source

and passive by using the header features from incoming packets. The passive operating system fingerprint is compared to the p0f database to determine the operating system. This operating system is than compared to the results of the active measurement. In case that it differs, the IP address was spoofed.

Akella et al. [12] consider fingerprint as a pre-defined threshold of a maximum number of bytes sent to an IP address (also called traffic profile). A sampling algorithms is used to build traffic profiles. The fingerprint is used for DDoS attack detection in the ISP network. This is a common approach used in anomaly based solutions.

Also Nigam et al. [13] use traffic information for getting a profile of DDoS attacks. They use the abnormality of packet reception rate and the inter-arrival time to characterise a packet as an attack in a Wireless Sensor Network (WSN). The results are based on a simulation in which sensor nodes are used to get information.

Fachkha et al. [14] consider fingerprint as a characterisation of DNS amplification DDoS attacks. They divided their analysis into two parts: the detection component including the packet count, scanned hosts, DNS query types and requested domains and the rate of the attack. For their analysis, they use real darknet data.

Beitollahi and Deconinck [15] consider fingerprinting for ports used by services in the network. In this case the fingerprint is used to find ports with purely good traffic and to remove the limited access of them. They analyse the traffic in four phases: Control phase (analysing the traffic rate), the negotiation phase, stabilisation phase and processing phase. In the last phase, the good traffic is isolated from the attack traffic. The attack traffic only gets limited access, while the purely good traffic can process as usual.

Hussain et al. [16] have a different approach for doing a fingerprint. Their fingerprint is used to recognize repeated DDoS attack. Their approach is based on spectral characteristics of attack streams. The attack packet is defined by the environment in which it is created and is influenced by cross-traffic from the network. A unique fingerprint can be generated based on these factors. This unique fingerprint will be the same for repeated attack.

Finally, Sanmorino and Yazid [17] consider fingerprint as the source IP, destination IP, source port, destination port, transfer protocol, flow size, and number of packets. The fingerprint is created by extracting information of the flow tables. Then a detection mechanism runs. In case that a DDoS attack is identified, a handling mechanism starts to drop the packets from the attacks. They use this fingerprint for anomaly-based detection of DDoS attacks.

In Table 2.4 the related work and their definition of fingerprint/characterisation are summarized. It is also mentioned which type of network measurement is used for their research.

Table 2.4: Related work of DDoS attack fingerprints

Literature	Meaning of Fingerprint	Usage	DDoS related	Type of Network Measurement
Lakhdari [6]	flow duration, direction, inter-arrival time, number of exchanged packets and packet size	detection, mitigation and attribution	general malicious IP traffic	flow-based
Shimoni and Barhom [7]	time difference from malware communication	identifying malware traffic	general malicious traffic	packet-stream
Lee and Shieh [8]	path of the traffic	identifying/filtering spoofed IP packets	yes	packet-based
Yaar et al. [9]	path of the traffic	mitigation	yes	packet-based
Saurabh and Sairam [10]	path of the traffic	mitigation	yes	packet-based
Osanaiye [11]	operating system	preventing DDoS Attack	yes	packet-based
Akella et al. [12]	traffic profiles (e.g. total number of bytes)	detection of attacks on ISP networks	yes	packet-based
Nigam et al. [13]	traffic information (e.g. packet reception rate)	protecting the WSN network from DDoS attack	yes	
Fachkha et al. [14]	packet count, scanned hosts, DNS query type, requested domain	inferring attacks and characterisation of these	yes, but only DNS amplification	flow-based
Beitollahi and Deconinck [15]	port interface of the defense router	dectect location of attack and mitigate this	yes	
Hussain et al. [16]	spectral characteristics of attack streams	identify repeated DDoS attacks	yes	packet-stream
Sanmorino and Yazid [17]	source IP, source port, destination IP, destination port, transfer protocol, flow size and number of packets	detection and mitigation	yes	flow-based

In most of the related work, the meaning of fingerprinting differs from the point of view of the author. In the opinion of the author there is no right or wrong definition. The definition of fingerprinting which is chosen for this research is comparable to the definition of Sanmorino and Yazid [17]: **IP addresses, ports, protocols, flow size and number of packets**. Although the work from Sanmorino and Yazid [17] has similarities to the proposal of this thesis, the thesis is intended to be generic and used by any signature or rule-based solution (e.g. network firewalls, Web Application Firewalls and Intrusion Detection/Prevention Systems). The DDoS fingerprint of this thesis is also intended to be used for e.g. legal attribution and reproduction of attacks (mainly for academic purposes). The fingerprint in this thesis uses consecutive filters based on characteristics that are most frequent in the network traffic (top 1 values). For example, the author identifies the target system by analysing the destination IP addresses. The destination IP address with the most incoming packets is classified as the target of an attack. The same is done for the protocol, the ports and additional service information. The author relies on the definition of DDoS attacks that the attacker uses several machines to send a huge amount of packet. All these research papers use either packet-based network measurement or flow-based network measurement. No research is done about using different measurement types for obtaining fingerprints as it is done in this work.

In this section it is described how others define/use fingerprints and how they differ from the approach of this theses. To conclude this chapter, the findings are summarized in the next section.

2.4 Concluding Remarks

The goal of this chapter was to understand the background of the research containing information about network measurement types, DDoS attacks and DDoS attack fingerprinting. This information is needed to develop a flow-based fingerprinting tool, to make it possible for network operators using flow-based network traffic to generate fingerprints.

In the first section the measurement types were introduced. There are mainly two different types: packet-based and flow-based. The packet-based approach (e.g. pcap) measures all information about the headers and the payload. It is the most precise measurement that can be done. However, it also needs much more storage than a flow. In flows packets with same characteristics (e.g. IP addresses, ports, protocol) are summarized. The information taken from flow is lesser than from packets, therefore it needs less hardware. In a comparison is shown that while a packet contained 1264 records, a flow contained 2 records. Tools as nfcapd or softflowd can be used for converting pcaps to flows, while for IPFIX the tool YAF can be used. This is needed to compare the flow-based and packet-based fingerprints of a DDoS attack on the same basis.

In a DDoS attack the attacker uses many machines to carry out an attack. The number of attacks is increasing, as in the fourth quarter of 2017 there was an increase of 14% compared to the fourth quarter of 2016. An attack consists of the attacker, the attack infrastructure and the victim. The attack infrastructure is divided into three parts: hander C&C, infected machines 'bots' and the public service. From each of these parts the attacker can attack the victim. The type of attack can differ. Examples for DDoS attacks are a fragmentation attack, a DNS attack or a SYN attack. By attacking the victim there are at least five sets of information that can be measured by a network operator: destination IP address, source IP addresses, protocol, source and destination ports and the service with payload information. This information can be called fingerprint. Important for a fingerprint is, that it is distinguished between port combinations: one port to many ports, many ports to one port, one port to many ports and many ports to many ports.

There is a misunderstanding in the meaning of fingerprint in previous papers. There are papers which define the path of a packet or the operating system as a fingerprint. One paper ([17]) has the same definition of fingerprint as used in this work: IP addresses, ports, protocols, flow size and number of packets. Although this definition is the same, the work of the thesis is intended to be generic and used by any signature or rule-based solution. Before starting with generating fingerprints of DDoS attacks, in the next chapter some requirements, an existing packet-based DDoS attack fingerprinting tool and the dataset is explained.

Chapter 3

Requirements and Dataset

The goal of this chapter is to describe the requirements, the existing packet-based fingerprinting tool and the dataset. First, the requirements are explained, that are crucial for generating a flow-based fingerprint and compare them to packet-based fingerprints. It follows the explanation of an existing packet-based DDoS attack fingerprinting tool, that will be used to get fingerprints from packets and will be adapted to get flow-based fingerprints. It follows an analysis of the threshold for stopping the measurements (Stopping threshold) to define when no more DDoS attack vectors will be expected. Then the dataset will be described that will be used to evaluate the approach of this thesis. The chapter ends with concluding remarks.

3.1 Requirements

The main goal of this thesis is to propose a novel flow-based approach for extracting DDoS fingerprints that is as precise as a packet-based approach. Therefore, three requirements have been named for precision: (1) the number of attack vectors extracted from both network trace types should be similar, (2) the types of attack vectors extracted from both trace types should be similar, and (3) the set of source IP addresses within the DDoS fingerprints extracted from both traces should also be the similar. In the best case 'similar' represents the same value achieved in both the packet-based and the flow-based approach. To achieve these goals an existing packet-based DDoS attack fingerprinting tool [18] is used and adapted for the flow-based approach. This tool and this thesis are part of a large researching umbrella. The packet-based approach is explained in the next section followed by an analysis of the stopping threshold of this approach.

3.2 An Existing Packet-Based DDoS Fingerprint

There is an 'under development tool' used by Dutch organisations for DDoS fingerprint extraction from packet-based measurements, called *DDoS dissector*, publicly available at `https://github.com/ddos-clearing-house/ddos_dissector`. Although this tool is widely used, even by organisations such as the Dutch National High Tech Crime Unit police and several Dutch ISPs, it lacks documentation. In this section, an overview of this tool is provided which is used as basis for comparison with the flow-based DDoS fingerprinting approach. For this thesis this existing tool is assumed as ground truth. In addition, it is contributed with the improvements of this tool by adding a more suitable documentation and propose some improvements in the implementation.

The DDoS dissector is a python script that mimics how a network operator would dissect a network traffic for finding the main characteristics of a DDoS attack (for further mitigation purpose). Currently, July 2019, the tool receives as input an (offline) pcap file containing both potentially legitimate traffic and DDoS attack traffic. Then it analyses the traffic aiming to identify characteristics that have a very high frequency compared to the rest of the traffic. Overall, the tool relies on the assumption that a DDoS attack is a repetition of a network traffic with similar characteristics from a set of IP addresses towards a single destination IP address.

The DDoS dissector works performing successive filters as described in Algorithm 1. First, (line 1) the procedure receives as input a file.pcap(ng) and outputs (line 42) a list of 'attack vectors'. Note, each attack vector is a single DDoS fingerprint. Second, (in line 2), a set of fields are filtered from the original input file. Then, (line 3) it is searched for the destination IP address with the most received packets, called as the target. The data_remaining is filtered to include only data with this destination IP address. Then a loop starts. This is done to search attack vectors that targeted the found destination IP address. Then it is searched and filtered for the 'protocol' (line 7 to 9). In case that the protocol is UDP or TCP, it is searched and filtered for the top 1 port (line 11 to 18). This port is either the source or destination port. In the next step it is searched and filtered for some additional/payload information (line 23 and 24). After this step (line 26) the source IP addresses that send traffic to (1) the destination IP address, (2) using the found protocol, (3) from/to the ports found, (4) with the same payload are selected. This found characteristics including the source IP addresses are a DDoS attack fingerprint.

In case that the number of source IP addresses is lower than two, the analysis stops and it is no longer searched for more attack types. At the end, the current packets of that attack types are filtered out from the overall packet stream (save_data). Then the analysis restarts and searches for other attacks.

Algorithm 1 packet-based DDoS dissetor

```
1: procedure DDoS_DISSECTOR(input : file.pcap(ng))
2:     data_remaining[] ← feature_selection(file.pcap(ng), features)
3:     dst_ip_freq[] ← frequency(data_remaining[], dst_ip)
4:     data_remaining[] ← filter(data_remaining[], dst_ip_freq[0])
5:     save_data[] ← data_remaining[]
6:
7:     while len(data_remaining[]) > 1 do
8:         protocol_freq[] ← frequency(data_remaining[], protocol)
9:                                   ▷ changed in Chapter 4: source port for obtaining the protocol
10:        data_remaining[] ← filter(data_remaining[], protocol_freq[0])
11:
12:        if protocol == UDP   or   protocol == TCP then
13:            src_port_freq[] ← frequency(data_remaining[], src_port)
14:            dst_port_freq[] ← frequency(data_remaining[], dst_port)
15:            if src_port_freq[0].value() > dst_port_freq[0].value() then
16:                data_remaining[] ← filter(data_remaining[], src_port_freq[0])
17:            else
18:                data_remaining[] ← filter(data_remaining[], dst_port_freq[0]) ▷ changed in Chapter 4: second port is also taken into account
19:            end if
20:        else
21:            continue
22:        end if
23:
24:        payload_info_freq[] ← frequency(data_remaining[], payload_info)
25:        data_remaining[] ← filter(data_remaining[], payload_info_freq[0])
26:        src_ips[] ← get_src_ips(data_remaining[])
27:
28:        if len(src_ips) < 2 then          ▷ changed in Chapter 3.3: different number for stopping threshold
29:            break
30:        end if
31:
32:        attack_vector[] ← attack_vector[] + [protocol_freq[0],
33:                                             src_port_freq[0],
34:                                             dst_port_freq[0],
35:                                             payload_info_freq[0],
36:                                             src_ips[]]
37:
38:        data_remaining ← filter(save_data[], attack_vector[])
39:
40:        save_data[] ← data_remaining[]
41:    end while
42:    return attack_vector[]
43: end procedure
```

In the algorithm there are three comments about changes that will be done in this thesis. At first the stopping threshold (line 28) will be changed. Then, instead of the protocol field other information is used to classify the attack (line 8). Also the second port is taken into account later in this thesis (line 17). A simpler description of the algorithm is shown in Figure 3.1.

A packet trace may contain multiple vectors of DDoS attack. Therefore, after the DDoS dissector identifies the first fingerprinting (attack vector), it takes again the original file and removes all the packets that match with the first fingerprint. After that it re-runs the successive filters for finding a possible second fingerprint. This process runs indefinitely until a predefined stopping event happens (stopping threshold). Currently, the stopping event was when a fingerprint contains only one source IP address. In this case the DDoS dissector considers a possible DoS instead of a DDoS attack, discards the current fingerprint and stops the process.

After all the single-vector DDoS fingerprints are identified, the post-processing

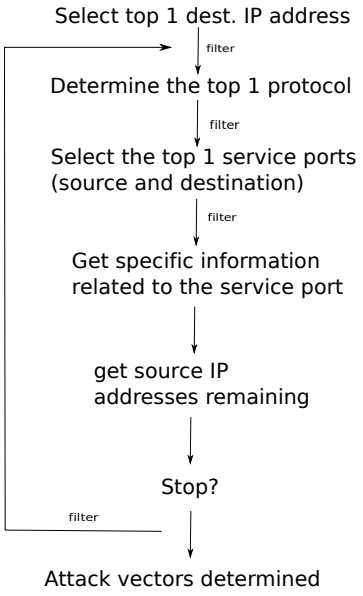


Figure 3.1: Steps of the analysis from the DDoS dissector

starts. Then each fingerprint is exported to a JSON-file. In addition, the original pcap file is filtered by the characteristics in the fingerprint, remaining a pcap with **only** those attack packets. In addition to that this pcap file is anonymized by removing the destination IP address and the source and destination MAC addresses. By making the pcap file anonymized, the data can be published and used for further research without publishing victim information. Finally, the JSON-file (fingerprint) and the pcap files are uploaded to the database on `http://ddosdb.org` to make it available for further use (e.g. mitigation, further research).

An example for the fingerprint can be seen in Figure 3.2. The first ten lines give the most important information about the attack vector. First, the multivector_key is given, which is a hash that identifies the file in which all the vectors were extracted. All different kinds of attacks within one DDoS attack have the same multivector_key. The singlevector_key represents a unique identifier for this attack type. Then, the source IP addresses are given. In this example 100.26.226.236 is shown, while the others are hidden for spacing purpose. In the next line, it shows that there is just one destination port: port 80. The source port is 123. Therefore, this is an example for a one port to one port attack. The IP protocol is 17 (UDP) and the service is NTP. As payload information the ntp.priv.reqcode 42 is given. Then some additional information is given. In total 1798 source IP addresses send 2,387,741 packets with the exact characteristics of the attack vector. In the following lines some information about the start time, the duration and the average bytes/packets per second are presented.

Limitations of the current DDoS dissector: As an 'under development' tool, the DDoS dissector has several limitations. First, the current fingerprint ends after the

```
{
 "multivector_key": "fa0a8f21a1816a6531acb543743124ec",
 "key": "fa0a8f21a1816a6531acb543743124ec",
 "src_ips": [
  "109.26.226.136",
  ... ],
 "dst_ports": [80],
 "src_ports": [123 ],
 "ip_protocol": "17",
 "service": "NTP",
 "additional": {"ntp_reqcode": 42 },


 "total_src_ips": 1798,
 "total_packets": 2387741,
 "duration_sec": 120.32017302513123,
 "start_time": "2014-12-22 11:12:56",
 "avg_bps": 9545941.59169052,
 "avg_pps": 19844.893337223457,
 "start_timestamp": 1419243176.663222
}
```


Figure 3.2: Fingerprint of a pcap file

investigation of a single value of the payload. It would be more suitable to add a step to investigate all the fields in the headers and payload for determining more details for the fingerprint. Second, the DDoS dissector relies on Tshark tool for reading and parsing the pcap files. Some versions of Tshark do not have the fields that are used by the DDoS dissector. Also in old versions, an attack can be falsely classified as an QUIC attack instead e.g. as an Chargen attack. To get the optimal results the newest version of Tshark should be used (2.6.6). Third, in the current code, the field 'protocol' from Tshark returns the protocol from the highest layer. For example, if a packet is DNS inside a UDP the field 'protocol' will return DNS. For a generic purpose (considering other packet-based and flow-based measurements) it would be more suitable to return UDP and then infer the DNS from the source or destination port number 53. Forth, the current 'stopping event' (when there are no more attack vectors) is triggered when a fingerprint has only one source IP address. This number is too relaxed. It is important to find a more representative number or other type of stopping event. Only the first limitation does not directly influence the comparison with the flow-based approach. The other limitations were solved and are described in the next chapter.

Before starting with evolving the flow-based approach, some analysis and improvements of the packet-based version must be performed. As named in the limitations the stopping event might be too low. In the next subsection, an analysis of this threshold is addressed. The results of this will be used for the flow-based approach.

3.3 Stopping Threshold

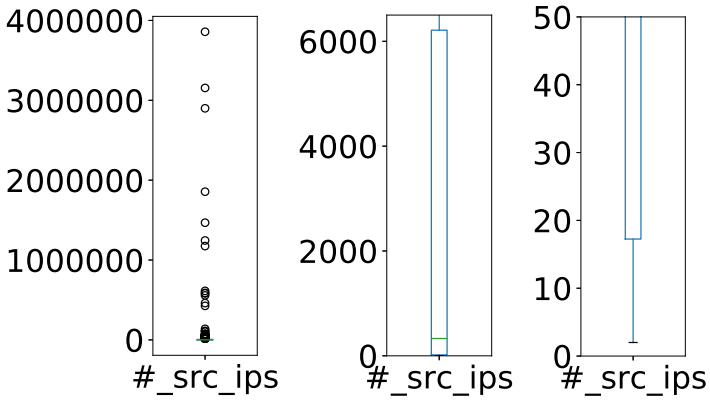
Originally, the DDoS Dissector stops executing if an attack vector is from one single source IP address, which is also listed in the limitation of Section 3.2. If the threshold is smaller it increases the chance of miss-classifying IP addresses that sent potentially legitimate traffic. This threshold is applicable and required to both the packet- and flow-based approach.

For investigating a more correct 'stopping threshold' all the 263 pcap traces (described in Section 3.4) are analysed in the packet-based DDoS dissector. This is done by investigating the number of source IP addresses involved in each attack. In total 520 attack vectors were found. This shows that on average each pcap trace contains around two attack vectors. Figure 3.3 shows a box-plot of the number of source IP addresses identified in each of the 520 attack vectors. The figure is split into three parts. First, the overall box-plot graph of the number of source IP addresses is shown. It can be seen that the box disappeared and that there are several attack vectors with an outlier number of source IP addresses. For example, there is one attack vector with almost 4 million IP addresses.

In Figure 3.3b it is zoomed into the overall graph and it appears that the median is 330 source IP addresses. It means that at least half of the attacks have up to 330 source IP addresses. While zooming into the bottom of the box-plot it can be observed that the minimum value is two. This is biased because the used stopping threshold was 'lower than 2'. A good value for the 'stopping threshold' would be the base of the 1st quartile in the box-plot (Figure 3.3c), which is 18 source IP addresses. The implication of this number is the higher chance that only source IP addresses involved in DDoS attack traffic is included in the fingerprint. However, some of the attack vectors ($<$ 18 source IP addresses) will not be considered or detected. It is up to the operator to decide which value to choose. For the remaining part of the evaluation a threshold equal to three will be used, which will not be able to classify attack vectors with two source IP addresses. The take-away of this analysis is that in an operational environment it is advised to use a threshold of 18.

3.4 Dataset

The dataset used in this master thesis is composed of 263 pcap traces in total. The traces used were extensively evaluated by Cardoso de Santanna [2] and by Bukac et al. [55]. Each of those traces contains attacks with one or more attack vectors. The number of total attack vectors varies in the different sections because of adaptations in the source Code. There is also a percentage of potentially legitimate traffic in each trace. Each of those 263 pcap traces were converted to Netflow v5



(a) Overall *(b)* Zoom to 6,000 *(c)* Zoom to 50

Figure 3.3: Number of source IP addresses considering 520 attack vectors

using the tools nfcapd [36] and softflowd [44]. The generated flow files were checked to guarantee that they contain the same number of packets as in the pcap files.

3.5 Concluding Remarks

The goal of this chapter was to describe the applicable requirements that are crucial for generating a flow-based fingerprint and to compare this to the packet-based approach, to explain the existing packet-based DDoS dissector and to explain the used dataset.

The comparison of the flow-based and packet-based fingerprints will be based on three requirements: the number of attack vectors, the types of attacks and the set of source IP addresses. There is already a packet-based DDoS attack fingerprinting tool called DDoS dissector, which is part of the researching umbrella of this thesis. The tool receives as input a pcap file containing both potentially legitimate and DDoS attack traffic. The dissector filters the file in four steps and outputs a set of attack vectors. Usually it stops its execution, if the number of source IP addresses is lower than two. The fingerprint then contains all traffic characteristics that match with the attack vector and is saved as a JSON file. The dissector has a few limitations in its current version (July 2019). First it only filters for one service information. More information could be included. Second, the output relies on the version of Tshark. The newest version should be used (2.6.6). Third, the DDoS dissector gives the service protocol. This is not generic as not all measurement types give this information. The IP protocol would be better. Fourth, the stopping event of the dissector might be too relaxed.

The last limitation is analysed in the third section. In total 520 attack vectors were analysed based on their number of source IP address. It was seen that there some

outliers inside. The median of the distribution of the number of source IP addresses is 330. While in the thesis a threshold of three is used as the stopping threshold, for an operational environment it is advised instead of using a threshold of 18.

The dataset used in the thesis consists of 263 pcap files. These files are converted to flow files(Netflow V5) to make a comparison possible. The generated flow files were checked to guarantee that they contain the same number of packets as the pcaps. After giving the dataset, explaining the basis of the comparison between the packet-based and flow-based approach and describing the packet-based approach, the flow-based approach for generating fingerprints will be presented in the next chapter.

Chapter 4

Flow-Based DDoS Attack Fingerprints

The goal of this chapter is to evaluate a flow-based DDoS attack fingerprinting tool defined in Chapter 3. The flow-based fingerprint should be as precise as the packet-based one, although it includes less information (e.g. no DNS queries). This lack of information needs to be minimized.

The evaluation of the DDoS dissector for flows is performed incrementally and the results are presented. The evaluation considers the packet-based DDoS dissector as ground truth. After applying a few changes, the impact of these changes is presented with some comparison between the flow-based and packet-based approach respecting the three requirements defined in Chapter 3 ((1) similar number of attack vectors, (2) similar attack types, (3) similar set of source IP addresses)

In the first section the same structure of code as from the packet-based DDoS dissector is applied. As the results of this can still be improved, in the second section the code for the packet-based DDoS dissector is adapted. To get a as precise result as the packet-based, the number of source IP addresses is adjusted in the third section. The chapter ends with concluding remarks.

4.1 Applying The Structure Of The Packet-Based DDoS Dissector

To develop the flow-based DDoS dissector, the packet-based approach is adapted. In this section first the improvements and corrections and then the used methodology are explained. At the end of this section the results for applying this methodology are given.

4.1.1 Improvements and Corrections

The fields of packets are different from the field of flows. To adapt the code of the packet-based DDoS to use it for flows, some corrections are needed to be done. The corrections are about determining the protocol, getting information about ICMP and finding a one port to one port attack. Following, these three corrections are explained.

Information About The Protocols And Fragmented Traffic

As explained in Chapter 3, in the current version of the packet-based approach the field 'protocol' returns the protocol in the highest layer. Therefore, it usually returns the service (from the application layer). Examples for these are DNS, NTP, UDP or Chargen attacks. However, this information is not explicitly available in the flow-based network traces. The field 'protocol' of Netflow gives the IP protocol as Internet Control Message Protocol (ICMP), UDP or TCP. For making the fingerprints comparable, a first correction needs to be done by using the source and destination ports to infer the service in the application layer. Internet Assigned Numbers Authority (IANA) maintains an updated list of 'service name and transport protocol port number' [56].

Preliminary experiments show that if the source and destination port numbers are zero (0) in flow records, it means that IP fragmentation happened. There was no document found emphasizing this finding, but this observation was used for classifying fragmentation-based attacks.

Information About ICMP

The second correction that needs to be done is about the ICMP type. Although there is no explicit information about the ICMP type and code in the flow-based network traces, according to Fullmer and Romig [57] this information can be extracted from the destination port value. In Table 4.1, the range of destination port values by Lucas [58] and ICM [59] are listed.

Note that although the destination ports from Lucas [58] and ICM [59] are different from each other, there is no overlap or inconsistencies among them. For example, if the destination port is in the range from 300 to 313 or from 768 to 781, the ICMP will be considered as a type 3 ('destination unreachable'). Therefore, to determine the ICMP type for the flow-based DDoS dissector the values from Table 4.1 are used. After having more information about some protocols, the attack needs to be classified as described in Section 2.2.

Table 4.1: ICMP types

ICMP type	Code	Decimal [58]	Decimal [59]	Description
0/8				Echo replay/request
	0	0/800	2048 to 2303	Echo
3				Destination unreachable
	0	300	768	Network unreachable
	1	301	769	Host unreachable
	2	302	770	Protocol unreachable
	3	303	771	Port unreachable
	4	304	772	Fragmentation Needed and Don't Fragment was Set
	5	305	773	Source Route Failed
	6	306	774	Destination Network Unknown
	7	307	775	Destination Host Unknown
	8	308	776	Source Host Isolated
	9	309	777	Communication with Destination Network is Administratively Prohibited
	10	310	778	Communication with Destination Host is Administratively Prohibited
	13	313	781	Communication Administratively Prohibited
5				Redirect
	0	500	1280	Redirect Datagram for the Network (or subnet)
	1	501	1281	Redirect Datagram for the Host
11				Time exceeded
	0	2816	2816	Time Exceeded/Time to Live exceeded in Transit
	1	2817	2817	Time Exceeded/Fragment Reassembly Time Exceeded

1 Port To 1 Port Attack

As explained in the Section 2.2 there are four kinds of attacks related to the source and destination port (1. from many source ports to one destination port; 2. from one source port to many destination ports; 3. from one source port to one destination port and 4. from many source ports to many destination ports). Originally the DDoS dissector analysed the frequency of both source and destination ports. Then, the follow-up filter would consider the port number (either source **or** destination) with highest frequency. The impact of this decision is that the attack will be either one to many or many to one. For improving the DDoS Dissector also, the *1 port to 1 port attacks* should be considered to make a better classification possible. For this the *threshold 1 port to 1 port attack* is introduced. To analyse the effect of different numbers for the threshold, several numbers were used and the number of attacks are compared. The analysis was done for 20%, 40%, 60% and 80%.

In Table 4.2 an example for a port distribution is given to exemplify the thresholds. In this case source port 53 is the top 1 port. To classify a *1 port to 1 port attack* now the percentage of the destination port 80 is taken into account. By choosing a threshold of 20%, the destination port 80 is taken into account ($25\% > 20\%$). Therefore a *1 port to 1 port attack* happened from port 53 to port 80. By taking a threshold of 40% the value of the destination port is not taken into account ($25\% < 40\%$). The dissector would give a *1 port (53) to many ports attack*. This result would be the same for a threshold of 60% or 80%.

Table 4.3 shows a different example of a port distribution. Again, source port 53 is the top 1 port chosen by the DDoS dissector. Now the value of the destination port 80 is 55%. Therefore, it is higher than 20% and 40%. Using a threshold of 20% or 40% a *1 port to 1 port attack* will be classified. For the thresholds 60% and 80% a

Table 4.2: First example of a port distribution

Source Port	%	Destination Port	%
53	90	80	25
123	10	1235	10
449	5	777	1
.	.	.	.
.	.	.	.

1 port to many port attack will be found as the value of the destination port is below the threshold.

Table 4.3: Second example of a port distribution

Source Port	%	Destination Port	%
53	99	80	55
123	10	1235	10
449	5	777	1
.	.	.	.
.	.	.	.

4.1.2 Methodology

The same structure as for the dissector for packets is used to make a comparison between flows and packets possible (see Algorithm 1). As the flow-based measurement gives less information, some corrections need to be done: (1) information about the protocols, (2) information about ICMP, (3) and the right threshold to specify between a *1 port to 1 port attack* and a *1 port to n ports/n ports to 1 port attack*. To analyse the effect of different numbers for the threshold and to choose the optimal value, several numbers are used (20%, 40%, 60% and 80%) and the number of attacks are compared.

In Chapter 3 it was observed that the packet-based approach for extracting a fingerprint from a network trace relies on five successive filters (1. destination IP address, 2. protocol, 3. source and destination ports, 4. specific header or payload of the protocol and 5. remaining source IP address). For a flow-based approach only the 4th step of the packet-based cannot be covered, as no field exists for additional information. An exception for this are the TCP flows, which contain a field for the TCP flags. For ICMP also some additional information can be extracted from the flows. To compare the results for the packet-based and flow-based approach 263 traces were used as described in Chapter 3.

4.1.3 Results

After applying the first corrections the results of these can be shown. First, the results of the analysis for the *threshold 1 Port to 1 Port attack* are shown. Then a first comparison between the packet-based and flow-based approach can be done.

Threshold 1 Port to 1 Port Attack

In Figure 4.1 the results for the *threshold 1 port to 1 port attack* for the flows are shown. It was expected that 20% would give the highest number of attacks as then usually the threshold is easier reached. But for this threshold the lowest number of attacks is shown (500 attack vectors), while 40% gives with a number of 520 attack vectors the highest number of attacks. Both, a threshold of 60% and a threshold of 80% have about 515 attacks.

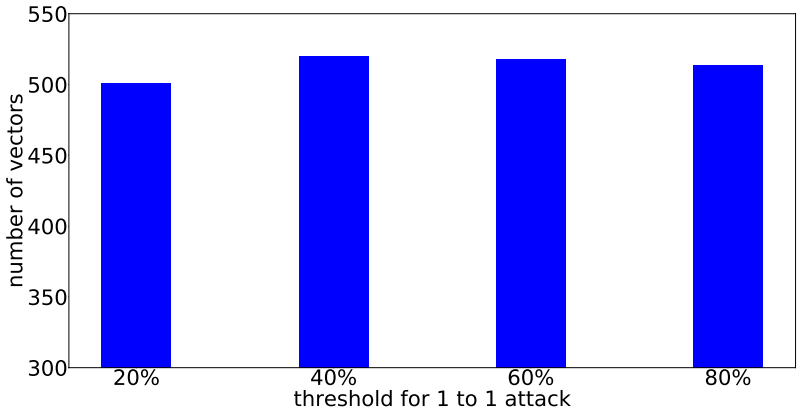


Figure 4.1: Threshold for a 1 to 1 attack

It was expected, that the lower the *threshold 1 port to 1 port attack* the higher the chance, that there will be given a *1 port to 1 port attack*, which is included in a *1 port to many ports* or *many ports to one port attack* for higher thresholds. In case of a threshold of 20% it seemed to be the case, that these *1 port to 1 port attacks* may consist of only a few source IP addresses, which might be lower than the *threshold minimum src IP addresses*. Then the analysis stops and the trace is not taken into account as an attack.

To exemplify, (while analysing an attack trace) consider that for higher thresholds *a 1 port (port 53) to many ports attack* was found. For the threshold of 20%, now first a *1 port (port 53) to 1 port (port 80) attack* could be found and after that the *1 port (port 53) to many ports (excluding port 80) attack* was found. In case that now the *1 port to 1 port attack* consists only of one source IP address, the analysis stops. Neither the *1 port to 1 port attack* nor the *1 port to many port attack* is taken into account. Then the overall number of attack vectors becomes smaller.

For a higher threshold, it is the other way around. A lower threshold might include a *1 port to 1 port attack* and a *1 port to many ports attack*. If the *threshold 1 port to 1 port attack* is too high, the *1 port to 1 port attack* would be included in the *1 port to many ports attack*. So instead of having two attack vectors, there is only one attack

vector left. An example for this: normally there is one attack from source port 53 to destination port 80 and also an attack from source port 53 to many ports (excluding port 80). In case of a high threshold, only one attack from source port 53 to many ports (including port 80) is found. Thus, for higher thresholds, the number of attack vectors becomes smaller.

Overall it can be said that the difference between the number of attacks for different thresholds is quite low, compared to the overall number of attack vectors. For the rest of the research a threshold of 40% is used. With this threshold, all necessary information needed to generate fingerprints is available. Therefore, the first comparison of the flow-based and packet-based approach can be done.

Comparison Of The Vectors

After having the best value for the *threshold 1 port to 1 port attack*, a first comparison between the flow-based and packet-based approach can be done. As described in Subsection 4.1.2 the comparison will be based on 263 network traces. In Figure 4.2 the number and distribution of attack types is depicted. The y-axis shows the number of traces and the x-axis shows the number of vectors/fingerprints observed inside a trace. In total 569 attack vectors for the packets and 553 attack vectors for the flows are found. The distribution of the vectors is shown alternately for packet-based and flow-based network traces starting with the vectors of the packet-based. For example, 36 packet-based network traces have only one attack vector, while over 40 traces from the flows have one attack vector.

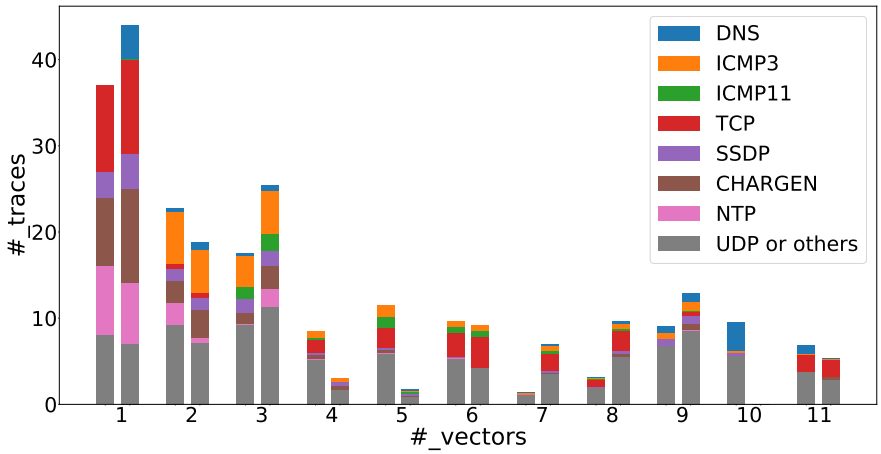


Figure 4.2: First comparison between packet-based and flow-based approaches to extract DDoS fingerprints. Depicting the number of traces containing one or multiple attack vectors (and its types of attacks)

For two attack vectors, there are more traces of packet-based than of flow-based network traces. For flow-based network traces there are about 18 traces and for the packet-based 23. For three attack vectors the difference is higher. There are about 25 flow-based network traces, while there are only 18 packet-based network traces. For four attack vectors there is a difference of five trace between packet-based and flow-based network traces. For six attack vectors the difference in the number of traces between flow-based and packet-based traces is only one trace. For vectors higher than seven, the difference become higher again. There is even the case for ten attack vectors that there is no flow-based trace.

In this graph also the distribution of the attack types per attack vector is shown. The distribution is shown for eight types of attacks (i.e., DNS, ICMP (type) 3, ICMP (type) 11). In the UDP also some attacks (e.g. SNMP attacks) are included which have only a few attack vectors in all the traces. For only one attack vector it can be seen that there are already some similarities.

For UDP, NTP and SSDP there is only a difference of one trace. For Chargen the packet-based network traces have about two traces more than flow-based. For TCP the flow-based have only a few more traces than packet-based. The biggest difference is, that there are traces with TCP attacks for the flow-based network traces, while there is no trace with TCP for packet-based. Explanations for the differences are given in the next paragraph. For two attack vectors per trace, packet-based and flow-based network traces include the same attack types, but the number of traces with the attack types differ for UDP, NTP, Chargen and DNS. There is a higher number of traces with the UDP and NTP attacks for the packet-based compared to the flow-based network traces, but more traces from the types Chargen and DNS for the flow-based. Except for a number of six attack vectors per trace (only a few differences), starting with a number of four attack vectors the differences in the attack types rises.

The differences in the classification of attacks is based on the differences in the methodology of naming the attacks. For the packet-based approach the application protocol is used to name the attack as e.g. 'DNS attack'. For the flow-based approach only the service port can be used to classify the attacks. This makes a difference as e.g. a packet-based network trace can give a UDP packet from port 53. This will be shown as a UDP attack in the figure. For the flow-based approach all packets from port 53 are defined as a DNS attack. Therefore, instead of having a UDP attack a DNS attack will be shown in the figure.

Also, there might be some other problems in using the application protocol as explained in the limitations of Section 3.2: In some versions Tshark might show a QUIC attack, while newer ones show another type of attack. To get around these problems and to improve the comparability of both version of the DDoS dissector

(existing packet-based and the developed flow-based), the packet-based DDoS dissector was adapted to also use the service port for classifying protocols. The results of this adaptation are shown in the next section.

4.2 Adapting The Code Of The Packet-Based DDoS Dissector

After obtaining the first results of the packet-based and flow-based approach, the results showed that some changes were necessary. These changes and their results will be explained in this section.

4.2.1 Methodology

The code is adapted to the structure of the flow-based DDoS dissector. Instead of using the application protocol field for determining the service, the ports are used for this as it is done for the flows. This change makes a better comparison possible. For the analysis 263 traces were used as described in Chapter 3.

The fingerprints from the flow-based and packet-based DDoS dissector are compared based on the three requirements explained in Section 3.1: (1) the number of attack vectors extracted from both trace types should be similar, (2) the types of attack vectors extracted from both trace types should be similar, and (3) the set of source IP addresses within the DDoS fingerprints extracted from both traces should also be similar. The results of applying this change are given in the next subsection.

4.2.2 Results

The results for the first two requirements are described in the first subsection followed by the results for the third requirement.

Distribution Of Vectors (number and types)

To examine requirement (1) (similar number of attack vectors) and (2) (similar number of attack types) the result for the distribution of vectors is shown in Figure 4.3. In total 571 attack vectors for the packets and 613 attack vectors for the flows were found. For one attack vector it can be seen that the result becomes more comparable. Both, the packet-based and the flow-based approach, have 51 traces with one attack vector. For two attack vectors there is still a difference visible as for the packet-based approach there are 25 traces, while for the flow-based there are 33.

For three attack vectors and for six attack vectors the numbers become more similar again. For three attack vectors it is 33 for packet-based and 35 for flow-based approach, while it is for six attack vectors ten for the packet-based and nine for the flow-based approach. For the other numbers of attack vectors there is still difference visible. For a number of four attack vectors in one trace, the difference is the highest with over 20 vectors. While there are some traces with eight or nine attack vectors for the flows, there are no traces for the packet-based approach. For ten attack vectors it is the other way around: There are six traces for the packet-base network traces, but none for the flows.

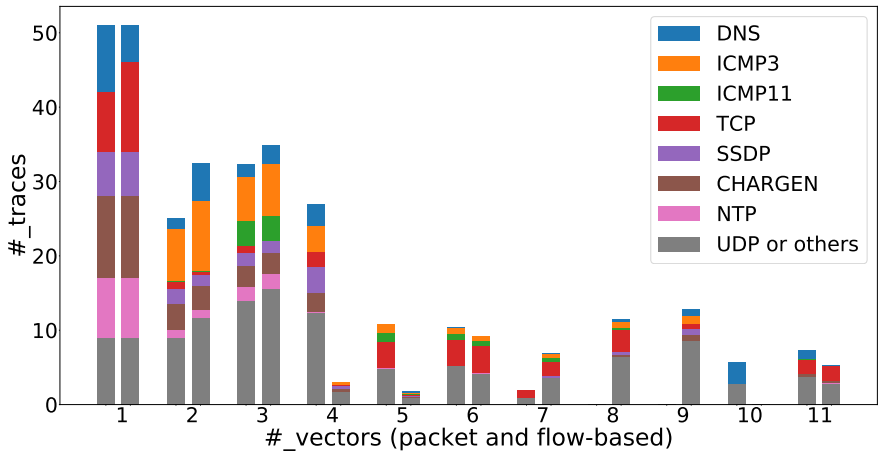


Figure 4.3: Second comparison packet-based and flow-based approaches to extract DDoS fingerprints. Depicting the number of traces containing one or multiple attack vectors (and its types of attacks).

For the number of attack vector one to three per trace, it can be seen, that the attack types become more similar. First, just the observations are described. The explanations follow in the next paragraph. For one attack vectors the number of UDP (or others), NTP, Chargen and SSDP attack is the same. For TCP and DNS the numbers are different. For two attack vectors in one trace there were more UDP attacks found for the flow-based approach than for the packet-based. The same happened for the ICMP (type) 3 and DNS. The number of NTP, Chargen, SSDP and TCP attacks are the same. For three attack vectors in one trace, the number of NTP, Chargen, SSDP and ICMP (type) 11 attacks is the same. For the flow-based network traces more UDP, ICMP (type) 3 and DNS are found, while for the packet-based network traces TCP attacks were found, but for the flow-based not.

The reasons for some differences are the DNS queries and the TCP flags. As mentioned in the previous section there is no additional information given about DNS in the flow-based network trace. Therefore, only one DNS attack can be found, while

in the packet-based network traces several DNS attacks with different DNS queries are shown. To exemplify, the flow-based DDoS dissector finds one DNS attack in a trace. Therefore, in this trace one attack is found.

The packet-based DDoS attacks also finds a DNS attack, but it can differ between different DNS queries. For example, consider that first a DNS attack with the DNS query 'example.com' is found. Then, a DNS attack with the DNS query 'mydomain.com' is found. The last attack found in the trace is a DNS attack with the DNS query 'university.com". Therefore, in the whole packet-based network trace three attacks are found. However for the flow-based approach, there is no distinction and only one DNS attack vector will be found. For the packet-based approach this trace is not taken into account for the one attack vector, but for three attack vectors. So the attacks move to the right. This happens for several traces, so that this leads to shifts in the graph.

Something similar happens to the TCP flags. The flow-based DDoS dissector also takes the TCP flags into account, but in most cases, they are different from the flags of the packet-based network traces. This makes it difficult, if not impossible, to compare these attacks. Not only the flags differ from those two measurement types, but also the number of TCP attacks with different flags inside of the traces differ. This leads to a change/shift in the distribution of attack vectors in the figure.

The overall take away is that, although there are still some differences, the results from the packet-based and the new flow-based approach are be compared and the requirements 1 and 2 (similar attack vectors and attack types) are met. Especially for the lower amount of attack vectors per trace (< 5) the results were similar, sometimes the same. The differences are based on a shift through the application information of DNS (no DNS queries in the flows) and TCP (different flags).

Source IP Addresses Of Each Attack

This subsection is based on the requirement 3 (similar source IP addresses found using packet-based and flow-based DDoS dissector). This analysis will be done by dividing the number of source IP addresses of the flow-based divided by the number of source IP addresses of the packet-based network traces in percent. The result show, if the number of source IP addresses from the flow-based is the same than from the packet-based network traces (100%), if it is lower ($< 100\%$) or higher ($> 100\%$). The number of source IP addresses from the packet-based network traces is the ground truth, as the information in the packets is more precise. Therefore, in case that the result is higher than 100%, it means that the flow-based DDoS dissector found more IP addresses than the packet-based. These IP addresses possibly were misclassified and were sending a legitimate traffic very similar to the

attack (possibly with lower frequency of packets). This result needs to be avoided as **no** potentially legitimate traffic should be included in the fingerprint. It is preferred that then there is a false negative (i.e., IP addresses that were part of the attack, but were not found by the dissector). In this case, less source IP addresses would be found in the flows than in the packets ($< 100\%$). So there might be a few attack IP addresses that are not found, but still most of the attack IP addresses are included in the fingerprint and no potentially legitimate traffic is included.

For the analysis 263 traces are used as described in Chapter 3. The Figure 4.4 shows the result for this comparison in a boxplot. TCP is not included because no comparison was possible as explained in Section 4.2.2.

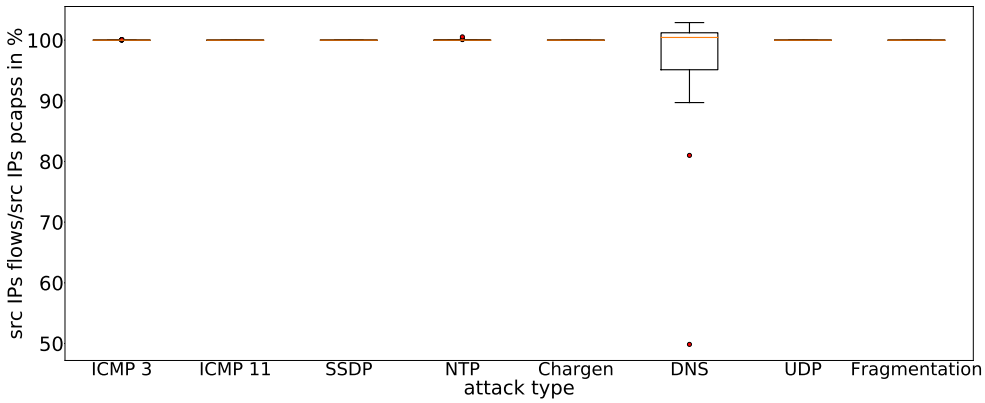


Figure 4.4: Number of source IP addresses from flows compared to packets

The comparison is done for each attack type, which are listed on the x-axis. On the y-axis the result of dividing the number of source IP addresses of the flows by the number of source IP addresses of the packets in percentage is shown. For all attack types shown in the figure the median is about 100%. So it can be said that in average the goal of having the same number of IP addresses for flow-based and packet-based network traces is achieved. For the ICMP (type) 11, SSDP, Chargen, UDP and Fragmentation only the line of the median is shown. This means that there are no other deviant number of source IP addresses. For all attack of these types the number of IP addresses is the same for the flow-based and for the packet-based approach.

For the ICMP (type) 3 attack there is one outlier at 100.1% because there was an error based on selecting the ICMP type. In one attack there were more IP addresses in the fingerprints of the flow-based than in the packet-based approach. But as this is just one outlier, it does not have to be considered.

The third quartile and the maximum of the boxplot are for NTP about 100.05%. It means that in worst case 0.05% of the IP addresses from the flows are potentially

sending legitimate traffic. This difference is based on the usage of additional information in the packet-based approach. The packet-based DDoS dissector filters in case of an NTP attack for the ntp.repriv.reqcode. For the flow-based network traces this information is not given. The attack cannot be filtered for this and therefore more IP addresses can be found.

For the DNS attack, there are two outliers, below 100% (50% and around 80%). The maximum is at 103%. This means that in worst case 3% of the IP addresses from the flows are potentially legitimate traffic. For the DNS also additional information is used for the packet-based DDoS dissector. If an DNS attack is found, it is searched for a DNS query. So only attacks which specific DNS queries are included in an attack. In the flow-based DDoS dissector all DNS flows are summarized in one attack. This leads to more IP addresses for the flows. As explained above, no potentially legitimate traffic should be included in the fingerprint. Therefore, the results for NTP and DNS attacks are not satisfying and are further investigated in the next section.

4.3 Adjusting The Number Of Source IP Addresses

In this last section the results from the comparison of the attack vectors and the number of source IP addresses were very similar for some attack types. For attack types which use application information (NTP and DNS) in worst case potentially legitimate traffic is included in the fingerprints of the flows. In case of NTP the false positive is lower than 1% and in case of DNS it is 3%. For making the flow-based approach as precise as the packet-based one, these numbers need to be adjusted. The IP addresses with potentially legitimate traffic need to be removed from the fingerprint. This can be done by a third threshold, the *Certainty Threshold*. This will remove a specific percentage of the IP addresses.

In the next subsection the methodology for obtaining correct values of this threshold is explained. After that, the number of source IP addresses from NTP and DNS attacks are compared again.

4.3.1 Methodology

Section 4.2.2 illustrated that in case of NTP and DNS attacks the number of source IP addresses is higher for the fingerprints of flow-based than for packet-based approach. This is based on the application information that is missing in the flows. The packet-based DDoS dissector uses for NTP a filter for the ntp.priv.reqcode and for the DNS the DNS query. To get the IP addresses with certainty a third threshold (*Certainty Threshold*) is applied to the code.

In case that a DDoS attack is from the type NTP or DNS, the IP addresses with the lowest number of packets are most likely the ones with potentially legitimate traffic as these would send packets with a lower frequency. Therefore these IP addresses should be discarded from the fingerprint. The right value for this threshold will be analysed in this section. As starting point for the threshold the maximum of Figure 4.4 is used (0.05% for NTP and 3% for DNS). If the maximum is still above 100% a higher threshold is used. If the maximum is below 100% a lower threshold is tried to see if it is sufficient to remove less IP addresses.

To exemplify the procedure, consider that: The packet-based fingerprint found ten source IP addresses involved in one attack vector. In case that the flow-based fingerprint also found ten source IP addresses, a perfect result is achieved and nothing has to be changed. If only nine source IP addresses are included in the fingerprint of the flows, it means that one IP address was not classified as part of the attack (false negative). In this case **no** legitimate traffic will be blocked. In case that the flow-based fingerprint consists of eleven source IP addresses, it means that one IP address is sending legitimate traffic similar to the attack traffic (false positive). This must to be avoided. Therefore, 10% of the IP addresses need to be removed. Then the flow-based fingerprint consists of 10 IP addresses, which is the same as for the packet-based. The number of IP addresses for flows is then adjusted successfully.

For the analysis the 263 traces described in Chapter 3 are used. For NTP the analysis is first done in steps of 0.01% and later in steps of 0.02%. For DNS the analysis is done in steps of 0.5%.

4.3.2 Results

The optimal number for the threshold for an NTP attack is investigated in Figure 4.5. On the x-axis the values of the Certainty Threshold in percent is shown. The starting value is taken from the value of the maximum from Figure 4.4 which is 0.05%. The y-axis shows the number of source IP addresses of the flow-based approach divided by the number of source IP addresses of the packet-based in percentage.

Figure 4.5 shows that by using a threshold of 0.05% the maximum is still above 100% (100.015%). This means that in a worst case still potentially legitimate traffic is included in the fingerprint. As this should be avoided, a higher threshold is needed. For a threshold of 0.1% the third quartile is now equal to 100%, but the maximum is still higher than 100% (100.07%). Again a higher threshold is needed. By using a threshold of 0.2% the maximum is equal to the third quartile with a value of 100%. Now all IP addresses from the flow-based network traces are included in the packet-based. Therefore, **no** legitimate IP addresses that send legitimate traffic will be included in the fingerprint. In a worst case 99.8% of the IP addresses from the

packet-based network traces are found (minimum of boxplot). This 0.2% of false negative ensures, that no potentially legitimate traffic will be blocked.

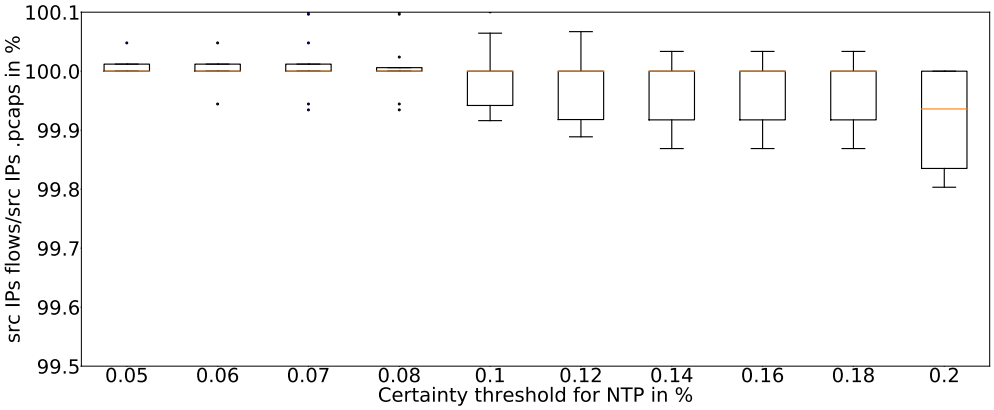


Figure 4.5: Analysing the *Certainty Threshold* for NTP attacks

The same adaptation needs to be done for the DNS fingerprints as the packet-based approach filters for DNS queries. In Figure 4.4 a threshold of 3% was seen. This value will be first used for the adaption of the IP addresses. The result is shown in Figure 4.6. By using a threshold of 3% the maximum of the boxplot is already equal to 100%. So no potentially legitimate traffic will be included in a flow-based fingerprint.

Still, for the *Certainty Threshold* the lowest possible value should be used for removing only the false positive source IP addresses. That why lower values for the threshold are tried. For 1.5% the maximum is still above 100%. The same applies to a threshold of 2% and 2.5%. For all these three thresholds in worst case potentially legitimate traffic would be included. Therefore 3% is the lowest possible threshold for not including potentially legitimate traffic. In the worst case (minimum boxplot value) 'only' 88% of the IP addresses of the packet-based fingerprint are included in the flow-based fingerprint. This means than 12% of attack IP addresses will not be found. This result is okay, as it makes sure that no potentially legitimate traffic will be included.

This section shows that the number of source IP addresses can be adjusted, so that they match to 100% or lower with the ones from the packet-based network traces. Therefore, **no** potentially legitimate traffic is part of the fingerprint anymore, even for attack types where application information is used by the packet-based approach. This fulfils requirement 3 (similar set of source IP addresses) from Section 3.1, whereby now all requirements are met. The next section gives a summary of the results achieved in this chapter.

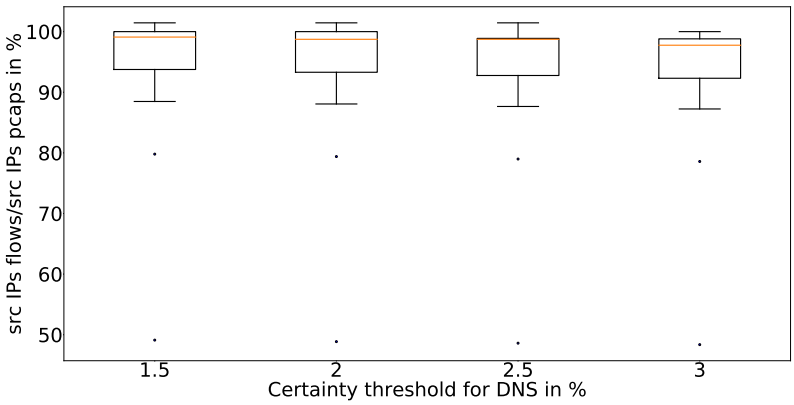


Figure 4.6: Analysing the *Certainty Threshold* for DNS attacks

4.4 Concluding Remarks

The goal of this chapter was to compare the flow-based and packet-based fingerprints and to make the flow-based approach as precise as the packet-based one (RQ3). The packet-based DDoS dissector was adapted to generate flow-based fingerprints.

Some corrections were needed to be applied for the flow-based fingerprint: protocols, Information about ICMP and 1 Port to 1 Port attack. (1) In flow network traces the protocol fields consists of the IP protocol. No service information is available. This information has to be taken from the ports. Also the only application information that can be taken from the flows is the TCP flag. (2) In the flow network traces there is no specific information field about the ICMP type, but this information can be taken from the destination port value.

(3) Originally, the DDoS dissector only takes the top 1 (either source or destination) port into account. To also classify a 1 port to 1 port attack, a threshold is needed. An analysis showed that a threshold of 40% gives the best result. After applying these correction, a first comparison based on the number of attack vectors and attack types was done. It showed that there were already some similarities for a lower amount of attack vectors, but there are still differences. Also the classified attack types differed.

To minimize this difference the packet-based DDoS dissector was adapted. After this adaptation the port is taken for determining the service and not the service protocol field. Again the number of attack vectors was compared. For one attack vector the same number of traces was found. For the other values the number still differed. By comparing the attack types it was seen that it was quite similar especially for a number of attack vectors from one to three. The difference is attributed to a shift

of the attack vectors from the packet-based network traces based on the additional information of DNS and TCP traces.

Then the numbers of source IP addresses of each attack were compared. For the attack types ICMP (type) 3, ICMP (type) 11, SSDP, Chargen, UDP and Fragmentation the same number of IP addresses were found. For NTP and DNS attacks in a worst case more IP addresses were found for the flow-based approach than for the packet-based as the packet-based DDoS dissector filters the traffic with application information (ntp.priv.reqcode and DNS query). By using application information the attacks can be specified and potentially legitimate traffic with the same protocol can be filtered out. In the flow-based fingerprints these potentially legitimate traffic is included.

This difference is mitigated by adjusting the number of source IP addresses for these kinds of attacks. For this a specific threshold in form of a percentage is used. This specific percentage of IP addresses will be removed from the fingerprints as this is most likely potentially legitimate traffic. For NTP a threshold of 0.2% and for DNS a threshold of 3% is used.

After adjusting these, no potentially legitimate traffic is included in the fingerprints anymore. In the worst case, the flow-based fingerprint will include 99.8% of the source IP addresses from the packet-based fingerprint for NTP and 88% for the DNS. The remaining 0.2% / 12% are false negative, which means that **no** potentially legitimate traffic will be blocked in case such a fingerprint would be used for blocking traffic. The next chapter will summarize the whole thesis and also some recommendations will be given.

Chapter 5

Conclusions And Future Work

The goal of this thesis was to make a flow-based approach for creating DDoS attack fingerprints as precise as a packet-based approach. The following chapter provides the conclusions related to the previous chapters in this thesis. It ends with future work.

5.1 Conclusions

Most operators prefer the flow-based network measurement for monitoring their network although it has less information because it needs less hardware. In a DDoS attack the attacker uses many machines to carry out an attack. The number of attacks is increasing, as in the fourth quarter of 2017 there was an increase of 14% compared to the fourth quarter of 2016.

There are mainly five sets of information network operators can measure from attacks: destination IP address, source IP addresses, protocol, source and destination ports and the service with payload information. This information can be called fingerprint. The DDoS dissector is a tool which generates fingerprints based on packets. For flows there was no tool for generating fingerprints. Therefore, a flow-based approach was developed by adapting the (packet-based) DDoS dissector.

To reach the goal of this thesis (Making a flow-based approach as precise as a packet-based), three Research Questions (RQ) were defined: (1) What is the state of the art on DDoS attack fingerprinting and its relation to different types of network measurements? (2) How to generate a DDoS attack fingerprint based on flows? (3) How comparable are DDoS attack fingerprints generated from flows and packets?

To answer RQ 1 and give a background about the topic of fingerprints, a literature study was done on the topic of network measurement types, DDoS attacks and DDoS attack fingerprints.

The DDoS dissector created fingerprints of packets based on four filtering steps:

destination IP address, service protocol, source and destination ports and additional information. After finding a first attack, the DDoS dissector filters out the packets of that attack and searched for another type of attack. It stoped, if the number of source IP addresses was lower than two. This stopping event was too low. An analysis with more than 200 attack traces was performed. For this thesis a threshold of 3 was chosen. But the analysis of the stopping threshold showed, that for operational environments a stopping threshold of 18 is advised.

To develop a flow-based fingerprinting tool and answering RQ 2, the packet-based DDoS dissector was adapted. After applying the same structure than for the packet-based approach, three corrections were needed to make: First, the ports determine the service, instead of the application protocol. Second, the ICMP type has to be taken from the destination port value. Third, originally the DDoS dissector classified an attack either as one port to many port attack or many port to one port attack. A one port to one port attack is not considered. To also get an attack from one specific source port to one specific destination a threshold was investigated. This threshold gives a percentage, when the other port should also be taken into account to get a one port to one port attack. An analysis using more than 200 attacks showed that 40% as one-port-to-one-port threshold would give the best result.

Then, a comparison between the flow-based and packet-based fingerprints was performed which is based on three requirements that were formulated to answer the RQ 3: (1) the number of attack vectors should be similar, (2) the type of attack vectors should be similar, (3) the set of source IP addresses identified as part of the attack should be similar. A first comparison between the packet-based and flow-based approach showed that the number of traces for lower attack vectors (<4) looks similar, but it is not the same. Also mostly the same attack types were found, but the number of each attack types is different. Therefore, this result can be improved.

To mitigate this difference, the packet-based DDoS dissector was adapted to also use the port for determining the service instead of the service protocol field. The comparison of the attack vectors and the attack types showed, that the numbers were more similar. The number of traces for one attack vector are the same for the packet-based and flow-based approach. For two and three attack vectors the numbers move closer together.

Also the number of attack types in each number of attack vector become more similar. The difference in the attack vectors is based on using application information for DNS, which is not available for flows, and for TCP, which were completely different from the flows and packets. With this analysis the requirements 1 and 2 were met (similar number of attack vectors and attack types).

Then the numbers of source IP addresses of each attack are compared to in-

vestigate requirement 3 (similar set of source IP addresses). By comparing the number of source IP addresses, for ICMP 3, ICMP 11, SSDP, Chargen, UDP and Fragmentation the same amount was found for the packet-based and flow-based approach which represents a great result. However, for NTP and DNS the flow-based approach showed more IP addresses than the packet-based approach, as flows do not contain application layer information, for example the DNS queries or the ntp.priv.reqcode. This means that in worst case potentially legitimate traffic is included in the flow-based fingerprints of NTP and DNS.

By using application information from packets the attacks can be specified and potentially legitimate traffic with the same protocol can be filtered out. In the flow-based fingerprints these potentially legitimate traffic is included. This difference was compensated after proposing the *Certainty Threshold*. The source IP addresses with the lowest number of packets were removed from the attack measured in the flow-based network traces to match the number of the source IP addresses of the packet-based network trace. The removed packets were the ones with the highest probability to be potentially legitimate traffic.

For NTP a threshold of 0.2% and for DNS a threshold of 3% was found. After adjusting these, no potentially legitimate traffic was included in the fingerprints anymore. By using this threshold, in the worst case the flow-based DDoS dissector identifies 99.8% of the source IP addresses for NTP and 88% of the source IP addresses for DNS from the packet-based approach. Hence, only 0.2% and 12 % respectively of the IP addresses from the packet-based are not found in the flow-based approach. It also means that **no** potentially legitimate traffic will be included in the fingerprint.

By applying these changes, all three requirements are met and therefore the goal (Making a flow-based approach as precise as a packet-based) is achieved. Operators measuring the internet traffic as flows can use this flow-based DDoS dissector to search for DDoS attack in their traffic. They will only suffer from not getting application information (e.g. DNS queries), but the attack types and the source IP addresses will be taken as an output. This information can then be used to e.g. mitigate DDoS attacks. Some recommendations about future work for this research will be given in the next section.

5.2 Future Work

The thesis shows that fingerprints can be generated with the flow-based DDoS dissector. Although good results have been achieved for the flow-based DDoS dissector (in worst case 88% of the source IP addresses from the packet-based approach and **no** potentially legitimate traffic included), there are still some points that needs

to be adapted. It was seen that there was a problem with comparing the TCP attacks as the flags were different from packets and flows. A reason might be based on problems within the conversion of network traces from pcap to Netflow v5. In future work it should be focused on this problem and ideas to solve this problem should be implemented. Besides this version, there is also v9 and IPFIX, which should verify the results, as they are similar to Netflow v5.

In the packet-based DDoS dissector only one additional information is taken and analysed. In future work more information of the packets should be extracted and used for the analysis.

Another point is that a lot of operators use sampling when measuring flows. In this thesis no sampling is considered. Therefore, the whole traffic is used. By using sampling, less packets are included and therefore there is less data to analyse. The results of a fingerprint might be different. In future work the impact of sampling should be analysed.

Some operators also use other measurement types such as sFlow. In future work these types should also be integrated into the DDoS dissector, so that operators using e.g. sFlow can also benefit from the fingerprinting of DDoS attacks.

Bibliography

[1] Netflow architecture. URL https://docs.ipswitch.com/NM/86_Traffic%20Analyzer%20v1/03_Help/index.htm?26563.htm?toc.htm.

[2] José Jair Cardoso de Santanna. *DDoS-as-a-Service Investigating Booter Websites*. PhD thesis, University of Twente, 2017.

[3] Vern Paxon. A defence mechanism: Dns based ddos attack. In *Computer Communication Review*, 2001.

[4] Saman Taghavi Zargar, James Joshi, and David Tipper. A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks. In *IEEE Communications Surveys and Tutorials*, 2013.

[5] Andrew Ross. Ddos attack volumes increase by 110 % in q3 2018, according to link11s new report. URL https://www.information-age.com/link11-ddos-attacks-123476662/.

[6] Nour-Eddine Lakhdari. Fingerprinting malicious ip traffic. Master's thesis, Concordia University, 2014.

[7] Arnon Shimoni and Shachar Barhom. Malicious traffic detection using traffic fingerprint, 2014. Ben-Gurion University.

[8] Fu-Yuan Lee and Shiuhyng Shieh. Defending against spoofed ddos attacks with path fingerprint. In *Computers and Security*, 2005.

[9] Abraham Yaar, Adrian Perrig, and Dawn Song. Pi: A path identification mechanism to defend against ddos attacks. In *IEEE Symposium on Security and Privacy (SPi03)*, 2003.

[10] Samant Saurabh and Ashok Singh Sairam. Pt: A path tracing and filtering mechanism to defend against ddos attacks. *Computer Networks and Intellgent Computing*, 2011.

[11] Opeyemi A. Osanaiye. Short paper: Ip spoofing detection for preventing ddos attack in cloud computing. In *18th International Conference on Intelligence in Next Generation Networks*, 2015.

[12] Aditya Akella, Ashwin R. Bharambe, M. Reiter, and Srinivasan Seshan. Detecting ddos attacks on isp networks. In *MPDS*, 2003.

[13] Varsha Nigam, Saurabh Jain, and Dr. Kavita Burse. Profile based scheme against ddos attack in wsn. In *International Conference on Communication Systems and Network Technologies*, 2014.

[14] Claude Fachkha, Elias Bou-Harb, and Mourad Debbabi. Fingerprinting internet dns amplification ddos activities. In *6th International Conference on New Technologies, Mobility and Security (NTMS)*, 2014.

[15] Hakem Beitollahi and Geert Deconinck. A four-steptechnique fortackling ddos attacks. In *Proceida Computer Science*, 2012.

[16] Alefiya Hussain, John S. Heidemann, and Christos Papadopoulos. Identification of repeated denial of service attacks. In *25th IEEE International Conference on Computer Communications*, 2006.

[17] Ahmad Sanmorino and Setiadi Yazid. Ddos attack detection method and mitigation using pattern of the flow. In *International Conference of Information and Communication Technology (ICoICT)*, 2013.

[18] José Jair Santanna. Ddos dissector and ddosdb, 2018. URL `https://github.com/jjsantanna/ddosdb`.

[19] Hashem Alaidaros, Massudi Mahmuddin, and Ali Al Mazari. An overview of flow-based and packet-based intrusion detection performance in high speed networks. In *12th Arab Conference on Information Technology*, 2011.

[20] Luca Deri. Improving passive packet capture: Beyond device polling. URL `http://luca.ntop.org/Ring.pdf`.

[21] Pcapng. URL `https://wiki.wireshark.org/Development/PcapNg`.

[22] Wireshark. Wireshark. URL `https://www.wireshark.org`.

[23] tcpdump and libpcap, 2018. URL `http://www.tcpdump.org`.

[24] Netflow v9 datagram knowledge series: Part 1 - netflow overview, 2012. URL `https://thwack.solarwinds.com/community/solarwinds-community/geek-speak/blog/2012/09/06/netflow-v9-datagram-knowledge-series-part-1--netflow-overview`.

[25] riverbed. What is netflow? URL https://www.riverbed.com/faq/what-is-netflow.html.

[26] Michael. Netflow v9 vs. netflow v5: What are the differences?, 2009. URL https://www.plixer.com/blog/netflow/netflow-v9-vs-netflow-v5/.

[27] Peter Haag. nfdump, 2018. URL https://github.com/phaag/nfdump.

[28] RFC7011. Specification of the ip flow information export (ipfix) protocol for the exchange of flow information. URL https://tools.ietf.org/html/rfc7011.

[29] Michael. What is ipfix vs. netflow v9?, 2009. URL https://www.plixer.com/blog/netflow/what-is-ipfix-vs-netflow-v9/.

[30] Elisa Jasinska. sflow, 2006. URL https://fahrplan.events.ccc.de/congress/2006/Fahrplan/attachments/1137-sFlowPaper.pdf.

[31] Inmon corporation's sflow, 2001. URL https://www.ietf.org/rfc/rfc3176.txt.

[32] sFlow.org. Traffic monitoring using sflow, 2003. URL https://sflow.org/sFlowOverview.pdf.

[33] Don Jacob. Traffic analytics in the new it landscape: Netflow vs. sflow, 2013. URL https://searchnetworking.techtarget.com/opinion/Traffic-analytics-in-the-new-IT-landscape-NetFlow-vs-sFlow.

[34] sfcapd. URL https://manpages.debian.org/stretch/nfdump-sflow/sfcapd.1.en.html.

[35] sflow. Print binary sflow feed to ascii, or forward it to other collectors., 2019. URL https://github.com/sflow/sflowtool.

[36] nfcapd. URL http://manpages.ubuntu.com/manpages/bionic/man1/nfcapd.1.html.

[37] CERT NetSA Security Suite. Yaf. URL https://tools.netsa.cert.org/yaf/docs.html.

[38] nprobe™. URL https://www.ntop.org/products/netflow/nprobe/.

[39] Flowtraq. URL https://www.flowtraq.com.

[40] Libpcap file format. URL https://wiki.wireshark.org/Development/LibpcapFileFormat.

[41] tshark: Terminal-based wireshark. URL https://www.wireshark.org/docs/wsug_html_chunked/AppToolstshark.html.

[42] pcap vs pcap-ng. URL https://osqa-ask.wireshark.org/questions/1891/pcap-vs-pcap-ng.

[43] Michael. Cisco netflow v5 vs. netflow v9: Which most satisfies your hunger pangs? part 2, 2009. URL https://www.plixer.com/blog/scrutinizer/cisco-netflow-v5-vs-netflow-v9-which-most-satisfies-your-hunger-pangs-part-2/.

[44] Damien Miller. softflowd. URL http://manpages.ubuntu.com/manpages/xenial/man8/softflowd.8.html.

[45] Rfc 7012 ipfix information model. URL https://tools.ietf.org/html/rfc7012.

[46] Elisa Jasinska. sflow samples flow data. URL https://sflow.org/developers/diagrams/sFlowV5FlowData.pdf.

[47] Akamai. state of the internet / security q4 2017 report, . URL https://www.akamai.com/de/de/multimedia/documents/state-of-the-internet/q4-2017-state-of-the-internet-security-report.pdf.

[48] Martin McKeay. What were the ddos numbers for q2 and q3 2018?, 2018. URL https://blogs.akamai.com/sitr/2018/11/what-were-the-ddos-numbers-for-q2-q3-2018.html.

[49] Christos Douligeris and Aikaterini Mitrokotsa. Ddos attacks and defense mechanisms: classification and state-of-the-art. In *Computer Networks*, 2004.

[50] Akamai. state of the internet / security q2 2017 report, . URL https://www.akamai.com/de/de/multimedia/documents/state-of-the-internet/q2-2017-state-of-the-internet-security-report.pdf.

[51] Jelena Mirkovic and Peter Reiher. A taxonomy of ddos attack and ddos defense mechanisms. In *ACM SIGCOMM Computer Communication Review*, 2004.

[52] Donald Shin. How to defend against amplified reflection ddos attacks, 2018. URL https://www.a10networks.com/resources/articles/how-defend-against-amplified-reflection-ddos-attacks.

[53] DDoS-Guard. Terminology / attack types. URL https://ddos-guard.net/en/terminology/attack_type.

[54] Dns amplification. URL https://www.incapsula.com/ddos/attack-glossary/dns-amplification.html.

[55] Vit Bukac, Vlasta Stavova, Lukas Nemec, Zdenek Ríha, and Vashek Matyas. Service in denial - clouds going with the winds. In *Network and System Security*, 2015.

[56] Internet Assigned Numbers Authority (IANA). Service name and transport protocol port number registry. URL https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml.

[57] Mark Fullmer and Steve Romig. The osu flow-tools package and cisco netflow logs. In *14th Systems Administration Conference (LISA 2000)*, 2000.

[58] Michael Lucas. *Network Flow Analysis*. 2010. URL https://books.google.nl/books?id=5MDucc0LwiUC&pg=PA54&lpg=PA54&dq=2816+port+icmp&source=bl&ots=BEMm-wnKG4&sig=ACfU3U1h87ULncmkh8iSnwFnJapAPIB_2A&hl=de&sa=X&ved=2ahUKEwjyzrmb4_7gAhUFZVAKHQC4DfgQ6AEwA3oECAcQAQ#v=onepage&q=2816%20port%20icmp&f=false.

[59] A prefix map for using descriptive icmp labels, 2008. URL https://tools.netsa.cert.org/confluence/display/tt/A+Prefix+Map+for+Using+Descriptive+ICMP+labels.

***Note: all URLs were accessed on the 1st August 2019.**