

Recommendations for Network Traffic Analysis Using the NetFlow Protocol

Best Practice Document

Produced by the AMRES/RCUB-led working group NMS

Authors: Ivan Ivanović (RCUB), Slavko Gajin (RCUB)

April 2016

© AMRES/RCUB, 2016

© GÉANT, 2016. All rights reserved.

| | |
|--------------------------|---|
| Document No: | GN4-1-NA3-T2-AMRES-BPD-104 |
| Version / date: | V2.1 / 19-04-2016 |
| Original language : | Serbian |
| Original title: | <i>“Preporuka za analizu mrežnog saobraćaja pomoću NetFlow protokola”</i> |
| Original version / date: | Revision 1 (of the document dated September 2010) / 11 November 2011 |
| Contact: | helpdesk@rcub.bg.ac.rs |

AMRES/RCUB is responsible for the contents of this document. The document was developed by the NMS Group organised by AMRES with the purpose of implementing joint activities on the development and dissemination of documents containing technical guidelines and recommendations for network services in higher education and research institutions in Serbia. Parts of the report may be freely copied, unaltered, provided that the original source is acknowledged and copyright preserved.

The research leading to these results has received funding from the European Union’s Horizon 2020 research and innovation programme under Grant Agreement No. 691567 (GN4-1).

Table of Contents

| | |
|---|----|
| Executive Summary | 1 |
| Introduction | 2 |
| 1 Network Traffic Analysis | 3 |
| 1.1 An Overview of the Technologies Applied in Network Traffic Analysis | 3 |
| 1.1.1 Advantages and Shortcomings of the Analysis of Exported NetFlow Statistics | 4 |
| 1.2 The Architecture of the NetFlow Statistics Analysis System | 5 |
| 1.3 The Location of the NetFlow Collector in the Network | 7 |
| 2 Configuring the NetFlow Exporter | 8 |
| 2.1 Group I – Only the Central Device Supports the NetFlow Technology | 8 |
| 2.2 Group II – Devices at the Edge of the Network Support the NetFlow Technology | 10 |
| 2.3 Group III – Duplication of Traffic | 11 |
| 3 Exporting the NetFlow Statistics Using Layer 2 Devices | 13 |
| 4 Correctly Setting the Export Time Intervals | 17 |
| 5 Virtualisation and the NetFlow Protocol | 20 |
| 6 Analysis of NetFlow Statistics Using the NetVizura NetFlow Analyzer Application | 22 |
| 6.1 Configuration and the Analysis Results | 24 |
| 7 Practical Examples | 30 |
| 7.1 Analysis Using a Graphic Overview | 30 |
| 7.2 Direct Analysis of Raw Files | 32 |
| References | 34 |
| Glossary | 35 |

Executive Summary

This document presents the procedures used for network traffic analysis, which provide a clear overview of the structure of traffic and enable the efficient detection of potential problems and irregularities.

The document first presents the technologies applied in network traffic analysis, including their advantages and shortcomings. It then turns to detailed recommendations for traffic analysis based on statistics obtained through the NetFlow protocol. The recommendations include examples of the correct configuration of the NetFlow protocol on network devices, as well as examples of the indirect implementation of the NetFlow protocol in situations where network devices do not support it.

The document also includes an overview of the implementation of the NetVizura NetFlow Analyzer system for analysing the NetFlow statistics, which is used as a Network Management System in the Academic Network of Serbia and in other NRENs.

Introduction

A number of technologies have been developed to increase our understanding of the behaviour of network traffic. NetFlow technology enables an overview of the statistics of the traffic passing through our network and is recommended for environments where the network devices can support this technology.

Today, NetFlow technology has become a standard, and the majority of network equipment manufacturers implement it in their devices. Thus, the existing network infrastructure can deliver an insight into network traffic characteristics without additional investment and the installation of specific devices. This document describes the basic principles and provides guidelines that administrators should follow to configure the NetFlow protocol.

Although this technology is often readily available on simple router platforms, there are also free software solutions that enable the use of NetFlow technology in situations where it is not supported by the network equipment. Some of these solutions are described in this document.

1 Network Traffic Analysis

1.1 An Overview of the Technologies Applied in Network Traffic Analysis

In today's networks, traffic analysis should be conducted not only up to Layer 2 (SNMP – the number of bytes transferred, or the number of packets transferred at the level of the device interface), but also on Layers 3 and 4.

There are a number of tools that can be used for traffic analysis. They are divided into those that require specialised hardware and those that are based on software solutions that are not dependent on the hardware.

The solutions that rely on hardware are rather expensive, but also significantly faster because they have hardware support. Most often, they are placed in the network so that they are transparent to the rest of the network. Such solutions offer Deep Packet Inspection (DPI) functionality and provide the possibility of taking action if a problem has been detected. Figure 1.1 shows an example of the positioning such a device in the network. Some versions of these devices are used not only to detect problems (IDS – Intrusion Detection System), but also to prevent them (IPS – Intrusion Prevention System).

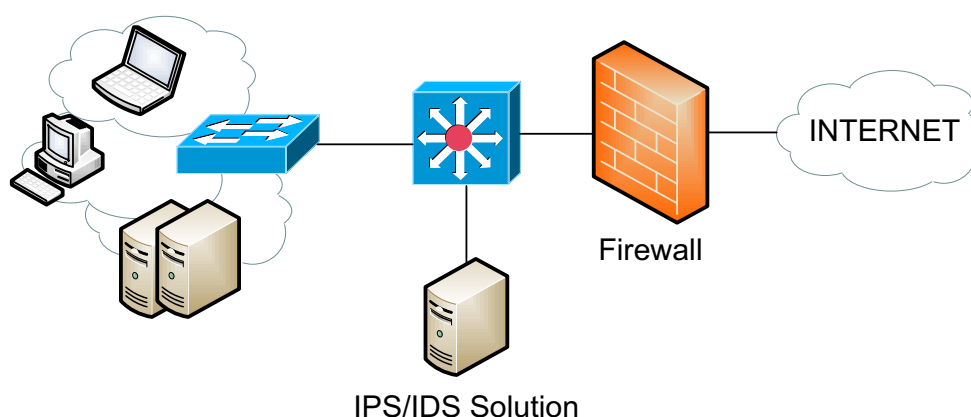


Figure 1.1 – An example of the placement of the traffic analyser in the network

The IDS/IPS device should be placed as close to the end users as is practicable, in order to attain the best possible insight into the network traffic statistics. Therefore, the implementation of the IDS/IPS solution should cover only one link (network segment). In order to cover the entire network, it is usually necessary to install several IDS/IPS devices. A downside of such an implementation is in that the data gathered for the individual network segments are stored in several locations (IDS/IPS devices). In order to ensure overall awareness of the network traffic, it is necessary to centralise the data management, i.e., to collect all the data in one location and co-ordinate the automatic procedures from that location.

Another solution is based on analysing the statistics collected and exported by the network devices. Devices capable of data processing at Layers 3 and 4 (such as, routers and L3 switches) can gather the network traffic statistics up to Layer 4 and export them to the server, where additional data processing is conducted. The NetFlow protocol enables this solution.

NetFlow is the term used to describe the technology, the protocol and the format that defines and records the statistics gathered by network devices. The name, NetFlow, as well as the relevant technology, is related to Cisco Systems, Inc., which was the first company that offered this solution. Other manufacturers have developed similar protocols under different names (such as, Juniper – jflow and Huawei – netstream). Since all of these protocols are compatible with the NetFlow technology, the term NetFlow is commonly used.

Today, the most commonly used versions of the NetFlow protocol are version 5 and version 9. In version 5, the statistics gathered are recorded in a fixed format, while version 9 supports flexible formats corresponding to the set of selected parameters for which the statistics are being collected. This is why version 9 is also called flexible NetFlow.

Due to the need for a universal standard, the Internet Engineering Task Force (IETF) has defined the IPFIX protocol as the protocol for exporting network traffic statistics. The IPFIX protocol is equivalent to the V9 NetFlow protocol developed by Cisco.

1.1.1 Advantages and Shortcomings of the Analysis of Exported NetFlow Statistics

Collected NetFlow statistics are exported to a server running an application tasked with processing the data in accordance with a set criteria. The results of the analysis are usually presented in the form of charts and tables. The charts and results thus obtained provide an easy insight into existing problems. The analysis of the NetFlow statistics provides possibilities for the automatic detection of problems and attacks on the network.

The results of this analysis provide us with the following information:

- information on the total amount of traffic between individual subnets (bytes, packets, connections);
- information on the total amount of local traffic (protocols, servers and hosts);
- information about external access to our network (protocol, service, host);
- the detection of traffic blocked by an access list;
- the detection of traffic rejected due to black hole routing;
- a prediction of future traffic behavior;
- communication between autonomous systems (AS);

- traffic information divided by QoS markers;
- an overview of IPv6 traffic characteristics.

The problems that may be identified in this manner include:

- the existence of a virus in the network (a large amount of incoming or outgoing traffic is being generated);
- DoS attacks (a large amount of traffic is being generated towards DNS or Email servers);
- bandwidth abuse (such as, YouTube, Facebook, or Torrent);
- access to forbidden websites;
- attempts to attack/access protected network devices;
- link overload.

Advantages:

- centralised data collection;
- the existing equipment may be used;
- easy configuration;
- the possibility of collecting other parameters during communication, such as, delays, variation of delays or lost packets;
- free applications for collecting the NetFlow statistics.

Shortcomings:

- the applications are not capable of solving the problem themselves (by blocking communication);
- a lot of time may elapse between the moment the problem emerges and its detection;
- only information up to Layer 4 can be collected;
- familiarity with the network is required in order to properly configure the export of data via the NetFlow protocol.

1.2 The Architecture of the NetFlow Statistics Analysis System

Most of the systems that collect and process the NetFlow statistics consist of devices that generate and export the traffic information and devices that analyse the collected information. There is usually only one system that processes such information in the network, while the traffic information may be collected from a number of devices.

Most commercial and non-commercial software uses the components shown in Figure 1.2 to create the application for collecting and processing NetFlow statistics, as defined in the list below:

- Exporter – a device that collects information about the traffic passing through it and exports the information to the analysis system in the NetFlow format;
- Collector – a part of the analysis system that only collects the NetFlow statistics from all exporters;

- **Aggregator** – a part of the analysis system that processes the collected NetFlow statistics according to a set criteria and keeps the obtained results stored in a database or in some other manner;
- **Raw Files** – binary files, in which the analysis system keeps all the collected NetFlow statistics;
- **Database** – a part of the analysis system that stores the information obtained from the raw files and processes it according to predefined requirements (such as, MySQL, Oracle or PostgreSQL);
- **User Interface** – a web application used to view the processed statistics.

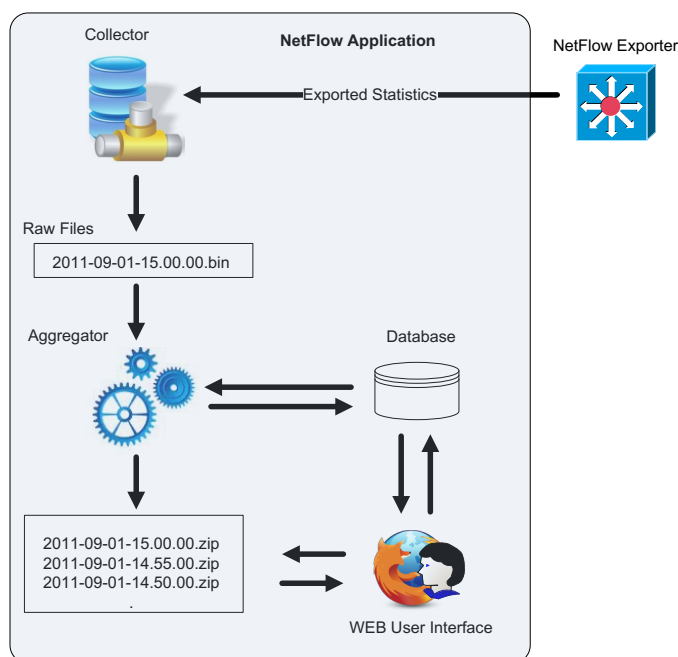


Figure 1.2 – Components of the NetFlow system for analysis of the statistics

The explanations provided below are based on the architecture of the NetVizura NetFlow Analyzer application used for collecting and processing NetFlow statistics at AMRES.

Although NetFlow statistics can be exported to a remote central server from a number of network devices, Figure 1.2 shows an example of the export of the NetFlow statistics from one device only (NetFlow Exporter) as well as the entire course of processing the collected NetFlow information. The first component of the application used for collecting the exported NetFlow statistics sent from the exporter is called a collector. Within set time intervals, this part of the application generates raw files that contain the NetFlow information gathered in the relevant timeframe and stores them in binary format. Once the collector populates the raw file, the file is passed on to the second component in the system, which is called an aggregator. The aggregator receives the file from the collector and processes it using predefined information from the database. The data thus processed (aggregated) is stored in the database. The aggregator then compresses the raw file previously received from the collector and keeps it in the archive. Compression is used to save space. Experience has shown that the compression ratio is 1:10.

The user interface is a web application that enables us to obtain information on the status of the network, based on the data aggregated in the database. If it is necessary to get more detailed

information about a specific communication, the user may open the relevant raw file via the web and filter it according to the desired criteria.

1.3 The Location of the NetFlow Collector in the Network

The location of the device collecting NetFlow statistics depends on the architecture of the network itself. The amount of NetFlow information exported by network devices is directly dependant on the amount of traffic passing through that device (exporter). Experience has shown that the amount of NetFlow traffic does not exceed 1% of the total amount of traffic through the network, so the “distance” between the server (collector) and the network device exporting the data (exporter) is not relevant. The accessibility and the security of the server are the more important parameters.

Usually, the server is physically connected to the main node because most of the main traffic passes through the node. The following is recommended:

- positioning the server in a separate VLAN (management VLAN);
- installing firewall protection on the server;
- ensuring that the NetFlow server is available for traffic analysis in the event of the failure of certain network devices. Therefore, it is necessary to have a separate Uninterruptable Power Supply (UPS) system to provide power for the NetFlow server.

2 Configuring the NetFlow Exporter

Configuring the export of NetFlow statistics on devices depends on the characteristics of the devices themselves, as well as on the network architecture. On newer devices, it is possible to set the collection and export of the NetFlow statistics at the interface level, although only in one of two directions – in/ingress or out/egress. Some devices enable the export of information in both the ingress and the egress directions at the same time. Most older devices support the collection and export of the NetFlow statistics on all interfaces in the ingress direction only.

The most frequently encountered problems in configuring the NetFlow protocol are those related to the duplication of the exported NetFlow statistics. Depending on the network architecture, these problems may be divided into three groups. These groups and the solutions to the problems are described below.

2.1 Group I – Only the Central Device Supports the NetFlow Technology

The first group of problems emerges in a situation where the NetFlow protocol is started on the central device. This is usually the only device that supports the collection and export of NetFlow statistics in smaller campus networks.

Figure 2.1 shows an example of an improperly configured NetFlow export, while Figure 2.2 depicts an example of correctly configured NetFlow export. The **green arrows** show the direction in which the export of information has been configured on the interfaces.

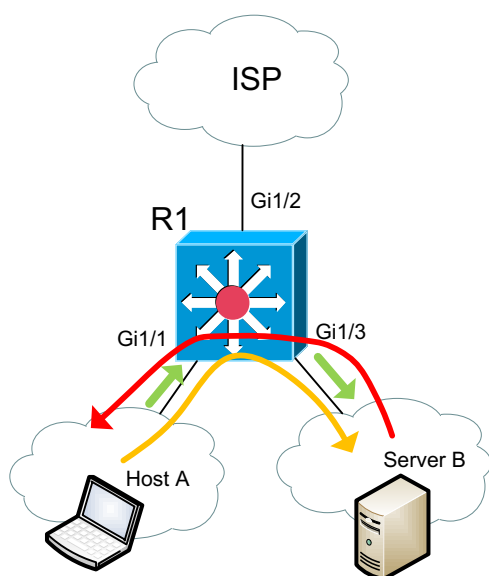


Figure 2.1 – Improperly configured NetFlow export

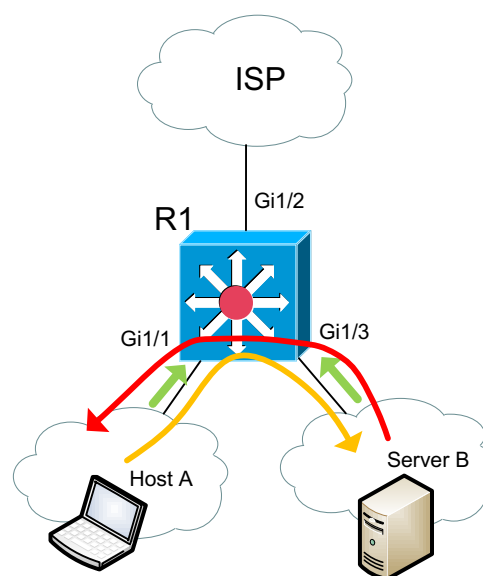


Figure 2.2 – Properly configured NetFlow export

The configuration shown in Figure 2.1 is incorrect because the same flow will be processed and exported to the NetFlow collector twice on its way from Host A to Host B – the first time when it enters the Gi1/1 interface, and the second time when it exits the Gi1/3 interface. Thus, the information about this communication will be duplicated when analysing the collected NetFlow statistics. In this case, data on the communication sent from point B to point A will not be collected.

In order to collect the information correctly, it is necessary to configure NetFlow on all the interfaces, either in the ingress or in the egress direction.

Figure 2.2 shows an example of NetFlow statistics being collected correctly. The collection of the NetFlow information has been configured in the egress direction on both Gi1/1 and Gi1/3 interfaces. The information about a communication sent from point A to point B will be collected at the Gi1/1 interface, while the information about a communication sent from point B to point A will be collected at the Gi1/3 interface. In this way, the statistics about all of the traffic passing through the central device will be collected and exported.

If the communication between certain subnets in the network does not pass through the central device, as shown in Figure 2.3, the NetFlow statistics about that communication will not be collected. The R2 device in Figure 2.3 does not support the NetFlow technology.

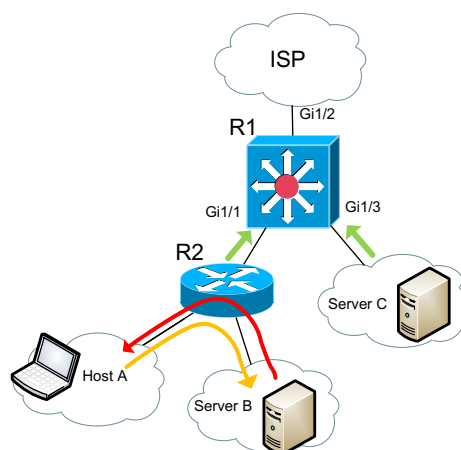


Figure 2.3 – Information about the local traffic shown above is not being exported

2.2 Group II – Devices at the Edge of the Network Support the NetFlow Technology

Figure 2.4 illustrates a situation in which the devices located at the edge of the network support the NetFlow technology. Such network architecture is typical of institutions with parts of their network in several locations, which are connected through an MPLS network of a telecom service provider. Configuration and collection of NetFlow statistics should be carried out on all remote devices, but only on interfaces leading towards the end subnets. The **green arrows** in Figure 2.4 show the interfaces on which the collection of NetFlow statistics has been configured and the head of each arrow indicates the direction. Configuration must be done in the same direction for the entire network, i.e., either ingress or egress. In Figure 2.4, the ingress direction has been selected.

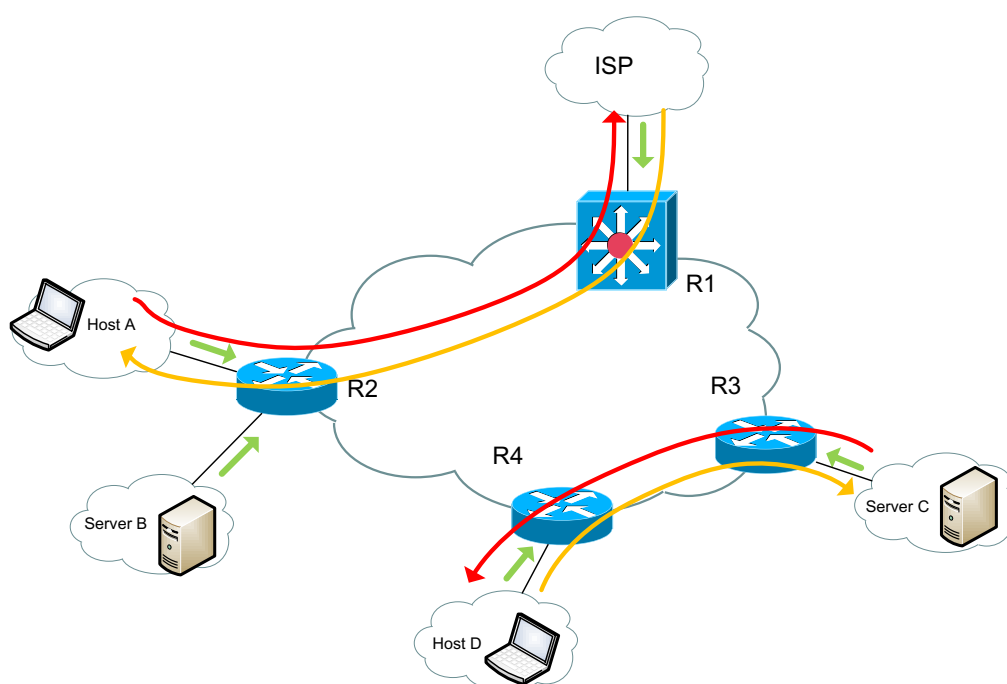


Figure 2.4 – Configuring the collection of NetFlow statistics on devices at the edge of the network

Figure 2.4 shows that the information about traffic between Host A and the ISP will be collected, as well as information about traffic between the local subnets, i.e., the communication between Host D and Server C. In this case, it is assumed that all of the network devices support the NetFlow technology and that they enable configuration of the direction in which the NetFlow statistics will be collected at the interface level.

2.3 Group III – Duplication of Traffic

In more complex network configurations, it is sometimes impossible to avoid the duplication of the NetFlow statistics on the hardware level, i.e., by only configuring the network device itself. Figure 2.5 illustrates an example of such a situation, involving two network devices at the core of the network.

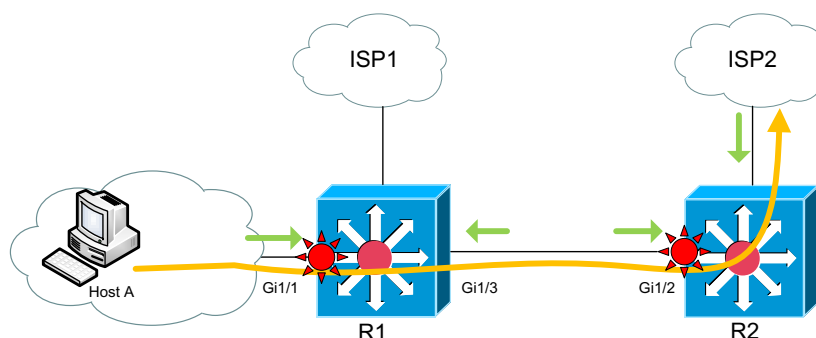


Figure 2.5 – Duplication of the NetFlow Statistics

Figure 2.5 shows a situation in which the central devices, R1 and R2, enable the collection of NetFlow statistics at all interfaces, but only in the ingress direction. The communication leading from Host A to ISP2 will be processed twice, the first time when it enters device R1 at the Gi1/1 interface, and the second time upon entering device R2 at Gi1/2. The communication leading from ISP2 towards Host A will also be processed twice. Since it is not possible to prevent the duplication of statistics at the hardware level, duplication must be detected in the application that analyses the NetFlow statistics. There are several ways to do this.

One of the solutions to this problem requires an application that collects the statistics in order to detect the duplicated information and reject it. Some of the NetFlow fields of the duplicated information will be identical (*src* and *dst* IP addresses and protocols, as well as the *src* and *dst* ports), making it possible to detect the duplication.

Another solution to this problem requires an application that collects the information necessary to be able to filter the collected statistics. Since the exported NetFlow statistics contain information about the IP address of the exporter and the input and output interfaces passing through the exporter for each flow, it is possible to use these fields for the purpose of filtering. In the example shown in Figure 2.5, it is possible to exclude (filter) the traffic passing through the link between devices R1 and R2 from the analysis, as the information is duplicated at this point. The problem of duplication is solved by excluding all of the NetFlow statistics generated by exporter R1 with Gi1/3 in the analysis as input interface, and all of the NetFlow statistics generated by exporter R2 with Gi1/2 as input interface.

3 Exporting the NetFlow Statistics Using Layer 2 Devices

When devices on the network layer do not support the NetFlow protocol, it is possible to use the following solution. Suitable software must be installed on the server, and the NetFlow probe (daemon) must be started in order to analyse the traffic and generate the NetFlow statistics. Then, the network devices should be configured and the traffic from its ports forwarded (mirrored) to the server. Figure 3.1 illustrates this situation.

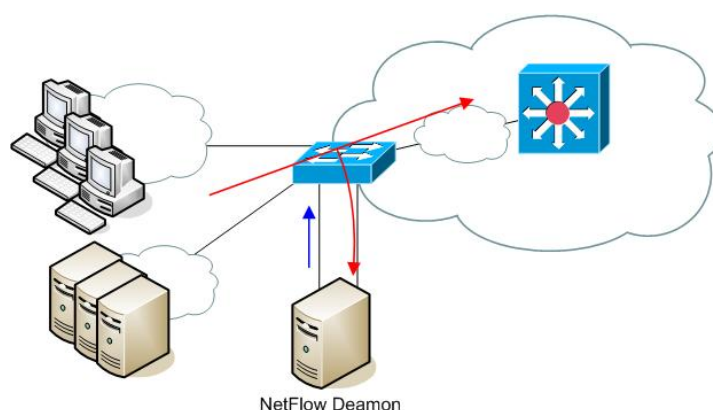


Figure 3.1 – Port mirroring the traffic to the NetFlow server

Figure 3.2 contains a more detailed illustration of port mirroring from uplink port Ge0/0 to port Ge0/1. This solution provides information about the traffic entering at any Fa0/X interface of the L2 device and exiting at the Gi0/0 interface, as well as information about the traffic entering at the Gi0/0 interface. However, it is not possible to collect statistics about the local traffic between ports Fa0/X and Fa0/Y.

In smaller campus networks, a standard workstation computer can be used as a server. It is necessary for the server to have two network interface cards.

The software required by the server to collect the traffic statistics is called NetFlow probe. An example of free probe-software is the Softflowd probe (<http://code.google.com/p/softflowd/>). This probe can be used on both Windows and Linux platforms. Free software for working with NetFlow statistics can be found at <http://www.switch.ch/network/projects/completed/TF-NGN/floma/software.html>. Table 3.1 shows an example of starting the Softflowd application.


```
[root@linuxserver /]# softflowd -i eth2.4 -n 192.168.99.6:2055 -v  
5 -t udp=1m30s -t tcp=1m30s -t maxlife=4m -m 6553
```

Table 3.1 – An example of starting the softflowd application

Tables 3.2 and 3.3 show examples of how to configure port mirroring on Cisco and Juniper devices.

```
switch#(config)# monitor session 1 source interface  
GigabitEthernet 0/0  
switch#(config)# monitor session 1 destination interface  
gigabitEthernet 0/1
```

Table 3.2 – An example of the configuration of port mirroring on a Cisco device

```
ivke@switch#edit ethernet-switching-options  
ivke@switch#set analyzer my-monitor input ingress interface ge-0/0/0.0  
ivke@switch#set analyzer my-monitor input egress interface ge-  
0/0/0.0  
ivke@switch#set analyzer my-monitor ratio 1  
ivke@switch#set analyzer my-monitor output interface ge-0/0/10.0
```

Table 3.3 – An example of the configuration of port mirroring on a Juniper device

In order to avoid duplicating the rejection of traffic, it is necessary for the ports used for copying to have the same characteristics. Likewise, it is necessary that the port on the server, to which the copied traffic is forwarded, has the same characteristics as the port on the switch. Figure 3.2 shows Gigabit Ethernet ports used for collecting the statistics – two at the switch and one at the server. The second network interface card at the server (in this case, FastEthernet) is used to connect the server to the network and to provide regular communication with the server.

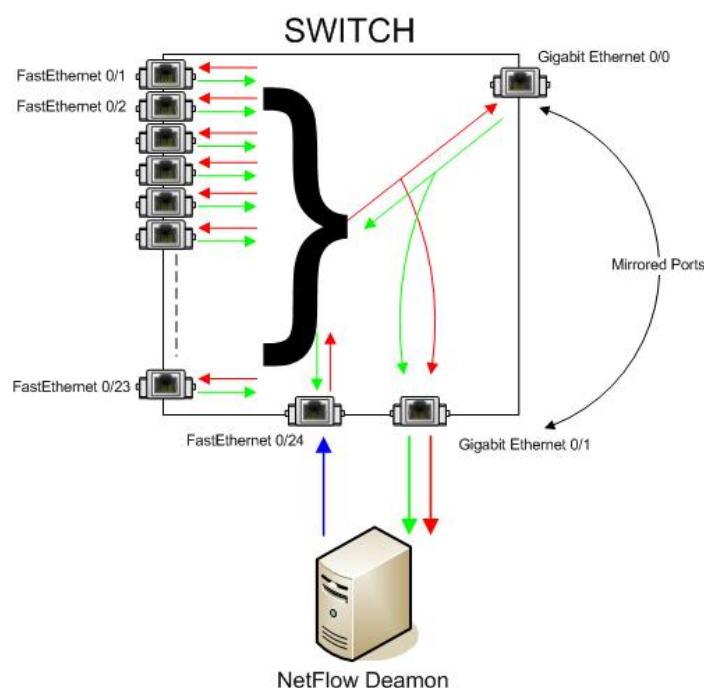


Figure 3.2 – A more detailed illustration of the position of the server and the port connections

Figures 3.1 and 3.2 show the duplication of traffic towards the server when the NetFlow probe is activated. When duplication of traffic is configured on the switch, the interface where all the traffic is mirrored, i.e., duplicated (Gi0/1), becomes unusable for normal communication between devices. Where traffic duplication has been configured, it only forwards the entire incoming and outgoing traffic to and from the interface. The problem is how to export the NetFlow statistics if the interface of the server to which the NetFlow probe is connected has become unusable for normal IP communication. This problem can be solved by adding another network interface card to the server and connecting it to the switch. The **blue arrow** in Figure 3.2 indicates the export of NetFlow statistics from the second network card of the server. This type of configuration enables the export of NetFlow statistics, even from an L2 device.

A disadvantage of this solution is that it takes up additional ports at the switch, and another is that it needs another server. When exporting the collected statistics in this way, some information that is standard for the NetFlow protocol will be lost. The information about AS numbers, the IP address of the exporter, the next-hop address, and input and output interfaces will no longer be available. The applications that analyse the collected NetFlow statistics based on the IP address of the exporter and input and output interfaces cannot be used in this situation, because this information is unavailable when NetFlow statistics are exported in this way.

The solution illustrated in Figure 3.2 allows us to export the collected statistics to the central NetFlow server, which can be located anywhere in the network. If no centralised collection of NetFlow statistics is used in the network, the server shown in Figure 3.2 can also be used as the location where the application that processes the NetFlow statistics is installed. An example is shown in Figure 3.3. However, it is necessary to configure the NetFlow probe to export the statistics locally to IP address 127.0.0.1. Thus, administrators at isolated locations can have an insight into the characteristics of the traffic passing through their switch.

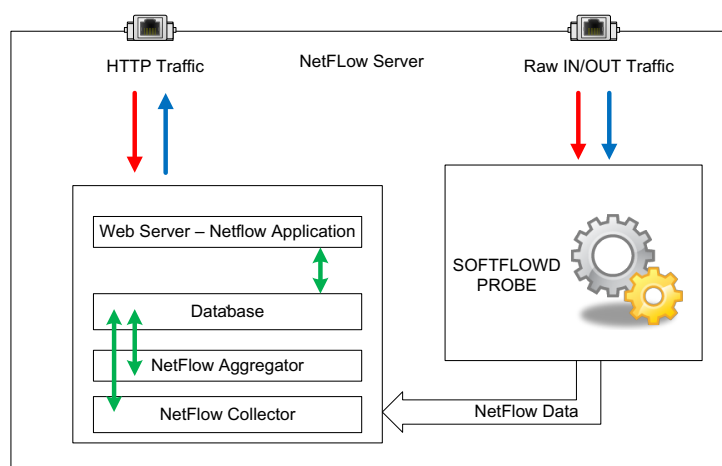


Figure 3.3 – An example of the local export of information

4 Correctly Setting the Export Time Intervals

Applications that process the NetFlow statistics use two different methods to process the information about the time a certain flow in the network started and how long it lasted.

1. The first method gathers the information about the time from the NetFlow raw format, i.e., by reading the timestamp field. In essence, this information contains the exact moment that the relevant flow started and how long it lasted.

2. Applications designed to support a large amount of exported information cannot read the NetFlow timestamp field for each flow during the analysis, as this would slow down the application and increase the size of the database in which the information is stored after processing. These applications define and use a minimum time interval, for example, five minutes, and store the aggregated statistics collected in the last time interval for each individual host as one, five-minute sample. An example of this is shown in Figure 4.1.

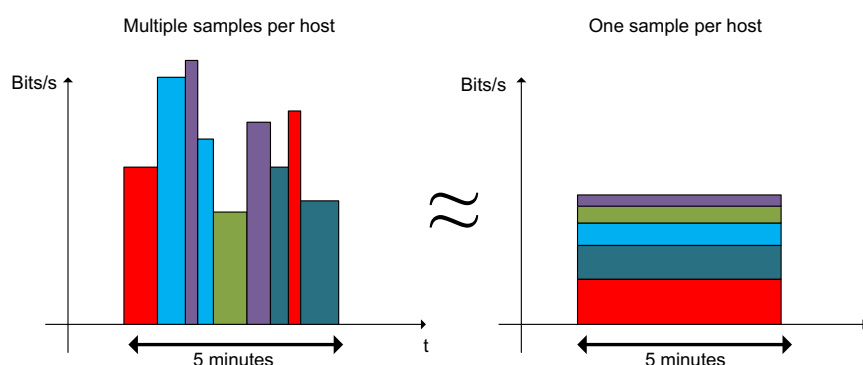


Figure 4.1 – An example of the storage of the statistics in the database

In Figure 4.1, different hosts (IP addresses) are shown in **different colours**. During a five-minute interval, the hosts could generate data flows once or several times. Instead of saving information in the database for each host individually, only the total (summary) information about the amount of traffic transferred within the five-minute interval is saved. Averaging the collected statistics is introduced for the set time interval. The time interval in the example shown in Figure 4.1 is five minutes.

When using a processing application that averages the collected NetFlow statistics for the set time interval, attention should be paid to the time interval in which the NetFlow information is exported.

The time interval should be set in the exporters so that it is shorter or equal to the average time interval used by the application.

In order to obtain correct information about the events on the network during long data flows, as well as during data flows that are too short, the time interval for exporting the information on the NetFlow exporter also requires adjusting when traffic duplication has been configured.

Figure 4.2 shows a situation in which there is a flow lasting for a longer time (twenty minutes). The exporter will send the collected NetFlow information to the collector only after the end of that flow. If the application processing the statistics does not use the timestamp field, which is included in the information received, it will assume that the event took place within the last five minutes and the information about the amount of data transferred will be saved in the database as though the flow lasted for five minutes, instead of twenty minutes. Thus, a higher amount of flow will be shown, which does not correspond to the actual situation.

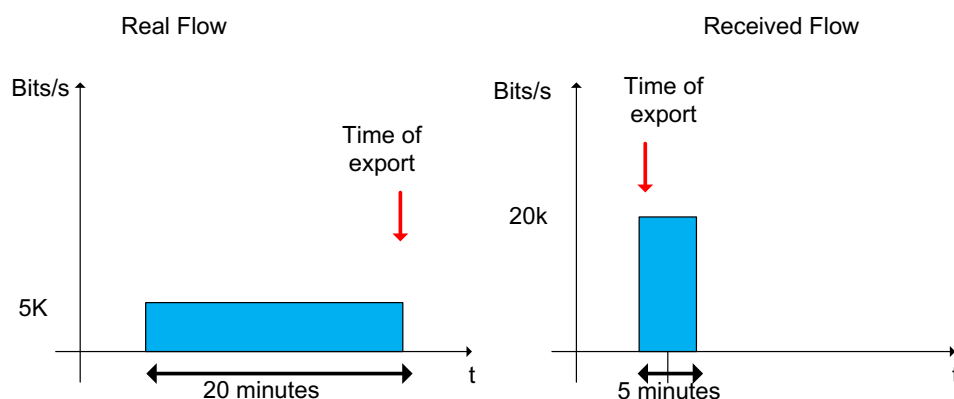


Figure 4.2 – An example of NetFlow statistics being collected incorrectly

There are three types of flow-aging timers in the various devices, and an example of the configuration of aging-timers on Cisco devices is shown in Table 4.1.

- Normal aging is predefined as five minutes.
- Fast aging needs to be set to a shorter time interval than used in normal aging. This is used in combination with the criterion of the amount of transferred packets in order to mark a flow as completed. The idea is to have the short queries, such as, DNS or ping, marked as completed, expired and exported to the collector as soon as possible. In this manner, network attacks can be identified quickly.
- Long aging is used to mark long-duration flows as expired and export them to the collector. An example of this situation is illustrated in Figure 4.2.

```
router#(config)#mls aging fast time 30 threshold 100
router#(config)#mls aging normal 300
router#(config)#mls aging long 300
```

Table 4.1 – An example of setting the timer on Cisco devices

These timers should be set correctly so that the exporters can work properly. When a DOS attack is generated (ping sweep, DNS sweep), the cache memory containing the NetFlow statistics on the exporter may become full within a short time. When the NetFlow cache memory has become full, the exporter will start aging all the information and exporting it to the collector, whether or not the flows are completed. In this way, the exporter frees the cache. This may load the device's processor significantly. Rapid aging of the NetFlow information enables the exporter to quickly mark the short-lived queries as expired, export them, and remove them from the cache, thus freeing the cache for new NetFlow statistics.

5 Virtualisation and the NetFlow Protocol

Certain virtualisation software producers (Citrix, VmWare) also enable the placement of NetFlow probes, as described in Chapter 3.

Figure 5.1 illustrates an example of the placing the Softlowd probes.

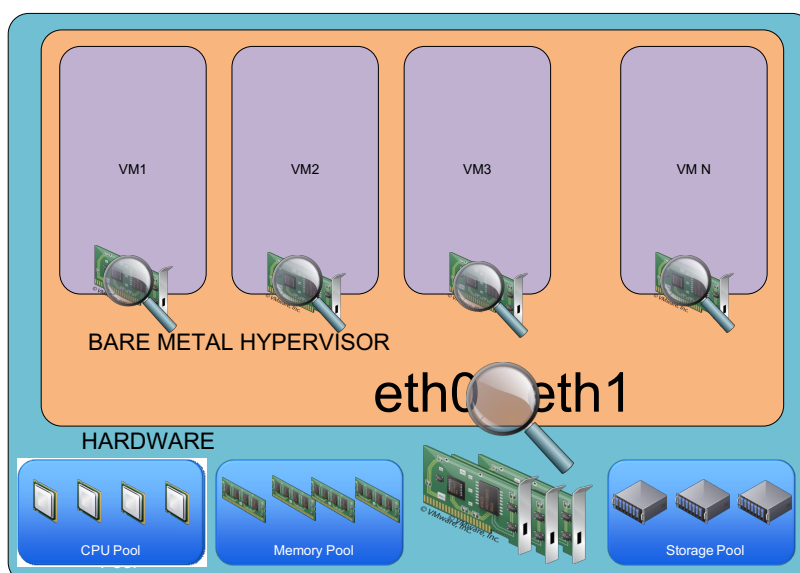


Figure 5.1 – Placing the NetFlow probes in a virtual environment

Figure 5.1 shows that the probes can be placed either on virtual machines or even on a lower layer (bare metal). In the former, they are placed on virtual interfaces, while in the latter, the probes are placed directly on the physical cards.

In this way, the ingress/egress traffic information can be collected even if it is not possible to collect the statistics on the network devices. Figure 5.1 illustrates an example of this type of placement. The collected information can be exported to the system that collects the NetFlow statistics, which may be in the local network or in a remote location, as shown in Figure 5.2.

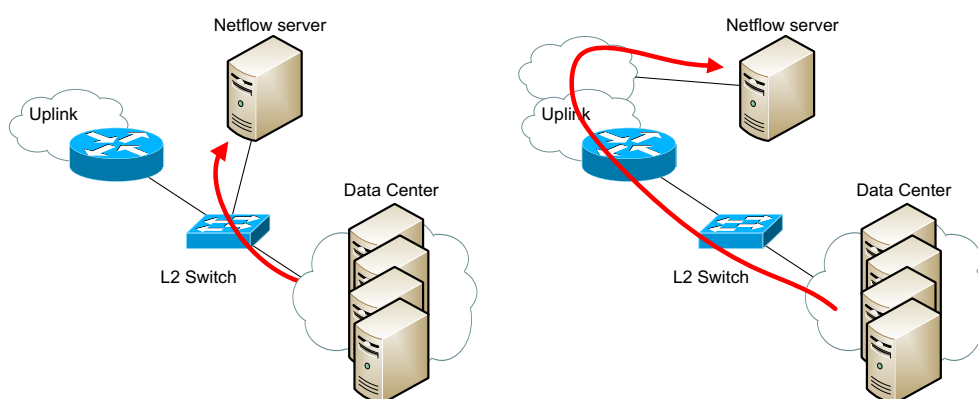


Figure 5.2 - Collecting statistics from a server farm

6 Analysis of NetFlow Statistics Using the NetVizura NetFlow Analyzer Application

Traffic analysis in most applications that are available for the processing of NetFlow statistics is based on the information on input/output interfaces and the IP address of the exporter. If the traffic statistics are collected using NetFlow probes or by way of port mirroring, this information will be lost. During the export of the NetFlow statistics, the NetFlow probe will write the IP address of the interface from which it is sending the statistics in the exporter field (the IP address of the server on which it is installed). The information about input/output interfaces will also be lost since all forwarded traffic enters through one interface server. For the sake of comparison, Figure 6.1 shows the information exported from the router and the information exported using the NetFlow probe installed on the server.

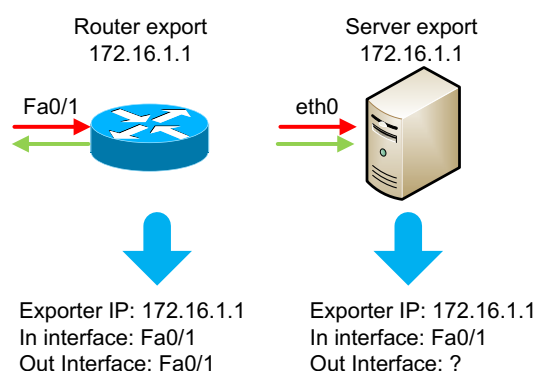


Figure 6.1 – The problem encountered when using the port mirroring option

If the server is used as an exporter, as shown in Figure 6.1, the application cannot distinguish between incoming and outgoing traffic because the application interprets all traffic as incoming, since it comes to the server. If the router is used as an exporter, as shown in Figure 6.1, the application can differentiate between outgoing and incoming traffic, because it receives the information on the exporter during exporting, as well as receiving the information about the input/output interfaces for each flow.

In order to gather and properly analyse the traffic exported using the mirroring method, it is necessary to modify the approach to analysing the collected information. The traffic analysis cannot be based on the information about the input/output interfaces and the IP address of the exporter. The application has to perform the analysis based on other parameters in the exported NetFlow

statistics. One of the applications enabling analyses based on other parameters (traffic source and destination IP addresses) is NetVizura NetFlow Analyzer.

The NetVizura NetFlow Analyzer application is used at AMRES as part of the network infrastructure monitoring system. The application is free for all academic institutions. The use of the NetVizura NetFlow Analyzer application for collecting and analysing the NetFlow statistics is described below.

The NetVizura NetFlow Analyzer application enables the analysis of the NetFlow statistics based on, for instance, traffic source and destination IP addresses. If the source and destination IP addresses are used as criteria in the analysis, it can be established, for each flow, where it has come from and where it is going. Figure 6.2 illustrates this example.

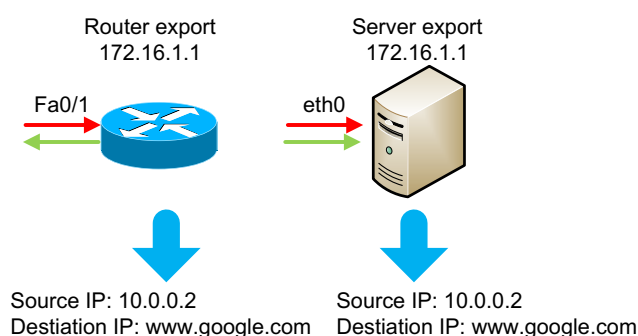


Figure 6.2 – An example of analysis based on src/dst IP addresses

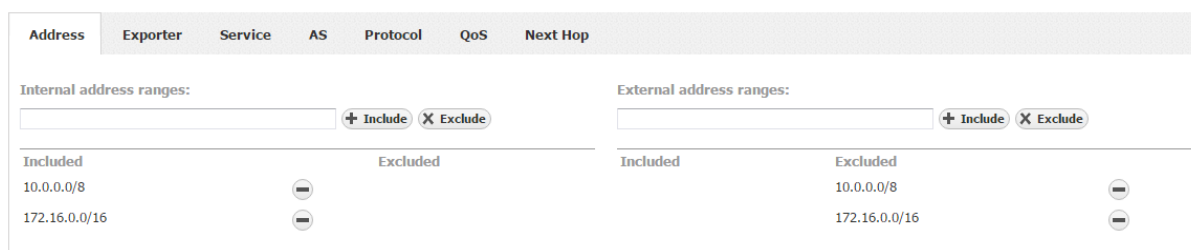
For analysing the collected statistics, the NetVizura NetFlow Analyzer application uses a logical element called Traffic Pattern. This logical element is formed by defining the network range (the local address range), which includes the address space for the relevant network. An external network can also be defined if traffic going towards that network is also monitored. This is the basic configuration required for the Traffic Pattern, although additional filtering criteria can also be defined, such as:

- IP address – IP address of the exporter;
- Service – monitoring the TCP or UDP ports;
- AS – source or destination AS;
- Protocol – L3 protocol;
- QoS – QoS markers;
- Next Hop – Next Hop IP address

6.1 Configuration and the Analysis Results

In the case of campus networks, we typically need to monitor the traffic from our local network towards the Internet and vice versa. The example below explains the configuration of the application and an analysis of the NetFlow statistics.

We have defined the internal address range, 172.16.0.0/16, as the range used on our local subnet. Any other address range can be defined as the external range, except for the range corresponding to the local subnet (in our case, everything except address range, 172.16.0.0/16). Thus, we have covered all the traffic leaving our network and ending in another network that is not our local network. Figure 6.1 shows an example of the local network with two subnets, or two address ranges – 10.0.0.0/8 and 172.16.0.0/16. In this way, we can gather information about the traffic coming from networks 10.0.0.0/8 and/or 172.16.0.0/16 and ending in another network that is neither 10.0.0.0/8 nor 172.16.0.0/16.



| Address | Exporter | Service | AS | Protocol | QoS | Next Hop |
|---|----------|---------|----|----------|-----|----------|
| <div> <div>Internal address ranges:</div> <div> <input type="text"/> + Include X Exclude </div> <div> <div>Included</div> <div>10.0.0.0/8</div> <div>172.16.0.0/16</div> </div> <div> <div>Excluded</div> </div> </div> <div> <div>External address ranges:</div> <div> <input type="text"/> + Include X Exclude </div> <div> <div>Included</div> </div> <div> <div>Excluded</div> <div>10.0.0.0/8</div> <div>172.16.0.0/16</div> </div> </div> | | | | | | |

Figure 6.1 – An example of the configuration of the application to analyse the exchange of Internet traffic

The figures below show an overview of the results of the analyses of the collected statistics by different parameters, such as, subnets, hosts, services, protocols, and QoS markers.

Figure 6.2 shows an overview of the analysis of the collected NetFlow statistics by subnet. In order to obtain detailed information of traffic by each subnet in the internal network, all subnets in the network need to be defined in advance.

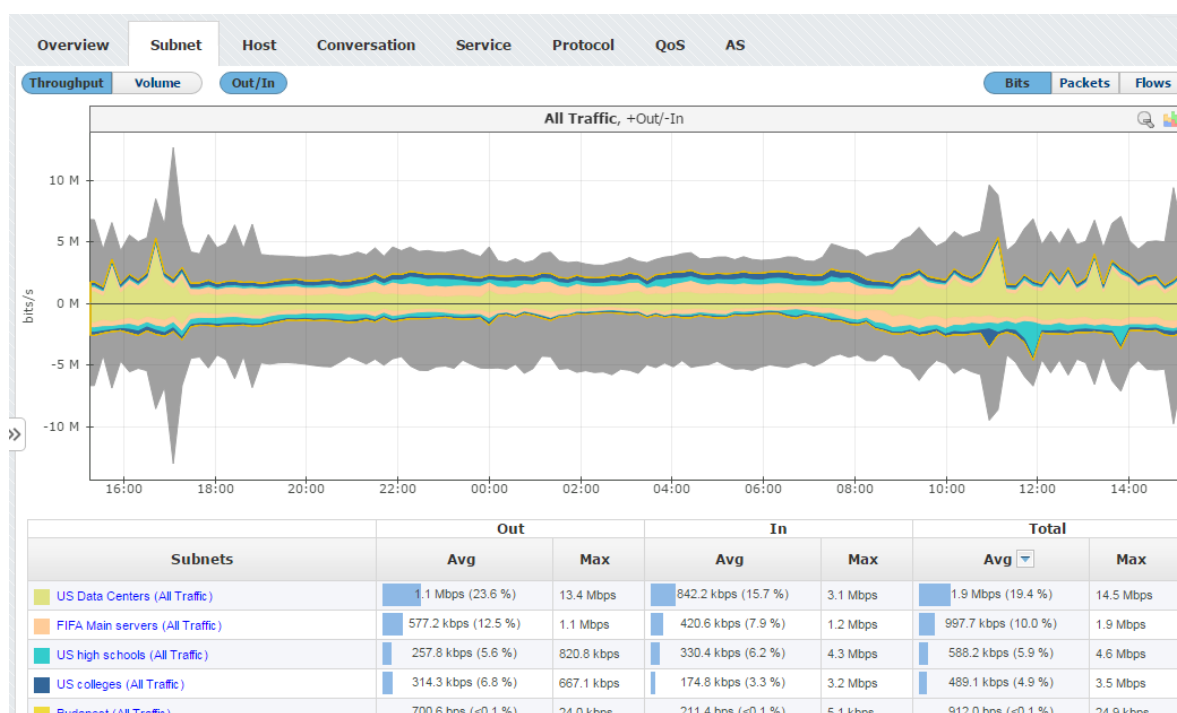


Figure 6.2 – An overview of the results of the analysis by subnet

Figure 6.3 shows an overview of the analysis of the collected NetFlow statistics by host. The positive side of the Y axis shows the traffic leaving the internal network, while the negative side of the Y axis shows the traffic entering the internal network.



Figure 6.3 – An overview of the results of the analysis by host

Figure 6.4 shows an overview of the analysis of the collected NetFlow statistics by conversations.



Figure 6.4 – An overview of the results of the analysis by conversation

Figure 6.5 shows an overview of the analysis of the collected NetFlow statistics by service.

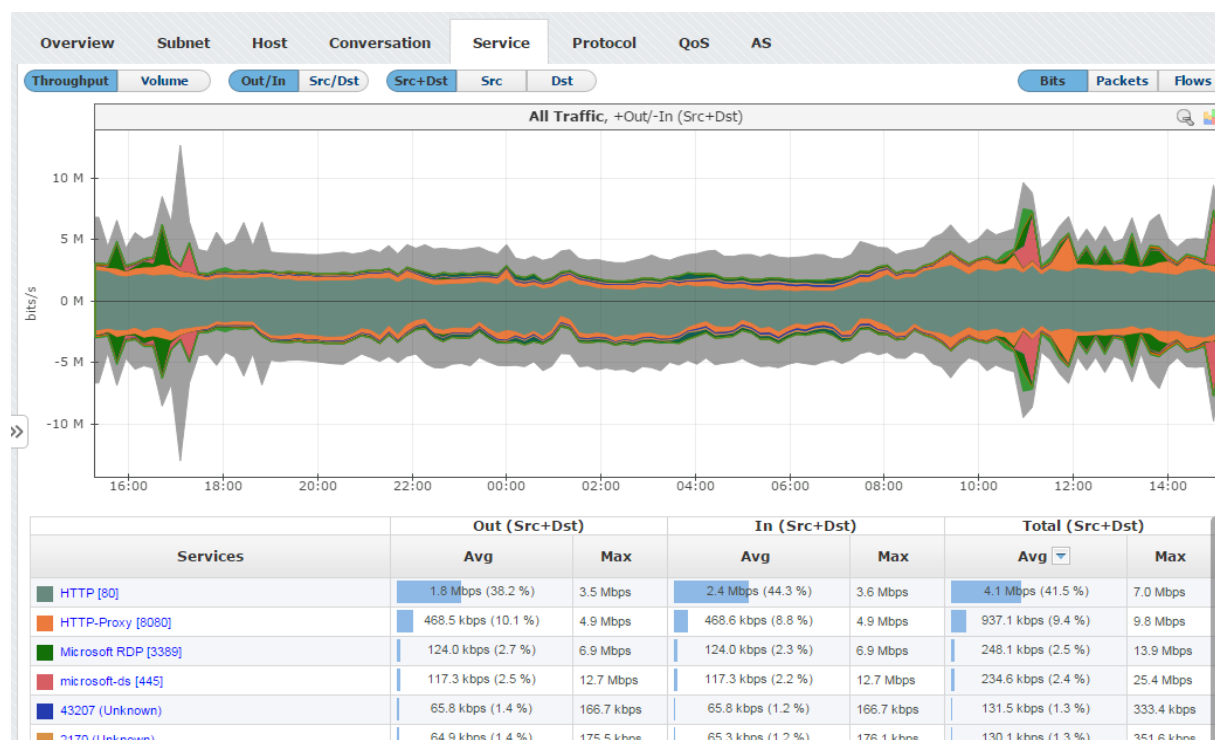


Figure 6.5 – An overview of the results of the analysis by service

Figure 6.6 shows an overview of the analysis of the collected NetFlow statistics by protocol.

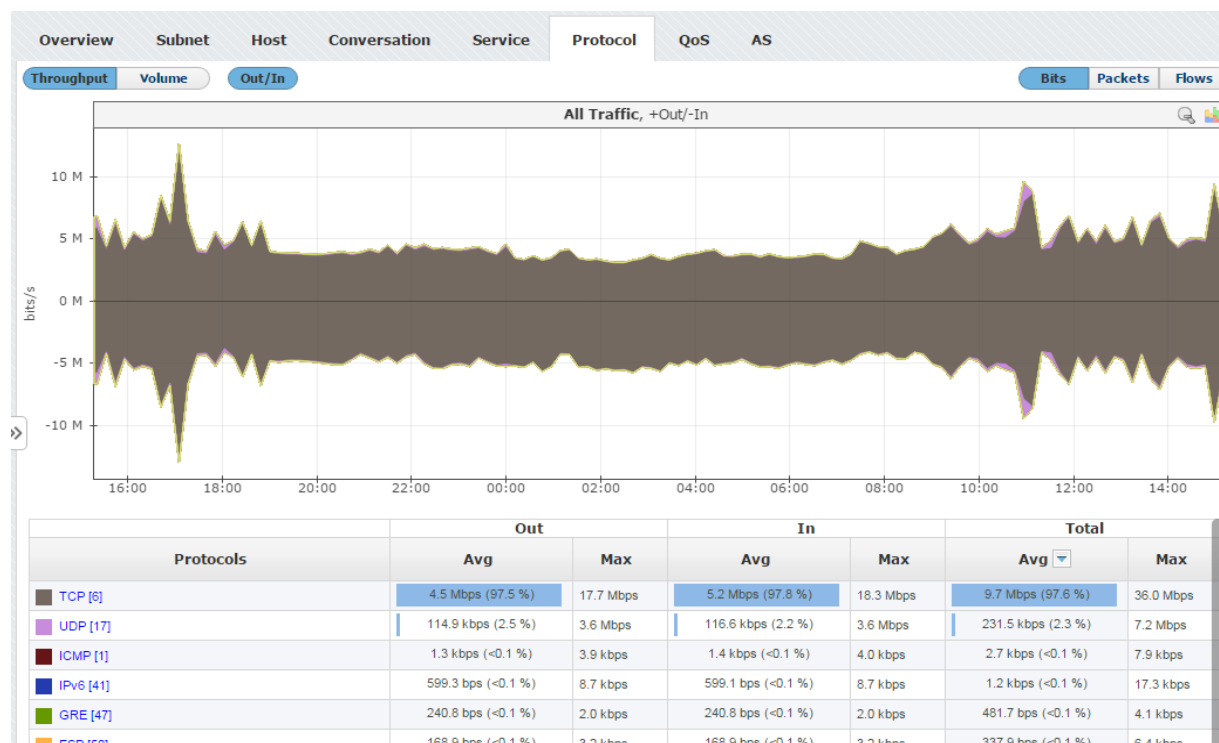


Figure 6.6 – An overview of the results of the analysis by protocol

Figure 6.7 shows an overview of the analysis of the collected NetFlow statistics by QoS markers.



Figure 6.7 – An overview of the results of the analysis by QoS markers

Figure 6.8 shows an overview of the analysis of the collected NetFlow statistics by AS system. In order to populate the AS fields when exporting the NetFlow statistics, the exporter needs to have the BGP protocol running and the entire BGP table.

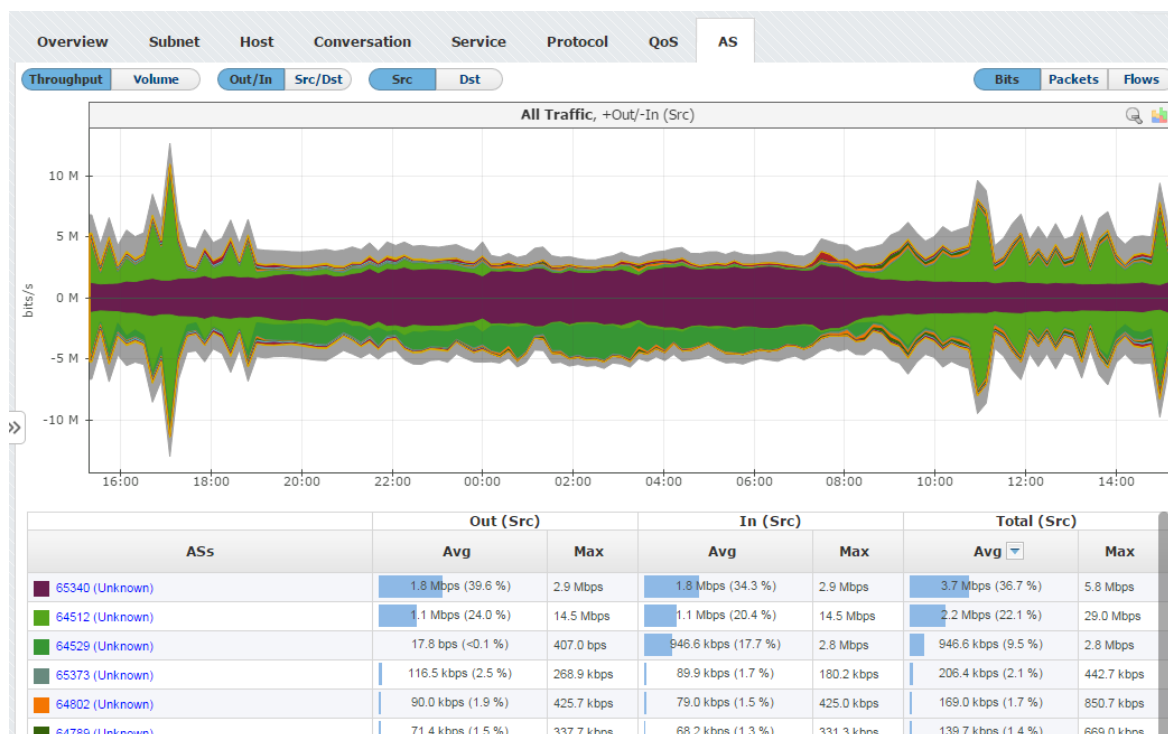


Figure 6.8 – An overview of the results of the analysis by AS value

The above figures provide information about the traffic generated from the internal network and the traffic generated towards an external network. The information that is missing is the IP addresses towards which the traffic is generated, or from which the traffic to our network was generated. This information can be obtained by reading the raw files directly, as shown in Figure 6.8. This figure provides details of the NetFlow statistics collected for each generated communication passing through a specific exporter. Further analyses of these statistics can be performed by filters that can be set at the beginning of each column of the table shown in Figure 6.9.

The application works in the following manner. When a traffic anomaly is detected in the graphs that analyse the traffic by one of the parameters (such as, those shown in Figures 6.1 through 6.8), the problem is first analysed based on the available graphic results. Based on the information obtained from the graphs, an additional, highly detailed filtering of the raw NetFlow statistics is performed in order to find the cause of the anomaly.

| Request Names Details X Clear Export | | | | | | | | | | | |
|--------------------------------------|-------------------------|-----------------|--------|----------|---------|----------|----------|-----|-----------|-------|----|
| Start Time | End Time | Duration | Src IP | Src Port | Dst IP | Dst Port | Protocol | TOS | TCP Flags | Flows | |
| 28-03-2016 14:31:48.00 | 18-04-2016 08:34:35.296 | 1792967.296 sec | 149.85 | 80 | 1.45 | 45897 | 6 | 0 | A | 1 | 2 |
| 28-03-2016 14:30:59.00 | 18-04-2016 08:34:17.296 | 1792998.296 sec | 149.85 | 80 | 1.45 | 58966 | 6 | 0 | A | 1 | 28 |
| 28-03-2016 14:31:19.00 | 18-04-2016 08:34:21.296 | 1792982.296 sec | 149.85 | 80 | 1.45 | 47239 | 6 | 0 | A | 1 | 8 |
| 28-03-2016 14:31:30.00 | 18-04-2016 08:34:17.296 | 1792967.296 sec | 149.85 | 80 | 9.150 | 4931 | 6 | 0 | A | 1 | 5 |
| 28-03-2016 14:31:00.00 | 18-04-2016 08:33:53.296 | 1792973.296 sec | 149.85 | 80 | 79.173 | 4360 | 6 | 0 | A | 1 | 6 |
| 28-03-2016 14:31:41.00 | 18-04-2016 08:34:28.296 | 1792967.296 sec | 86 | 52444 | 27.237 | 55859 | 6 | 0 | APF | 1 | 1 |
| 28-03-2016 14:30:54.00 | 18-04-2016 08:34:09.296 | 1792995.296 sec | 86 | 0 | 81.252 | 2816 | 1 | 0 | none | 1 | 7 |
| 28-03-2016 14:31:18.00 | 18-04-2016 08:34:05.296 | 1792967.296 sec | 86 | 80 | 198.10 | 41989 | 6 | 0 | A | 1 | 3 |
| 28-03-2016 14:31:18.00 | 18-04-2016 08:34:05.296 | 1792967.296 sec | 86 | 80 | 198.10 | 41989 | 6 | 0 | AP | 1 | 1 |
| 28-03-2016 14:31:17.00 | 18-04-2016 08:34:04.296 | 1792967.296 sec | 86 | 1333 | 354.203 | 9200 | 6 | 0 | A | 1 | 2 |
| 28-03-2016 14:31:33.00 | 18-04-2016 08:34:20.296 | 1792967.296 sec | 86 | 18947 | 87.8 | 2140 | 6 | 0 | A | 1 | 1 |

| Request Names Details X Clear Export | | | | | | | | | | | |
|--------------------------------------|-------|---------|--------|------------|----------|--------------|---------------|----------|--------|--------|--|
| IP Flags | Flows | Packets | Bytes | Throughput | Exporter | Interface In | Interface Out | Next Hop | Src AS | Dst AS | |
| | 1 | 2 | 2,928 | - | 0.127 | 209 | 207 | 6.89 | 65235 | 65340 | |
| | 1 | 28 | 40,992 | - | 0.127 | 209 | 207 | 6.89 | 65235 | 65340 | |
| | 1 | 8 | 11,712 | - | 0.127 | 209 | 207 | 6.89 | 65235 | 65340 | |
| | 1 | 5 | 7,320 | - | 0.127 | 209 | 205 | 6.85 | 65235 | 65340 | |
| | 1 | 6 | 8,784 | - | 0.127 | 209 | 205 | 6.85 | 65235 | 65340 | |
| | 1 | 1 | 240 | - | 0.127 | 209 | 205 | 6.85 | 64864 | 65340 | |
| | 1 | 7 | 392 | - | 0.127 | 0 | 205 | 6.85 | 65340 | 65340 | |
| | 1 | 3 | 4,392 | - | 0.127 | 209 | 207 | 6.89 | 65248 | 65340 | |
| | 1 | 1 | 834 | - | 0.127 | 209 | 207 | 6.89 | 65248 | 65340 | |
| | 1 | 2 | 80 | - | 0.127 | 205 | 209 | 106.254 | 65340 | 65447 | |
| | 1 | 1 | 1,464 | - | 0.127 | 209 | 205 | 6.85 | 64512 | 65340 | |

Figure 6.9 – An overview and detailed analysis of the collected statistics using filters

7 Practical Examples

7.1 Analysis Using a Graphic Overview

The following example describes the analysis of an attack on the AMRES network. The NetVizura NetFlow Analyzer application is used and its graphs are shown in Figure 7.1. The page contains graphs that indicate that this attack has come from the internal network.

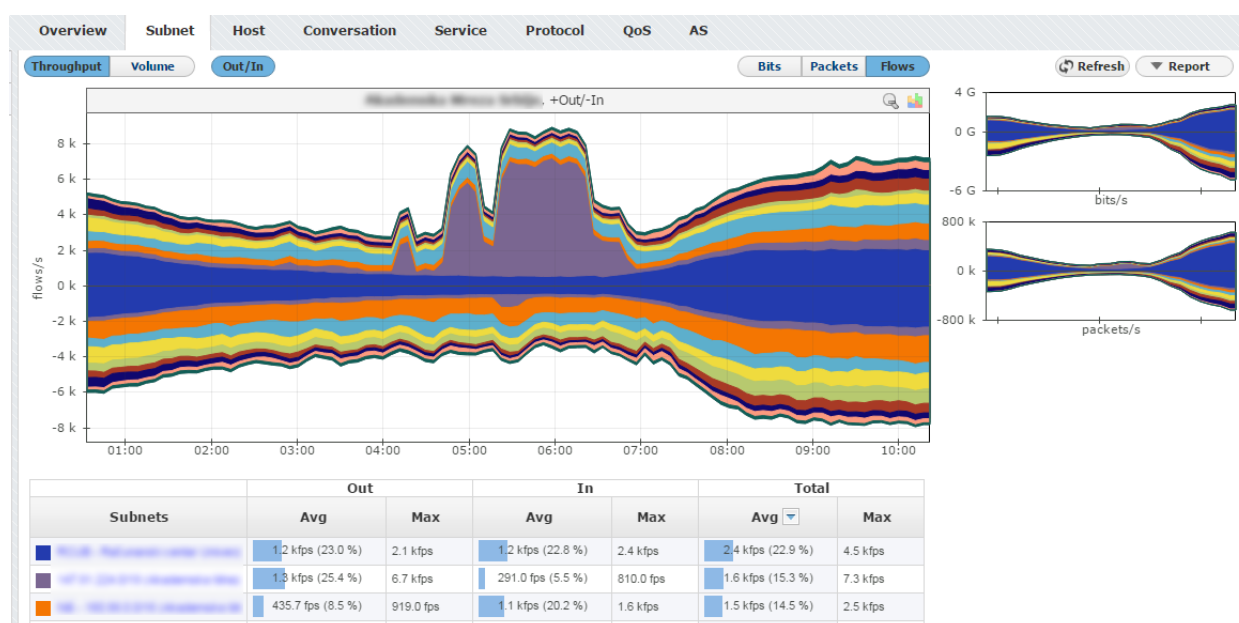


Figure 7.1 – NetVizura NetFlow Analyzer graphs

An irregularity can be seen immediately in the graph shown in Figure 7.1, which displays the quantity of established flows (connections). The detected anomaly is colored in purple. The anomaly is on the positive side of the Y axis in the graph that displays the total quantity of flows per second, which implies that it is traffic generated from our internal network.

In order to obtain information about the host that generated this traffic, as well as about the quantity of traffic in the time period selected in the graph, we need to select the host-display option. Figure 7.2 shows the host-display option. If we need to know the service towards which this data has been generated, we can select the services tab and obtain that information. Figure 7.3 shows an overview of traffic distribution by service.

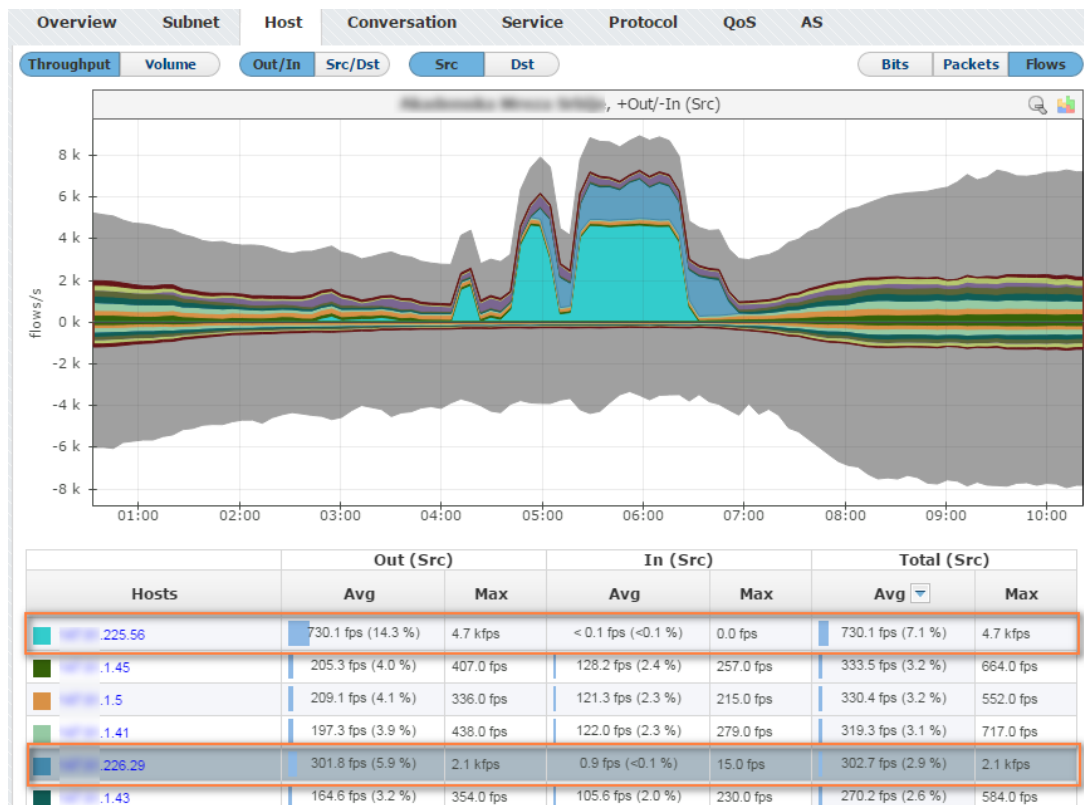


Figure 7.2 – Information about the host that generated a large quantity of traffic

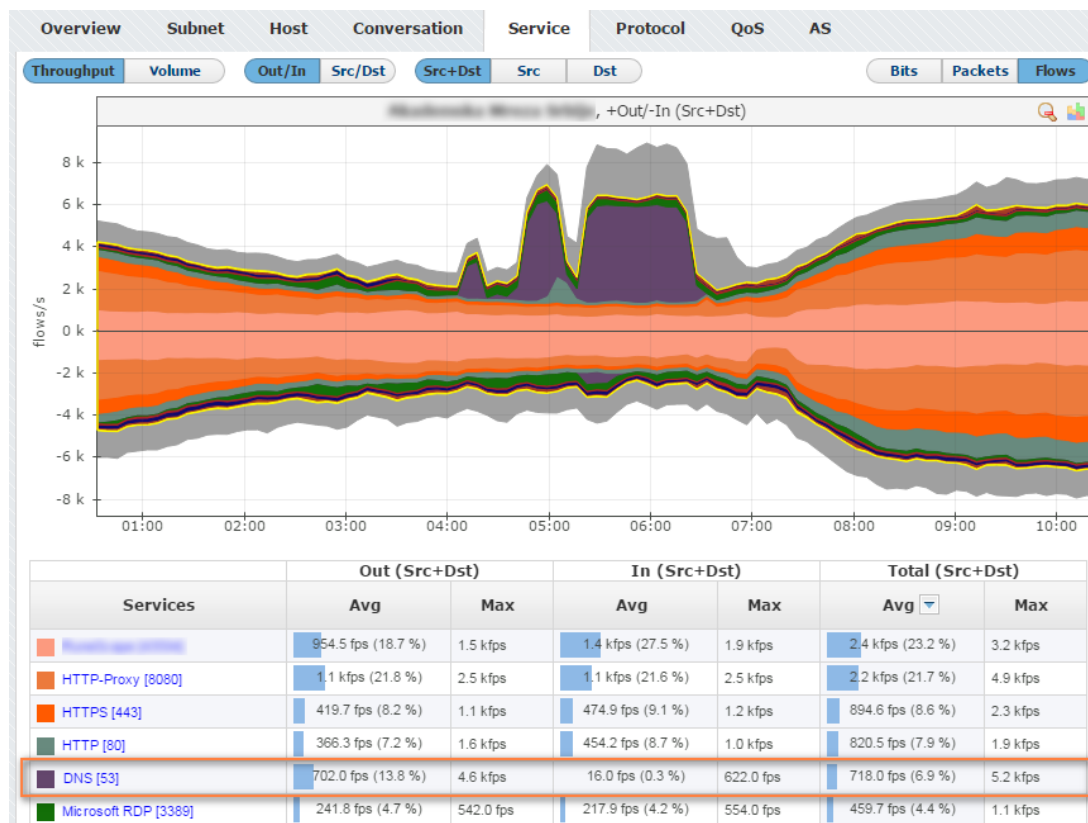


Figure 7.3 – The distribution of traffic, shown by service

Figure 7.5 shows that it is the DNS service. The only information that is now missing is the information about the server to which this traffic is being sent. In order to obtain this information, we need to filter all the collected NetFlow data using the information we previously managed to obtain by analysing the graphs. Therefore, filtering (of some of the raw files within the detected time period) needs to be performed using IP address of the identified source from the internal network and destination port 53. Figure 7.4 shows the result of the filtering.

| Src IP | Src Port | Dst IP | Dst Port | Protocol | TOS | TCP Flags | Flows | Packets |
|---------|----------|---------|----------|----------|-----|-----------|---------|-----------|
| .225.56 | | | 53 | | | | | |
| .225.56 | - | .53.227 | 53 | 6 | 0 | S | 149,540 | 1,641,163 |
| .225.56 | - | .53.169 | 53 | 6 | 0 | A | 147,097 | 1,612,117 |
| .225.56 | - | .53.54 | 53 | 6 | 0 | SF | 231,141 | 1,562,218 |
| .225.56 | - | .0.188 | 53 | 6 | 0 | A | 148,409 | 1,630,040 |
| .225.56 | - | .0.106 | 53 | 6 | 0 | AF | 229,344 | 1,545,302 |
| .225.56 | - | .0.15 | 53 | 6 | - | - | 10,270 | 180,465 |
| .225.56 | - | .17.169 | 53 | 6 | 0 | A | 150,887 | 1,594,486 |
| .225.56 | - | .17.138 | 53 | 6 | 0 | S | 149,418 | 1,555,899 |
| .225.56 | - | .17.115 | 53 | 6 | 0 | A | 147,776 | 1,547,468 |

| Exporter | Interface In | Interface Out |
|----------|--------------|---------------|
| .12.1 | 544 | 537 |
| .12.1 | 544 | 537 |
| .12.1 | 544 | 537 |
| .12.1 | 544 | 537 |
| .12.1 | 544 | 537 |
| .12.1 | 544 | 537 |
| .12.1 | 544 | 537 |
| .12.1 | 544 | 537 |
| .12.1 | 544 | 537 |

Figure 7.4 – The result of the raw-file filtering

The information obtained for the selected five-minute file shows that ~13M of packets have been generated from the internal network by the host x.x.255.56. Figure 7.3 shows that this attack lasted for more than two hours. Figure 7.4 also shows that this traffic has not been forbidden by the access lists, i.e. it has passed towards another AS peer because the output interface of this connection is not 0 but regular VLAN (vl101 with Interface ID 537).

7.2 Direct Analysis of Raw Files

The direct analysis of raw files is illustrated by another simple example showing a very common type of attack. It is an attack that generates a small amount of data in the network so that nothing indicative of an attack can be detected in the graphs, which show the traffic in bytes. However, if the raw files are filtered by the logical criteria corresponding to the description of the attack, interesting results are obtained. In this example, we used a raw file containing the NetFlow information collected in the time interval of five minutes.

This example is based on the assumption that a large number of attacks originate in the internal network and that use the TCP protocol can be stopped by access lists, which are defined in a part of the network. If there are viruses and bots in the internal network that are constantly trying to perform an attack using the TCP protocol, a large number of unsuccessful attacks aimed at

establishing a TCP connection should show up in the raw files, i.e., a lot of traffic should be registered with a TCP SYN flag.

Figure 7.5 shows the results of traffic filtering according to the criterion that only identifies connections with a SYN flag, grouping them by source IP address and sorting them by packets transferred. For the sake of clarity, only the first three rows from the table are shown below.

| Src IP | Src Port | Dst IP | Dst Port | Protocol | TOS | TCP Flags | Flows | P |
|-------------|----------|--------|----------|----------|-----|-----------|--------|--------|
| 10.242.1.98 | - | - | 445 | 6 | 0 | S | 12,968 | 13,035 |
| 10.242.1.98 | - | - | 445 | 6 | 0 | S | 2,608 | 2,657 |
| 10.242.1.4 | - | - | 445 | 6 | 0 | S | 2,261 | 2,284 |

| Flows | Packets | Bytes | Throughput | Exporter | Interface In | Interface Out |
|--------|---------|---------|------------|----------|--------------|---------------|
| 12,968 | 13,035 | 625,680 | 3.8 Kbps | 10.124 | 179 | 0 |
| 2,608 | 2,657 | 127,536 | 3.4 Kbps | 10.124 | 179 | 0 |
| 2,261 | 2,284 | 118,768 | 3.1 Kbps | 10.124 | 63 | 0 |

Figure 7.5 – An example of filtering the raw files

Figure 7.5 shows that the top of the list is populated by addresses from the local network that are trying to establish connections to several destination IP addresses on TCP port 445, but they are blocked by an access list (0 is shown in the column, interface out). Interface 0 means that the exporter has rejected the flow, i.e., that it has been blocked by the access list.

If we select one of the IP addresses in the list shown in Figure 7.1, e.g., the first source IP address, and if we filter its connections with the TCP flag set on SYN and sort them by destination IP addresses, we will obtain the result shown in Figure 7.2. This shows that a packet of 48 bytes has been generated from the IP address of the selected source to various IP addresses. Figure 7.1 indicates that the total number of packets generated during 5 minutes is 13,035, which further shows that the first IP address on the list has generated a huge amount of packets towards about 13,000 different IP addresses on TCP port 445. The final conclusion is that a device in the local network has been infected with a malicious software that is constantly attempting to find and attack a device whose TCP port 445 is open.

| Src IP | Src Port | Dst IP | Dst Port | Protocol | TOS | TCP Flags | Flows | Packets | Bytes |
|-------------|----------|--------------|----------|----------|-----|-----------|-------|---------|-------|
| 10.242.1.98 | - | - | - | - | - | S | - | - | - |
| 10.242.1.98 | 2898 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 1864 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 1168 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 4285 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 1361 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 1122 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 2741 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 1051 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 4325 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 4333 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 1386 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 2184 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |
| 10.242.1.98 | 3373 | 223.104.12.1 | 445 | 6 | 0 | S | 1 | 1 | 48 |

Figure 7.6 – Filtering by the IP address of the infected machine (the first eighteen rows)

References

- [1] Introduction to Cisco IOS NetFlow - A Technical Overview.
http://www.cisco.com/en/US/prod/collateral/iosswrel/ps6537/ps6555/ps6601/prod_white_paper0900aecd80406232.html
- [2] NetFlow Version 9 Flow - Record Format.
http://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9_ps6601_Products_White_Paper.html
- [3] Softflowd NetFlow Probe. <http://code.google.com/p/softflowd/>
- [4] NetVizura NetFlow Analyzer. <https://www.netvizura.com/products/netflow-analyzer>

Glossary

| | |
|--------------|---|
| AMRES | Academic Network of Serbia |
| AS | Autonomous System |
| BPD | Best Practice Document |
| DNS | Domain Name Service |
| DoS | Denial of Service |
| IDS | Intrusion Detection System |
| IETF | Internet Engineering Task Force |
| IPFIX | IP Flow Information Export |
| IPS | Intrusion Prevention System |
| ISP | Internet Service Provider |
| MPLS | Multiprotocol Label Switching |
| NMS | Network Monitoring System |
| NREN | National research and education network |
| QoS | Quality of Service |
| TCP | Transmission Control Protocol |
| UDP | User Datagram Protocol |

