Network Monitoring Based on IP Data Flows

Best Practice Document

Produced by CESNET led working group
on Network monitoring
(CBPD131)

Authors: Martin Žádník
March 2010

Document No: GN3-NA3-T4-CBPD131
Version / date: 24.03.2010
Original language : Czech
Original title: “NETWORK MONITORING BASED ON IP DATA FLOWS”
Original version / date: 1.2 of 3.12. 2009
Contact: izadnik@fit.vutbr.cz

CESNET bears responsibility for the content of this document. The work has been carried out by a CESNET led working group on Network monitoring as part of a joint-venture project within the HE sector in Czech Republic.

The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 238875, relating to the project 'Multi-Gigabit European Research and Education Network and Associated Services (GN3)'.

Table of Contents

Executive Summary

Do you monitor your network? Try to answer the following questions. Which users and which services use the most network bandwidth, and do they exceed authorised limits? Do users use only the permitted services, or do they occasionally "chat" with friends during work hours? Is my network scanned or assaulted by attackers? NetFlow will answer these and other questions.

In the network world, NetFlow is synonymous with monitoring IP data flows. A flow is generally defined as a sequence of packets which share a common feature and pass through an observation point. In the NetFlow terminology this definition is narrowed down to a one-way packet sequence with identical source and destination IP addresses, source and destination ports and protocol number. Various indicators are monitored for each such quintuple, for instance, the duration or the amount of data transferred for a flow.

1 Approaches Used for Network Monitoring

Monitoring of present-day networks can be divided into two basic groups. The first one is based on inspection of packet contents. The contents of the packet are compared to a fairly large database of known samples (regular expressions), and if a match is found a relevant action takes place, for example, communication from the computer that sent the packet is blocked. Most contemporary *IDS* (Intrusion Detection Systems) are based on this principle.

The second group concerns collection and analysis of statistics describing network behaviour. Statistics are gathered with various level of detail, depending on what information we are willing to omit. Basic information is obtained by monitoring the status of key network components. For instance, by monitoring values of *SNMP* counters at network interfaces. The collected data are very approximate because the counters aggregate information about all traffic. Another option is to use the *RMON* (Remote Monitoring) architecture. *RMON* agents are able to carry out several actions, such as collecting statistics about interfaces (network load, *CRC* errors etc.), creating history out of selected statistics, specifying alarms for statistics thresholds being exceeded and generating events (sending alerts). Some agents allow setting up statistics monitoring for several selected users.

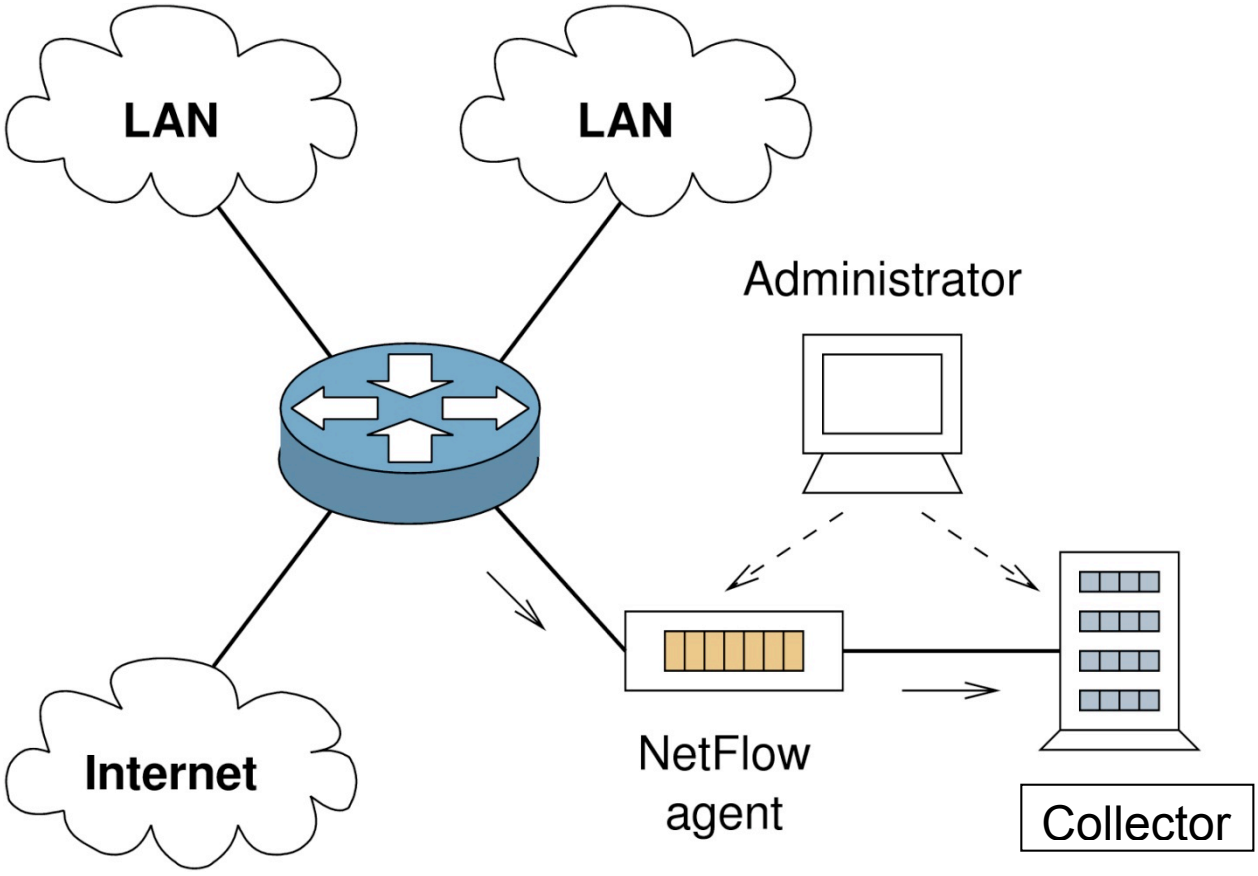
This is insufficient from the perspective of present-day networks. Nowadays companies build their infrastructure (stock exchange and payment transactions, IP telephony, e-commerce) on reliable and secure networks, and hence advanced technologies must be used to get detailed statistics.

The NetFlow technology introduced by Cisco in the late 1990s is among the most popular technologies today. The popularity of NetFlow is due to its convenient level of abstraction. The whole traffic mix is divided into flows based on a quintuple of key data, and besides the quintuple (source and destination IP address, source and destination port, protocol number) other statistics are also monitored for each flow, such as the number of packets and bytes, the time of flow start and end, set *TCP* flags and more. The collected data can help you identify and locate network incidents, show network load, tune QoS settings etc. The NetFlow data can be further aggregated and thus various views of network traffic may be created and important information about applications and users may be obtained, and these can be used later for strategic planning of company development or to verify conformance to network usage policy.

Examples include observing limits for the amount of incoming/outgoing traffic from/to a local network, where the IP-address-based aggregation and sorting according to the number of bytes will display the statistics of the users that violate their limits most heavily.

2 NetFlow Architecture

NetFlow architecture is based on two types of components. The first type are components (let us call them agents) able to collect statistics (records about flows) and send them through the NetFlow protocol. The other type of component is a collector that receives statistics measurements and saves them for further analysis. Agents can be implemented in routers or autonomous probes placed at important network locations. For example, at the gateway of a local network to the Internet, at campus network nodes or data centres etc. A collector can be placed anywhere on the network and collect information from several exporters at the same time. The network administrator accesses the data through a web interface or a terminal.



Picture 1: Architecture to measure flows based on the NetFlow protocol

3 NetFlow Agent

The agent runs two important processes, a measuring one and an exporting one. The measuring process consists of several subtasks. The most basic subtask is receiving the packet, assigning a unique timestamp to it and extracting important items from the packet header. Then the relevant flow record must be found in the memory. Records are usually organised into a field of lists and the address of the relevant list is obtained by calculating the hash value out of the five key flow items (addresses, ports, protocol). The correct record is found sequentially in the list. If a specific record does not exist, it is the first packet of the flow and a new record is therefore created. The next packets of the given flow will contribute to the statistics of the same record. When the flow ends, the record must be released from the memory so that it does not needlessly occupy space for newly created records. This is not so simple. For *TCP* connections, the end of communication can be detected in packets by looking for the *FIN* and *RESET* flags, but if such a packet goes through a different route or gets lost, the record will be in the memory for quite a long time. Furthermore, such flags do not exist for other protocols (*UDP*, *ICMP* and others). Besides the flags mentioned above, heuristics is also used; heuristics detect the end of flow if no packet came to a given record for a long time. If the agent is saturated with new flows, the records are also released. The record can be handed to the exporting process after it was released. The exporting process will process the records and create NetFlow protocol packets out of them.

3.1 Agent Parameters and Configuration

We can tailor the measuring and exporting processes to the network and administrator needs. For instance, if the agent runs on a router the parameters must be set to let the router carry out its main function, i.e. routing.

Sampling is the first parameter that has an impact on the performance of the agent. Sampling determines the probability of the passing packet being the subject of monitoring. Decreasing the probability lowers the load of the measuring process, but unfortunately the quality of the exported statistics is also lowered. When the probability is too low (usually less than 1:10), some characteristics that can be deduced from unsampled data disappear (e.g., the original number of flows). Some NetFlow agents explicitly require sampling with a low probability; therefore it is advisable to learn about the agent capabilities before you start monitoring.

Inactive timeout is another parameter that influences performance; specifically it influences the size of the allocated memory. Inactive timeout is an interval that is used as a heuristic to release records from memory (if the record was not updated during this interval it is released). Too low inactive timeout causes a premature release of the record. In that case several records are created for a flow, which is similar to the broken spaghetti effect (lots of short flows). Increasing the timeout interval will improve the situation, but the allocated memory space will increase. Optimal timeout for present-day networks is 10 to 30 seconds.

So far we described how to release the record after the flow ends. But what if the flow lasts too long? Imagine a user with a slow Internet connection downloading the latest Linux distribution. To let the administrator learn about such events, you must introduce so-called active timeout, and if the flow takes more than that interval then it is released from memory and reported.

For the export process it is necessary to set the destination of the measured statistics, i.e. the collector's IP address and port, or more collectors if the exporter supports that. For some exporters the sampling of outgoing NetFlow data can be configured to prevent the collector being overloaded.

The agent configuration is not provided by the NetFlow protocol, but rather by proprietary methods using a terminal or web interface. For example, with Cisco routers you can use the following commands to launch and configure NetFlow (C2800, IOS 12.4):

Global NetFlow configuration at the router (definition of destination collector and export protocol version)

```
Router(config)#ip flow-export destination 192.168.0.100 60001
Router(config)#ip flow-export version 5
```

Configuration of monitored networks

```
Router(config-if)#ip flow egress
```

Checking flow memory and export

```
Router#show ip cache flow
Router#show ip flow export
```

4 Collector

Just like an agent, a collector runs several processes. The basic process is saving data received from NetFlow agents in a defined format to specified storage. Depending on the type of collector you can encounter other processes such as the presentation process displaying saved data (usually through a web interface) or a process analysing received data (development trends, calculations of long-term statistics, discovering network anomalies). Apart from these processes the administrator can access the stored NetFlow data at any time and collect information he is currently interested in.

The saving process may be implemented differently for different collectors. The two most important formats for saving NetFlow data are saving to standard databases (for example *MySQL*), where database server services are used and the subsequent analysis is run via *SQL* queries, or saving directly into binary files in a specific format depending on the collector.

The advantage of database collectors is easy query processing for saved data, because the database system does most of the work. Conversely, for binary collectors the queries are implemented in the application and the creator of the collector decides what query types to support. Poor performance while accepting NetFlow data, i.e. inserting them into the database, is a clear-cut disadvantage of database-based collectors.

Approximate volumes of saved NetFlow data are around 300 MB per hour for a loaded 100-Mb/s network and 600 MB per hour for a 1 Gb/s moderately utilized network, but it always depends on the specific composition and type of traffic. Most collectors therefore provide advanced tools for long-term administration of saved NetFlow data, such as automatically replacing the oldest data with new data if a predetermined level of data storage allocation is reached or decreasing granularity (level of detail) of older data while maintaining all the details for the newest flows. This also reflects the way NetFlow data are used, i.e., incidents are dealt with immediately or within a few days at most, while older are used for top-N statistics and trend monitoring..

The presentation and analysis processes always depend on a specific collector implementation. Usually there is a graphic interface accessible through a web interface, with optional display of graphs from received data on various time scales, filtering the defined traffic type only, displaying waveforms according to the amount of transferred data, number of flows or packets, summary statistics for a selected period, list of the largest data transfers and IP addresses with the highest load, trend estimates etc.

4.1 Collector Parameters and Configuration

Collectors can be differentiated according to the amount and intensity of the incoming NetFlow data or according to the subsequent usage of stored data.

Sampling of incoming NetFlow data is the first critical parameter. A surplus of performance capacity should be maintained, especially for heavily loaded collectors. For database collectors almost always a longer sampling interval must be set to prevent unexpected packet loss. This is of course also true for collectors which use their own method to save data to disk storage. But these are less sensitive to the amount of NetFlow data and they are usually able to process all data without sampling.

Time intervals according to which the data are saved are another parameter. We commonly meet one- or five-minutes' intervals, but other intervals can be defined as well.

Most collectors let you specify NetFlow data sources and ports where NetFlow data can be accepted. These countermeasures make it harder for an adversary to pollute your NetFlow storage with forged data, which is relatively easy because the *UDP* transport protocol is used.

5 NetFlow Protocol

The most widespread protocol today is NetFlow v5, which is the de-facto standard to transmit flow data. Thanks to its simple structure it is widely supported by both exporters and collectors. The NetFlow v5 record format contains only the source and destination IPv4 address, source and destination port, protocol number, start and end timestamp, the number of transferred packets and bytes, TCP flags, the ToS/DiffServ field and the number of network interfaces at which the flow was measured.

Nowadays this format appears to be restrictive, and a flexible record format defined by the NetFlow v9 protocol is being introduced. Introduction of templates is the most important difference between v5 and v9. Templates are used to define your own record structure and record items. The user will thus define which values she wants to export and how much space she wishes to assign to them. The collector will process these templates and interpret the incoming records based on them. Moreover, the exporter can send much more information about the agent itself (number of received packets, discarded packets, number of sent flows etc.) when using NetFlow v9. Unfortunately, neither NetFlow v5 nor NetFlow v9 supports secure data transfer from agent to collector, because they are assumed to be placed on the same private network. This is not always possible and then records are transmitted over the Internet. In such cases the protocol is susceptible to eavesdropping and submission of forged records. Moreover, the records can get lost because the UDP transport protocol is used; for NetFlow v9 *SCTP* (Stream Control Transmission Protocol) can also be used.

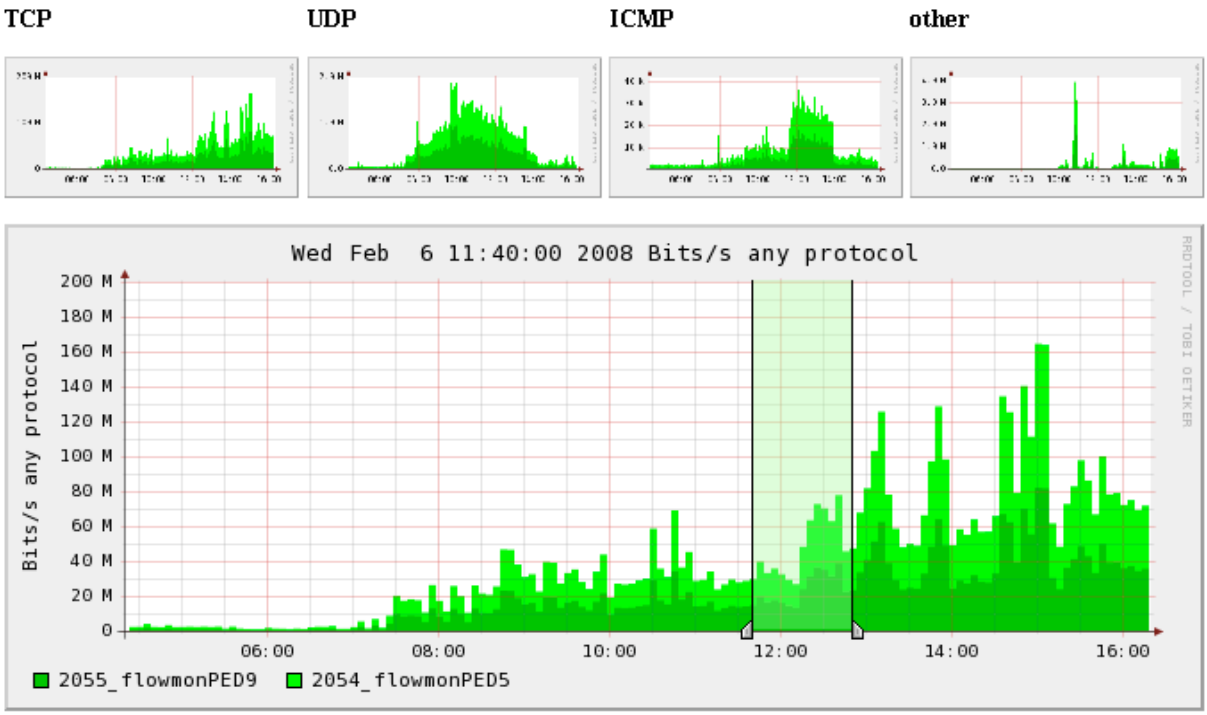
The NetFlow protocols were developed by Cisco and as such are a proprietary solution (NetFlow v9 is described in the informational RFC 3954). The IETF (Internet Engineering Task Force) therefore started to work on a more general, broader definition of a protocol to transfer flow records called *IPFIX* (Internet Protocol for Flow Information Export). This definition is based upon NetFlow v9, from which it adopts the use of templates but it defines possible record items more precisely and defines more measurable values. Unlike NetFlow, IPFIX requires *PR-SCTP* (RFC 3785) to transport data, which is a reliable protocol and it prevents congestion. The definition has been published in several RFC documents (RFC 3917, 5101, 5472, and others). IPFIX is expected to replace NetFlow and presently there are several prototype implementations of exporters and collectors which are able to work with the IPFIX protocol.

6 NetFlow Data Analysis

NetFlow data are received at the collector side where they are saved to disk and subsequently analysed. Either the analysis runs automatically or the user runs the analysis with queries. This creates various views on the network traffic and important data are obtained, for instance, daily traffic distribution, information about users and many more.

An example shows how to discover users who violate the limit for the amount of outgoing traffic from the local network to the Internet. The process is shown on the publicly available NfSen collector.

1. First we select a time interval of interest in the chart (picture 2).



Picture 2: Traffic Amount vs Time Chart

2. We will select aggregation of NetFlow records according to source IP addresses and sorting according to number of bytes, and for brevity's sake we will be interested in the first five only.

3. Listing (pic. 3) will display a table of the users who violate their limit most heavily.

4. If we want to learn whom those users communicated with, we can list all records containing the user's source IP address and sort these records again according to the amount of transferred data.

```
Top 10 Src IP Addr ordered by bytes:
Date first seen          Duration Proto       Src IP Addr    Flows  Packets    Bytes      pps      bps   bpp
2008-02-06 05:49:41.984   334.976 any     121.201.189.240      953     6064    6.4 M       18   159998  1104
2008-02-06 05:49:58.512   297.457 any       142.241.112.8       77     1850    2.1 M        6    59766  1201
2008-02-06 05:50:09.069   299.911 any       51.20.192.101       10     8148   350643       27     9353    43
2008-02-06 05:50:07.102   235.578 any        172.58.120.1       93      881   319750        3    10858   362
2008-02-06 05:50:00.831   291.591 any       204.145.193.9       39      422   235179        1     6452   557
2008-02-06 05:52:34.667    58.697 any        205.191.73.7       10      142   151929        2    20706  1069
2008-02-06 05:52:34.999    53.128 any       205.191.78.32        4      114   136503        2    20554  1197
2008-02-06 05:52:26.652    62.278 any       205.191.76.89        6      120   113490        1    14578   945
2008-02-06 05:52:35.758    48.410 any       205.191.72.51       51      381   112531        7    18596   295
2008-02-06 05:50:14.209   274.068 any       204.69.181.16        9     2470   106226        9     3100    43

IP address anonymized
Summary: total flows: 3240, total bytes: 10.9 M, total packets: 29113, avg bps: 272216, avg pps: 86, avg bpp
Time window: 2008-02-06 05:49:41 - 2008-02-06 05:55:16
```

Picture 3: Top-10 users with the largest outgoing traffic

7 Usage

If you recall the NetFlow v5 record definition it is clear that it is possible to discover who talks to whom, for how long, and what application he uses. And many other kinds of information can be gathered with the advent of the NetFlow v9 and IPFIX protocols.

Measuring network traffic based on flows has many practical applications which contribute to network reliability and security. NetFlow implementations and use differ according to traffic and network characteristics. The most popular applications of NetFlow include the following areas.

- Observing limits and security policies. NetFlow data can be used to monitor how the users comply with network usage policy. For example, whether the limits for incoming and outgoing traffic per user are exceeded. Moreover, with the help of NetFlow data we can display a spectrum of network traffic, i.e., distribution of data between services, and we can focus on the most used services.

- Locating illegally installed servers on the network is an example of complying with security policies. Imagine a common user who brings her laptop to work and connects it to the company network. An *FTP* server is running on this laptop, and Internet users connect to this server and download data, which increases the amount of outgoing traffic and decreases the available capacity for legitimate traffic. Such a server can be exposed by pairing NetFlow data about incoming and outgoing flow and comparing which flow happened first. If the incoming flow initiated the communication then we just revealed a server.

- Another group of undesirable applications are *P2P* networks. Locating them on the network is advisable for several reasons: the data sent and parties communicating over these networks are not trustworthy, illegal data are shared (BitTorrent), applications are disturbing people (*ICQ*, Messenger, ...). Analysis of NetFlow data makes it possible to reveal such applications by observing known ports, for instance, BitTorrent traditionally opens several connections on ports 6881-6889. However, most modern P2P applications are not permanently bound to specific ports, and if standard ports are blocked they scan ports and try to connect to the central server using an open port. Such applications can be detected by a specific IP address (address prefix) of the central server to which the application normally connects to log in to the network. Unfortunately, once such applications become blocked on the firewall, they start to mask their activity in various ways and their detection becomes hard. A blocked Skype can create a *HTTP* or *HTTPS* connection to a secret proxy and through such a proxy it can get to the central server.

- Detection of attacks and suspect activities is a very broad subject, and this report therefore covers only some examples for which it describes how they appear in NetFlow data and how to find them in NetFlow data. A search for an incident can be carried out directly on flows coming from the attacker or on data which are the reaction of attacked computers to the attack.

- Most attacks are preceded by scans of IP addresses and ports. The attacker is in this way looking for ports on which applications listen so that she can exploit their vulnerability.

  There are two types of scans: vertical and horizontal. Vertical scan means scanning ports of a single computer. The list of open ports gives the attacker information about the type and version of the operating system and the possibility to exploit a known vulnerability. Conversely, with a horizontal scan the attacker selected a particular application (port) and she tries to discover which computers are using this application.

  Both types of scans are revealed by an increased number of flows (if the scan is intensive enough). The administrator can then focus on the relevant section and search via aggregation which target user accounts for the largest increase in flows, and then find out who the scanner is (vertical scan discovery). Aggregation by ports can discover an unusual increase in flows for a specific port and locate a scanning user (horizontal scan discovery). Scanning flows will contain only a small number of packets (usually one packet).

  When analysing the reaction of attacked computers we should focus on any unusual increase in TCP RESET packets which a computer generates in response to a TCP SYN packet on a blocked port (vertical scan). To detect a horizontal scan we monitor increased numbers of ICMP Host Unreachable. UDP traffic can be analysed in a similar manner.

- Detection of Denial-of-Service (DoS) attacks. The attacker tries to deny legitimate users access by using server resources or even network resources.

  A well-known attack is TCP SYN-flood, where the attacker keeps opening new TCP connections, which depletes the victim's resources. The victim may try to close the connection by sending a TCP packet with an RST flag set.

  This can be found in NetFlow data by looking for a large number of TCP flows containing a single packet, or in a roundabout way by monitoring an increased number of RST flags in the opposite direction of communication. Trying to identify the attacker is useless, because the source addresses are usually forged.

  DoS attacks are nowadays usually run from so-called botnets, networks of computers belonging to normal users that were attacked by the attacker and to which the attacker gained unauthorised access. Such computers generate legitimate requests for services (for instance, a web page request), but the number of requests saturates the server. Such attacks show up as a flash-crowd effect, where many users try to connect to the server, for instance, to download the newest music video. In both cases, the administrator should be informed.

- Flow monitoring can expose the spread of worms. Worms, unlike viruses, spread using the file system or network. The infected computer opens new connections to other computers and the worm tries to exploit the application vulnerability to spread. In order to identify compromised computers, look for large numbers of unexpected open connections to other computers.

- Quality of Service (*QoS*) monitoring is another field where you can use flow monitoring. Unlike active *QoS* measurement where special packets are inserted into the network traffic this is a passive measurement. *QoS* parameters are thus measured on real traffic.

  This is advantageous on the one hand because measurement runs on user data and the network load does not increase, but on the other hand the experiments cannot be controlled precisely and the measured data might be biased. Usually, data from multiple agents must be compared for so-called one-way measurement, which requires an exact time synchronisation at all measurement points (using *GPS*

for instance). Delay can thus be measured for all network flows on components and transport lines between agents. This can be used to monitor varying times to process specific traffic, for instance, processing of multicast packets on routers may take longer as their processing is more complicated than that of normal traffic.

A flow is considered one-way in NetFlow; however, the two-way connection features can be measured as well, such as *RTT* (round-trip-time), but then the relevant flows must be paired - outgoing and incoming.

- Flow monitoring can be used for optimisation purposes. Imagine that a company has a *SLA* (Service Level Agreement) with its Internet Service Provider to guarantee bandwidth for VoIP traffic. Using NetFlow, it is possible to verify the assigned bandwidth and to plan a decrease or increase of bandwidth according to a maximum load.

  NetFlow statistics can also help balance routing between ISPs and plan peering strategies thanks to the knowledge of source, destination and intermediate nodes that data travel through.

- NetFlow records can be useful for accounting of transferred data or services. Since the statistics contain information about communicating parties, time and amount of transferred data, it is possible to charge individual users according to the communication interval, time of day and amount of data transferred.

8 Ethical Perspective of NetFlow

If we compare flow measurement to packet content analysis, we find that NetFlow describes network traffic from the perspective of its behaviour and not from the perspective of the data itself. Thus it is not possible to use NetFlow to block a specific flow that a worm uses to spread, which is generally possible when inspecting packet contents. In NetFlow data, it is possible to see that the attacked computer initiates too many outgoing connections, which is suspect and then we can focus on such a computer and possibly block its communication at the firewall.

Monitoring flows appears to be most acceptable by all parties concerned (ISPs and users) from the ethical point of view. A simple explanation of the difference between NetFlow and packet inspection can be given by making a comparison with the postal services. Looking at packet contents is de-facto opening the envelope and searching through its contents, while flow monitoring is reading and copying the information on the envelope about sender and addressee, which are publicly known anyway.

In this respect, monitoring based on flows will often meeting the requirements of national legislations about collection by network operators of operational and localisation data of electronic communications.

9 Available Solutions

Cisco routers were originally the only components capable of collecting and exporting NetFlow data. Thanks to the popularity of NetFlow, software agents were written for common PCs, and thus autonomous NetFlow probes were created. Real usage has shown that ordinary computers with no hardware support suffer from packet loss during peaks of network traffic or on lines with intense traffic. Hardware-accelerated probes therefore appeared; these consist of a special network card and a computer.

If a Cisco device is on the network, the first option for flow monitoring is to configure *IOS* to monitor and export NetFlow data. NetFlow can run on Cisco *IOS* routers series 800 to 7500, and also on Cisco Catalyst 6500 Switch and routers series 7600, 10000, 12000 a CRS-1 devices. The measured statistics are exported to one or more collectors in the NetFlow v5 or v9 format. The most interesting features of Cisco NetFlow include incoming traffic filtering (measurement runs only on a subset of the total traffic), support of monitoring *MPLS* (Multiprotocol Label Switching) packets and definition of additional items for security analysis. Features of individual agents can differ with different versions of *IOS* and types of device.

NetFlow data collection consumes the routers' computing resources according to current network traffic and configured parameters of the monitoring process. Before starting experimentation it is advisable to visit a Cisco page (www.cisco.com/go/netflow, NetFlow Performance Analysis) where you can find details about allocated computational resources for individual devices and types of network traffic.

The amount of allocated resources can be one of the reasons to use autonomous NetFlow probes. The advantage of such a solution is the fact that the router carries out its primary function, i.e. routing, and is not burdened with another task. As a consequence, experimenting with a NetFlow probe has no impact on the network traffic. NetFlow probes will naturally be used on networks with no source of NetFlow data, e.g., not built upon Cisco technology.

The Czech company INVEA-TECH (www.invea-tech.com) offers an interesting solution in this area. Its portfolio includes probes (called FlowMon) capable of measuring networks from 10 Mb/s to 10 Gb/s. Probes create statistics fully compatible with NetFlow v5, v9 and IPFIX and send them to an embedded or external collector. Probes are connected to the network through a mirror port of the router/switch or by direct insertion of an optical or metallic fork (*TAP*). The FlowMon product series includes standard probe models for ordinary networks and hardware-accelerated models for critical and heavily loaded lines. Exporters of these probes can send data to several collectors and also filter them according to a scope of IP addresses at the same time. An ISP which provides connectivity to several companies can hand over relevant NetFlow data directly to the relevant companies, which can use them for applications as those mentioned above. If NetFlow data need to be presented to third parties they can be made anonymous (IP addresses or ports can be modified) and users can thus be protected from potential misuse.

NetFlow data generated with the FlowMon probe are sent to an integrated or external collector. Any third-party application or a FlowMon monitoring centre that is part of the package can be used as a collector.

Publicly available software NetFlow agents can be downloaded from the Internet and installed on an ordinary computer. It is advisable to carefully optimise such agents, so that no large packet losses occur. The first tunable parameter is cutting the size of the received packet (snap length). The reason is that only the packet header must be processed during monitoring. Normally, capturing the first 96 bytes is sufficient; this saves unnecessary memory allocation and speeds up the monitoring. It is also advisable to limit the maximum number of monitored flows. Such a countermeasure prevents the measuring process exhausting the computational resources of the probe in critical situations (DoS). Additional optimisation requires recompilation of the operating system kernel, so that classic reading of packets from the network card is replaced with constant probing of the network card to check if any packets are available. This removes a system bottleneck caused by an interrupt storm (in a classical system, one interrupt is generated per packet). After these optimisations, the probe can measure even Gigabit lines with normal traffic.

nProbe (http://www.ntop.org/nProbe.html) is one of the popular, commercially available agents. Its distinctive features are an export format of NetFlow v9 or IPFIX, and that it is offered for Unix and Windows systems. Compared to standard NetFlow/IPFIX items it also contains proprietary items which focus on VoIP monitoring (specifically on *SIP* and *RTP* protocols).

Publicly available software NetFlow agents include fprobe (http://fprobe.sourceforge.net/), which provides data in NetFlow v5 only (and also in v1 and v7, which are not described in this report).

Once the proper probe is selected we need to focus on a collector. To choose the right collector we need to make sure that both agent and collector can process data in the chosen format: NetFlow or IPFIX. It pays off to be especially careful with the IPFIX protocol. Even though both exporter and collector may support IPFIX, this protocol is still under development and interoperability might be an issue.

An example of a typical publicly available collector with an advanced graphical interface is the NfSen collector (http://nfsen.sourceforge.org). It is able to process NetFlow v5 and v9 protocols or the multipurpose tool ntop (http://www.ntop.org) with a special plug-in to collect NetFlow and IPFIX.

The benefit of commercial solutions is technical support and often sophisticated advanced functions such as automatic generation of detailed exports, detection of network anomalies and attacks. Popular collectors include Cisco NetFlow Analyzer (http://manageengine.adventnet.com/products/netflow/cisco-netflow.html) and Caligare Flow Inspector (http://www.caligare.com), or the FlowMon solution mentioned above.

The FlowMon monitoring centre is accessible through a secure web interface and it offers many options such as displaying network statistics as charts and tables with various time scales, generation of top-N statistics, filtering of data according to required criteria, creating user profiles, running security analyses, or setting the generation of automatic alerts to required events such as breaches of security policies. By using expansion modules these functions can be further extended to include *SNMP* monitoring or automatic anomaly detection.

10 Future Outlook

The development of applications based on flow monitoring keeps advancing. Search and discovery of network incidents (port scanning, attacks, exceeding limits or faulty network configurations) becomes automated. Some alerts can be generated by the NetFlow agent itself (record memory overflow can indicate a DoS attack), but thorough analysis of NetFlow data runs on a collector, most often on a fixed interval (5 minutes).

Automated searching in NetFlow data for suspect activities is usually based on exceeding limits for allowed deviation from normal traffic. This means that the search method first starts to learn after it is deployed, and later searches for unusual network traffic behaviour.

The second approach is searching known behaviour patterns for certain anomalies. This could be used when exposing network scans for instance. Such an activity will appear as a line segment in a four-dimensional space built upon source IP address, destination IP address, source port, and destination port (we draw a point in this space for each NetFlow record). By finding all such line segments we also find all scanning activities during a given interval.

The agents themselves are developing, and they are modified to deliver the best quality data about current network traffic. One example is embedding an application decoder into the agent monitoring process. Its task is to locate the application which generated the data transferred. This is simple for applications which use known ports. Unfortunately, if another application uses the same ports (popular port 80 for *HTTP* traffic) or unassigned ports, then it is misidentified or not identified at all. Blocked applications may exploit this and hide their traffic from the firewall or monitoring device behind traffic of another application. If we expand the NetFlow monitoring process to include searching in the package contents before the contents is thrown away, it is possible to more precisely identify the communicating applications (a pattern to detect an ssh connection looks like this: ^ssh-[12]\.[0-9]). If an application has no significant pattern or it encrypts its traffic, then pattern detection is rendered useless. As an alternative, feature analysis (statistical indexes) of such flows needs to be used. Data measured include maximum, minimum, variance and average packet length, and the same indexes are measured for intervals between packets of each flow. This can give you a behaviour fingerprint that you can use to estimate the type of application. For example, interactive voice communication would have the following fingerprint: regular intervals between packets up to 150 ms, low data volume, interval at least 10 seconds, and the same pattern in the other direction.

Statistics about intra-packet intervals are very valuable information not only to detect the application, but they can also be used to measure QoS by determinig uneven delay between packets (so-called jitter).

The future of devices that measure flows lies in extending the monitoring statistics and implementing pattern detection in the packet contents. This will add valuable information to data reported by agents, which will allow related methods such as locating suspect traffic or measuring line quality to perform better.

11 Conclusion

Detailed network monitoring is becoming more important nowadays because the amount of illegal activities increases every year. The attackers become professional and making money is their motivation. It is useful to recall that attackers keep adapting to new challenges, they change types of attacks and try to mask behind legitimate traffic.

From this perspective, flow monitoring appears to be a robust and promising method which makes automated search and differentiation of network incidents possible. Moreover, the trend is to expand the functionality of agents and collectors (such as Cisco *MARS security system*) to provide reliable sources of data about network traffic.

12 List of Figures