Brno University of Technology

NetFlow Security

SEC6NET

Adam Štěpánek

August 29, 2012

1 Introduction

The NetFlow architecture mainly consists of Exporter and Collector. Exporters listen to the network traffic and generate NetFlow records with the information of monitored network. Collector is generally a machine or a device where the records are sent to and stored. They can be later processed and viewed by the administrators for easy maintenance of the network.

As an exporter, either a specialized device or a computer with an installed software can be used. Specialized devices use custom hardware and so they are usually high performance. Software is supplied by the vendor and therefore it is not easy to make changes.

A computer with an installed software is not so high performance and is unable to process huge amounts of data as expensive special purpose devices. The advantage is that a custom solution can be made, especially as a piece of code running on Linux or FreeBSD operating system.

These exporters can be attached to the network by the Test Access Port (TAP) or the Switched Port Analyzer (SPAN) as seen in the Figure 1.

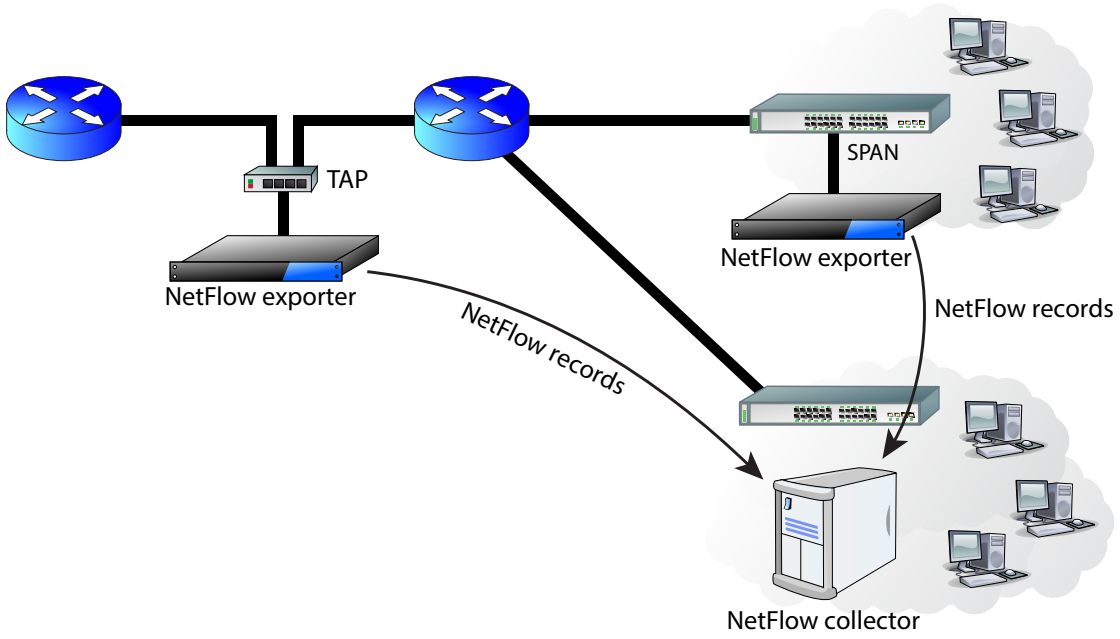

Figure 1: NetFlow architecture with Exporter and Collector

Collector is usually a server with a high capacity storage where NetFlow records are kept. It is running a specialized software which allows to receive, store and presents records in a suitable form to administrators, including filtering by multiple criteria.

2 Conceptual Solution of Secure Transport

Traditionally, data from an exporter to a collector is transfered by an unreliable channel. The situation is depicted in the Figure 2. Exporter creates a record and sends it to Collector. The unreliable channel is either repsented by a dedicated link or by any other infrastructure. It is the Internet in many cases. Data is transfered by the IP protocol which doesn't guarantee a reliable delivery. This is called best effort delivery and it is based upon the current network load.

The solution for the reliable delivery adds two components to the architecture: Sender and Receiver. Sender receives NetFlow records from Exporter and sends them securely to Receiver which creates

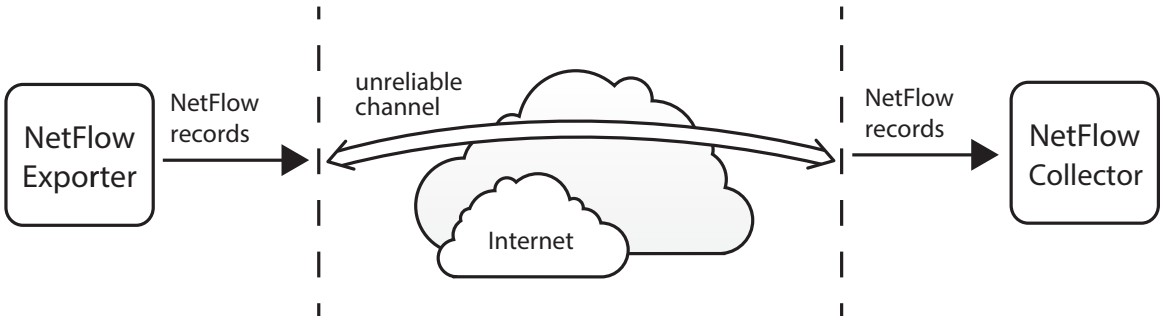

Figure 2: Traditional architecture with the unreliable channel

original records and sends them further to Collector. If there is a failure in the connection between Sender and Receiver records are stored in the buffer in Sender. When the connection is re-established, records are sent and removed from the buffer. The architecture is depicted in the Figure 3.

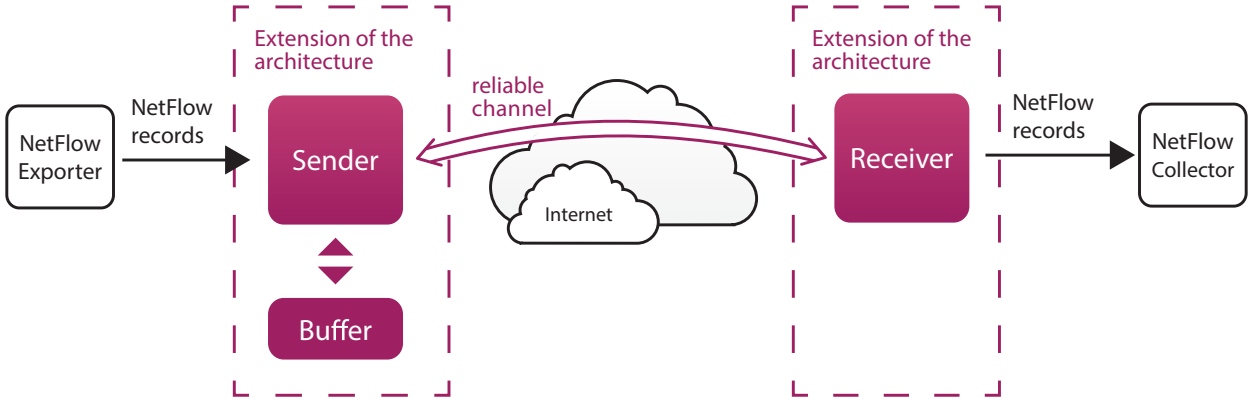

Figure 3: Reliable architecture with two components: Sender and Receiver

Sender and Receiver don't need to be necessarily standalone components. Sender can be a part of the machine where Exporter resides as well as Receiver can be a part of the Collector device.

Regardless of the Sender and Receiver components, the architecture with reliable channel is the same as the traditional one. The physical channel stays untouched. The abstract reliable channel is made up of appropriate technics of communication and encapsulation of data transfered between Sender and Receiver. The whole solution is a noninvasive one and it is not necessary to make extensive alterations to the existing architecture.

3 Implementation

Conceptual Sender and Receiver are implemented, respectively, as `nf_sender` and `nf_receiver` daemons. Implementation language is Perl.

3.1 Sender

Sender consists of two threads: UPD server and TCP client. UDP server receives NetFlow records from the exporter and saves them to Buffer. TCP client picks up records from Buffer, sends them to Receiver and provides a general communication with Receiver.

Buffer as seen in the Figure 3 is a directory `/tmp/flow.queue/` where the cache files are saved.

Tady by mělo být popsané jak se tam vytváří a mažou ty dočasné soubory. Nejsem si úplně přesně jistý, jestli jsem to správně pochopil.

3.2 Receiver

Receiver has only one thread. It ensures receiving of NetFlow records and passing them to the collector.

3.3 Communication

If the connection between Sender and Receiver fails, Sender tries to re-establish the connection after one second. Every next unsuccessful attempt makes the time three times longer, ie. three seconds, nine seconds, twenty seven seconds and so on. Maximum waiting time is 243 seconds. It means once in 243 seconds Sender tries to establish the connection to Receiver until it succeeds.

A někde by tady určitě mělo být ohledně toho šifrování.

4 Installation and Usage

Both `nf_sender` and `nf_receiver` daemons are distributed in a single source RPM package from which two RPMs are generated: for `nf_sender` and `nf_receiver`. Steps for the installation depends on the distribution you use. For more information type: `man rpm`.

When the installation is complete, navigate to `/etc/sysconfig/` and edit the files `nf_sender` and `nf_receiver` according to the instructions in them. These configuration files set the options for running daemons in your desired configuration. If not set, default options will be used. For more information about the available options, see manual pages (type `man nf_sender` or `man nf_receiver`).

In order to start the daemons, you can either reboot your computer and the daemons will start automatically or you type `/etc/init.d/nf_sender start` and `/etc/init.d/nf_receiver start`. The same way you start the daemons, you can stop them using the word `stop` instead of `start`. You can also restart the daemons using the word `restart`.