

18.09.06

## Deliverable DJ2.2.3: Interim Requirements Document

*Requirements for the NetFlow Toolset in JRA2/WI2 and its  
Proposed Architecture*



### Deliverable DJ2.2.3

Contractual date:	31/01/06
Actual Date:	18/09/06
Contract Number:	511082
Instrument Type:	Integrated Infrastructure Initiative (I3)
Activity:	JRA2
Work Item:	WI2
Nature of Deliverable:	R (Report)
Dissemination Level	PU (Public)
Lead Partner	SURFnet
Document Code:	GN2-06-245v4

**Authors:** Maurizio Molina (DANTE), Jacques Schuurman (SURFnet), GEANT2 JRA2 participants

### Abstract

This document describes the vision of the core collaborating NRENs in JRA2/Work Item 2 on how the eventual structure and architecture of the set of tools (hereafter to be called 'the Toolset') should look like. The purpose of this document is twofold: firstly, to align the ideas and thoughts within the working community itself, and secondly, to unambiguously describe the architecture of the Toolset and its individual components (modules) to external parties who are possibly interested to collaborate on developing parts thereof.

# Table of Contents

0	Executive Summary	iv
1	Document rationale	1
2	A Generic Toolset for Traffic Analysis	2
2.1	Background	2
2.2	Toolset Desired Functionality	3
2.2.1	Detection of Anomalies	4
2.2.2	Investigation of Anomalies	8
2.2.3	Response to Anomalies	9
2.2.4	Billing	10
2.2.5	Traffic Engineering / Planning	11
3	Requirements	12
3.1	Generic Requirements	12
3.2	Specific Toolset Architecture Requirements	13
3.2.1	Collection	14
3.2.2	Storage	15
3.2.3	Analysis	15
3.2.4	Anomaly Database	17
3.2.5	Detection	17
3.2.6	Policy Database	18
3.2.7	Data Fusion Module	18
3.2.8	Action Module	18
3.2.9	User Interface	19
4	Conclusion	20
5	References	21
6	Acronyms	22
Appendix A	NREN Feedback on Toolset Aspects	23

## Table of Figures

<b>Figure 2.1:</b> The paradigm of traffic data analysis	3
<b>Figure 3.1:</b> General context of the Toolset	13
<b>Figure 3.2:</b> Overall Toolset architecture	14
<b>Figure 3.3:</b> Real-time monitoring process flow (red) versus forensic post-mortem analysis process flow (green)	16

## 0 Executive Summary

This document outlines the specific requirements for the Toolset to be further developed in the context of JRA2/WI2. It is a joint effort from all the project partners to draft a baseline document onto which the activities of the project can be carried out, as well as to serve as the set of specifications that can be used for any external party interested in either incorporating the Toolset into their own solutions, or parties providing specific components that may be fit for usage within the project's context.

The most important development area identified in this deliverable is to enhance the Toolset's capability to determine new types of anomalies based on what has been monitored from the network. Other enhancements to the Toolset include a wider range of analysis possibilities to the patterns seen, e.g. second or even third degree of derived information, i.e. to be able to dynamically determine anomalies, not only based on a single pattern, but in a series of patterns and their respective change in any of the relevant dimensions.

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

## 1 Document rationale

The reason for writing this document is twofold. Firstly, the community actively involved in JRA2/Work Item 2 felt the need for documenting (and agreement on) the requirements for the Toolset that this Work Item eventually aims to achieve. Based on what has been done so far, we now have a good deal of elements that are either already incorporated in the Toolset, or, might be at a later stage, depending on further development and interoperability testing. To facilitate this, a widely agreed-upon architecture specification is needed. Secondly, it was deemed necessary (and beneficial) to describe the architecture of the Toolset and its individual components (modules) to different external parties who are interested to collaborate in developing the existing or new parts thereof. It should be noted that this document is aimed solely at the developers of Toolset elements and those closely working with them.

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

## 2 A Generic Toolset for Traffic Analysis

### 2.1 Background

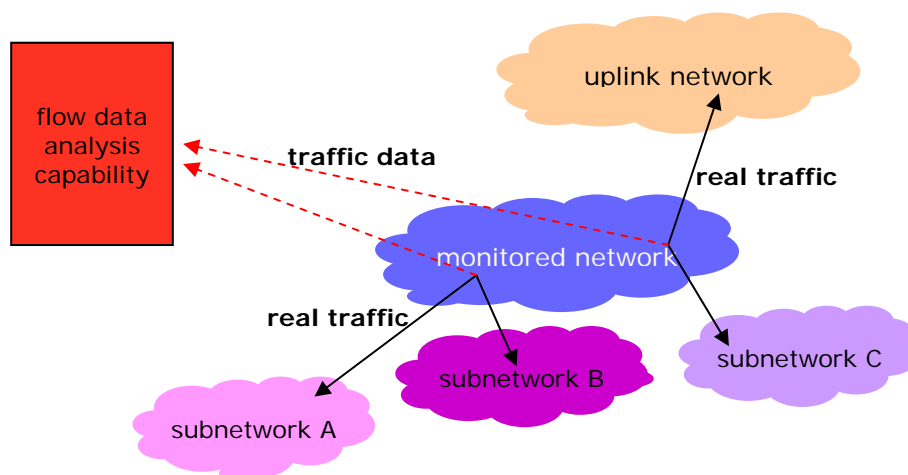
When network traffic is being transported (routed) over the Internet using the Internet Protocol (IP)<sup>1</sup>, specific traffic related information will result as a derivative of this operation. Traffic, being routed on the paradigm of connection-less packet switching as is the case with IP, can be grouped or bundled in so-called *flows*. A flow is essentially a series of packets (*datagrams*) that logically belong to the same transaction or communication of a certain application, which can be identified at one or more abstraction levels higher. Therefore, routing IP traffic will, in itself, generate traffic data that come available to the network operator providing these routing services. This paradigm is depicted in Figure 2.1

While flow data generation, essentially, was designed for accounting, billing and network capacity planning purposes, it soon turned out that the same flow data (actually, *metadata*, as it can be regarded as data *about* data), were very well fit for security purposes. The idea behind this perhaps surprising approach lies in the fact that flow data bears surprisingly accurate information about the nature of the flows being reported on, and therefore, about the behaviour of the application (and ultimately, the user) generating these flows. Specifically, flow data contains, per each flow, the source and destination IP addresses, the number of packets and bytes belonging to the flow and the timestamp of the first and last packet. It soon became apparent, that for example traffic patterns could be made visible by adequately analysing the resulting flow data. In quite the same way, specific patterns strongly indicating malicious<sup>2</sup> traffic could be identified (detected and ascribed to specific sources) as easily. This led to the assumption that, even in the high capacity environment of research networks such as is the case in the GÉANT2 community, the proper collection and analysis of flow data could be a powerful method to initially detect and subsequently respond to various security threats and attacks. Therefore, it was decided in the JRA2 community to undertake the development of a Toolset which would be capable of performing exactly this task. Of course, this requires a well-designed approach of the blueprint of such a Toolset. While several (potential) modules are already under development, and in some cases even deployed, it was felt that there is a need for an overall document describing precisely the requirements of the Toolset, and a draft architecture for fulfilling them.

<sup>1</sup> The IP protocol is meant in a generic way here, it may refer to IPv4 as well as to IPv6.

<sup>2</sup> The term malicious is of course a biased term. It depends on the policies of the entity operating the network whether specific traffic patterns are to be defined as malicious.

This document is to serve as the baseline from which all further activities in the context of JRA2/WI2 involving the development of elements of the Toolset and the integration of (possibly) existing elements are to be carried out. Streamlining the various activities already underway with the JRA2/WI2 project partners, as well as interested (relevant) third parties who in some way want to consider collaborating with these activities should be facilitated on the basis of this document.



**Figure 2.1:** The paradigm of traffic data analysis

Figure 2.1 shows the process of transporting IP traffic via a monitored network (from and to connected uplink or subnetworks) will yield metadata (traffic data) which can subsequently be collected and analysed. These metadata will provide an insight into the nature of the routed traffic, and hence the behaviour of the underlying applications or users.

## 2.2 Toolset Desired Functionality

Before stating the Toolset requirements, we list down its desired functionality. In this section, we summarise the result of input collected from NRENs in several ways (simple e-mail interaction, response to a short questionnaire, verbal comments) and in different times over the past few months. We have tried to capture commonalities in the replies and to organise it in a list of (desired) functionalities (what the Toolset should do). For each of these functionalities we try to:

1. describe it;
2. rank its relevance (of course, we do this without the ambition of giving a precise measure; the ranking we give mainly reflects the number of NRENs expressing interest in the functionality);

3. describe if and how (and to what extent) it is possibly accomplished already in some ways by existing tools (which might therefore be eventually integrated in the Toolset);
4. describe what are the planned/desired ways for the implementation and/or evolution of this functionality.

Note that not all the replies come from people working in CERTs or otherwise involved in security. Therefore, some functionality is not strictly security related. We do, however, choose to include this type of functionality here, as it may pinpoint opportunities for synergy elsewhere in the GÉANT2 project.

## 2.2.1 Detection of Anomalies

### 2.2.1.1 Description

This functionality aims at automatic detection of anomalous traffic in the network. In some of the received feedback, “proactive” was also used instead of “automatic”. While one may argue that there is a difference between the two terms, we preferred to consider them as synonyms, and group all the related inputs in one category, as opposite to the following one (investigation of anomalies). We can say that anomaly detection is concluded to “raising” an alert (it does not matter whether this is done completely automatically or after specific human intervention), while anomaly investigation is concluded as the process of finding out additional information about something that has already been flagged as “anomalous”.

There is of course the issue with defining the very term “anomalous”: based on the received feedback, we can assume all of the following as such:

- **Malicious traffic:** traffic related to network attacks or activities alike (e.g. port scanning, botnet control), traffic generated by worms or viruses;
- **Illegal or doubtful<sup>3</sup> traffic:** traffic generated by applications exchanging or retrieving illegal content or doubtful content, i.e. traffic that is not *certainly* illegal (it would require more investigation) but the probability of this is *high*. The problem is that “illegal” may mean different things according to the concerning NREN’s policies and contracts with their end users. Furthermore, even if traffic is formally legal according to a certain NREN’s policies, it may still be illegal in a more general sense. For example, quoting RENATER’s input: “P2P is allowed on RENATER, but as you know 90% of p2p traffic is used for illegal file transfer”;
- **Topologically undesired traffic:** traffic leading to congestions, or with a risk of inducing congestion. Note: this traffic is not necessarily violating transport contracts with end users. Furthermore, these conditions may be very volatile, and therefore hard to formalise.

---

<sup>3</sup> It is noted that “doubtful” is just a label expressing the fact that this kind of traffic should alert the security team of the network to further analyse or investigate the traffic. Other appropriate labels are “suspicious” or “dubious”. In any case, the reader should be aware of the fact that the underlying patterns deviate from what should be expected on the network, and might therefore violate a specific policy or (local) law.

### 2.2.1.2 Importance

This functionality is very relevant for almost all the NRENs giving input

### 2.2.1.3 Current practice

The methods for detecting anomalous traffic vary a lot, partly due to the wide spectrum of what can be defined anomalous as discussed above. We report here (as an example) some of the described current practices.

#### **Malicious traffic**

For the detection of malicious traffic the most exploited methods currently are the “threshold based” ones like the following approach, as used by GRNET. They consist of two steps:

Step 1: An analysis module (feature extraction modules) gets a NetFlow<sup>4</sup> feed and calculates certain metrics over a certain time window (e.g. 1 minute, or 5 minutes) which are stored in RRDs. Some examples may illustrate this concept: e.g the number of flows with duration <100ms, number of flows with size [32-64B], the number of total flows per second, the number of flows with only the SYN flag set, etc

Step 2: The data stored in the RRDs are processed by detection modules. The main detection techniques are threshold based: (i) Threshold on the difference of a metric from an average value of the recent past. (ii) Threshold on the difference of a metric from a fixed value (determined by a learning phase). (iii) Threshold on the difference of a metric from a predicted value divided by the standard deviation.

IUCC describes a (conceptually similar) methodology: the computed metric is volume, and this can be relative to specific (prefixed) IP source addresses and ports. The administrator can define a threshold for the maximum traffic that can be seen in a 1h or 24h period. The trick is to set initial parameters (prefix lengths, ports, volume thresholds) and then tune it over a period of a few days/weeks until one is at a point where the alerts are only true alerts (reduction of false positives). With this methodology, one can quickly spot botnet<sup>5</sup> hosts that start spewing out loads of void data or hosts that have been taken as *warez*<sup>6</sup> servers.

Note that most of the time these “threshold based” methods are not considered reliable enough to lead directly to the flagging of a security alarm. For example, one problem is that adapting thresholds of metrics involving the number of flows to different sampling rates is difficult, because the variation can be heavily non-linear. Therefore, more investigation (possibly requiring human intervention) is generally needed for deciding whether the anomaly is really a result of malicious traffic and for classifying it. This further step is actually an “Investigation of the anomalies” and will be described later.

<sup>4</sup> NetFlow is a format developed by Cisco and widely accepted as the de facto standard in which traffic information can be stored, shared and analysed. Major versions of NetFlow include version 5 and version 9.

<sup>5</sup> A machine (usually a pc-like workstation) is said to be botnet if it is part of a botnet: a probably very large dispersed set of connected and ill-administrated machines that are actively under the control of a single intruder who can misuse the machines’ (network) power to launch amplified attacks

<sup>6</sup> The term *warez* is generically used for contents that violates current legislation (usually copyrighted materials available on-line without the owner’s consent)

A not uncommon detection method can be represented by an end user signalling that something (e.g. a web server they run, or try to access) is not working or responding slowly, without much more additional detail.

### **Illegal or doubtful traffic**

For several NRENs, this is synonym for P2P traffic, because its content is generally not related to educational and research activities (which is what the NREN's networks should be used for), and because most of the time the exchanged content may be illegal (e.g. because of violated copyrights, etc.). Some P2P traffic can be detected by spotting exchanges using "well known" P2Pports (such as 4662, 6881 to 6889, etc.), but more often other heuristics methods are required. For example, some typical file sizes are "suspicious" because a large download of a DVD or DIVX are split in files of equal size (around 15MB), a Playstation game is split in 45MB files, etc. We emphasise that "illegal", "doubtful" or other terms indicating suspicion are, by definition, fashion oriented and policy based. Nonetheless we present techniques (usually heuristics) to enable the network monitoring instance to recognise such types of traffic.

### **Topologically undesired traffic**

This anomaly category includes human-induced events such as flash crowds generated by announcements of new software distributions, breaking news, etc. Detection procedures are similar to the threshold based ones for malicious traffic. However, only with a further investigation of the anomaly at hand, it is possible to differentiate it from malicious traffic.

Traffic volume anomalies can also be induced by events like routing changes due to faulty or misconfigured equipment. Normally, they can be spotted by comparing multiple instances of AS-AS traffic matrixes obtained at different times.

#### **2.2.1.4 Evolution / wish list**

Some NRENs just wish to evolve and consolidate their (possibly threshold based) anomaly detection methods, possibly refining them or implementing them within tools providing also visualisation features (like Nfsen, Stager, flowscan/JKFlow) and abandoning (or porting into these tools) *ad hoc* scripts.

Some others, already having more consolidated detection procedures in place, wish to move to more sophisticated anomaly detection methods reducing the need of manual intervention (i.e. merging the detection and investigation stage). One approach considered with interest is the PCA (Principal Component Analysis), which promises not only to reduce the false positives and false negatives ratio, but also provide a more automated anomaly classification. A good example of such an approach is published in the recent work of a research group based at the Department of Computer Science of Boston University, Boston, MA, USA<sup>7</sup>. Their work focuses on mathematical analysis of flow data in various dimensions (time, address range, port range), thereby yielding results that otherwise might pass unnoticed. We anticipate on closer collaboration with this group.

Obtaining an AS-AS traffic matrix and visualizing it is a feature not widely available in all the NetFlow analysis tools. Enhancements of the tools in this aspect are highly desired.

---

<sup>7</sup> <http://cs-people.bu.edu/anukool/pubs/sigc05-mining-anomalies.pdf>

## 2.2.2 Investigation of Anomalies

### 2.2.2.1 Description

This phase comes as soon as the detection phase has evidenced some anomaly. It has the purpose of finding more information about the anomaly, functional to:

- decide on the nature of the anomaly;
- decide, with an acceptable degree of confidence, that this was actually an anomaly (and not a normal, statistical fluctuation of the traffic volume or nature);
- collect more information about the anomaly, possibly correlating it with the analysis of other anomalies, possibly even pointed to by indicators obtained from entirely different sources (i.e. sink hole reports<sup>8</sup>, etc.) This information is normally useful in the following phase (reaction to anomalies), if present;
- rank the anomaly, with respect to other ones, according to some severity level.

### 2.2.2.2 Importance

This functionality is very relevant for almost all the NRENs having given feedback.

### 2.2.2.3 Current practice

Depending on the technique used during the anomaly detection phase, some metadata can already be available for classifying and further investigating the anomaly. For example, a threshold base method detecting an increase of short flows towards a web site under attack might already come with a list of the sources (or the most frequent sources) of these flows. Contrast to this, anomaly detection triggered by off line information (e.g. a user complaint) without additional information requires the collection of metadata during this phase itself (e.g. looking back to an archive of raw NetFlow data collected before and after the user spotted the problem).

Some detailed examples of current Investigation of anomalies practices are listed below:

#### **Syn flood attacks - GRNET**

---

<sup>8</sup> A *sink hole report* is a usually automated report on malicious traffic seen from a specific IP address. The reports are usually sent to the CSIRT responsible for that specific address. Sink holes themselves do not solicit any traffic, so traffic seen by sink holes is *by definition* suspicious. The IP address of a sink hole (usually kept secret for obvious reasons) comes from unassigned IP address space, sometimes referred to as *darknets*. An excellent pioneering team working with darknets is *Team Cymru*: <http://www.cymru.com/>

During their analysis they not only focus on the initial TCP SYN packets from the attacker but also try to calculate the corresponding REPLY (SYN/ACK) packets in order to distinguish legitimate from malicious traffic. This requires unsampled NetFlow.

### **P2P Investigation - RENATER**

Each day (the night), flows are parsed by address and the results are sent to CERT RENATER which performs a verification before signalling the anomaly to the identified end user site. They signal all P2P traffic (based on the top ten of each day). However, CERT RENATER's actions are limited to signalling only, they do not claim that illegal traffic was seen, but do indicate that it is *probably* illegal. This can be seen as a piece of advice to the site involved. All connected sites acknowledge in their contracts with RENATER that the traffic originating from them must be for education and research.

#### **2.2.2.4 Evolution / wish list**

Most NRENs wish or are planning or are experimenting methodologies trying to:

- Link the anomaly investigation more closely to the (preceding) detection phase, possibly avoiding or limiting human intervention
- Automate the generation of alerts towards who can react to the anomaly (see later). For example, quoting the CESNET contribution "our intention is to have some sort of alarm emails when a traffic anomaly is detected for a longer period of time. Currently we cannot make accurate assumptions about this duration."
- Correlate alerts, keeping track of IP addresses included in each alert so that the tool would be intelligent enough to mark and treat them differently if necessary when more activity is detected regarding these addresses

## **2.2.3 Response to Anomalies**

### **2.2.3.1 Description**

The response to an anomaly should use all the data, collected during the investigation phase, to produce alerts or automatic equipment configurations intended to remove the cause of the anomaly, and/or to remove its effect, and/or to identify its origin, and/or mitigate the resulting damage.

### **2.2.3.2 Importance**

We did not receive sufficient information about this, one possible reason being that it was not explicitly asked. However, we assume that response to an anomaly must be important for the NRENs which already mentioned

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

detection and investigation as important issues, since the response is the natural and expected next step. Without it, the first two efforts would be largely in vain.

### 2.2.3.3 *Current practice*

Not explicitly asked, not enough information collected. However, this is more an area for WI3.

### 2.2.3.4 *Evolution / wish list*

Not explicitly asked, not enough information collected. However, this is more an area for WI3.

## 2.2.4 **Billing**

### 2.2.4.1 *Description*

NetFlow gives an easy and flexible way to associate the traffic volume sent/received by a certain IP address or pool of addresses in a certain time. The Port information can be also used to associate the traffic with an application (but the reliability of that depends on the type of application). NetFlow can therefore be used as a basis for flexible billing schemes.

### 2.2.4.2 *Importance*

Not so high. Two NRENs (IUCC and SWITCH) state that they are currently using it for billing purposes, and for another one (RedIRIS) accounting is on the wish list.

### 2.2.4.3 *Current practice*

One of the few NRENs having described an existing utilization of NetFlow for billing purposes is IUCC. They use their NetFlow system mainly for billing purposes. They exclude all intra-Israel traffic and only analyze the international traffic and base a specific percentage of their billing on the traffic split of the eight universities who are members of IUCC.

SWITCH has been using NetFlow data for billing since 2000. An internally designed system called "Fluxoscope" is used for this.

### 2.2.4.4 *Evolution / wish list*

RedIRIS puts this on the wish list.

## 2.2.5 Traffic Engineering / Planning

### 2.2.5.1 Description

NetFlow can be used to obtain flexible traffic matrixes among generic couples of “end points”. Traffic matrixes are useful for traffic engineering (short term reaction to routing changes and unbalanced loads in the network) and planning (longer term perspective). Such end points may be (prefixed) IP addresses, but they can also be Autonomous Systems (AS). The AS information is directly obtainable from the NetFlow records, but with the caveat that there are two ways for configuring it in a router:

- end src/dst AS: these are the ASs corresponding to the src and dst IP addresses of the flow
- previous/next AS: these are the ASs corresponding to the previous and next AS relative to the interface where NetFlow record is collected

### 2.2.5.2 Importance

From a security point of view, it is felt that the importance is medium or even low. This of course notwithstanding the importance in the various contexts in which feedback was received. SWITCH currently uses NetFlow data to support traffic engineering and planning. RedIRIS and DANTE mention this as important.

### 2.2.5.3 Current practice

SWITCH uses NetFlow data to support traffic engineering and planning. Their Fluxoscope system visualises aggregated traffic per peer (neighbouring Autonomous System), customer, and per heuristically derived "application protocol".

DANTE noted that the feature for an efficient visualisation of AS-AS matrix is absent in all the benchmarked visualisation tools (flow-tools, nfsen, stager, nerd).

### 2.2.5.4 Evolution / wish list

RedIRIS would like to have an AS/AS traffic matrix, and distinguish IPv6, mcast and MPLS traffic. DANTE would like to have an AS/AS traffic matrix between NRENs. DANTE already collects AS information in the “previous/next AS mode, and the previous/next AS are the bordering NRENs. So, it would be straightforward to obtain such a matrix, but the existing tools do not have a ready-to-use functionality for its (good) visualisation”

## 3 Requirements

Requirements for the Toolset and its proposed architecture come both from general, “sound” principles, as well as from practical, functional desires. Both categories are addressed here.

### 3.1 Generic Requirements

There are some general requirements and constraints for the Toolset to be developed. However, most of the requirements will have a specific profile, sometimes even attached to a particular element of the Toolset. For these requirements (and possibly constraints), we refer to the following sections of this document.

The general requirements include some that are of a non-technical nature (rather organisational or political), which nevertheless have a relevant impact on the design and development stage of the Toolset. These requirements include:

- ***Use of open standards***

Whenever the Toolset is to be deployed in a networking environment, the user must be able to incorporate and/or integrate this Toolset with an existing network infrastructure. Therefore, the Toolset and its comprising elements must use open standards. For the purpose of this criterion, open standards are assumed to be standards that are well described with specifications that are open to the development community, and that are widely adopted in relevant (similar) contexts.

- ***No payload inspection***

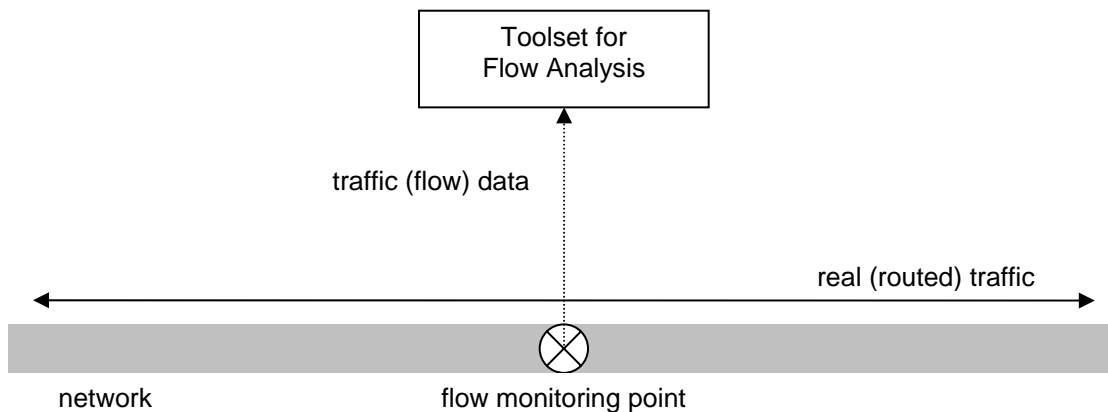
To incorporate tools capable of performing payload inspection in the Toolset may lead to two types of problems. Firstly, this will inevitably cause policy conflicts among the different NRENs regarding privacy issues that are hard or impossible to overcome. Secondly, the technical feasibility (at reasonable costs) of deploying payload inspection tools in the high bandwidth environments such as the GÉANT2 research community has still to be proved. For these reason, tools capable of payload inspection are not considered for the Toolset, at least not at this stage of the project.

- **Readiness to commit to Open Source policies**

When developing specific elements of the Toolset, project partners shall, in a way yet to be decided exactly, adhere to the principles of Open Source. It may be deferred as well as defined later in which way these principles will exactly be implemented, but the fundamental idea of Open Source is considered to be in line with the interests of the GÉANT2 / JRA2 community. The expectation that a wider, global community will emerge from the project partners actively contributing to the development of the Toolset is an important and intrinsic asset of these activities.

### 3.2 Specific Toolset Architecture Requirements

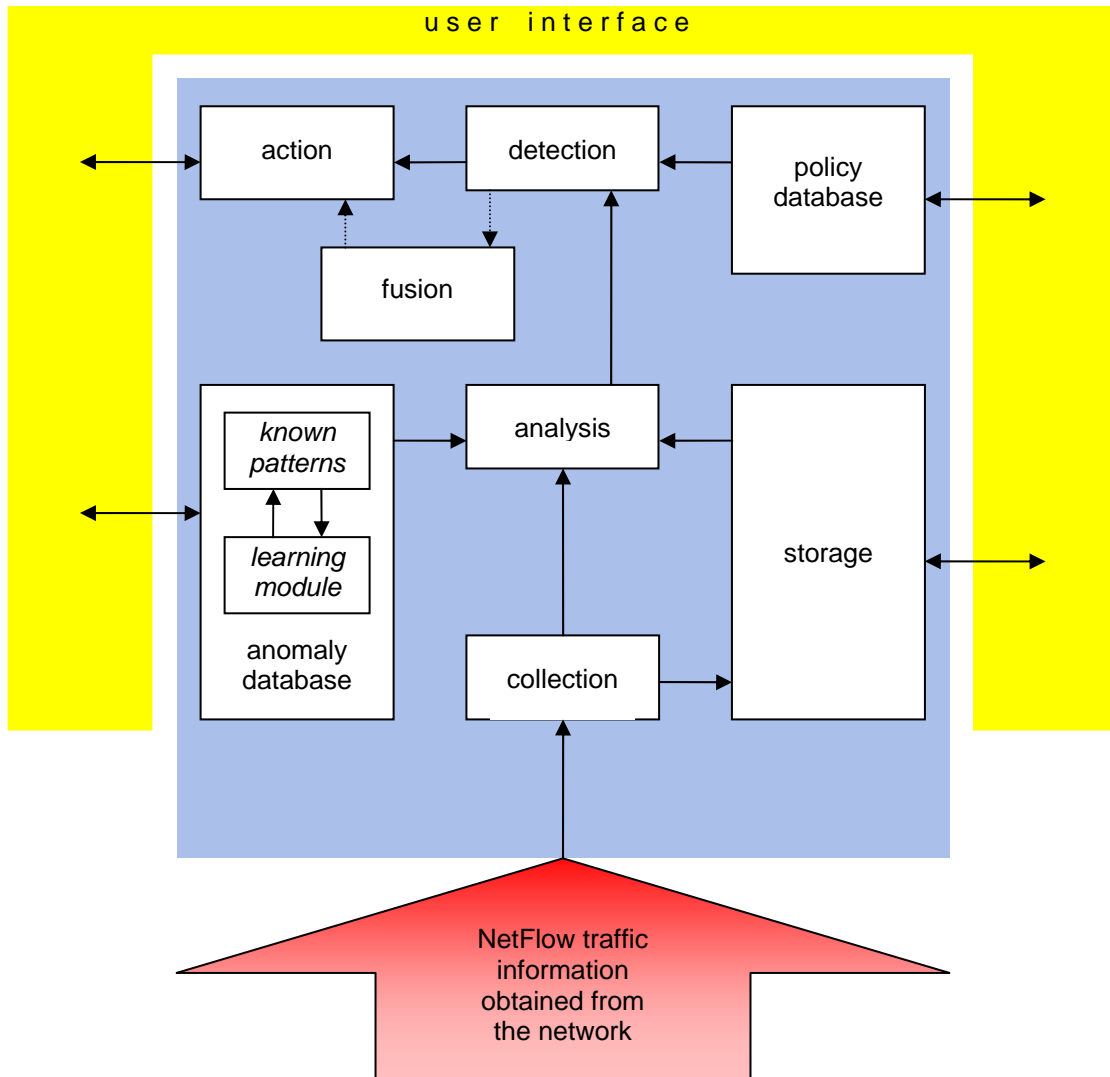
In order to understand the various elements of the Toolset, it is essential to appreciate the overall architecture and the context in which the Toolset is to be developed and deployed. Figure 3.1 depicts the general context.



**Figure 3.1:** General context of the Toolset

In essence, traffic is being transported over a (possibly large scale) network, and information about the process of packets being forwarded (routed) on the network is generated and sent to the Toolset. For pragmatic purposes, the Toolset will accept only information in a well-defined form NetFlow. This is, however, not a principal limitation. The generic functionality of the Toolset itself is sketched in Figure 3.2.

In the remaining sections, all of the modules sketched in Figure 3.2 will be discussed in more detail.



**Figure 3.2:** Overall Toolset architecture

### 3.2.1 Collection

The collection module is where the whole process starts, and is the basis for the operational input to the Toolset. The collection module has the role of extracting per flow information from the packets containing it. Normally, information about each flow is put in a flow record. Several flow records are then (for efficiency) wrapped together in a single packet and sent from the router (or a dedicated probe) to the collection module. Basic requirements for this module are speed and accuracy. The module itself does not need any additional "intelligence", since all possible functions of the Toolset that need such knowledge will be delegated to other modules further down the process. The information that is being gathered by the collection module should be traffic flow information, minimally including:

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

- source IP address
- destination IP address
- source port number
- destination port number
- protocol type
- time stamp (first seen)
- time stamp (last seen)
- number of flows in time frame
- number of packets seen
- number of bytes seen
- sampling frequency

The aforementioned set of parameters is sufficient to support the complete functionality of the Toolset in terms of deployment.

### 3.2.2 Storage

The storage module should be designed to cope with the rates of input delivered to it by the collection module, and should feature an underlying storage method that will enable the Toolset to perform the following functions:

- full backup of data for disaster recovery
- exchange of data with other Toolsets (peer-to-peer)
- fast search options for data mining purposes by the analysis model

Obviously, a database format that is compliant with other open database architectures should be used. It can be debated what the level of refinement of Toolset data should be in order to be “valuable” enough to warrant a strong full disaster recovery capacity. It is pointed out that raw data in itself may be of only temporary value; and from the technical as well as the functional perspective it might be just not worth the trouble to recover from (raw) data loss.

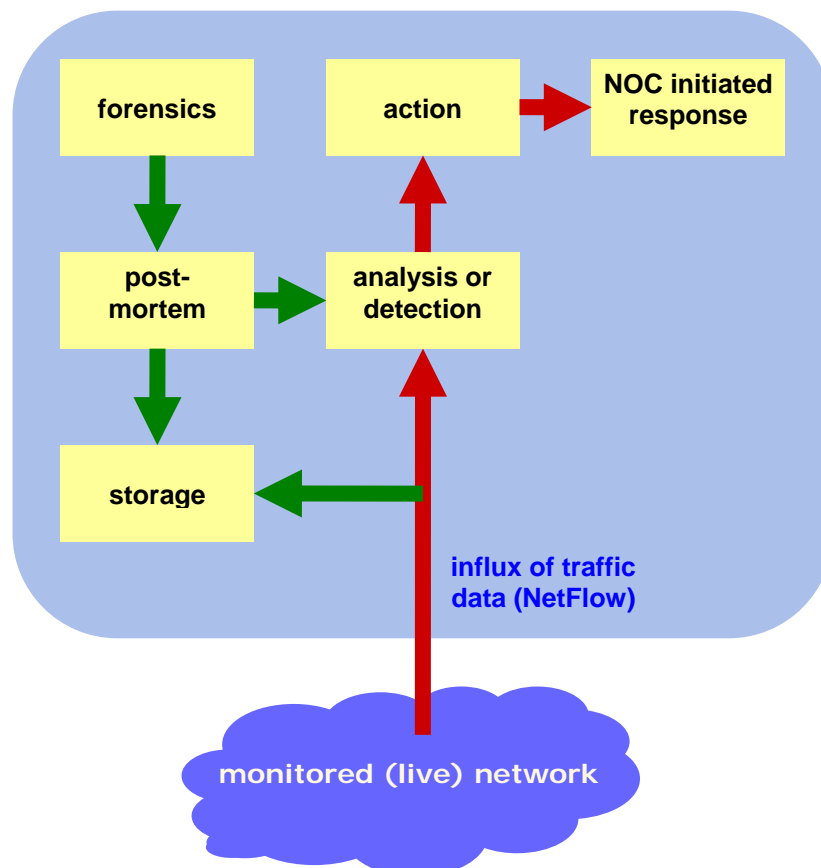
### 3.2.3 Analysis

The analysis module is the core of the Toolset. It performs the actual work of feature extraction, and correlating all the incoming flow data (delivered to it by the collection module), using the information in the anomaly database, to determine what type of traffic is being transported over the network. In addition to that, the same functionality should be applied to flow data that refers to traffic in the (relatively far) past, in which case the relevant flow data are retrieved from the storage module. This functionality is essential to give the Toolset a relevant role in situations where forensic analysis on the network is required. Early developments of tools capable of doing exactly this type of functionality have proven to be very useful in determining malicious traffic in the past, thereby providing valuable input for the anomaly database (see section 3.2.4 below). The results of the feature extraction layer should be stored in RRD format, which is an open database format and supports time alignment for the data that were measured by the various information sources. The feature extraction

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

module should be independent from the detection module in terms of the implementation details in order to allow for a diverse set of detection functionalities (i.e. one module per different functionality).

There is an interesting observation to be made with respect to the relationship between the storage module and the analysis/detection module. This relationship, not explicitly incorporated in Figure 3.2, is depicted in Figure 3.3 below.



**Figure 3.3:** Real-time monitoring process flow (red) versus forensic post-mortem analysis process flow (green)

In the functional sense, NetFlow data obtained from a live network is a key source for two closely related processes but very different in their flow:

- **Near real time analysis:** In the (near to) real time response, a Toolset user may want to have a flow activated from detecting an anomaly to shutting down parts of a network (functionally by a NOC or similar). This may be implemented to happen in a matter of hours or even minutes. This is indicated by the red flow.

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

- **Post-mortem analysis:** If a serious security incident is to be investigated a relatively long time after it happened (typically the case with forensics), post-mortem analysis needs to be applied possibly days or weeks after the detection of the anomaly: the green flow in the diagram.

The storage module (in terms of accessibility and scalability) as well as the analysis module (probably applying the same set of rules) should both be fit for both process flows. Note that in Figure 3.3, the various blocks correspond to operational processes rather than Toolset modules. For the storage, analysis/detection and action processes, these do, of course, have a close relationship with their counterpart modules in the Toolset architecture.

### 3.2.4 Anomaly Database

The database of anomalies is the "back office" for the analysis module. It should contain a set of patterns to which flow data can be matched, alongside a description of the type of anomaly at hand. This set (referred to as the *known patterns* in Figure 3.2) should be easily adjustable, both based on (new) insights from an external context, as well as from an additional sub-module (*learning module*) that may feed newly discovered anomalies into the known patterns. The analysis module should be able to regularly "refresh" its understanding of anomalies by reloading the set of known patterns from the anomaly database. Therefore, the anomaly database and the analysis module should agree on the form of the data to be exchanged, which should be done on the basis of the feature extracted (correlated data).

Furthermore, the format of the data in the anomaly database should be chosen in such a way that data from this database can be exported. It will be essential for the viability of the Toolset to be able to exchange anomaly data (hence, exchanging "ideas" as to anomaly patterns) with "peer tools" that might use different technologies but similar approaches.

### 3.2.5 Detection

The detection module is an independent unit (implementation-wise) of the modular architecture. This should be done in order to employ different detection functionalities in parallel or a combination of them, driven by the parallel data feed (in our case RRD<sup>9</sup> data) and policy database rules. The process of retrieving and analysing flow data, thus far, is an independent (context free) process. The detection module has the duty of linking it to the context of the environment in which the Toolset is to be deployed. In other words, the detection module will link the facts found on the network itself to the policies, which are to be applied on behalf of the network operator.

It will therefore compare the actual flow metrics computed by the detection modules reported to it by the analysis module to what the network operator has defined as acceptable, unacceptable, etc., in the policy

---

<sup>9</sup> RRD is the acronym for Round Robin Database. It is a system to store and display time-series data, for example traffic flows detected on a network. For generic information on RDD, see <http://oss.oetiker.ch/rrdtool/>.

database as described in section 3.2.6. After matching and marking the relevant data, the detection module passes this on to the data fusion module, specified in section 3.2.7.

### 3.2.6 Policy Database

The policy database is the module in which the context relevant information is stored on the basis of which the detection module can flag specific traffic with assessment information. The policy module reflects the operator's understanding of what is allowed on their network and what is not allowed. However, the policy database should be able to distinguish multiple layers of assessment information, allowing the network operator to "fine-tune" the behaviour of the Toolset. One can think of various levels of "seriousness" that can be ascribed to different types of anomalies.

### 3.2.7 Data Fusion Module

Based on the results of the various detection modules, a data fusion system is used to build consensus on the existence of a network anomaly. Simple data fusion algorithms like voting or 1st order logic expressions or more sophisticated algorithms like the Dempster – Shafer (D-S) method (see [D-S]), based on the theory of evidence or Artificial Neural Network (ANN) fusion may be considered and/or tested. The results (decisions) are matched against the policy database and a proper activation is ignited.

### 3.2.8 Action Module

The action module is responsible for performing the required action upon detection of a specific anomaly. An action in this context might include all sorts of steps (or series thereof), such as:

- doing nothing
- logging
- alerting by e-mail
- alerting in a real-time interface
- alerting using an out-of-band method such as text messaging
- operational functions on the network without human intervention

It goes without saying that the choice from any of the aforementioned actions is to the discretion of the entity that is responsible for the operation of the network where the anomaly occurs. Put in other words, a specific anomaly may be reason to block a specific IP address range in one environment, while the same or similar anomaly detected somewhere else may only lead to mere logging of the event. Therefore, the Toolset architecture should facilitate various actions in a generic sense, leaving the actual choice of response to the network operator.

### 3.2.9 User Interface

The Toolset is not designed to be used by "ordinary" end users. The aim for its deployment resides with Computer Security Incident Response Teams (CSIRTs<sup>10</sup>) and/or Network Operation Centres (NOCs). In the usual setting, CSIRTs monitor the network of their parent organisation, using tools to detect anomalies such as denial-of-service attacks, virus outbreaks, or other types of unwanted traffic. Upon detection of an unacceptable traffic pattern (in the live context), or possibly upon receiving a specific external complaint (in the forensic context), a CSIRT may want to perform specific defensive actions on the network. In some cases, the CSIRT can do this independently, in other cases, the CSIRT needs the NOC to perform such an operational action on the CSIRT's behalf. In any case, the Toolset should enable the user (i.e. a CSIRT team member) to easily use the Toolset as a valuable addition to their capabilities to effectively monitor the network and initiate defensive (counter) measures whenever possible. The interface to this particular (demanding) user should reflect this understanding.

Additionally, the Toolset itself resides on context-dependent information, such as the anomaly database (see section 3.4) and policy database (see section 3.6). These databases will be supplied with information and maintained by various types of users, with different specific characteristics. In this case, too, the user interface should resemble the way in which the Toolset is perceived by such user categories.

---

<sup>10</sup> Computer Security Incident Response Teams (CSIRTs) perform the overall operational security surveillance of a network. The internationally accepted acronym is CSIRT, although, for historical reasons, most teams bear the acronym Computer Emergency Response Team (CERT) in their name. CERT™ is a registered trade mark by the CERT Coordination Center of Carnegie Mellon University in Pittsburgh, PA, USA. For the relevance of this document, CSIRTs and CERTs are the same.

## 4 Conclusion

So far, tools that are available (either or not developed as part of this project) have been deployed, possibly in a pilot context, and tested. Specifically, [MM1] and [DJ2.2.1,1] will give overviews of what has been tested, and with what results. We refer to these papers for specific information on the (test) deployment of various tools in this area. In general, however, we could observe that some of the functions of the Toolset are already developed, sometimes partly, sometimes extensively, and sometimes overlapping. The coherence among the various modules (linking and interfacing them together), however, is far from complete, and should be improved. At the same time, the user interface in the generic, overall sense, does not exist yet. This should also be further developed.

Most importantly, more work should be done to the learning module, to make it capable of actually determining new types of anomalies based on what has been monitored from the network. Other enhancements to the Toolset include a wider range of analysis possibilities to the patterns seen, e.g. second or even third degree of derived information (i.e. to be able to dynamically determine anomalies, not only based on a single pattern, but in a series of patterns and their respective change in any of the relevant dimensions (time, address space, port range)).

## 5 References

- [D-S] Dempster – Shafer Theory: [http://en.wikipedia.org/wiki/Dempster-Shafer\\_theory](http://en.wikipedia.org/wiki/Dempster-Shafer_theory)  
[DDoSVax] <http://www.tik.ee.ethz.ch/~ddosvax/>

## 6 Acronyms

<b>AS</b>	Autonomous System
<b>CERT</b>	Computer Emergency Response Team
<b>CSIRT</b>	Computer Security Incident Response Team
<b>(D)DoS</b>	(Distributed) Denial of Service
<b>FTP</b>	File Transfer Protocol
<b>MPLS</b>	Multi-Protocol Label Switching
<b>NOC</b>	Network Operations Centre
<b>P2P</b>	Peer To Peer
<b>RRD</b>	Round Robin Database
<b>RTIR</b>	Request Tracker – Incident Response

## Appendix A NREN Feedback on Toolset Aspects

During Y1, feedback was collected from participating NRENs in order to further define and shape the outline of the Toolset, including its (desired) functionality. For completeness purposes, an overview of the received feedback is given in this Appendix.

### A.1 Common NetFlow Analysis Practice

NetFlow analysis for security purposes is already used by security teams all over the world, including those of some of the NRENs participating to the GN2 project. Here we summarise the “common practice” about NetFlow data analysis for security, as reported by some NRENs. Looking at the state of the art of NetFlow Data analysis for security (and its current limitations) is the starting point for putting together a set of requirements for the Toolset which is really functional to the goals of JRA2.

#### A.1.1 RENATER

##### **DoS alarm:**

They had not implemented a true algorithm in their old collector, and it is still the case in the new one. Only alarm based on a threshold is defined. The threshold is configured according to the site flow average during one or five minutes.

##### **FTP warez and P2P traffic detection:**

Their detection is based on characteristics like the utilised port (e.g. P2P) and/or the size (e.g. FTP warez) of the flow:

- port-based: the commonly used P2P ports (4662, 6881 to 6889, etc.)
- size-based: all flows with a size between 14MB and 16MB, and between 44MB and 55MB are logged.

At RENATER, they use these values based on the observation of how people download such files: often a video (DVD or DIVX) is divided into several files of the same size (15MB, hence, they keep activate sampling NetFlow values between 14 and 16MB; 45MB is for playstation games etc.).

## A.1.2 RedIRIS

Until recently, RedIRIS had used CISCO NetFlow Collector and Analyser. It was installed and managed by the RedIRIS NOC team. Because of the tool's limitations, IRIS-CERT just used NetFlow information for incident verification and/or incident post analysis. This was done using a selection of home made scripts (perl) that went through the flow records in raw files looking for activity related to one or several IP addresses in a given time frame.

After evaluating different tools in a test environment, Nfdump/Nfsen and flowtools/flowsan/JKFlow were selected to be deployed (in operation since May 2006). Nfdump/Nfsen is used for security purposes and is operated by the CERT, while flowtools/flowsan/JKFlow are operated and used by the NOC. The reason to make this distinction is that RedIRIS could not find a set of tools able to fit both requirements.

From a security point of view, NetFlow is used both to proactively detect anomalies and attacks in the network (releasing alerts from them) as well as for network forensics and post-mortem analysis.

The proactive detection implemented nowadays includes port scan detection (machines within our constituency scanning the network) and P2P traffic detection (the latter one based on activity related to the more common P2P ports), as well as DoS threshold based detection. Detection of specific attack patterns can also be added on the fly easily, thanks to the backend plugin facility implemented in the selected Tool. The detection plugins are in a phase of continuous improvement, and more efforts are in place to add more intelligence to the Tool by detecting different kinds of attacks (detection of e.g. botnet zombies or specific malware).

Nfdump information is also being used to get daily reports by different types (src/dst IP, src/dst ASs and /24 address aggregation). This project is currently in a testing stage within RedIRIS.

### Future

From the security point of view (as articulated by their CSIRT), adding more intelligence to Nfsen for detecting more kinds of attack patterns and anomalies is more than desirable. Improving the alarm system is another goal, automating alerts as much as possible, integrating these with the Incident Response Tool used by IRIS-CERT (RTIR).

Furthermore, RedIRIS has expressed its interest in a mechanism to correlate alerts keeping track of IP addresses included in each alert so that the tool would be intelligent enough to mark and treat them differently if necessary where more activity is detected regarding these IP addresses. Another issue RedIRIS would like to work on involves minimising the alerts, i.e. adding to the Toolset the intelligence needed to keep track of the events and just send an alert when a configurable threshold is reached.

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

From the RedIRIS NOC point of view, and planned for deployment, the application is to be configured for identifying IPv6, mcast and MPLS traffic. Traffic would also be sorted by origin and peer AS. The stored data would be used for accounting and detecting routing anomalies.

### A.1.3 GRNET

GRNET aims at finding some relationship by analysing sampled and unsampled NetFlow data and is now working on a specific NetFlow data set recently obtained from the network. Java and perl are used for data manipulation and analysis. So far, these efforts have not yielded fruitful results yet.

The analysis is not only focused on the initial TCP SYN packets from the attacker but also tries to calculate the corresponding REPLY (SYN/ACK) packets in order to distinguish legitimate from malicious traffic.

### A.1.4 CESNET

At CESNET, there are two main areas where NetFlow can be (and partially is) effectively used for primary security purposes (omitting other areas like security devices tuning - firewalls, targeted filtering setup & verification).

#### 1. Incident verification, incident post-analysis

This is deemed important by CESNET, especially from the NREN backbone perspective. Strict source based routing/forwarding policies are usually not (or cannot be in some cases) widely configured in connected networks (or at the backbone border).

Actions to verify or (post-mortem) analyse incidents include:

- Investigating whether the reported identification of the attacker (usually host-names, IP addresses) corresponds with real traffic coming from network(s) which are using such identification legally.
- After first detecting specific kinds of attacks (DDoS), providing large scale analysis to stop its distribution as quickly as possible. This usually means (step by step) to choose some significant flow record values (e.g. attack related - protocol & port numbers) and provide either interactive queries or configure a filter to find other (possibly infected) sources.
- Trying to analyse some sophisticated attacks very deeply to find out their real origin.

In order to facilitate the aforementioned (or similar) actions, CESNET feels that a full NetFlow record browser would be needed - CESNET are currently using their FTAS system for this

#### 2. Pro-active attacks detection

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

According to CESNET, considering the usual NetFlow configuration (time parameters, sampling) it only makes sense for a limited set of attacks. Basic methods are well known (sampled, on-fly aggregation by source, flow count based ordering) and at CESNET they will probably implement some targeted attacks-detection functions into the FTAS system in the future - currently they already use the side effect of its sampling and aggregation features. There are many related important parameters like real traffic structure (single purpose lines vs. common backbone, strict vs. liberal administration point of views), traffic amount (packet rates), flow generation parameters (sampling, time outs), computing power, etc...

In general CESNET is optimistic on new possibilities and more flexibility in the pro-active detection area in relation to next steps of their hardware NetFlow probe development.

### A.1.5 IUCC

IUCC collects NetFlow from two core routers with no sampling (Cisco 7600s) and creates all the standard NetFlow reports: top 100 senders, top 100 receivers (both by individual IP and by subnet), AS matrix; and all reports are available for last hour, last day, last week, last month and last year.

The NetFlow system is mainly used for billing purposes. It therefore excludes all intra-Israel traffic and only analyzes the international traffic.

For security purposes IUCC creates a top 50 ports used report which is occasionally reviewed. But the most important tool for security purposes is an automated email add-on that analyses the traffic on any boundary as defined by the netadmin (could be a /32 or a /24 or any length prefix).

To illustrate this with a specific example, the following IUCC-specific scenario is sketched. A netadmin defines the prefix, length, and the maximum amount of traffic that should be seen over a 1 hr period and over 1 day. For example, a netadmin could state that for their 128.139.0.0/16 - the hi-mark is 100GB over 1 hr and 500GB/day, and the server farm located at 128.139.10.0/24 should have a limit of 10GB over 1 hr and 30GB/day and the proxy server located at 128.139.10.5/32 should have a limit of 2GB/hr and 5GB/day. If any limit is exceeded an email is sent to the list of users as defined by the netadmin. Since some servers need to be excluded, one can define a prefix and length to be excluded. Example: if 128.139.0.0/16 has a limit of 100GB/hr but there is one segment like 128.139.128.0/22 that will generate lots of traffic that should be excluded - that can be done. The trick here is to set initial rules and to tune it over a period of a few days/weeks until reaching a point where the alerts are only true alerts.

With this tool IUCC is able to quickly spot botted hosts that start spewing out lots of data or hosts that have been taken as warez servers.

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

## A.1.6 SWITCH

SWITCH collects NetFlow data from all border routers (currently 4). They use the tools nfsen and nfdump for collection, analysis and visualisation of NetFlow data. These tools are used to validate incident reports and to support sys-admins in their investigations into hacked systems.

Plugins to nfsen and nfdump are used to accomplish some more complex tasks:

- Systems within SWITCH's constituency scanning the network are detected by searching for typical patterns in NetFlow data
- Detection of single large flows helps pinpoint systems engaged in (D)DoS activity
- Correlating information about known command and control systems (botmasters) of botnets with NetFlow data allows SWITCH to find systems in their constituency controlled by those botmasters

SWITCH is planning to develop additional plugins to nfsen and nfdump to add advanced anomaly detection capabilities. They envisage the implementation of promising recent research results in this area, e.g. the work of the DDoSVax team of ETH Zurich, Switzerland (see [DDoSVax]).

## A.1.7 ISTF

The main tool in use at ISTF for network monitoring has traditionally been NTOP. For a long time it was used as a NetFlow collector/analyzer; later – as packet “sniffer”. Currently, a return back to NetFlow is envisioned, and different tools like nfsen and nerd are evaluated with the intent to replace NTOP (though not necessarily completely).

Such changes have been prompted by various shifts in the requirements imposed upon the network monitoring process. Those, in turn, reflect the constant development of the network – its topology, available external and internal bandwidths, typical traffic patterns, user base, etc. Packet capture, for instance, was introduced as an answer to the mass spread of P2P networks, which at some point tended to saturate the whole infrastructure with unacceptable traffic. The ability to identify even the exact names/hashes of the exchanged files turned out to be very handy in some tougher cases.

NTOP is a nice tool, even though not very well suited for larger environments (ISTF is still able to use it at the backbone, but it is already becoming overloaded). It is also more generally oriented, making it less-than-ideal for security purposes. Still, the statistics for network/transport/application layer protocol distribution and activity over time, the TCP/UDP port usage, and especially the detailed packet information per host, indicating various suspicious patterns: exceedingly high number of host “contacts”, various fragmentation issues, SYN-FIN packets, other unusual combinations of flags, etc., are of much use. Unfortunately, P2P hosts do not yield that much information anymore, and are often misidentified.

Project:	GN2
Deliverable Number:	DJ2.2.3
Date of Issue:	18/09/06
EC Contract No.:	511082
Document Code:	GN2-06-245v4

Although NTOP does a good deal of analysis, many of the actions in the context of the current document are still processed manually. This means that for a certain anomaly in the traffic (or rather – misbehavior of certain user) a human operator should first notice it (i.e. “generate an alarm”), then perform a detailed inspection and further analysis, and eventually, upon confirming the incident, take appropriate actions. This system is obviously not very efficient, is time consuming, and also prone to failures. Indeed, it works in practice, but at least one of the reasons is that there are simply not many real security incidents happening in our network so far. This, undoubtedly, is going to change though, which calls for new tools and practices to be employed, as was mentioned in the beginning.

Being “human-based”, the current analysis practice itself does not conform to some hard-coded rules. Usually, the general traffic volumes are followed, together with the distribution by types. If any perceivable anomalies are found (e.g. sudden peak in the total volume, or unusually large amount of traffic on a port known for recent security vulnerability), effort is made to identify the affected hosts, and to determine whether there are indeed security issues involved. Sometimes, a full inspection of the captured anomalous traffic is made. The “usual suspects” – hosts generating suspiciously high volumes of traffic, known “villains” from previous cases, etc. – are also kept under constant surveillance. Special attention is paid of course to the sensitive points: core network equipment, critical servers, etc. Once again – there are no strict rules: most of the steps rely on the technical expertise, experience, and even the “sharp-eye” of the human operator.