NetSEC: metrology-based application for network security

Jean-François Scariot - Bernard Martinet

Centre Interuniversitaire de Calcul de Grenoble.
Domaine Universitaire
BP 53, 38041 Grenoble cedex 9
{Jean-Francois.Scariot, Bernard.Martinet}@grenet.fr

Abstract:
We know how to use metrology to discover network functional problems and how to measure error rate to detect active or passive abnormal behaviour items. Even, we can use it to observe the network use and to detect abnormal resources consumption. This leads us to develop a flow analyses application to secure networks. Build around a relational data base system, filled by metrology platform and supporting multi-criteria sort queries, NetSEC allows analyzing unusual traffic in order to find its roots. A user friendly web interface allows using NetSEC application close to final users.

Keywords: metrology, security, relational database, data mining

1 Network Metrology

Since a long time, the only solution for network problems was to increase bandwidth capacity to have a bigger pipe and faster routers to improve traffic flow. In the middle of 90's, metrology has contributed to change this behaviour, putting new focuses on network management. Today, metrology can provide new improvement, particularly about network security. This section is an introduction to computer network metrology. It describes the measurements, the qualitative and quantitative parts, the analyses and the hierarchical organization problems.

1.1 Measurements: Why, What and How?

Metrology is, and can be, used for several reasons:

- To know network usage: “who does what, when and how?”
- To know network availability. Is there any traffic congestion? Are servers and consequently services correctly distributed on the network?
- To detect dysfunction. Is there an abnormal rate of collision, frame error, broadcasts, etc.?
- To do cost sharing. To issue accounting in proportion to network usage, a detailed measurement of traffic is necessary.

In addition, metrology can also be used to detect malicious actions.
We distinguish two measurement types: qualitative and quantitative measures.

1.2 Qualitative parts

We use the word qualitative because we collect device performance indicator values, which provide information about traffic quality, either in absolute terms or in relation to network physical characteristics like available bandwidth.

The aim of these measures is to know how data-flow takes place. The objective is to understand and manage traffic, to ensure quality and to forecast network evolution. I/O rate and volume, memory or CPU load for various devices are typical measures. In this approach, we collect also physical fault indicators like collision rates, framing or alignment errors, packet discards, broadcast rates…
Measures are done by reading counter values collected by routing equipments (SNMP queries, MRTG graph…). This gives, on the fly, a snapshot of the overall of network status and allows, by graph use, to monitor its development (improvement, deterioration, fever spike…).

1.3 Quantitative parts

In that context, the goal is to detail and to characterize the traffic. For that purpose, the traffic is split up according to data types, protocols (ICMP, TCP, and UDP…), source or destination hosts, etc. I/O volumetric values are detailed by host, group, service, etc. Traffic evolution is tracked according previous parameters. All these values can be obtained by extracting relevant information from frame. It is more complicated than collecting a simple counter value and it generally requires the use of dedicated and specialized equipment. Results are used to know and manage the network, e.g.: which services are important? Where servers may be placed to avoid flow congestion? Which network structure modifications may improve quality of service?

1.4 Measurement to watch

Another feature of metrology, may be less developed, is using the huge amount of data provided by metrology platforms to watch on the network in order to take care of its integrity. The goal is not indeed to spy but to ensure network availability and quality of services. So, it is necessary to oversee and control information flow. Any abnormal situation is expected to be discovered through data extraction and analysis. At this point it should be stated that an anomaly is not necessary a trouble! For example, one-Megabit traffic increase during the night on a 100 Megabits link does not produce a disruption, but it may be the sign of a particular service or host attack.
Metrology-based supervision is built on the following principles:

- Supervision must be done **daily**, even shortly; sometimes a fifteen minutes cycle is needed.
- It must be done using **curves** or **histograms**.

The idea is using graphical format that human can easily understand and remember. Diagrams and values can be, of course, explicit but it is very difficult to memorize series of numbers. The appearance of a curve, bell-shaped, with morning connection peak, lunch break, etc., is easy to remember. We can summarize as follow:

"To control and manage a network, you must visualize its behaviour »

Figure 1 and figure 2 illustrate MRTG [1] graph supervision.

1 – A normal day

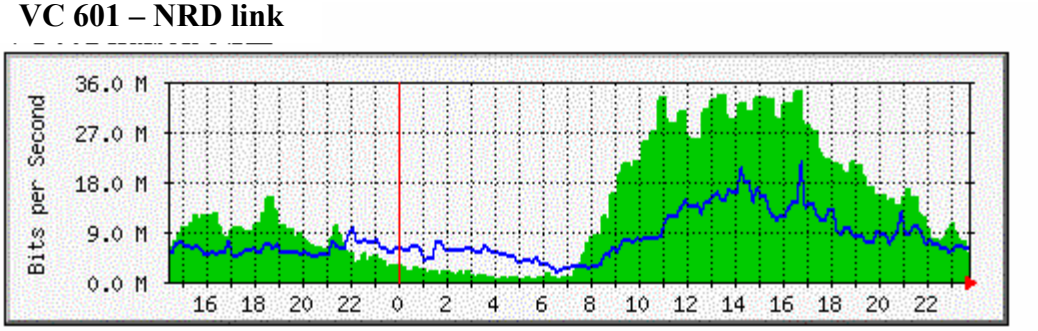


Figure 1 - NRD link - Monday April the 2nd 2001

2 – A day with potential problems.

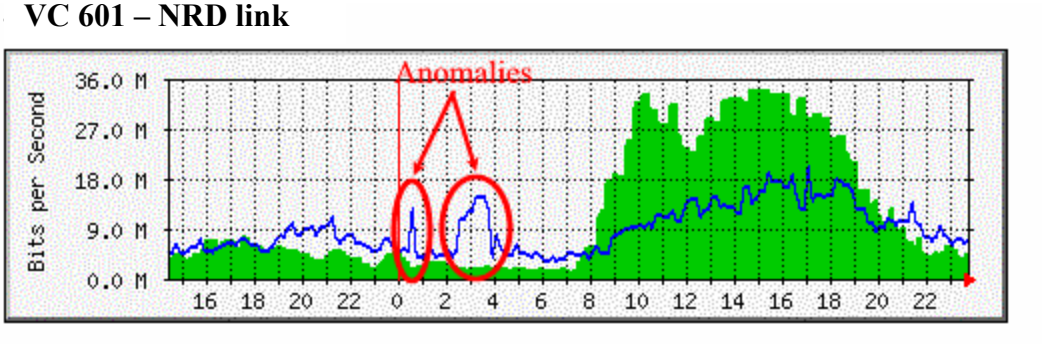


Figure 2 - NRD link - Monday April the 9th 2001

On the second graph, we note two abnormal peaks. First, around 00:30 am and second more obvious between 02:30 am and 04:00 am. Unfortunately, at this stage, we are helpless and we can only take note of anomalies. We have to go back and analyze the traffic of the involved period.

1.5 Analyze

At this point, analyze means:

- To identify the host(s) involved as source and/or destination of the abnormal traffic.
- To determine the precise date and time.
- To go through the logs, if centralized logs exist.
- To determine the attack or intrusion method, when it's relevant.
- To evaluate the extent of the damage (how many hosts are concerned…).

To do that, one can use metrology data and sort them using several criterions. The amount of data to process might be very important; therefore one must have powerful sort tools to quickly obtain a result.

1.6 Organize into a hierarchy

Another problem must also be tackled: how can we know that traffic on a host is abnormal? Within a small organization, the network manager can know the answer. But how can we envisage, on a multi establishment campus, to control the normality of all hosts traffics? With powerful metrology tool like NetMET [2], one can easily know the "talkative hosts top-ten" on a network. For sure, one will find establishment web sites or ftp servers. One can easily keep in mind such a list, but how can discover unusual traffic for hosts between the 30th and the 50th place?

Moreover, several megabytes traffic may be non significant for an ftp server, and abnormal for another host of the same site. Only a detailed knowledge of a site allows bringing to the fore abnormal behaviour on a graph and discovering it with only one quick look.

2 NetSEC

This section presents the NetSEC platform, its goals, its implementation and the actual state of development. The previous section explained why we need a software platform to help analyzing existing metrology data. The tools must support multi-criteria and powerful sort queries. These queries have to be done in a declarative way. This leads to a relational database system which provides such a possibility with the SQL language. It is easier to write a SQL query than to modify a classical language or script program to make complex sorts. Another advantage of relational database systems is their ability for data manipulation and management.

2.1 NetSEC goals and foundations

There are three main goals:

- To have platform, scalable and adaptable to different sites. The tools must be usable for network ranging from small campus network to metropolitan network.
- To analyze data provided by metrology platforms commonly used in university environments, such as NetMET and IPtrafic [3].
- To support open standards to ensure easy distribution to everyone especially to network administrators.

All NetSEC project activities aim at these goals. At the beginning, we have designed the main NetSEC foundations to be as simple and opened as possible.
They are:

- Using a relational database, well adapted to the multi-criteria searches we are doing.
- Using SQL language (powerful and standard to extract selective data).
- Querying and viewing results with web browsers.
- Having a simple network description fitting any organization.
- Building a set of predefined queries. It is expected that the user community will populate and extend this set**.**
- Having a scalable architecture (like a "Lego®") to allow every contributor to adapt it to the use of its framework software.

2.2 Using open source software

All platform modules are open source software. This choice is made to ensure agreement about the solution foundations and easy distribution even for small entity.
The components are the following:

- The system platform is a Linux RedHat distribution; however use and development are possible without problems on any other Linux distribution.
- The relational database system is MySQL because it is commonly used and very efficient. Using another relational database system, like Oracle, is possible if it provides JDBC/ODBC drivers.
- An Apache web server with Jserv module, which implements Java Servlets API, is used to display results.

On the database side, the rules are the following:

- For standard constraints, database accesses should be Java JDBC or ODBC API.
- A simple relational schema is used because the structure to be modelled is simple and hierarchic.
- SQL ANSI without embedded queries is used to allow basic database systems and to easily write and understand queries.
- A data loader is built per data collector based on the same principles wherever data come from NetMET, IPtrafic or even Traffic Director. This allows having an open approach to the different metrology tools.

2.3 Adaptability to the network

The basic NetSEC adaptability rules allow its use for a metropolitan network as well as for a small entity network. The CPU power and the data storage server capacity are dependent on traffic network and history that we want to have. The web interface simplicity allows NetSEC use by any network administrator.
The description is very simple. Each element of a description is made of four items:

- A network.
- An organism.
- An entity: it's an organism part. Its size depends on the scalability that we want, on the organism, …
- A locality.

To detect problems or intrusions, many searches are possible. For example:

- Outside networks having
  - the biggest network traffic volume (amount of accesses),
  - with an internal organism, entity, host,
  - In a time slot.
- « Top » uses of network ports between outside
  - and an internal organism, entity, specified host,
  - In a time slot.
- An internal organism, entity, or host having
  - the biggest network traffic in term of accesses,
  - with an identified internet host,
  - in a time slot.

The searches check the source and/or destination addresses (grouped by organism entity, locality), the ports, the protocols and the data volume. These searches are done for different time slots. The network traffic is accounted in volume, or in term of amount of accesses to discover network or port scans (a huge number of accesses to hosts or ports using small frames). When the right SQL query to check some problems is not available, it's easy to build it e.g. by modifying an existing one. Once tested and validated, the query is expressed using an html form (similar to the one shown in section 2.5) and then, registered in order to be reused later if needed.

2.4 System architecture

The platform is built around a data base system. To populate it, a specific loader has been developed for each data collector (NetMET and IPtrafic in our case). NetSEC provides services through modules. The database system used is MySQL [4], release 3.23.41. Today we dispose of three specialised modules as shown in figure 3:

- A data query module.
- A graphic generation module.
- A data mining module.

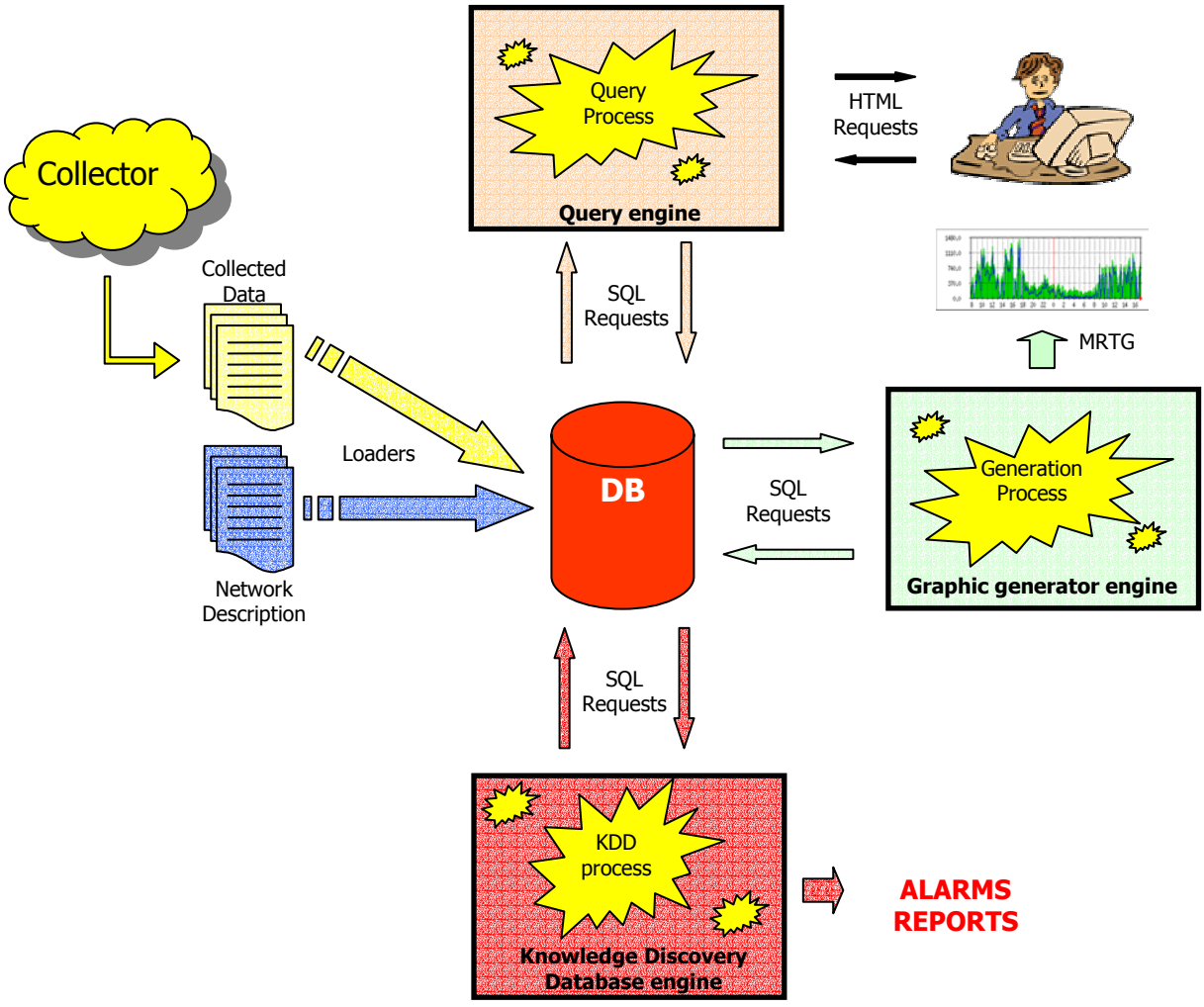


Figure 3 - NetSEC architecture

2.5 Database structure

The data are recorded in daily tables (one table per day). These tables are named by the string “day” followed by the date format “YYMMDD”.

Example:

Tables_in_netmet
Day010608
Day010609
Day010610
Day010611

The tables are created automatically every day at 00h00. The loader inserts the data extracted from NetMET files in the current table every ten minutes.
The table structure is composed of the following attributes:

- The source IP address: four fields each on one byte.
- The destination IP address: four fields each on one byte.
- The timestamp encoded on four bytes,
- The port encoded on two bytes,
- The protocol encoded on one byte,
- The data volume exchanged between the source and the destination encoded on eight bytes.

Example of a daily table:

Field	Type
srcA	tinyint(3) unsigned
srcB	tinyint(3) unsigned
srcC	tinyint(3) unsigned
srcD	tinyint(3) unsigned
dstA	tinyint(3) unsigned
dstB	tinyint(3) unsigned
dstC	tinyint(3) unsigned
dstD	tinyint(3) unsigned
sequence	timestamp(10)
port	smallint(5) unsigned
protocol	tinyint(3) unsigned
volume	bigint(20) unsigned

Every table entry is twenty three bytes long. The timestamp depends on the frequency. A loading every minute will allow a very small timestamp, but will involve breaking in several entries the same data flow when it exceeds this time unit (large ftp transfer for example). On the contrary, increasing time between two loadings decreases these duplications, but event precision is also decreased.

2.6 Data queries

The first module proposed by the NetSEC platform is the data query tool. It provides a web interface which allows choosing a query from a list of HTML forms. The query is parameterized with the information that the user provides using combo boxes and text fields.

Example:
Produce the top 10 host access to the CICG organism on December the 10th 2001.

The information query process is activated when the form is validated. The process is the following:

- DB driver loading and connexion opening
- Start and end date processing
- Entity and/or subnet processing
- Network number resolution
- Services filtering
- Results number processing
- SQL building and executing
- Result displaying

Example:
Figure 4 shows a form asking for the list of the ten machines which having issued the biggest data transfer to the CICG organism, on the http port on February the 17th 2002.

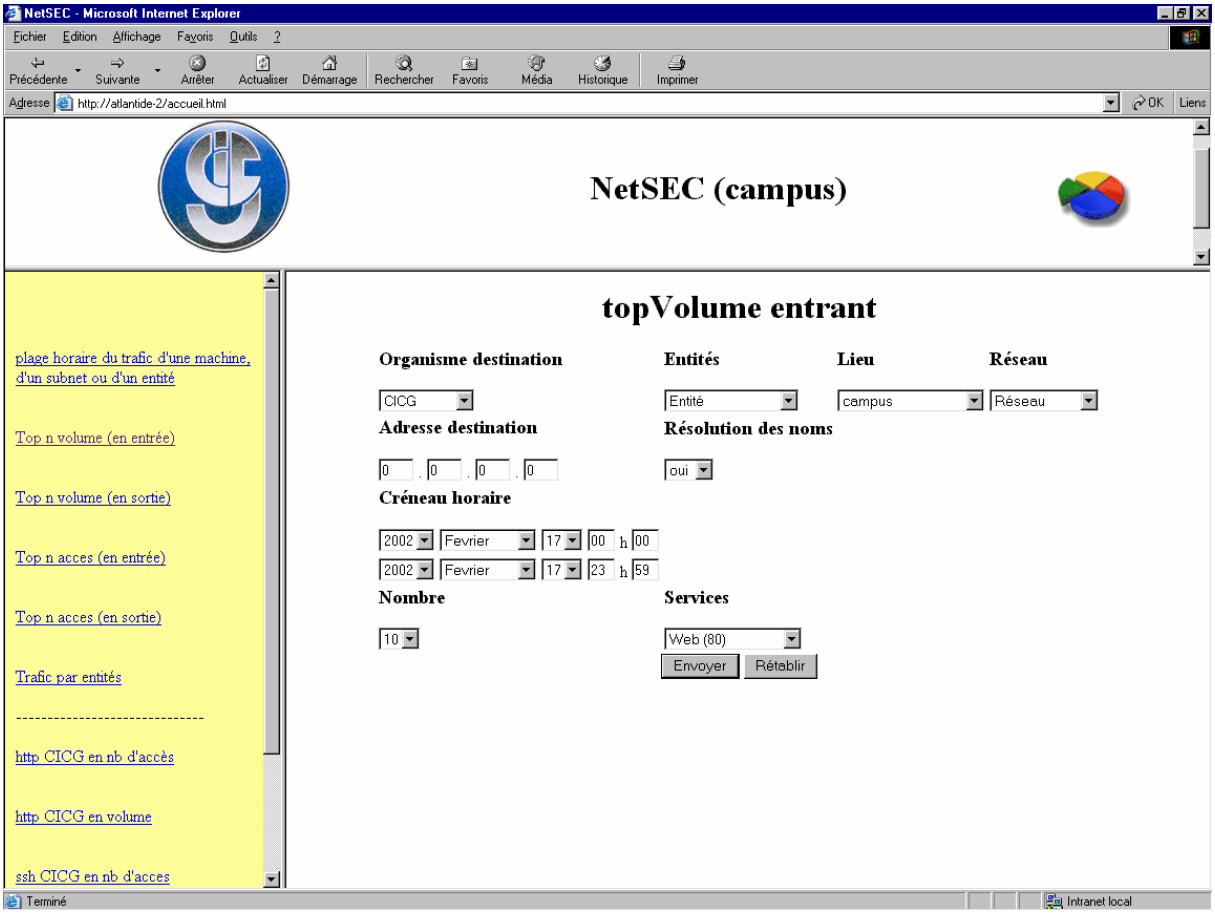


Figure 4 - Web query interface

Figure 5 shows the result obtained from the previous form. The SQL request generated is the following:

SELECT *<destination adresses>*, COUNT (*)
FROM *<February the17th 2002 daily table and network description table>*
WHERE *<organism name ('CICG')>*
AND *<source address is a 'CICG' network address>*
AND *<on port 80>*
AND *<time between '00:00' and '23:59'>*
GROUP BY *< destination addresses >*
ORDER BY 5 desc limit 10

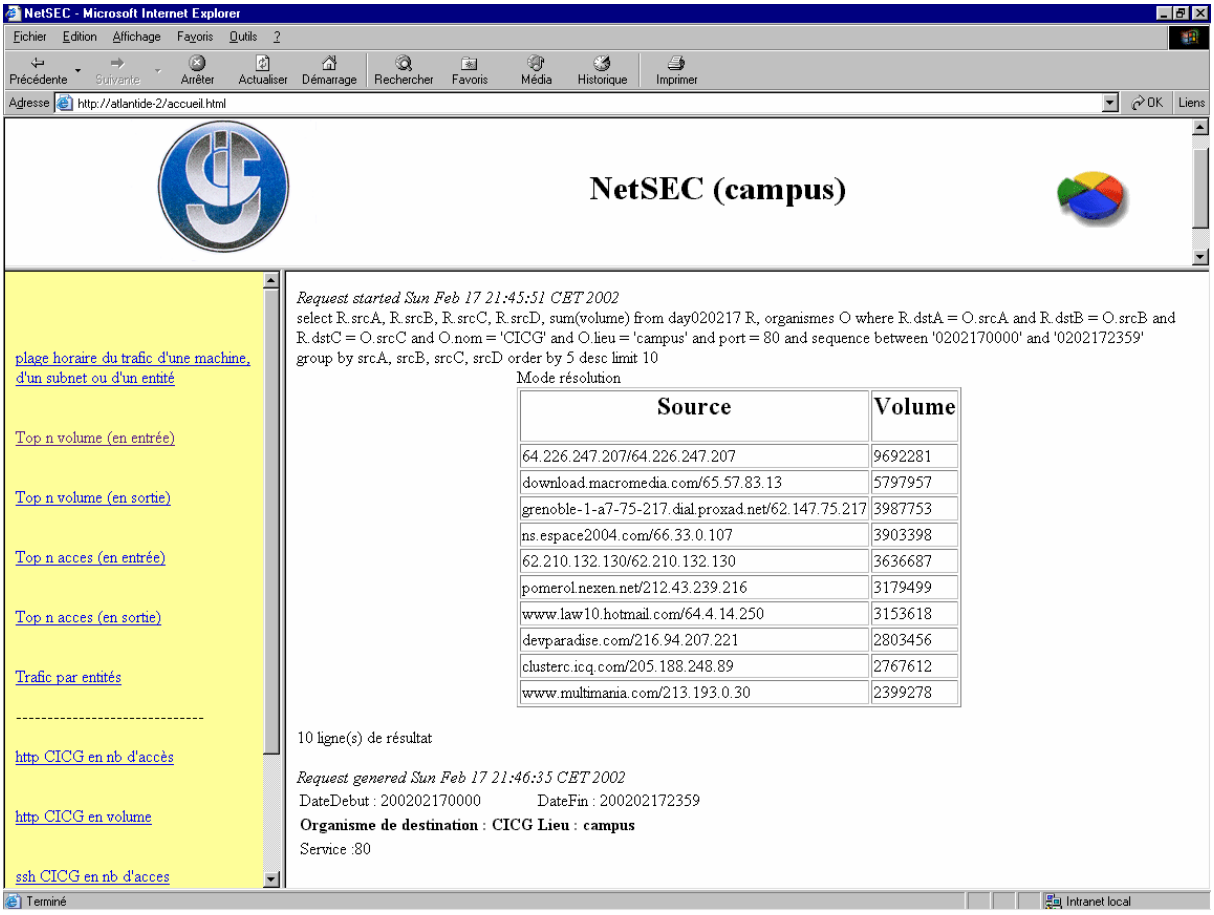


Figure 5 – Results

2.7 Graphics generation

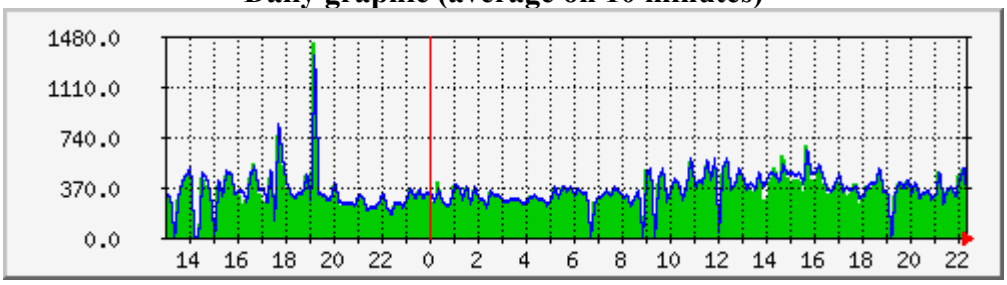
The NetSEC software provides a graphic generation module. This functionality allows a zoom on a network on demand.

Example:
Produce the SMTP graphic for the in and out access number from and to CICG entity

To realize this functionality, we use three components:

- The data base system to provide the data.
- A data query program which has been developed for this need. This program queries the database system and returns the results to the graphic generator software for displaying.
- The MRTG software, tool which builds the daily (figure 6), weekly, monthly and yearly graphics.

Statistics updated on Tuesday the 26th February 2002, 22:34
Daily graphic (average on 10 minutes)



Max smtp IN : 1442.0 access Moyenne smtp IN : 348.0 access Actuel smtp IN : 255.0 access
Max smtp OUT :1334.0 access Moyenne smtp OUT :362.0 access Actuel smtp OUT :273.0 access

Figure 6 - CICG HTTP access daily graphic view

2.8 Data mining

We are currently developing a data mining module. This module is intended to produce non trivial (unknown) and useful information about stored data. The database system holds information about communication flows around the network. For the moment, detecting malicious actions is done a posteriori, that is after the action is performed. We are expecting to detect them by viewing on-going transactions in the database system. Data mining goals are to anticipate in real time the security problems and to raise alarms to administrators. Then, appropriate anticipative actions can be taken. In fact, NetSEC will behave as a network intrusion detector as describe in [5].

The programs query the data to extract the ingoing and outgoing traffic of each organism. We use a free implementation of the Apriori algorithm [6] to produce association rules with a minimum support and a maximum confidence. To produce these rules, we use descriptors which (we think) are the most significant. Some descriptors are time dependent like the day of the week, other are relevant of the port vulnerability level or of the volume transferred.

Example:
"] 14h-19h]" **AND** "SCAN/REGULAR_SERV" **AND** "[0-1KB]" **AND** "TUESDAY" ➔ 53
(14.8%, 90.4%)

This rule comes from thousand rules which have been produced using data of one week. It shows an association between frequent items. The frequent items are the following:

-] 14h-19h]: slot time
- 53: type of service
- SCAN/REGULAR_SERV: level of vulnerability on this service
- [0-1KB]: volume in bytes
- TUESDAY: day of the week

With each rule, a quality level is provided through the frequency and the confidence. Here, the frequency is 14.8% and the confidence is 90.4%. This rule indicates that, the DNS is a current service provided by the organism and regularly tested to find some vulnerability. This rule also shows that, most of the time, the request appeared on Tuesday in the afternoon with a size between 0 and 1 KB. As a first approach, it seems to be correct even if the DNS service is potentially vulnerable. DNS administrator must especially take care of the service. The rest of these rules must be, of course, studied to bring a global security evaluation.

3 Evolution and conclusion

This section presents the features of the NetSEC system that are still under development. Some parts are particularly focused on e.g. data mining, however the following points should be improved:

- Report production:
  One important aspect is to produce reports. They can be of several types and are scalable according to the user needs e.g.:
  - Weekly or monthly reports for people in charge of entity, organism and so on…
  - Daily reports for system and network administrators who can check easily the traffic coherence of the networks they are responsible of.
- Automatic archive:
  Another important problem is to create data archive. Every day, the size of the created tables is important. What is the amount of daily tables which should be kept in the database system? There is no simple and unique answer. As a matter of fact, it depends on the security policy that we want to enforce and the available staff we have to apply this security policy. If we detect an anomaly about the traffic of an entity, and if we discover that one "warez" server has been installed on a host, we can clean this host, inform the CERT and close the problem. But we may wish to find when the corruption was done, who made it, and search details about another possible attack. This last objective is the more interesting, but it requires searching what happened in the past. To do that, we need a history flow. We can imagine for example an automatic archive solution on CD (because they are very cheap). This solution is easy to use, but we need to solve the CD loading problem.
- To develop the graphic generation:
  Today, the graphic generation tool provides only curves about access and volume traffic. We will improve the information produced by graphics. For example, when an anomaly appears on a graphic (see section 1.4), it will be possible to click on it to get detail about the problem, e.g., what sort of traffic is suspected? In which entity?
- To develop data mining:
  We just have initial results. Now, we need to analyse the thousand rules which have been produced and may be finding and developing solutions to help the rules sorting. So, we have to make experiments and gain experience on data mining. We may also find other descriptors to improve the data mining model we have.

To conclude, the last activity is the distribution and the documentation. As a first approach, we plan distributing the platform by the mean of a training session. We think that this mode will allow to exchange with future users before distributing the software through Linux packages on an ftp server. People interested may contact us by mail at: netsec@grenet.fr

References

[1] Tobias Oetiker. MRTG - The Multi Router Traffic Grapher. *Proceedings of the Systems Administration Conference (LISA '98)* http://people.ee.ethz.ch/~oetiker/webtools/mrtg/. December 1998.

[2] A. Simon. NetMET: une solution de métrologie générale pour les réseaux régionaux, métropolitains et de campus. *Actes JRES 2001,* http://www.jres.org/. December 2001.

[3] Y. Autran, C. Chassagne and J.P Leguigner. Le manuel des spécifications de IPtrafic, http://www.urec.cnrs.fr/iptrafic/specifs.html. May 1998.

[4] MySQL home page: http://www.mysql.com/products/index.html

[5] W. Lee, S. J. Stolfo, K. W. Mok. A data mining framework for building intrusion detection model. *Proceeding IEEE Symposium on Security and Privacy,* http://www.cs.columbia.edu/~sal/hpapers/ieee99.ps.gz. 1999.

[6] R. Agrawal, T. Imielinski et A . Swami. Mining association rules between sets of items in large databases. *Proceedings of the 1993 ACM SIGMOD international conference on Management of Data (SIGMOD'93)*, *pages 207-216*. ACM Press. May 1993.

Vitae

Jean-François Scariot is graduated in electrical and computer science at Joseph Fourier University. He is working in network and system at CICG (Grenoble universities computing centre) from 1993. He is currently in charge of the NetSEC project and works on supervision of universities campus backbone network and of the research metropolitan network of Grenoble.

Bernard Martinet received his PhD thesis in computer science in 1992. From 1992 to 1999 he worked in network and system engineering at C.N.R.S. (French National Scientific Research Centre). Since 1999, he works in U.R.E.C (Network Unit of C.N.R.S.). He is in charge of the supervision of the universities campus backbone network and of the research metropolitan network of Grenoble.