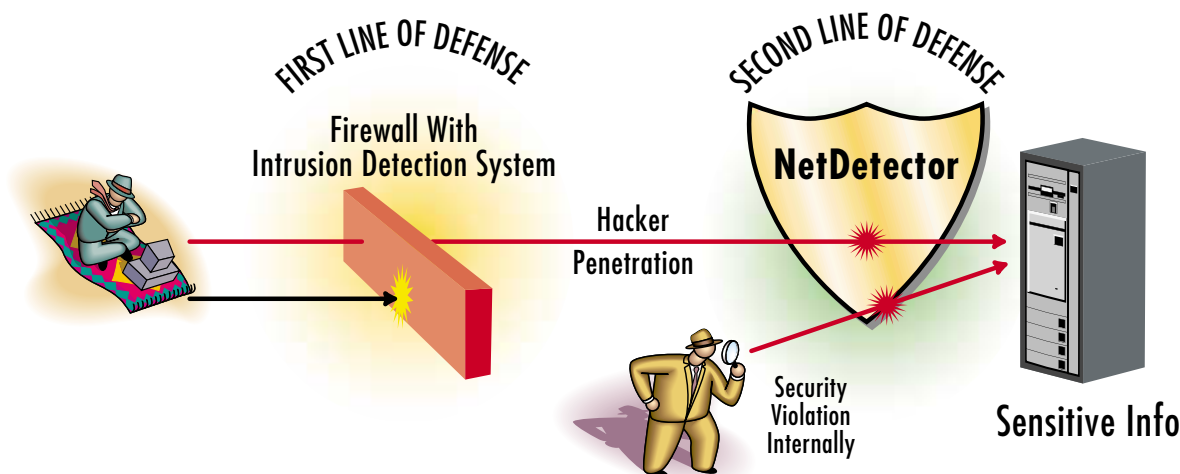


NetDetector™

Infrastructure Equipment for Security Enhancement



- ▶ **Data Recording, Analysis and Playback of Any Network Activity**
- ▶ **Intrusion Detection - Next Generation Intrusion Detection with Sophisticated & Customizable Analysis**
- ▶ **Automated Alarming - Sophisticated Alarms can be made available via JAVA/SNMP**
- ▶ **Role-Based Access Control - Data sets are secure and access is restricted by user community**

Non-Intrusive Equipment

NetDetector does not slow down the network it is monitoring. Intruders cannot “detect” the device which can surreptitiously gather all information in real-time.

Security Breach Analysis

NetDetector can monitor all activity on the network and can analyze traffic patterns for security breaches. Network security breaches can also be analyzed post-event to assess the damage done and the extent/method of the damage.

Hooks into State-of-the-Art Scanning and Signature Detection Systems

NetDetector provides a strong capability to “hook” onto sophisticated “text-analysis” and signature detection systems

Lawful-Recording and Analysis of Data

Suspect activity can be monitored for validation, evidence, and legal prosecution by law-enforcement agencies.

Complete Playback Capability

Applications can be played back to analyze user actions on the network. Examples include: reconstruction of web activity, replay of Voice over IP, scanning of e-mails, telnet sessions, etc.

Historical Archiving

Terabytes of data can be archived, analyzed and processed using NIKSUN’s high bandwidth processing engine.

NIKSUN, INC.
 111 North Center Drive
 North Brunswick, NJ 08902 USA
 T: +1 (732)-821-5000
 F: +1 (732)-821-6000
 E: info@niksun.com
www.niksun.com

NetDetectorTM Concepts

NIKSUN, Inc.
WWW.NIKSUN.COM

NetDetector Concepts

NetDetector is a powerful network surveillance appliance for IP networks. NetDetector provides non-intrusive, continuous traffic recording and real-time traffic analysis. NetDetector makes continuous copies of data from the network, accurately timestamps the recorded data, analyzes every packet, detects the activities of intruders, sets alarms for real-time alerting, and gathers evidence for post-event analysis and legal prosecution

NetDetector is based on NIKSUN's high-performance recording and analysis technology. NIKSUN's monitoring products have been designed and optimized to record traffic at very high-rates and analyze the traffic in real-time. NetDetector runs simultaneously processes for data recording, data processing, alert detection, data analysis and data exporting.

NetDetector provides network administrators with the ability to record large amounts of traffic, to define alerts based on certain events or thresholds, and to analyze the recorded traffic once a network event has occurred. By providing complete post-event analysis and application reconstruction capabilities, NetDetector becomes a very effective way to carry out network security and event investigation.

NetDetector is composed of four main elements: the *Traffic Recorder*, the *Query Processor*, the *Alerter* and the *Web GUI*. The users access all the capabilities of the NetDetector using a Web browser.

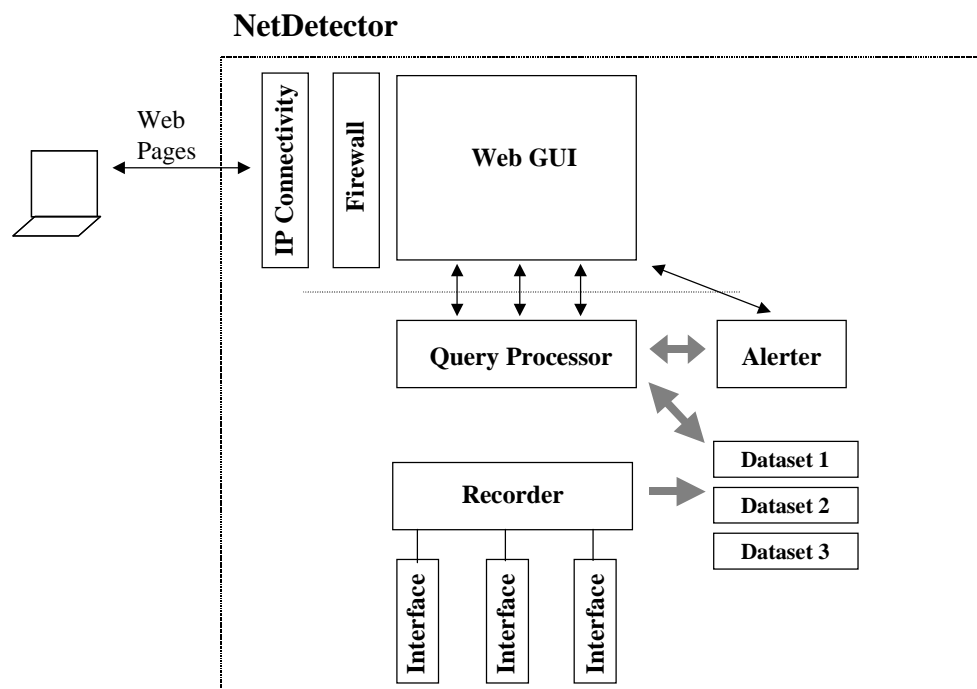


Figure 1. NetDetector Components

The *Traffic Recorder* is the process that collects all the traffic from the network interfaces and puts it into permanent storage. The *Query Processor* analyzes the traffic, once it has been recorded, to respond to queries by the *Alerter* or by the users via the *Web GUI*. The *Alerter* is a background process that calculates traffic statistics to detect and alarm on traffic anomalies and thresholds. The *Web GUI* is the application that controls the interaction with the users, including configuring the system, displaying the alerts, formulating the queries to the *Query Processor*, formatting the results, and presenting the tables, plots and screens.

Once the network traffic has been recorded, users can query the data anyway they desire. NetDetector provides flexible query architecture and intuitive *Web GUI* to visualize the traffic and perform complex traffic analysis. The query processor has pre-processed most of the traffic statistics, so any ad-hoc query is answered quickly by the NetDetector.

Traffic Recorder

The *Traffic Recorder* reads the data from various physical interfaces and copies the data into *datasets* stored on the attached storage subsystem. The Recorder deals with two types of interfaces, physical interfaces and virtual interfaces.

Physical interfaces are high-performance network adapters. They copy the traffic from the network non-intrusively, accurately timestamping each packet, and forwarding the packets/ cells / frames to the *Traffic Recorder*.

Physical interfaces are recognized by a unique name that is made of the type of interface and a sequential number.

Type	Interface Name
10/100 Ethernet interfaces	<i>fxp0, fxp1, fxp2, fxp3, ...</i>
10/100/1000 Ethernet interfaces	<i>ti0, ti1, ...</i>
T1/E1 interfaces	<i>fdm0, fdm1, fdm2, fdm3, ...</i>

fxp0 is management interface. This interface is used only to communicate to the NetDetector and is not used for monitoring.

T1/E1 interfaces support various encapsulation protocols, including Frame Relay, PPP, Cisco HDLC, and Bay PPP. The encapsulation protocols for each T1/E1 interface are configured at the factory.

Virtual Interfaces are user-defined interfaces that correspond to a subset or a superset of traffic recorded by physical interfaces. In other words, virtual interfaces reference to traffic on other physical interfaces. When the virtual interface is created, it is given a system-generated name (i.e. *fxp1_vi0*). Virtual interfaces can expand various physical interfaces only when the physical interfaces of the same type.

Security administrators are able to define virtual interfaces for a subset of the recorded traffic. For example, users could create a virtual interface that only contains broadcast traffic and use the virtual interface to monitor the level of broadcast packets on a network. If a user detects an increase in the level

of broadcast traffic over a short time frame, he might want to investigate it further as it might be caused by a broadcast storm attack.

Each physical interface for Ethernet and Gigabit Ethernet represents a half-duplex link. When the physical interface is connected to a shared hub or a mirrored port on a switch, it will only monitor half-duplex links. However, when the NetDetector is connected to non-intrusive full-duplex link through Ethernet taps, the users should create a virtual interface that combines the two uni-directional interfaces. With that virtual interface, the users could analyze the bi-directional traffic. For example, a full-duplex virtual interface `fxp1_vi0` can combine the traffic of `fxp1` and `fxp2` half-duplex physical interfaces.

In some instances, security administrators might want to analyze unidirectional traffic. For example, when security administrators want to detect *host scans* or *ports scans* coming from the outside, the alerts only need to be configured on the incoming traffic. The associated *host scan* or *port scan* alert should be defined on the physical unidirectional interface, as opposed to on the virtual bi-directional interface.

In the case of T1/E1 interfaces, each physical *fdm* interface represents up to 8 unidirectional links, which can monitor simultaneously up to 4 full-duplex T1/E1s. In order for a security administrator to have an interface that represents each individual T1/E1, he needs to create virtual interfaces on the *fdm* interface that select only certain links¹. In the case of channelized T1/E1 interfaces, the security administrator could also create virtual interfaces that refer to each individual channel on a channelized T1/E1 by specifying the appropriate qualifier filter².

Datasets: They are NIKSUN proprietary file structures that contain all the traffic information associated with an interface, including the raw traffic captured by the *Recorder* and the statistics caches generated by the *Query Processor*. As described in Figure 2, NetDetector stores in each dataset the raw traffic and a set of pre-calculated statistics.

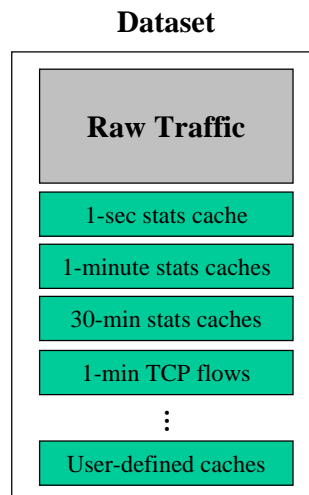


Figure 2. Structure of Traffic Dataset

¹ The filter language contains the qualifier *tlink* to identify the appropriate link. For example, the filter “*tlink 0* or *tlink 1*” selects the first full-duplex T1 on a T1 card.

² Each channel within the T1/E1 can also be identified by qualifier *tchannel* in the filter language

Each dataset is named in reference to the recorder hostname and the interface name (physical or virtual). For example, netdet1.acne.com/fxp1 refers to dataset recorded by the host netsec1.acne.com and the physical interface fxp1. When the hostname is changed, NetDetector creates new datasets with the new hostname and the name of the recoding interfaces.

There are four types of datasets:

- a. **Datasets linked to Physical Interfaces:** These datasets own the recorded raw traffic and the associated statistics caches. The configuration screens for these datasets control the recording parameters of a physical interface, including capture filters, bytes to keep per packet, and amount of raw data to store. The dataset configuration screens also allow stopping and starting the recording of the physical interface.
- b. **Datasets linked to a Virtual Interface:** These datasets that do not contain raw data; their raw data information is only a reference to physical dataset with the raw data. These datasets, however, do contain statistics caches generated by the Query Processor for the filtered traffic (the traffic that matches the qualifier filter defined when the virtual interface was created).
- c. **Cloned datasets:** These datasets do not contain raw data or statistics. There is just a time interval reference to a dataset linked to a physical interface. Cloned datasets are named fxp1_copy1, fxp1_copy2, ...
- d. **Archived datasets:** These datasets contain archived raw data and statistics, but are not associated with a physical interface. These could include datasets recorded by other host, datasets recorded under a different hostname, or datasets generated by an alarm.

Each dataset can have the following four states:

State	Explanation
Recording	Recording has started and the data is being aged
Stopped	Recording has stopped and the data is being aged
Archived-Static	Recording has stopped and the data is not being aged
Archived-Recording	Recording has started and data is not being aged

- Datasets linked to Virtual Interfaces only age out statistics caches. They do not age out raw data because they do not own the raw data (the datasets linked to the Physical Interfaces do)
- Cloned datasets do not own any raw data or statistics caches. However, if their state is Archived-Static, the time interval that the cloned dataset references to it will not be aged out.
- Archived or Cloned datasets cannot take the state Recording or Stopped, since they do not have an interface associated with them.

When the recorded data has filled its allotted space, NetDetector overwrites the new raw data over the oldest raw data recorded. The overwritten data is said to have been 'Aged Out'. This Data Aging process is performed by the *Space Management process*.

Statistics for the raw data are calculated as soon as the defined resolution period, or aggregation window, has elapsed and saved in a separate disk space (by default, NetDetector has statistics aggregating caches

for 1-sec and 1- minute windows). Loss of the raw data therefore does not mean loss of the statistics. The statistics are also recorded cyclically within their defined disk space; as data ages, the older high resolution statistics are overwritten by new statistics of the same resolution.

Space Manager: The space manager is the process that cleans up the file system for new incoming raw traffic and statistics. Depending on how each dataset has been configured, the space manager will age out raw data or statistics not marked as archived. The space manager process only starts to clean up the file system when it becomes full. However, since the NetDetector is expected to monitor traffic indefinitely, having a fairly full file system is a normal state.

Query Processor

The *Query Processor* is the process that generates all the traffic statistics required by the NetDetector. It supports a powerful and flexible query language, accessed through the *Web GUI* and to the *Alerter*. The Query Processor is also responsible for generating the statistics caches stored within the datasets. The caches enable the Query Processor to respond to any ad-hoc query generated by the user very quickly.

Alerter

The *Alerter* is the program that monitors the alerts defined by the user. The *Alerter* is constantly running queries to the *Query Processor*, determining if the thresholds defined in the alarms has been reached.

Once the specified threshold of the alert has been exceeded, the Alerter will take the appropriate action defined by the user. The actions could include a screen pop-up, email, SNMP trap and/or archiving the traffic for the appropriate time interval.

In the release 1.1, NetDetector supports seven alert categories. These categories were chosen to provide generic yet powerful and flexible means of detecting suspicious traffic patterns without resorting to script writing or signature downloads. The alert categories are as follows:

Utilization: This alert type tracks the average utilization used by the network traffic on the given interface over a given time interval. If the traffic utilization exceeds the given threshold, the alert is considered breached. You can further analyze the breach to identify the traffic sources and destinations that contributed to the higher-than-expected utilization. Depending on your site's network security policy, utilization breaches could be the sign of Denial of Service (DoS) attacks (either as a victim or unwitting attacker), or other inappropriate usage.

TCP Connections: This measures the number of active TCP connections (i.e., TCP state transitions) per host pair on a given monitored link. If the number of connections exceeds the threshold over a certain period of time, the alert is triggered. Once a baseline value for the number of such connections is determined, host pairs that exceed the specified number of TCP connections are considered anomalous.

Host Flood: This alert monitors the number of host pairs involving a common destination. Such a pattern could arise from a Distributed Denial of Service (DDoS) attack or an attack with spoofed source addresses. When this alert occurs, analysis can be done on the common destination to assure these acts are not malicious.

Host Scan: This alert monitors the number of host pairs involving a common source. Such a pattern could arise from a reconnaissance scan (where a hacker is attempting to map the hosts on your network) or from a single source DoS attack.

Host Pair Bytes: This alert monitors the number of bytes exchanged per host pair on a given monitored link. This could indicate inappropriate usage such as a bulk data retrieval or submission.

Invalid Address: This alert monitors valid IP ranges for your site's network or Intranet. The alert is useful on assuring only valid IP numbers are within your network. The alert creates a fishbowl, which allows you to contain, isolate, and monitor an unauthorized user within a system. If the network is connected to the Internet, this alert should be defined in uni-directional links.

Port Scan: This alert detects Port Scans being done on your network. It is useful to identify when a potential intruder is gathering information to be used in a future exploit. This alert gets triggered whenever the threshold for the number of ports scanned on any host from a common address is exceeded.

Web GUI

The NetDetector Web GUI is application that controls the user interface. It receives user requests and responds to those requests. The Web GUI consists of the Surveillance screen (and associated screens), the Configuration screens and the Traffic Analysis screens.

The *Surveillance Screen* provides a centralized view for all alerts that are being monitored by the NetDetector. The Surveillance screen allows access to "children pages", including the Alert Detail Screen and the Alert Log Screen. The *Alert Detail Screen* contains all the instances that the selected alert has been breached. The *Alert Log Screen* consolidates all the breaches for all the alerts and the respectively action taken when the breaches occurred.

The *Configuration Screen* contains four configuration screens. The *Surveillance Configuration Screen* allows users to create, edit, and delete alerts. The *Network Services Screen* allows users to Enable/Disable FTP or Telnet access to the host. The *Traffic Recording Configuration Screen* allows users to configure all the interfaces or datasets and its corresponding recording properties. The *IP Networking Configuration Screen* contains the network settings for the management port of the unit.

The *Traffic Analysis Screens* present and visualize traffic statistics. As illustrated in Figure 3, the Traffic Analysis Screens adhere to a standard format, displaying tables on the left and timeline plots on the right. However, there are some protocol-specific differences on the displayed tables.

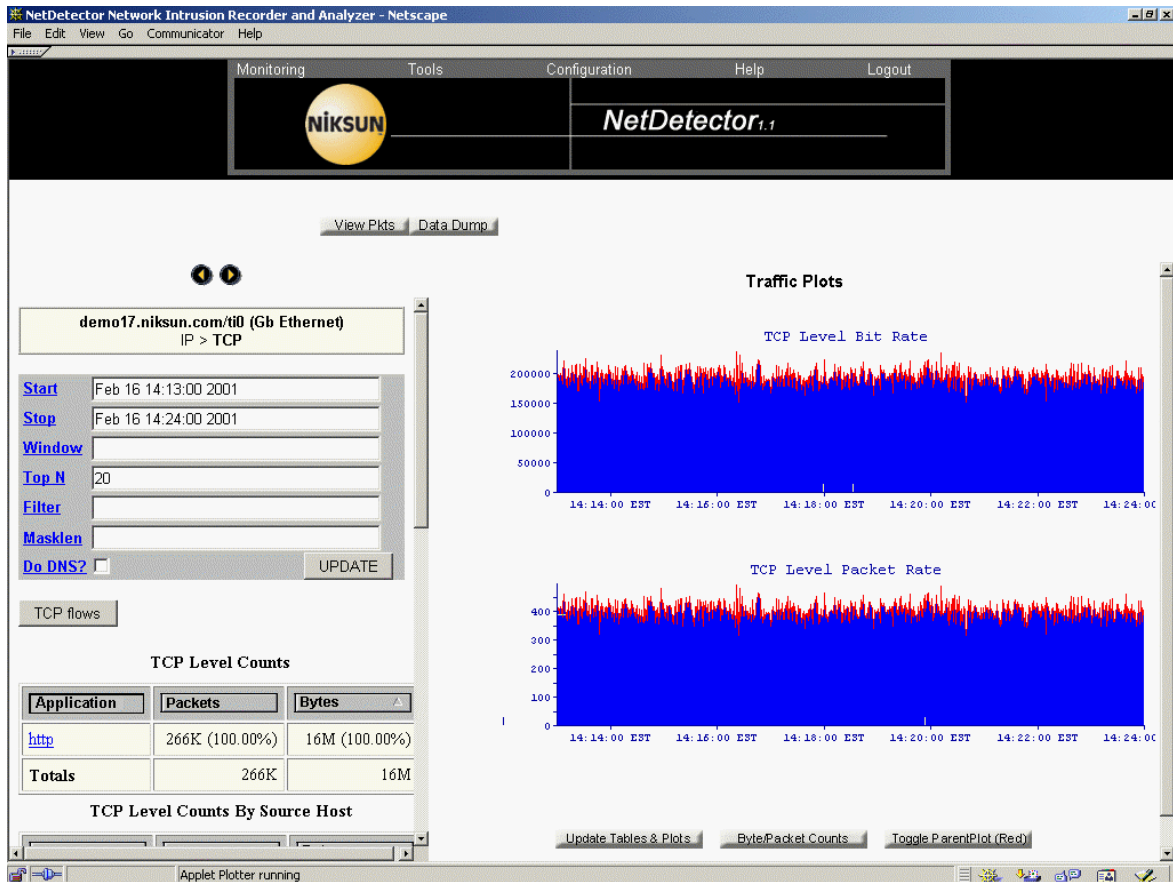


Figure 3. Traffic Analysis Screen

The *Traffic Analysis Screens* have two very compelling concepts that make it very intuitive and powerful for security applications:

- *Multi-timescale analysis*: Security administrators can analyze the recorded traffic at any time scale, by specifying the start and end time of the analysis. The plots and tables always reflect the time scale selected by the user. For example, if a user selects 1 hour time span, the time plots will contain a resolution of about a 10 second window, while if the user selects 6 seconds of time span, the plot will use a 10 milliseconds window.

Since some network attacks occur over long periods of time while others occur during very short periods of time, having the ability to perform traffic analysis at any time scale is critical to investigating security events.

- *On-demand statistics based on traffic filters*: The *Traffic Analysis* screens display the traffic statistics (tables and plots) on the user-selected traffic. The security administrators can select the traffic filter by typing the filter explicitly or by using the available hyperlinks. Once the user has changed the traffic filter, the new Traffic Analysis screen will show the traffic statistics associated with that new filter. In addition, the traffic dump and sniffer export capabilities are applied to the current time interval and traffic filter.

The traffic filters on a Traffic Analysis screen specify the subset of the traffic the user wants to analyze. Security administrators can isolate certain activities from the rest of the traffic. In a very

busy network with large amounts of hosts, application and flows, security administrators need to be able to isolate and analyze certain subset of traffic that might look suspicious.

The investigation process is very interactive and ad-hoc. Security administrators generate certain hypotheses about the events that occurred, perform analysis on the traffic and confirm or reject the hypothesis based on the results. Security administrators can then investigate further the events that have being verified.

For example, the security administrator detects a spike in ARP traffic during a particular timeframe, which might represent a host scan. The security administrator can't determine the source of the scan, since the ARP packets only contain the IP address of the immediate router. To determine the source, the security administrator might look for the hosts that have large number of host pairs during that particular time interval or look for a common source on the packets that followed the ARP reply packets.

Traffic Filters

NetDetector uses a very flexible and powerful filtering language³. Users can define filters that identify traffic by direction, protocol, application, host, network, etc. The filters have the purpose to determine if a packet complies with a certain defined criteria. Filters sort on a packet-by-packet basis.

NetDetector supports filter expressions in: capture filters, virtual interface qualifiers, traffic analysis filters, and alert filters.

1. *Capture filters*: Capture filters are used to reduce the number of packets that are kept by NetDetector. For example, one could set a capture filter to gather only web traffic. Security administrators can set up a capture filters to filter out traffic that does not need to be recorded because it does not represent a security threat. For example, recording the transactions between the database server and the web server might not be interesting.
2. *Virtual Interface Filters*: These filters define the traffic that forms part of a virtual interface.
3. *Filters for Traffic Analysis*: Traffic analysis filters are applied to the recorded traffic after the packets have already been captured. These are used to display traffic statistics for a subset of the traffic.
4. *Alert Filters*: Alert filters are defined for each alert, so the alert is only applied to a subset of the traffic. Security administrators are able to define alerts on events related to certain traffic patterns. For example, if the users is concerned about the amount of bandwidth for a particular application, it can define an alert that tracks the utilization of that particular application.

³ The filter language is very similar to the one used by *tcpdump*, a well-know Unix program for traffic capturing and filtering