AUTOMATYKA • 2010 • Tom 14 • Zeszyt 3/1

Artur Sierszeń*, Łukasz Sturgulewski**, Michał Dubel**,
Tomasz Marciniak**, Adam Wójciński**

Network Behavior-Analysis Systems with the Use of Learning Set and Decision Rules Based on Distance

1. Introduction

The most critical problem for network administrators is the thorough analysis of all traffic flowing through their networks. Apart from the operation of traditional firewalls, detection systems, and intrusion prevention systems, administrators have to identify and block traditional attacks as well as detect malicious traffic and new attacks in real time, regardless from the fact whether the reasons for these activities result from more and more demanding regulations or from new, more directed attacks and the latest techniques of malicious code hiding.

In order to detect malicious activity in a network, most widely applied security technologies (antivirus software, IDS/IDP, or firewalls) are based on lists of known patterns or statistic rules. Although protection based on signatures is a significant component of the protection arsenal, it is essential to be able to identify hidden attacks which escape notice of traditional protection systems. Network Behavior Analysis (NBA) systems are becoming more and more popular.

NBA is an ability to identify traffic patterns which do not occur during normal operation of a network. In other words, it is an attempt to identify irregularities in a network, one which goes beyond simple settings concerning exceeding parameters for traffic of a given type.

NBA systems focus mainly on internal traffic. They can view packets at network connection, similarly as IDS/IPS do. However, NBA attempts to recognize a threat which occurs again. Owing to this ability, the mechanism tries to intercept threats omitted by IDS/IPS or antivirus software. NBA devices are dedicated to detecting anomalies in

* Computer Engineering Department, Technical University of Lodz, Poland
** Student, Computer Engineering Department, Technical University of Lodz, Poland

network traffic which go beyond standard behavior patterns. NBA systems focus on behaviors or symptoms. Updates of NBA analyzing engine are required considerably more rarely than in IDS/IPS systems.

2. Traditional methods of network monitoring

The first protocol via which it was possible to manage a network (through proper management of network devices) was Simple Gateway Monitoring Protocol presented in November 1987 [4]. Its new version [5] called Simple Network Management Protocol appeared in short time afterwards (August 1988). In the following years subsequent specifications of this protocol appeared, ones which developed its functionality and eliminated errors discovered in the protocol. The simplicity of implementing the first version of the protocol (SNMP v1) resulted in wide application of this solution in network management; virtually each producer of network equipment includes a SNMP agent in its software. Subsequent versions of the protocol did not change much concerning the general principle of operation; they focused on adding new functions and increasing security level.

The SNMP v2 improved the function of reading more than one parameter at a time and the way of propagating traps. The SNMP v3 introduced a correct security level to the protocol; however, its complexity of implementation is the reason why many administrators are not willing to use full possibilities of security mechanisms.

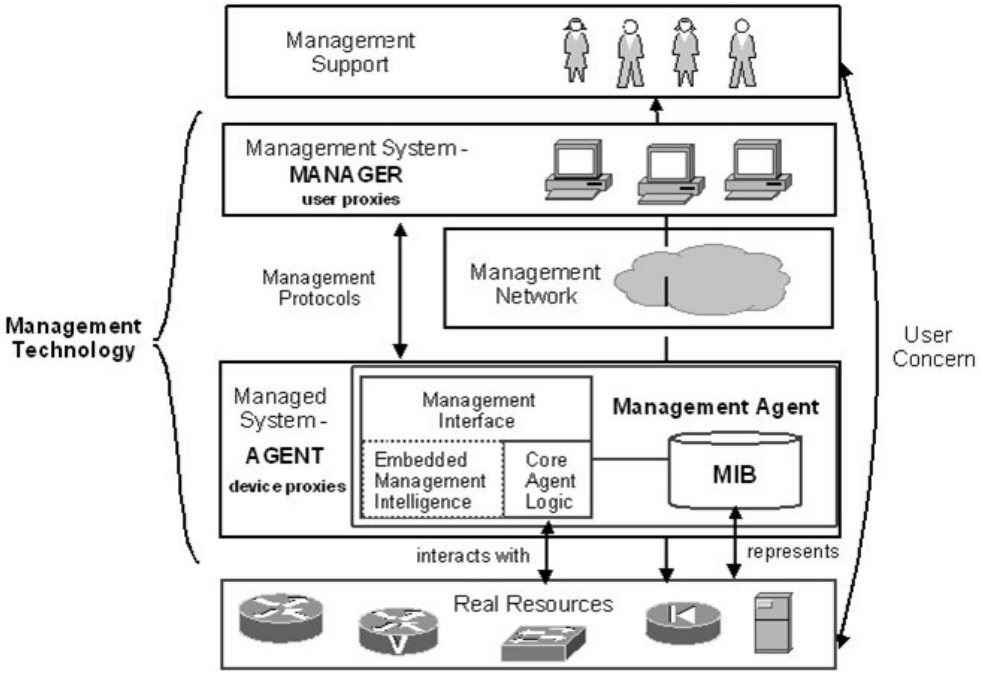

Fig. 1. Anatomy of Simple Network Management Protocol

Monitoring with the use of SNMP protocol is conducted through a set of applications complementing the software of the managing station, applications which present information about the state of devices gathered in the central MIB base (Fig. 1) [7].

Other systems of network monitoring, which are currently used, are Intrusion Detection System (IDS) and Intrusion Prevention System (IPS). These are systems which increase computer network security by detecting intrusions (IDS) or detecting and blocking attacks (IPS) in real time.

Intrusion Detection Systems operate basing on the analysis of network traffic with the use of two methods:

– Heuristic analysis – an analysis which consists in defragmenting, combining packets in data streams, analyzing packet headers, and analyzing application protocols. It enables one to select packets which may result in the destabilization of the target application in case it contains implementation errors. In most IDS systems heuristic analysis is performed simultaneously with data normalization before the data is subject to signature analysis.
– Signature analysis – an analysis which consists in searching packets for data streams characteristic of known network attacks. A key element is a base of signatures which are constructed when new attacks appear and which are updated frequently enough.

Typical elements of IDS/IPS system are the following:

– sensor – an element which analyzes network traffic and detects attacks,
– data base – a component which gathers information about attacks from a group of sensors,
– log analyzer – an element which makes it possible to visualize and analyze logs from a group of sensors.

Depending on the location of a sensor and the scope of analyzed events, the following kinds of IPS systems are distinguished [3]:

– *HIDS/HIPS* (Host-based IDS/IPS) operates as an application in one protected operating system, analyzing events coming from system logs and from local network interfaces.
– *NIDS/NIPS* (Network IDS/IPS) analyzes network traffic for all systems in a network segment to which it is connected. NIDS can recognize attacks directed against systems without HIDS installed. However, it has limited abilities of analyzing traffic sent in SSL channels or events which occur locally in a system (e.g. shortage of memory, local attacks from a console).
– *CBIPS* (Content-based IPS) inspects the content of network packets for unique sequences, called signatures, to detect and hopefully prevent known types of attack such as worm infections and hacks.
– *RBIPS* (Rate-based IPS) is primarily intended to prevent Denial of Service and Distributed Denial of Service attacks. Through real-time traffic monitoring and comparison with stored statistics, RBIPS can identify abnormal rates for certain types of traffic e.g. TCP, UDP or ARP packets, connections per second, packets per connection, packets to specific ports etc. Attacks are detected when thresholds are exceeded.

IPS network systems can operate in the following network topologies:

- *passive sensor* – a sensor which is connected to the monitoring port of a switch. It analyzes the copies of all packets in a relevant network segment. A sensor in this topology has limited abilities of reacting to attacks. In passive topology two techniques of blocking attacks are applied. The first one consists in sending false TCP RST packets to both communicating parties and breaking the connection; the second one consists in dynamic reconfiguration of the firewall with which a sensor can cooperate. In the first case only TCP traffic can be blocked; in the second one reaction may be too late.
- *Inline* – a sensor is located between two segments of a network; it lacks IP addresses. It operates as a transparent bridge and it analyzes and directly participates in transferring all packets in the network. In this mode a sensor can block 100% of packets recognized as dangerous (false TCP RST packets are still sent in order to avoid retransmission). Operation in this mode is much more demanding for sensor software concerning efficiency and stability.

3. Flow system of network monitoring

Programmers of modern applications try to design them in a way that makes them even easier in operation. This is inseparably connected with the increase in data amount sent through a computer network. It turned out that the methods of network traffic analysis used so far do not perform well when detecting defects as a result of which new applications operate incorrectly. The concept of flow became the basis to develop systems which would be able to monitor network efficiently.

Flow is a set of IP protocol packets going through an observation point in a determined period of time.

The first producer who implemented functions for monitoring based on flows in its devices was the network devices tycoon, namely CISCO Systems, Inc. It is this company that introduced the notion of NetFlow, which describes the informal standard of gathering information about flows and the way of exchanging these information. Functional solutions anticipated a formal standard.

Despite the leading role of Cisco as a network devices producer, it became necessary to develop universal methods of gathering information on flows in networks using technologies of other producers. IETF (The Internet Engineering Task Force) formed the working group called IPFIX which was responsible for preparing the Information Model (one which would describe IP flows) and IPFIX protocol (one which would define a protocol for exchanging information on flows). The result of this group's work was the publication of the standard called Requirements for IP Flow Information Export (IPFIX) [6].

Preparing a new standard, the group working within IETF introduced a specific set of notions which describes the architecture of the new monitoring system.

IP Traffic Flow – defined as a set of IP protocol packets going through an observation point in a determined period of time. All packets forming one flow have a common set of defined parameters, e.g.:

- the content of one or more IP packet header fields (e.g. target IP packet address), transport layer header field (e.g. no. of the source port) or application layer header field,
- a parameter which characterizes a packet itself (e.g. MPLS protocol label number),
- information on a way of redirecting a packet (e.g. next hop IP address, output interface through which a packet is going to leave the device).

A packet is assigned to a given flow if its properties meet conditions defining the flow.

observation point – a logical point in a network (switch, router port, a place of connecting a device dedicated to flow monitoring) in which it is possible to observe IP packets. Observation points may exist within other points (e.g. observation points on individual router interfaces are contained in the observation point which covers the whole router).

metering process – a process which creates logic flows (grouped in records) basing on physically observed packets (in an observation point) which meet criteria of defined flows. This process uses functions which intercept packet headers, perform timestamping, sample, and classify. This process also initiates functions responsible for management of flow records (which contain statistics of individual flows). Their operation consists in the creation of new records, updating the existing ones, detecting outdated flows, transferring flows to the exporting process, and deleting flow records.

flow record – a structure storing information on flows observed and classified in a given observation point. A record contains measurable information on a flow (e.g. a total byte number in a flow) and data characterizing a flow itself (e.g. IP source address).

exporting process – a process which sends flow records (generated by the classifying process) from an observation point to one or more collecting processes.

collecting process – a process receiving flow records sent by one or more exporting processes. After having received records, it may store the records or process it further.

Using the notions defined above, the architecture of the new solution may be presented in the diagram next page (Fig. 2).

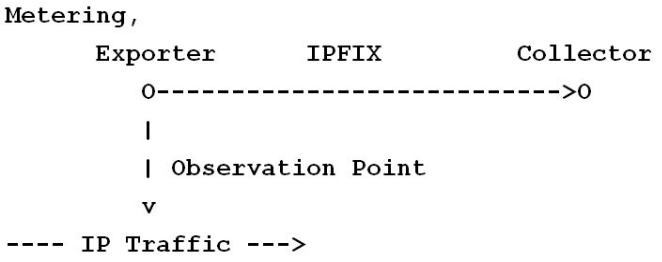

Fig. 2. Anatomy of Simple Network Management Protocol

Many commercial solutions have been developed basing on IPFIX architecture. The most popular ones are presented in the table below (Tab. 1).

Table 1
Squashing internal threats

Vendor	Product	Features
Arbor Networks	Peakflow X	Stop known and emerging threats; eradicate phishing solicitations; and control user access and eliminate insider misuse
Cisco	MARS	Reduce false positives; define mitigation responses; recommend threat removal; and visualize attack path to identify source of threat
Granite Edge Networks	ESP	Create a virtual security zone around key business processes; expose suspicious activity; and pinpoint threat progress to trigger mitigation
Lancope	StealthWatch	Identify rogue traffic, hosts, devices, applications, users and top talkers; and provide security and network analysis
Mazu Networks	Profiler	Detect and mitigate threats; isolate activities that pose risk; and audit and enforce network use policies
Q1 Labs	QRadar	Detect and respond to anomalous network behavior; build real-time network asset profiles; and correlate events from enterprise security products

4. Pattern Recognition

The authors of this article propose to use mechanisms developed within a discipline called Pattern Recognition to construct algorithms which would search for flow patterns and detect anomalies.

Among methods of pattern recognition, two main groups may be distinguished:

– *Structural recognition***,** known in literature as linguistic or syntactic recognition methods;
– *Decision-theoretical recognition*, frequently called deterministic recognition or, sometimes, statistic recognition.

In the first case, a constructor, using his knowledge and experience, describes classes. If the description deduced from the learning set enables one to recognize objects well enough (objects not only from the learning set but also from another independent testing set), it is regarded that the construction of a classifier has been successful. This type of classifier construction heavily depends on specific applications. In this approach, the most difficult part of the task is played by the block of properties separation. The classifier is constructed very simply; it only verifies to which of the classes the description of the recognized object corresponds most.

The second and much larger group of methods is presented in a simplified way in the figure (Fig. 3). In the figure, a bloc was marked connected with algorithms based on the Nearest Neighbor Rule (k-NN). Algorithms from this family come from supervised classification, in other words from classification in which unknown class labels of recognized objects are determined based on a set of objects with known labels, i.e. the learning set (Fig. 4). Algorithms from this group will be used to develop mechanisms for controlling data flow.

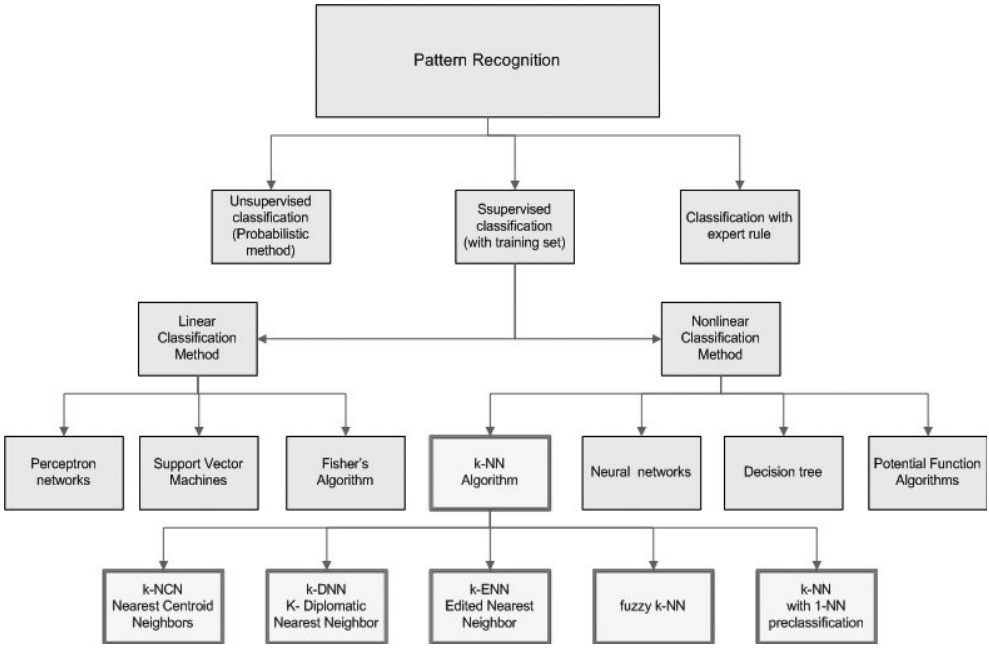

Fig. 3. Division of pattern recognition methods

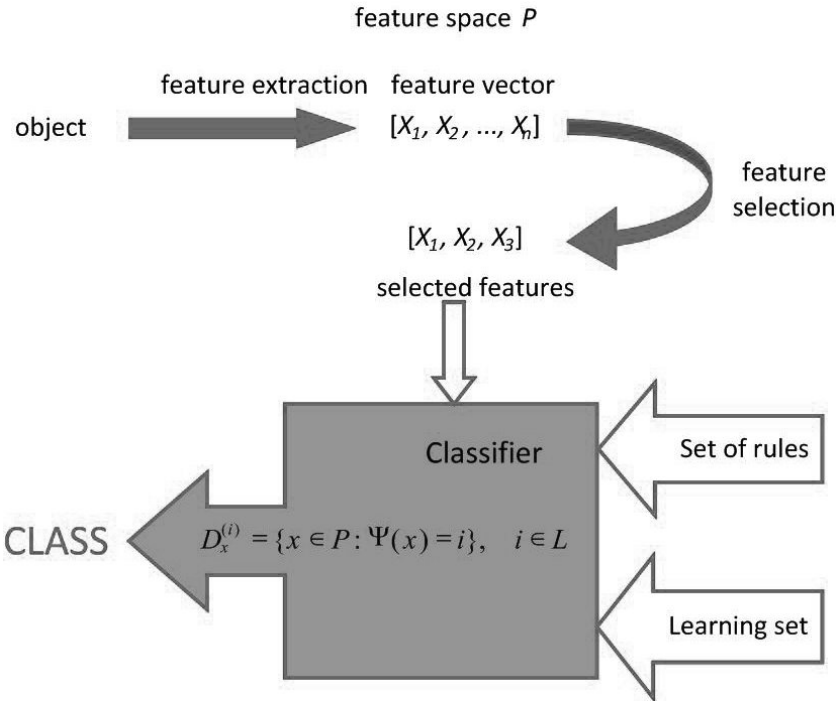

Fig. 4. Supervised classification

A pattern recognition algorithm is a decision rule which assigns one of c classes to each test sample to which this rule has been applied. Each sample is usually described with a property vector $x \in R^S$ and a class label from the set $I_c = \{1, 2, ..., c\}$. Each function

$$D : R^S \rightarrow I_C$$

is a classifying function or decision function on R^S. The result of a decision function is decomposition of R^S space into c separate regions I_C [1].

There are two main kinds of classification:

– *supervised classification* – unknown class labels of recognized objects are guessed based on a set of objects with known labels; this is the learning set.
– *unsupervised classification* – in this case a set of objects is given; however, there is no information about a class to which an object belongs. The task consists in dividing the set of objects into two or more subsets. Objects in a subset have to be similar to each other as much as possible (in the space of given properties and terms of defined metrics); objects from different subsets have to be possibly least similar to each other. Subsets created in such a way are called clusters, and the process of their creation is

called clustering. Examples of clustering are e.g. the segmentation of objects in 2- and 3- dimensional images or categorization of text documents e.g. for the needs of network browsers.

The basic rule of classification may be expressed with the following sentence: "you are the same as your surroundings". Some kinds of classifiers use the neighborhood of a sample in a more direct way (e.g. the rule of *k* nearest neighbors), other in a more disguised way (e.g. neural networks). It is not even possible to imagine that this paradigm could be rejected. However, in order to talk about similarity, a measure of similarity should be introduced, in other words a measure of distance.

In case of numerical properties the most popular measures are Euclidean metrics and urban metrics (Manhattan). They are specific cases from the family of Minkowski metrics, which can be described by the following formula:

$$d_p(x, y) = \left[\sum_{i=1}^{d} (x_i - y_i ($$

where $d_p(x, y)$ is a distance between vector x and vector, and d is a dimension of space. For $p = 1$ urban metrics is received, for $p = 2$ Euclidean metrics is received and for $p \to 8$ the maximum measure is received (the border case).

Apart from two specific cases of Minkowski metrics, many more other metrics are known, e.g. Czebyszew metrics, Mahalanobis metrics, or Camberra metrics [2]. However, incomparably higher costs of computing distances in these metrics results in the fact that their application is limited in practice.

5. Concept of NBA system with the use of k-NN rule

Our goal is to create an analyzer which would be properly included in a network's structure and which would make it possible to gather data flowing through the network and to perform the following tasks: gathering and processing data in order to learn the standard computer network operation, detecting (based on gathered patterns) the patterns of events related to network traffic, and presenting the obtained results in the form of an interactive dialogue with a network engineer via an interface constructed as a dynamic website with the use of php technology. Data should be gathered in real time and should be passed to the data base (at this stage, we use mysql base for the construction and management) where they would be processed into a clear report after having been taken by a relevant module.

Relevant sub-modules have been sectioned off the project according to the architectonic design pattern, i.e. Model View Controller (Fig. 5). Ultimately, the platform, the programming language, and the data base type should be chosen freely by a programmer and compatible with each other (they should facilitate the implementation of new solutions in the preferred programming environment).

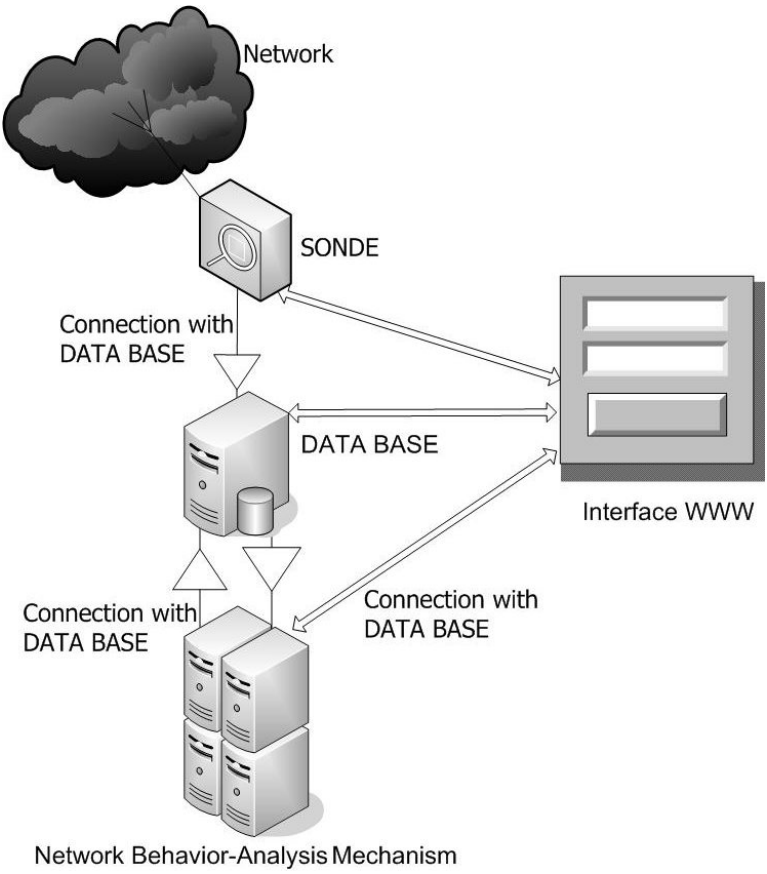

Fig. 5. Operation diagram of proposed solution

The sonde taking data from a chosen network segment as well as the engine processing the data are Controller modules. The sonde uses Pcap library (or Winpcap i Windows systems). The current version uses the wrapper of this library called jpcap which enables the implementation of solutions using the sonde's modules via applications written in Java language. The sonde is able to intercept network traffic on a chosen interace (dynamically recognizing additional connected interfaces and network modules if necessary) according to the filtering criterion chosen by the engineer (supporting various the second and third of OSI layer network protocols as well as protocols of higher layers and making it possible to filter the traffic flowing through connections using chosen ports or physical or logical addresses).

The data processing engine will be able to create a summary of received data according to filtration criteria given by the user interface (via www browser). Its task will be to compare the observed network traffic with the set of model behaviours of a network the opera-

tion of which is not disturbed by external factors (e.g. irregularities resulting from the physical topology fault, incorrectly configurated network service, or an attempt of malicious intrusion or network attacks). The model behaviours of a network will be gathered in two places, i.e. the database of received information and the database of models learnt through the analysis of the chosen computer network. Owing to this comparative analysis, it will be possible to conclude whether the network operates correctly and, if any irregularities are detected, the system should notify the user of their most probable cause.

View module is a dynamic website to constructed with the use of php (Fig. 6), which will be able to present results obtained by the engine as well as to reconfigure the engine itself (this function, however, will be available in future versions) and to directly edit the database, what will enable end users to control the operation of the sonde quickly and comfortably as well as to monitor and gather the results. This website will also graphically present reports notifying about various factors, i.e. network load, proportions of used protocols, detected potential attempts of malicious attacks in a chosen network segment etc.

The database will be the place where data flowing directly from the sonde module used by the engine will be stored. It will also contain patterns of network operation.

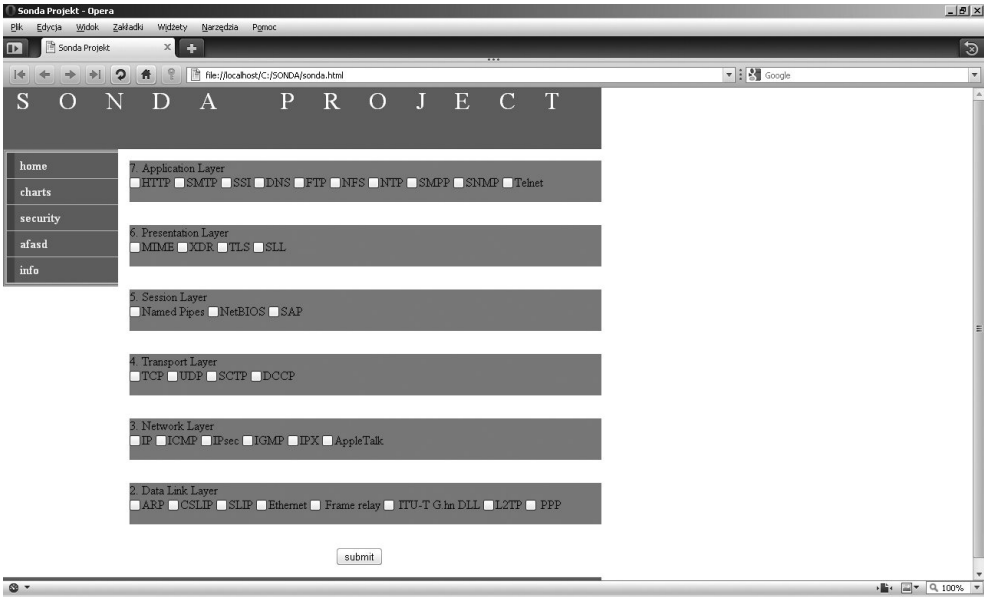

Fig. 6. Program interface

6. Conclusion

The most difficult task is to develop a model of well-working network. It should be possible by gathering appropriate properties describing a data unit sent through the network

based on the detailed analysis of a packet header. The authors are aware of a large amount of data (packets) which will be included in the analysis and the processing in the classification process. The operation speed of minimum-distance classifiers depends mainly on the size of the reference set. Therefore, the reduction of the reference set is the most common method of improving the classification speed. Its additional asset is a fact that in many cases the reduction of the reference set does not influence the classification quality or influences the quality only to the minimum extent. Numerous methods have been developed to separate a reduced set which would be smaller than the original set. Most of these methods generate sets consistent with the original sets what means that the reduced set enables correct classification of all points from the original set. The conformity criterion can determine whether, and to what extent, the reduced set approximates the original decision surfaces.

The authors are a scholarship holders of project entitled „Innowacyjna dydaktyka bez ograniczeń – zintegrowany rozwój Politechniki Łódzkiej – zarządzanie uczelnią, nowoczesna oferta edukacyjna i wzmacnianie zdolności do zatrudniania, także osób niepełnosprawnych" supported by European Social Fund.

References

[1] Duda R.O., Hart P.E., *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York City, NY, 1973.
[2] Michalski R.S., Stepp R.E., Diday E., *A recent advance in data analysis: clustering objects into classes characterized by conjuctive concepts*. W: L.N. Kanal & A. Rozenfeld (eds.), Progress in Pattern Recognition, Vol. 1, New York: North-Holland, USA, 1981, 33–56.
[3] NIST SP800-94, *Guide to Intrusion Detection and Prevention Systems.* (IDPS) http://csrc.nist.gov/publications/nistpubs/800-94/SP800-94.pdf, 2007.
[4] RFC 1028 – *Simple Gateway Monitoring Protocol* http://tools.ietf.org/pdf/rfc1028.pdf.
[5] RFC 1067 – *A Simple Network Management Protocol* http://tools.ietf.org/pdf/rfc1067.pdf.
[6] RFC 3917 – *Requirements for IP Flow Information Export (IPFIX)* http://tools.ietf.org/pdf/rfc3917.pdf.
[7] Stallings W., *SNMP, SNMPv2 and RMON*. Addison-Wesley Publishing Company, Inc, 1996.