Detecting Denial-of-Service And Network Probe Attacks Using Principal Component Analysis

Khaled Labib and V. Rao Vemuri

Department of Applied Science
University of California, Davis
U.S.A.
kmlabib@ucdavis.edu and rvemuri@ucdavis.edu

Abstract

In this paper, an analysis of a method proposed for anomaly detection is presented. The method uses a multivariate statistical method called Principal Component Analysis to detect selected Denial-of-Service and network Probe attacks using the 1998 DARPA Intrusion Detection data set. The Principal Components are calculated for both attack and normal traffic, and the loading values of the various feature vector components are analyzed with respect to the Principal Components. The variance and standard deviation of the Principal Components are calculated and analyzed. A method for identifying an attack based on the Principal Component Analysis results is proposed. After presenting related work in the field of intrusion detection using multivariate analysis methods, the paper introduces Denial-of-Service and network Probe attacks and describes their nature. A brief introduction to Principal Component Analysis and the merits of using it for detecting intrusions are presented. The paper describes the approach used to collect the various statistics and how the data sets are created and used. The results obtained using a proposed criterion for detecting the selected intrusions are discussed. The criterion can yield 100% detection rate. Finally, the paper presents the conclusion of the work done and proposes future enhancements to the current method.

I. Introduction

With the growing rate of interconnections among computer systems, network security is becoming a major challenge. In order to meet this challenge, Intrusion Detection Systems (IDS) are being designed to protect the availability, confidentiality and integrity of critical networked information systems. Automated detection and immediate reporting of intrusion events are required in order to provide a timely response to attacks.

Early in the research into IDS, two major approaches known as anomaly detection and signature detection were arrived at. The former relies on flagging behaviors that are abnormal and the later flagging behaviors that are close to some previously defined pattern signature of a known intrusion [1]. This paper describes a network-based anomaly detection method for detecting Denial of Service and network Probe attacks.

The detection of intrusions or system abuses presupposes the existence of a model [2]. In signature detection, also referred to as misuse detection, the known attack patterns are modeled through the construction of a library of attack signatures. Incoming patterns that match an element of the library are labeled as attacks. If only exact matching is allowed, misuse detectors operate with no false alarms. By allowing some tolerance in attack matching, there is a risk of false alarms, but the detector is expected to be able to detect certain classes of unknown attacks that do not deviate much from the attacks listed in the library. Such attacks are called neighboring attacks.

In anomaly detection, the normal behavior of the system is modeled. Incoming patterns that deviate substantially from normal behavior are labeled as attacks. The premise that malicious activity is a subset of anomalous activity implies that the abnormal patterns can be utilized to indicate attacks. The presence of false alarms is expected in this case in exchange for the hope of detecting unknown attacks, which may be substantially different from neighboring attacks. These are called novel attacks.

Detecting novel attacks while keeping acceptably low rates of false alarm, is possibly the most challenging and important problem in Intrusion Detection.

IDSs may also be characterized by scope, as either network-based or host-based. The key difference between network-based and host-based IDSs is that a network-based IDS, although run on a single host, is responsible for an entire network, or some network segment, while a host-based IDS is only responsible for the host on which it resides [3].

In this paper, a method for detecting selected types of network intrusions is presented. The selected intrusions represent two classes of attacks; namely Denial of Service attacks and network Probe attacks. The method uses Principal Components Analysis to reduce the dimensionality of the feature vectors to enable better visualization and analysis of the data. The data for both normal and attack types are extracted from the 1998 DARPA Intrusion Detection Evaluation data sets [4]. Portions of the data sets are processed to create a new database of feature vectors. These feature vectors represent the Internet Protocol (IP) header of the packets. The feature vectors are analyzed using Principal Component Analysis and various statistics are generated during this process, including the principal components, their standard deviations, the loading of each feature on the principal components and Bi-plots to represent a graphical summary of these statistics. Based on the generated statistics, a method is proposed to detect intrusions with relatively low false alarm rates.

The rest of the paper is organized as follows: Section II discusses related work in intrusion detection using multivariate statistical approaches with emphasis on those using Principal Component Analysis. Section III provides an introduction to Principal Component Analysis and its applicability to the field of intrusion detection. Section IV describes Denial of Service and network Probe attacks with emphasis on the attacks selected for this study. Section V details the process of data collection and preprocessing and the creation of feature vectors. It also describes how the various statistics are generated using Principal Component Analysis results. Section VI discusses the results obtained using this method and suggests a method of detecting intrusions using these results. False alarm rates are also discussed here. Finally, Section VII provides a conclusion of the work presented in this paper and recommendations for future work.

II. Related Work

IDS research has been ongoing for the past 15 years producing a number of viable systems, some of which have become profitable commercial ventures [5].

There are a number of research projects that focus on using statistical approaches for anomaly detection.

Ye et al [6], [7] discuss probabilistic techniques of intrusion detection, including decision tree, Hotelling's T^2 test, chi-square multivariate test and Markov Chains. These tests are applied to audit data to investigate its frequency property and its ordering property.

Taylor et al [8], [9] present a method for detecting network intrusions that addresses the problem of monitoring high speed network traffic and the time constraints on administrators for managing network security. They use multivariate statistics techniques, namely, Cluster Analysis and Principal Component Analysis to find groups in the observed data.

DuMouchel et al [10] discuss a method for detecting unauthorized users masquerading as a registered user by comparing in real time the sequence of commands given by each user to a profile of the user's past behavior. They use a Principal Component Regression model to reduce the dimensionality of the test statistics.

Staniford-Chen et al [11] address the problem of tracing intruders who obscure their identity by logging through a chain of multiple machines. They use Principal Component Analysis to infer the best choice of thumbprinting parameters from data. They introduce thumbprints, which are short summaries of the content of a connection.

Shah et al [3] study how fuzzy data mining concepts can cooperate in synergy to perform Distributed Intrusion Detection. They describe attacks using a semantically rich language, reason over them and subsequently classify them as instances of an attack of a specific type. They use Principal Component Analysis to reduce the dimensionality of the collected data.

III. Principal Component Analysis

Principal component analysis (PCA) [12] is a well-established technique for dimensionality reduction and multivariate analysis. Examples of its many applications include data compression, image processing, visualization, exploratory data analysis, pattern recognition, and time series prediction. A complete discussion of PCA can be found in several textbooks [13], [14]. The popularity of PCA comes from three important properties. First, it is the optimal (in terms of mean squared error) linear scheme for compressing a set of high dimensional vectors into a set of lower dimensional vectors and then reconstructing it. Second, the model parameters can be computed directly from the data - for example by diagonalizing the sample covariance. Third, compression and decompression are easy operations to perform given the model parameters - they require only matrix multiplication.

Multi-dimensional hyper-space is often difficult to visualize, and thus the main objectives of unsupervised learning methods are to reduce dimensionality, scoring all observations based on a composite index and clustering similar observations together based on multi-attributes. Summarizing multivariate attributes by, two or three variables that can be displayed graphically with minimal loss of information is useful in knowledge discovery. Because it is hard to visualize multi-dimensional space, PCA is mainly used to reduce the dimensionality of *d* multi-attributes to two or three dimensions.

PCA summarizes the variation in a correlated multi-attribute to a set of non-correlated components, each of which is a particular linear combination of the original variables. The extracted non-correlated components are called Principal Components (PC) and are estimated from the eigenvectors of the covariance or correlation matrix of the original variables. Therefore, the objective of PCA is to achieve parsimony and reduce dimensionality by extracting the smallest number components that account for most of the variation in the original multivariate data and to summarize the data with little loss of information.

In PCA, the extractions of PC can be made using either original multivariate data set or using the covariance or the correlation matrix if the original data set is not available. In deriving PC, the correlation matrix is commonly used when different variables in the data set are measured using different units or if different variables have different variances. Using the correlation matrix is equivalent to standardizing the variables to zero mean and unit standard deviation.

The PCA model can be represented by:

$$u = Wx$$

Where u is the m-dimensional projected vector and x is the original d-dimensional data vector where $m << d$.

It can be shown that the m projection vectors that maximize the variance of u, called the principal axes, are given by the eigenvectors $e_1, e_2, \ldots, e_m$ of the data set's covariance matrix C, corresponding to the m largest non-zero eigenvalues $\lambda_1, \lambda_2, \ldots \lambda_m$.

The data set's covariance matrix S can be found as:

$$S = \frac{1}{n-1}\sum_{i=1}^{n}(x-\mu)(x-\mu)^T$$

and the eigenvectors can be found by solving the set of equations:

$$(S - \lambda_i I)e_i = 0 \qquad i = 1,2,...,d$$

After calculating the eigenvectors, they are sorted by their corresponding eigenvalues and choosing the m vectors with the largest eigenvalues. The PCA projection matrix is then calculated as:

$$W = E^T$$

Where E has the eigenvectors as its columns.

One of the motives behind the selection of Principal Component Analysis for the detection of network traffic anomalies is its ability to operate on the input feature vector's space directly without the need to transform the data into another output space as in the case of other self-learning techniques. For example, in Self-Organizing Maps, the transformation of a high-

dimensional input space to a low-dimensional output space takes place through the iterative process of training the map and adjusting the weight vectors. The weight vectors are typically selected randomly which makes the process of selecting the best initial weight vectors a trial-and-error one. While in Principal Component Analysis, dimensionality reduction is achieved by calculating the first few principle components representing the highest variance in the components of the input feature vector, without the need to perform any transformations on the input space. Thereby, the input data is analyzed within its own input space, and the results of the transformations are deterministic and do not rely on initial conditions.

IV. Denial of Service and Probe Attacks

In a Denial of Service (DoS) attack, the attacker makes some computing or memory resource too busy, or too full, to handle legitimate users' requests. But before an attacker launches an attack on a given site, the attacker typically probes the victim's network or host by searching these networks and hosts for open ports. This is done using a sweeping process across the different hosts on a network and within a single host for services that are up by probing the open ports. This is referred to as Probe attacks.

Table 1 summarizes the types of attacks used in this study.

Attack Name	Attack Description
Smurf	Denial of Service ICMP echo reply flood
Portsweep	Surveillance sweep through many ports to determine which services are supported on a single host
Neptune	SYN flood Denial of Service on one or more ports
IPsweep	Surveillance sweep performing either a port sweep or ping on multiple host addresses

Table 1 : Description of DoS and Probe attacks

Smurf attacks, also known as directed broadcast attacks, are a popular form of DoS packet floods. Smurf attacks rely on directed broadcast to create a flood of traffic for a victim. The attacker sends a ping packet to the broadcast address for some network on the Internet that will accept and respond to directed broadcast messages, known as the Smurf amplifier. The attacker uses a spoofed source address of the victim. If there are 30 hosts connected to the Smurf amplifier, the attacker can cause 30 packets to be sent to the victim by sending a single packet to the Smurf amplifier [15].

Neptune attacks can make memory resources too full for a victim by sending a TCP packet requesting to initiate a TCP session. This packet is part of a three-way handshake that is needed to establish a TCP connection between two hosts. The SYN flag on this packet is set to indicate that a new connection is to be established. This packet includes a spoofed source address, such that the victim is not able to finish the handshake but had allocated an amount of system memory for this connection. After sending many of these packets, the victim eventually runs out of memory resources.

IPsweep and Portsweep, as their names suggest, sweep through IP addresses and port numbers for a victim network and host respectively looking for open ports, that could potentially be used later in an attack.

V. Data Collection and Preprocessing

The 1998 DARPA Intrusion Detection data sets were used as the source of all traffic patterns in this study. The training data set includes traffic collected over a period of seven weeks and contains traces of many types of network attacks as well as normal network traffic.

This data set has been widely used in the research in Intrusion Detection, and has been used in comparative evaluation of many IDSs. McHugh [16] presents a critical review of the design and execution of this data set.

Approach

Attack traces were identified using the time stamps published on the DARPA project web site. Data sets were preprocessed to create feature vectors that were used to extract the principal components and other statistics. The feature vector chosen has the following format:

SIPx	SPort	DIPx	DPort	Prot	PLen

Where

- SIPx = Source IP address nibble, where x = [1-4]. Four nibbles constitute the full source IP address
- SPort = Source Port number
- DIPx = Destination IP address nibble, where x = [1-4]. Four nibbles constitute the full destination IP address
- DPort = Destination Port number
- Prot = Protocol type: TCP, UDP or ICMP
- PLen = Packet length in bytes

This format represents the IP packet header information. Each feature vector has 12 components. The IP source and destination addresses are broken down to their network and host addresses to enable the analysis of all types of network addresses.

Seven data sets were created, each containing 300 feature vectors as described above. Four data sets represented the four different attack types indicated in Table 1. The three remaining data sets represent different portions of normal network traffic across different weeks of the DARPA Data Sets. This allows for variations of normal traffic to be accounted for in the experiment.

One of the motives of creating smaller data sets for representing the feature vectors is to later enable studying the effectiveness of this method for real-time applications. Real-time processing of network traffic mandates the creation of small sized databases that are dynamically created from real-time traffic presented at the network interface.

Principal Component Analysis was performed on all data sets where each feature vector would be represented by its 12 components. An exploratory analysis and statistical modeling tool called S-Plus [17] was used to generate the required statistics for this study. The following statistics were generated for each data set:

- Standard Deviation for each component
- Proportion of variance for each component
- Cumulative proportion of variance across all components
- Loading value of each feature on all individual components
- A Bi-Plot representing the loading of the different features on the first and second components

VI. Results

Figure 1 shows the loading and variance of the first and second principal components for all data sets. Normal 1, 2 and 3 represent 3 randomly chosen data sets from normal traffic. IPsweep, Neptune, Portsweep and Smurf represent data sets for these attacks.

The principal component loadings are the coefficients of the principal components transformation. They provide a convenient summary of the influence of the original variables on the principal components, and thus a useful basis for interpretation of data. A large coefficient (in absolute value) corresponds to a high loading, while a coefficient near zero has a low loading [18].

The variance and standard deviation of a random variable are measures of dispersion. The variance is the average value of the squared deviation from the variable's mean, and the standard deviation is the square root of the variance.

If X is a discrete random variable with density function $f_X(x)$ and mean μ_X, the variance σ^2 is given by the weighted sum:

$$\sigma_X^2 = \sum_{i=1}^{n} (x_i - \mu_X)^2 f(x_i)$$

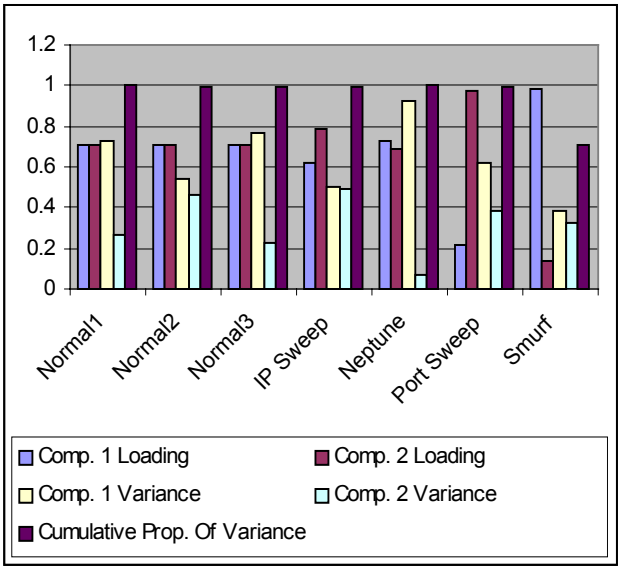

Figure 1: Component Loading and Variance

In the results above, the first 2 principal components consistently had their highest absolute value loading from SPort and DPort features across all data sets. This reflects the high variance in both source and destination port numbers for all data sets, except for Smurf at which the highest variance was due to source IP address components. Port numbers in TCP connections vary from 0 to 65534 and represent the different network services offered within the TCP protocol.

It is observed that the loading values for the first and second principal components in the three normal data sets are equal, with a value of 0.7. This represents the balance in variance in the packets flowing between a client and a server with respect to the source and destination ports. In TCP, the data and acknowledgement packets regularly flow between the client and the server, each using a designated TCP port number for the duration of the session.

For the four attack data sets, it is observed that the loading values for the first and second principal components are not equal, possibly representing the imbalance in variance in the packets flowing between a client and a server with respect to the source and destination port numbers.

In IPsweep attacks, one or more machines (IPs) are sweeping through a list of server machines looking for open ports that can later be utilized in an attack. While in Portsweep attacks, one machine is sweeping through all ports of a single server machine looking for open ports. In both cases, there is an irregular use of port numbers that causes the variance in the principle components to vary, with an associated irregularity in the loading values.

In Neptune attacks, a flood of SYN packets is sent to one or more ports of the server machine, but from many clients with, typically, non-existing (spoofed) IP addresses. The packets seen by the server appears to be coming from many different IP addresses with different source port numbers. This is represented by the irregularity in both loading and variance of the principal components.

In Smurf attacks, attackers utilize floods of Internet Control Message Protocol (ICMP) echo reply packets to attack a server. Using amplifying stations, attackers utilize broadcast addressing to amplify the attack. The packets seen by the server appear to be coming from many different IP addresses but to one source port. Therefore, 99% of the variance for this data set is represented by the first four principal components and has their loading values associated with SIP1, SIP2, SIP3 and SIP4, instead of the source and destination ports as in previous attacks.

Figure 2 shows the standard deviation for the first and second principal components for all data sets. In the case of IPsweep and Portsweep attacks, the standard deviation of both source and destination port numbers is almost similar. This is due to the similarity in utilizing source and destination port numbers in these attacks.

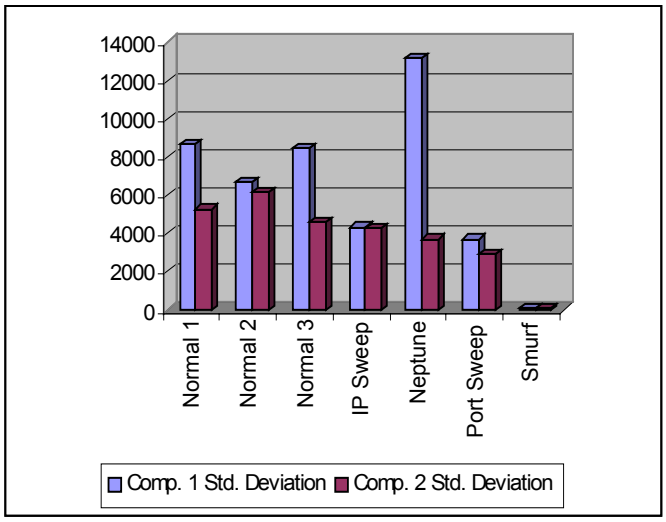

Figure 2: Standard Deviation Values for first 2 PCs

In Neptune attacks, the source and destination ports vary differently where the source port would have the highest variance. In Smurf attacks, the first two components, namely SIP1 and SIP2, represent only a portion of the variance and have a relatively small standard deviation value.

With these results, it is possible to use the loading values of the features on the first and second principal components to identify an attack. For normal traffic, loading values appear to be similar, while during an attack the loading values differ significantly for the first two principal components. A threshold value could

be used to make such a distinction. In addition, the decision could be further enhanced using the standard deviation values for first and second components. Whenever these values differ significantly, an additional data point could be obtained regarding the possibility of an attack.

Table 2 shows the results of a possible criterion C for the detection of an attack based on the loading values. This criterion is represented by the following equation:

$$C = abs\left(\left(l_1 - l_2\right)p_v * 100\right)$$

Where, l_1 and l_2 are the loading values for the first and second principal components, and p_v is the cumulative proportion of variance for the first and second principal components.

Attack Data Set	Comp. 1 Loading	Comp. 2 Loading	Cum. Prop. Of Variance	Attack Criteria
Normal 1	0.707	0.707	0.999	0.00
Normal 2	0.709	0.705	0.998	0.40
Normal 3	0.708	0.706	0.997	0.20
IP Sweep	0.617	0.787	0.998	16.97
Neptune	0.723	0.69	0.999	3.30
Port Sweep	0.221	0.974	0.998	75.15
Smurf	0.981	0.139	0.705	59.36

Table 2 : Attack Criteria calculation

If a threshold value of C = 1 is used given the above data sets, we could achieve a 100% detection rate using the selected criterion for detection.

In addition to the calculation of the attack criterion, Bi-Plots could be utilized to visually interpret the loading values of the principal components and to see which features had the highest loading on a given principal component value.

The Bi-Plot allows the representation of both the original variables and the transformed observations on the principal components axes. By showing the transformed observations, the data can be easily interpreted in terms of the principal components. By showing the variables, the relationships between those variables and the principal components can be viewed graphically.

Figure [3] shows two sample Bi-Plots generated for Normal 1 and Portsweep data sets.

Interpreting the Bi-Plot is straightforward: the x-axis represents the scores for the first principal component, the y-axis represents the scores for the second principal component. The original variables are represented by arrows, which graphically indicate the proportion of the original variance explained by the first two principal components. The direction of the arrows indicates the relative loadings on the first and second principal components.

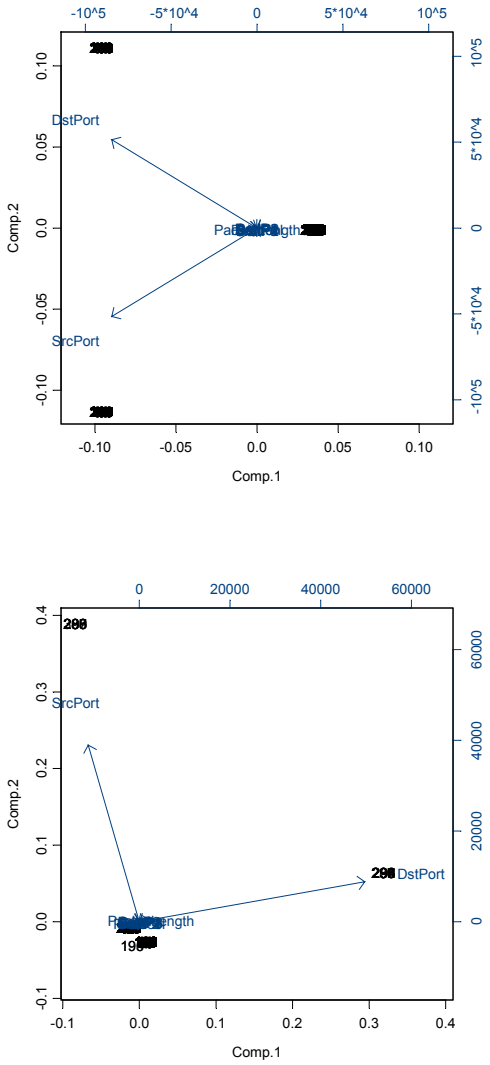

Figure 3: Bi-Plots for Normal 1 (top) and Portsweep (bottom) data sets

VII. Conclusion and Future Work

This paper presents a method for detecting Denial-of-Service attacks and network Probe attacks using Principal Component Analysis as a multivariate statistical tool. The paper described the nature of these attacks, introduced Principal Component Analysis and described the merits of using it for detecting intrusions. The paper described the approach used to extract the Principal Components and the related statistics. It also discussed the results obtained from using a proposed criterion for detecting the subject intrusions. This criterion can lend 100% detection rate. The paper presented a graphical method for interpreting the above results based on the Bi-Plots. Future work includes testing this model to work in a real-time environment at which network traffic is collected, processed and analyzed for intrusions dynamically. This may involve using a more comprehensive criterion that accounts for other statistics including standard deviation values of the Principal Components. In addition, an enhancement may be added to utilize Bi-Plots for visual interpretation of data in real-time. In this case the entire DARPA data sets will be used to qualify the results.

References

1. Axelsson S., "Intrusion Detection Systems: A Survey and Taxonomy". Technical report 99-15, Dept. of Computer Engineering, Chalmers University of Technology, Goteborg, Sweden, March 2000.
2. Cabrera J., Ravichandran B., Mehra R., "Statistical Traffic Modeling for Network Intrusion Detection"
3. Shah H., Undercoffer J., Joshi A., "Fuzzy Clustering for Intrusion Detection"
4. DARPA Intrusion Detection Evaluation Project: http://www.ll.mit.edu/IST/ideval/
5. Allen J. et al, "State of The Practice: Intrusion Detection Technologies". Carnegei Mellon, SEI, Tech. Report CMU/SEI-99-TR-028, ESC-99-028, January 2000
6. Ye N., Li X., Chen Q., Emran S., Xu M., "Probabilistic Techniques for Intrusion Detection Based on Computer Audit Data". IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems and Humans, Vol. 31, No. 4, July 2001
7. Ye N., Emran S., Chen Q., Vilbert S., "Multivariate Statistical Analysis of Audit Trails for Host-Based Intrusion Detection". IEEE Transactions on Computers, Vol. 51, No. 7, July 2002
8. Taylor C., Alves-Foss J., "NATE: Network Analysis of Anomalous Traffic Events, a low-cost approach". NSPW'01, September 10-13th, 2002, Cloudcroft, New Mexico, U.S.A.
9. Taylor C., Alves-Foss J., "An Empirical Analysis of NATE – Network Analysis of Anomalous Traffic Events". New Security Paradigms Workshop'02, September 23-26, 2002, Virginia Beach, Virginia.
10. DuMouchel W., Schonlau M., "A Comparison of Test Statistics for Computer Intrusion Detection Based on Principal Component Regression of Transition Probabilities".
11. Staniford-Chen S., Heberlein L.T., "Holding Intruders Accountable on the Internet".
12. Hotelling H., "Analysis of a Complex of Statistical Variables into Principal Components". Journal of Educational Psychology, 24:417–441, 1933.
13. Duda R., Hart P., Stork D., "Pattern Classification". Second Edition, John Wiley & Sons, Inc., 2001
14. Haykin S., "Neural Networks: A Comprehensive Foundation". Second Edition. Prentice Hall Inc., 1999
15. Skoudis E., "Counter Hack: A Step-by-Step Guide to Computer Attacks and Effective Defenses". Prentice Hall Inc., 2002
16. McHugh J., "Testing Intrusion Detection Systems: Critique of the 1998 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory". ACM Transactions on Information and System Security, Vol. 3, No. 4, November 2000, Pages 262-294
17. http://www.insightful.com/
18. S-Plus: Guide to Statistics, Volume 2. Insightful Corporation, 2001