



Identification of Effective Optimal Network Feature Set for Probing Attack Detection Using PCA Method



Peyman Kabiri¹, Gholam Reza Zargar²

¹School of Computer Engineering
Iran University of Science and Technology
Tehran, Iran
Peyman.Kabiri@iust.ac.ir

²Khozestan Electric Power Distributed Company
Ahwaz, Iran
Zargar@vu.iust.ac.ir

ABSTRACT: Existing intrusion detection techniques emphasize on building intrusion detection model based on all features provided. But not all the features are relevant ones and some of them are redundant and useless. This paper proposes and investigates identification of effective network features for Probing attack detection using PCA method to determine an optimal feature set. An appropriate feature set helps to build efficient decision model as well as a reduced feature set. Feature reduction will speed up the training and the testing process considerably. This paper proposes a strategy to focus on intrusion detection involving statistical analysis of both attack and normal traffics based on the DARPA 1998 dataset as the training data. DARPA 1998 dataset was also used in the experiments as the test data. Experimental results show a reduction in training and testing time while maintaining the detection accuracy within acceptable range.

Keywords: Intrusion detection, Principal components analysis, Data dimension reduction, Feature selection

Received: 19 January 2010, Revised 18 March 2010, Accepted 2 April 2010

© 2009 D-Line. All rights reserved

1. Introduction

Intrusion Detection System (IDS) deals with huge amount of data which contains irrelevant and redundant features. These two types of features will slow down training and testing processes, will cause higher resource consumption and reduced detection rate.

All feature selection methods need to use an evaluation function together with a search procedure to obtain the optimal feature set. The evaluation function measures how good a specific subset can be in discriminating between classes. These measures can be divided into two main groups: filters and wrappers. Filters measure the relevance of feature subsets independent of classifier type, whereas wrappers use the classifier's performance as the evaluation measure.

IDSs are typically classified as host-based or network-based. A host-based IDS will monitor resources such as system logs, file systems and disk resources; whereas a network-based IDS monitors the data passing through the network. Different detection techniques can be employed to search for attack patterns in the data monitored. Signature-based detection systems try to find attack signatures in the monitored resource. Anomaly detection systems typically rely on knowledge of normal behavior and flag any deviation from it [1]. Signature-based IDSs typically require human input to create attack signatures or to determine effective models for the normal behavior. Implementing the learning algorithms provide a potential alternative to the expensive human input.

Inefficient feature selection method not only reduces speed of its operation but also declines its accuracy [2]. The main goal in feature selection is to reduce the amount of data which are less important to the classification and can be eliminated. This has the benefit of decreasing storage requirements, reducing processing time and improving the detection rate. IDS has to examine a very large audit data in a short period of time. Therefore, any reduction in the volume of data may save the processing time [3]. This paper proposes a method based on TCP/IP Header parameter. In the proposed approach, Principal Component Analysis (PCA) is used as a dimension reduction technique. PCA is a statistical method applicable to feature selection. It is employed to reduce the high dimensionality in data and consequently improve the efficiency of the process and reduce usage of the system resources. In the proposed approach a reduced number of dimensions in the feature space is used for the detection of different types of attacks and normal activities. The low computational cost of this approach improves the real-time performance of the IDS.

2. Related Works

Data reduction can be achieved by filtering, data clustering and feature selection [4]. Generally, the capability of anomaly-based IDS is often hindered by its inability to accurately classify variation of normal behavior as an intrusion. Additionally, network traffic data is huge and it causes a prohibitively high overhead that often becomes a major problem for IDS [5]. Chakraborty in a published paper [6] mentions that, “the existence of these irrelevant and redundant features generally affects the performance of machine learning or pattern classification algorithms”. Hassan et al [7], proved that proper selection of feature set has resulted in better classification performance. Sung and Mukkamala [4], have exploited SVM [8] and Neural Networks to identify and categorize features with respect to their importance in regard to detection of specific kinds of attacks such as probe, DoS, Remote to Local (R2L), and User to Root (U2R). They have also demonstrated that the elimination of these unimportant and irrelevant features did not significantly reduce the performance of the IDS. Chebrolu et al [3], tackled the issue of effectiveness of an IDS in terms of real time performance and detection accuracy from the feature reduction perspective.

3. Problem of Irrelevant and Redundant Features

Basically, there are two approaches for designing IDS: Signature-based and Anomaly-based intrusion detection [9]. Signature-based intrusion detection is also called misuse detection. In principle, it is typically realized by modeling known attack behavior with prior understanding about specific attacks and system vulnerabilities. In this method, the IDS compares network traffic data versus well defined attack patterns to identify any possible penetrations into the system. Once an input pattern is detected to be similar to one of explicitly defined attack patterns in IDS, an alarm is raised. Defined attack patterns are frequently referred to as the signatures of intrusions. The signature could be a static string or a sequence of events.

While signature-based intrusion detection is achieved by modeling known attack behavior, on the contrary, anomaly-based intrusion detection models normal or expected behavior of computer users. It looks for malicious activities by comparing the observed data with these acceptable behaviors. If the data collected from the network traffic differs from the known normal behavior, an alarm is raised. In other words, anything will be suspected to be an attack if its behavior is deviated from the previously learned behaviors. However, there is one major problem in the collected network traffic database: problem of irrelevant and redundant features [2] Details of this problem are described in the following.

In general, the quantity of data processed by the IDS is large; it includes thousands of traffic records with a number of various features such as the length of the connection, type of the protocol, type of the network service and many other information. Theoretically and ideally, the ability to discriminate attacks from normal behavior should be improved if more features are used for the analysis. However, this assumption is not always true, since not every feature in the traffic data is relevant to the intrusion detection. Among the large amount of features, some of them may be irrelevant or confusing with respect to the target patterns. Some of the features might be redundant due to their high inter-correlation with one or more of the other features in the dataset [9]. To achieve a better overall detection performance, any irrelevant and redundant features should be discarded from the original feature space. Selecting a meaningful subset of features from network traffic data stream is therefore a very important and indispensable task at the early stages of an intrusion detection process.

4. Three Way Handshake

TCP needs to establish a connection between a source host and a destination host, before any data can be transmitted between them. The connection process is called the three-way handshake (see Figure 1). In the first step, a SYN packet is sent from

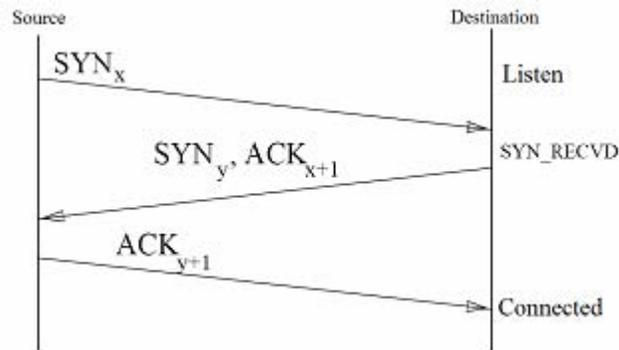


Figure 1. Three-way handshake

source node to the destination node. Then destination node sends a message to the source node with SYN and Ack Flags of it set. In the third step, source node sends a message with its ACK flag set to destination node. In this way, a connection between source node and destination node is established. The third message may contain user payload data.

5. Probing Attacks

Probing is a class of attacks in which an attacker scans a network of computers to collect information or find known vulnerabilities. An intruder with a map of machines and services that are available on a network can use this information to look for exploits.

There are different types of probes: some of them abuse the computer's legitimate features; and some of them use social engineering techniques. This class of attacks is the most common because it requires very little technical expertise. Worms and viruses may also execute probing attacks. Finally, horizontal (multiple hosts and same port number) and vertical (one machine, multiple port numbers) port scans are often performed by attackers as preliminary reconnaissance to identify a large number of vulnerable hosts in the Internet. This paper is focused on scalable detection of TCP scanning. Details of this type of attacks are described in the following.

Some examples of probing attacks are Ipsweep, Mscan, Nmap, Saint, Satan, ping-sweep and Portsweep attacks [10].

5.1 TCP SCAN

Hackers use TCP scans to identify active devices, TCP port status and their TCP-based application-layer protocols. TCP scans exploit either the TCP handshake process. Seven types of TCP scans are commonly used [10].

5.1.1 Vanilla TCP Connect scans (polite Scan)

Hackers use the Vanilla TCP connect scan to identify status of the ports on a victim machine. In this method, hackers use three-way handshake explained in section 4 and figure 1. During the Vanilla TCP connect scan, hacker sends the first packet with SYN flag set. The targeted machine sends a TCP reply with SYN and ACK flag set if the target port is open. If the target port is closed, then the target machine sends a TCP reply with the reset (RST) flag set. Finally, the hacker completes 3-way handshake process by sending a packet with ACK flag set [10].

5.1.2 TCP Half-Open Scan

Contrary to the Vanilla TCP connect scan, in TCP Half-open scan, hacker does not send the final ACK packet in the TCP three-way handshake. Therefore, the connection between source and target machines will stay open, so this method is similar to vanilla TCP connect to identify status of the port on a victim machine [10].

5.1.3 TCP FIN Scan and TCP SYN Scan

Hackers use these two methods to identify listening TCP port numbers based on how the target device reacts to a close transaction request for a given TCP port. In this method, hacker sends TCP packets with FIN or SYN flag set. The target device will send a TCP RST packet in reply if the TCP port in target device is close. The target device will discard the FIN and sends no reply if the TCP port in target device is open [10].

5.1.4 TCP XMAS Scan

Hackers use the TCP XMAS scan to identify listening TCP ports. This scan uses TCP packets with sequence number of 0 and Urgent (URG), Push (PSH) and FIN flags set. Some firewalls can detect this scan by checking the incoming TCP packets versus standard flag setting. The target device will send a TCP RST packet in reply if the target device's TCP port is close. The target will discard the TCP XMAS scan, send no reply if the target device's TCP port is open [10].

5.1.5 TCP NULL Scan

Hackers use the TCP NULL scan to identify listening TCP ports. In this scan TCP packets contain a sequence number of 0 and no flags. The target device will send a TCP RST packet in reply, if the target device's TCP port is close. The target will discard the TCP NULL scan, and send no reply if the target device's TCP port is open [10].

5.1.6 TCP ACK Scan

Hackers use the TCP ACK scan to identify listening TCP port. The TCP ACK scan uses TCP packets with the ACK flag and SYN flag set. This is the second step of 3-way handshaking process. If the target device is available and the port is open, the target device will send a TCP RST packet in reply. If TCP port is close, the target will discard the TCP ACK packet. Some filtering devices discard SYN flags received from TCP SYN scan, so some hackers exploit TCP ACK scan to identify the listening TCP port [10].

5.1.7. TCP SYN/FIN with Fragments Scan

Some hackers use the TCP SYN/FIN packet with fragments scan to bypass a filtering device. Hackers fragment a packet inside the TCP header. Unless the filtering device reassembles the packet, this device will not know that the incoming packet is a TCP SYN/FIN packet [10].

6. Data Reduction and feature selection using PCA

PCA is one of the most widely used dimensionality reduction techniques for data analysis and compression. It is based on transforming a relatively large number of variables into a smaller number of uncorrelated variables. This transformation is carried out by finding those orthogonal linear combinations of the original variables with the largest variance. The first principal component of the transformation is the linear combination of the original variables with the largest variance. The second principal component is the linear combination of the original variables with the second largest variance and orthogonal to the first principal component and so on. In many datasets, the first several principal components have the highest contribution to the variance in the original dataset. Therefore, the rest can be ignored with minimal loss of the information value during the dimension reduction process [11]. The transformation works as follows:

Given a set of observations x_1, x_2, \dots, x_n , where each observation is represented by a vector of length m , the dataset is thus represented by a matrix X

$$X_{n \times m} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1m} \\ x_{21} & x_{22} & \dots & x_{2m} \\ \dots & \dots & \dots & \dots \\ x_{n1} & x_{n2} & \dots & x_{nm} \end{bmatrix} = [x_1, x_2, \dots, x_n] \quad (1)$$

The average for each observation is defined by the equation (2).

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i \quad (2)$$

The deviation from the mean is defined in (3).

$$\phi_i = x_i - \mu \quad (3)$$

The sample covariance matrix of the dataset is defined in (4)

$$C = \frac{1}{n} \sum_{i=1}^n (x_i - \mu)(x_i - \mu)^T = \frac{1}{n} \sum_{i=1}^n \phi_i \phi_i^T = \frac{1}{n} A A^T \quad (4)$$

Where $A = [\phi_1, \phi_2, \dots, \phi_n]$

In applying PCA to reduce high dimensional data, eigenvalues and corresponding eigenvectors of the sample covariance matrix C have to be calculated [12]. Let $(\lambda_1, u_1), (\lambda_2, u_2), \dots, (\lambda_m, u_m)$ present m eigenvalue eigenvector pairs of the sample covariance matrix C . The k eigenvectors associated with the largest eigenvalues are selected. The dimensionality of the subspace k can be determined by the following equation [13]:

$$m \times k$$

The resulted matrix U , with k eigenvectors as its columns is called eigenvectors matrix or coefficient TCP/IP sampled dataset matrix. Data transformation using principal components into the k -dimensional subspace is carried-out using equation (5).

$$y_i = U^T (x - \mu) = U^T \phi_i \quad (5)$$

7. The Dataset and Pre-processing

In the following subsections the dataset used in the reported work and the pre-processing applied on the dataset are presented.

8. The Dataset used in this work

In 1998, under DARPA intrusion detection evaluation program, an environment was set up to acquire raw TCP/IP dump data for a network by simulating a typical US Air Force LAN. The LAN was operated like a real environment, but was blasted with multiple attacks.

In this work, basic features are extracted from TCP/IP packets [14]. These features can be derived from packet headers without inspecting the payload. In the reported work, TCP dump from the DARPA'98 dataset is used as the input dataset. Extracting the basic features, packet information in the TCP dump file is summarized into connections. Specifically, "a connection is a sequence of TCP packets starting and ending at some well defined times, between which data flows from a source IP address to a target IP address under some well defined protocol" [15].

DARPA'98 dataset provides around 4 gigabytes of compressed TCP dump data [16] for 7 weeks of network traffic [17]. This dataset can be processed into about 5 millions of connection records each about 100 bytes in size. The resulted dataset contains the payload of the packets transmitted between hosts inside and outside a simulated military base. BSM audit data from one UNIX Solaris host for some network sessions were also provided. DARPA 1998 TCP dump Dataset [17] was preprocessed and labeled with two class labels, e.g. normal and attack. The dataset contains different types of attacks. Satan and Portswep from Probing attack category are extracted. Probing is a class of attacks in which an attacker scans a network of computers to collect information or find known vulnerabilities. An intruder with a map of machines and services that are available on a network can use this information to look for exploits. Examples for this type of attack include Ipsweep, Mscan, Nmap, Saint, Satan, ping-sweep and Port sweep attacks.

9. Pre-processing

As displayed in table 1, 32 basic features are extracted from TCP, IP, UDP and ICMP protocols. Wireshark and Editcap softwares are used to analyze and minimize TCP dump files [14][18]. Finally, a Visual Basic program extracts the 32 basic features. In this work, intention is to reduce the processing and data transfer time needed for the intrusion detection. To do so, an accurate feature selection scheme is proposed to select important features with minimum loss of information. Work also aims to select features in such a way that their discrimination set to be categorical. This means that the selection criteria will be the same or with a low variance for the attacks in the same category. This property will increase the adaptability of the IDS that is using this feature set to the variation of the attack patterns that fall in the same category.

10. Experiments

Training data from the DARPA dataset includes "list files" that identify the timestamp, source host and port, destination host and port, and the name of each attack [19]. This information is used to select intrusion data for the purpose of pattern mining and feature construction, and to label each connection record with "normal" or "attack" label types. The final labeled training data is used for training the classifiers. Due to the large volume of audit data, connection records are stored in several data files. The traffic data on Thursday of the 6th week was selected as the sampled dataset, since it includes not only normal behaviors, but also a large number of Satan and Port sweep attacks that are types of probing attack activities. Table 2 shows number of

the connection records that are randomly extracted from the sequences of normal and probing attack connection records to create the normal dataset and attack dataset. Together they make the sampled dataset for the data analysis using PCA.

Dictionary table is used to convert text data into numeric data. Matlab software is used in this work for the calculations mentioned in section 4.

No.	Feature	Description
1	Protocol	Type of Protocol
2	Frame_lenght	Length of Frame
3	Capture_lenght	Length of Capture
4	Frame_IS_marked	Frame IS Marked
5	Coloring_rule_name	Coloring Rule name
6	Ethernet_type	Type of Ethernet Protocol
7	Ver_IP	IP Version
8	Header_lenght_IP	IP Header length
9	Differentiated_S	Differentiated Service
10	IP_Total_Lenght	IP total length
11	Identification_IP	Identification IP
12	MF_Flag_IP	More Fragment flag
13	DF_Flag_IP	Don't Fragment flag
14	Fragmentation_offset_IP	Fragmentation offset IP
15	Time_to_live_IP	Time to live IP
16	Protocol_no	Protocol number
17	Src_port	Source Port
18	Dst_port	Destination port
19	Stream_index	Stream Index number
20	Sequence_number	Sequence number
21	Ack_number	Acknowledgment number
22	Cwr_flag	Cwr Flag(status flag of the connection)
23	Ecn_echo_flag	Ecn Echo flag (status flag of the connection)
24	Urgent_flag	Urgent flag(status flag of the connection)
25	Ack_flag	Acknowledgment flag(status flag of the connection)
26	Psh_flag	push flag(status flag of the connection)
27	Rst_flag	Reset flag(status flag of the connection)
28	Syn_flag	Syn flag (status flag of the connection)
29	Fin_flag	Finish flag(status flag of the connection)
30	ICMP_Type	specifies the format of the ICMP message such as: (8=echo request and 0=echo reply)
31	ICMP_code	Further qualifies the

Table 1. Basic feature extracted from the header of TCP/IP

Category	Number of Records
PROBING ATTACK	10137
NORMAL	88860

Table 2. Number of records selected for the sampled dataset

Class name S Air For	Important features in descending order ce LAN. The LAN was ope	Total accumulated information value rated like a real ummarized into
Probing typical U	29in the TCP dump file is s	%39.03
Probing	29,26	%50.30
Probing	29,26,25	%60.89
Probing	29,26,25,28	%70.97
Probing	29,26,25,28,12	%79.87
Probing	29,26,25,28,12,13	%88.76
ProbingSolaris h	29,26,25,28,12,13,5ost for some network sessions were also	%92.84
Probing	29,26,25,28,12,13,5,27	%95.68
Probing	29,26,25,28,12,13,5,27,1	%97.04
Probingder with aAn intru	29,26,25,28,12,13,5,27,1, 10map of machines and s	%98.01ervices that are
Normalsweep attacks.	27	%98.22
Normal	27,25	%98.84

Table 3. List of the relevant features for the Probing attack and the normal state of the network

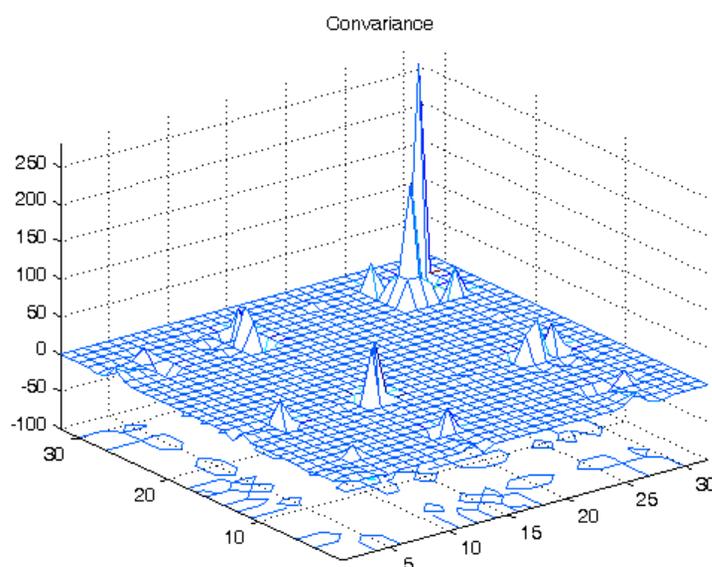


Figure 2. Covariance matrix

11. Experimental Results

Figure 2, shows correlation between features, the correlation is mainly between features 29 and 26. As the amount of correlation between the two parameters is increased, the relation between parameters and their behavioral similarity is more justifiable. Analyzing their correlation requires comparing observations on different parameters, such as source and destination IP address, an identifiable network route, commands entered by a suspected attacker and the time when activity began or ended.

Each behavior is constructed with some parameters. For example, in every TCP session, the “protocol” field of IP packet is equal to “6”. As another example, identification field in IP layer for each message have the same value. In a TCP Flood attack, and probing attacks, in addition to some other features, there are lots of TCP packets with “SYN=1”. But an ICMP flood attack never has these specifications. In fact, each attack has its own properties that are different from the properties of other attacks and even with the normal behavior. These features are used to compare each session versus the normal or a known attack behavior.

In Probing attack as explained in section 5, the attacker scans a network of computers implementing the three way handshaking method in TCP connection to collect information for detecting which port is open with TCP/IP vulnerabilities. As described in section 4, three way handshaking method in TCP connection uses SYN flag, ACK flag, FIN flag, and RST flag. As it is shown in Table 1, these flags are feature numbers 28, 25, 29 and 27. The result is reported in Table 3. Figures 2, shows the information value of the features with respect to their variances. Table 3 shows classes of the relevant features with their associated information value. In this table, probing attack is compared versus normal state of operation. As it is shown in Table 3, different feature sets are selected for the probing and the normal state. Component's 29, 26, 25 and 28, all TCP flags, are more effective and have more information value than the rest of the features. These components are fin flag, psh flag, Ack flag and syn flag respectively as they are presented in Table 1. As calculated by the PCA, their total information value is %70.97 of the total information in the feature set. If components 12, 13, 5, 27, 1 and 10 are added to them, the total information value will climb to %98.01.

Therefore, comparing results of these experiments and probing attack scenarios, intrusion detection systems are able to detect probing attack. In the normal dataset, component number 27 has the maximum information value with %98.22 of information. Once the component number 25 is considered as well, their total information value will rise to %98.84 of the total information value. Therefore, it can be said that the component number 25 does not have a significant effect in detecting the normal state. Figure 3, compares components of normal state and probing attack. The vertical axis in this figure shows absolute value of the eigenvalues.

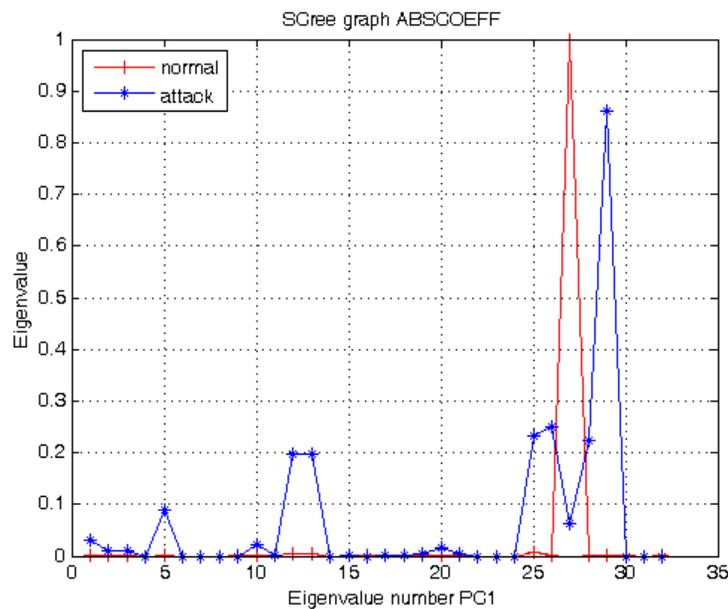


Figure 3. A comparison between information value in the feature space for the Probing attack and the normal state of the network

Figure 4 explains the method for the selection of the component 27 to detect the normal state and components 29, 26, 28 for detecting the probing attack. The SCREE graph for the PCA coefficient for probing attack is depicted in Figure 5.

In order to use these features, one should first calculate a threshold value (or range) for the features. Once the weights of the features are calculated, feature values should be multiplied by these weight values. As for example, the feature number 27 multiplied by its corresponding weight should be within the selected range or the state of the network is anomalous. Table 4, shows components of several attack scenarios as described in section 4, compared versus each other using effective parameters reported in this paper.

SCAN TYPE	Vanilla TCP Connect scans	TCP Half-Open scans	TCP FIN scans	TCP XMAS scans	TCPACK scans	TCP SYN/FIN with Fragments scan	UDP Port scans	ICMP Echo (Ping) scans
Feature number in this paper	28,25	28	29,25	26,29	25	28,29	10	1

Table 4. List of the effective features in detecting different probing attacks

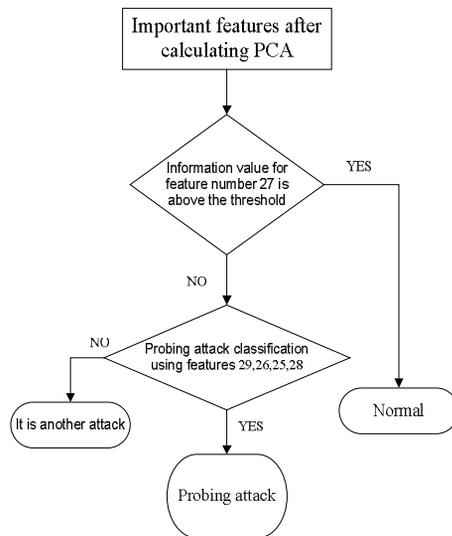


Figure 4. Flowchart for detecting probing attack

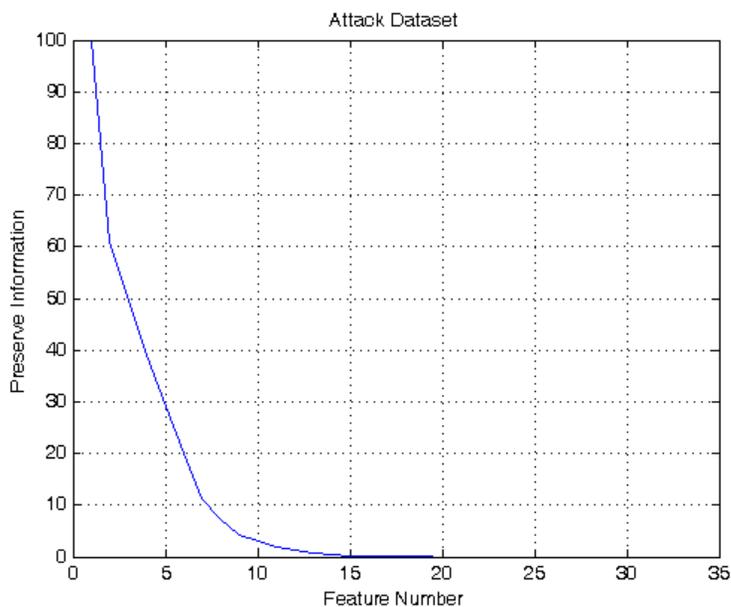


Figure 5. Scree graph for the PCA coefficients calculated using the sampled dataset

12. Conclusions

Results presented in this paper show that normal state of the network and the probing attack features can be selected to be discriminate. On the other hand, it is proven that certain features have no contribution to intrusion detection. This also indicates that there are analytical solutions for the feature selection that are not based on the trial and error.

Results from the experiments show that the 29th component in Table 1, i.e. Fin flag, has the highest information value. Hence, it is the most important component in this scenario for the detection purpose. As explained in section 5, hackers use the TCP FIN scan to identify the listening TCP port numbers. The TCP packets used in this scan have only their TCP FIN flag set. Comparing affects of this scenario on the TCP flag set, with TCP FIN scan, intrusion attempt by probing attack can be

detected. As reported in Table 4, in all of the previously explained probing attacks, hackers use TCP flags such as fin, Syn, Ack and psh to scan for any listening TCP port number.

Experimental results show that dimension reduction and identification of effective network features for this attack can reduce process time in intrusion detection system while maintaining the detection accuracy within acceptable range.

13. Future work

Plan for the future work is to calculate PCA for other attack categories and find feature sets for different attack categories. Later on, intention is to use classification methods to detect intrusions. Using the results derived from the intrusion detection and comparing it for the full feature set and the reduced feature set, one can analyze the different in accuracy and speed between them.

Acknowledgement

This work is carried out in the Intelligent Automation Laboratory, Department of Computer Engineering, Iran University of Science and Technology.

References

- [1] Guyon, I., Elisseeff, A (2003). An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research*, 3, 1157-1182.
- [2] Chou, T.S., Yen, K.K., and Luo, J (2008). Network Intrusion Detection Design Using Feature Selection of Soft Computing Paradigms. *International Journal of Computational Intelligence*, 4(3), 196-208.
- [3] Chebrolu, S., Abraham, A., and Thomas, J (2005). Feature Deduction and Ensemble Design of Intrusion Detection Systems, *Computers and Security*, Elsevier Science, 24(4) 295-307.
- [4] Sung, A.H., Mukkamala, S (2003). Identifying important features for intrusion detection using support vector machines and neural networks, In: *Proc. of the International Symposium on Applications and the Internet (SAINT)*, pages 209-216.
- [5] Sung, A.H., Mukkamala, S. (2004). The Feature Selection and Intrusion Detection Problems, LNCS, Springer Hiedelberg, 3321, 468-482.
- [6] Chakraborty, B (2005). Feature Subset Selection by Neurorough Hybridization LNCS, Springer Hiedelberg, 519-526.
- [7] Hassan, A., Nabi Baksh, M.S., Shaharoun, A.M., and Jamaluddin, H (2003). Improved SPC Chart Pattern Recognition Using Statistical Feature, *International Journal of Production Research*, 41(7), 1587-1603.
- [8] Vapnik, V (1995). *The Nature of Statistical Learning Theory* Springer, Berlin Heidelberg, New York.
- [9] Sabahi, F., Movaghar, A (2008). Intrusion Detection: A Survey, In: *Proc. of 3rd International conference on system and network communication*, (ICSNC08), pages.23-26.
- [10] <http://cisx1.uma.maine.edu/~wbackman/cis341/resources/scanning-explained.html> as visited on 19 August 2009.
- [11] Wang, W., Battiti, R (2009). Identifying Intrusions in Computer Networks based on Principal Component Analysis, <http://eprints.bibli.unitn.it/archive/00000917> as visited on 30 May.
- [12] Golub, G.H., Van Loan, C.F. (1996). *Matrix Computation*, Johns Hopkins Univ. Press, Baltimore.
- [13] Jolliffe, I.T. (2002) *Principal Component Analysis* 2nd Ed., Springer-Verlag, NY.
- [14] <http://www.wireshark.org> as visited on 29 Jan 2009.
- [15] Knowledge discovery in databases DARPA archive, Task Description. <http://www.kddics.uci.edu/databases/kddcup99/task.html>
- [16] <http://www.Tcpdump.org> as visited on 28 Jan 2009.
- [17] MIT Lincoln Laboratory <http://www.ll.mit.edu/IST/ideval/> as visited on 27 Jan 2009.
- [18] <http://www.wireshark.org/docs/man-ages/editcap.html> as visited on 20 Jan 2009.
- [19] Wenke Lee, (1999). A Data Mining Framework for Constructing Feature and Model for Intrusion Detection System, PhD thesis University of Columbia.

Authors Biographies



Gholam Reza Zargar received his B.Sc. in Computer Hardware Engineering from Iran's University of Science and Technology in 1991 and his M.Sc. in GIS (Geographic Information Systems) from Shahid Chamran University in 2008. He is currently a M.Sc. at the Iran's University of Science and Technology. He has 15 years of work experience in Khozestan water & power authority Co. IT department (a subsidiary of Iranian ministry of energy). His main research area is in computer and network security.



Peyman Kabiri received his PhD in Computing and MSc in Real time Systems from the Nottingham Trent University, Nottingham-UK in years 2000 and 1996 respectively. He received his B.Eng. in Computer Hardware Engineering from Iran's University of Science and Technology, Tehran-Iran in 1992. He was with the Faculty of Computer Science/ University of New Brunswick as project coordinator from early September 2004 until the end of September 2005. His previous academic positions were as follows: Assistant professor in Department of Computer Engineering Iran's University of Science and Technology (where he is currently an assistant professor) and Assistant Professor in Azad University – central branch – Faculty of Engineering both in Tehran-Iran. His research interests include Machine Learning, Remote Sensing, Robotics and Network Intrusion Detection.