An Efficient Detection Mechanism for Intrusion Detection Systems Using Rule Learning Method

K.SARAVANAN

Abstract—**Nowadays Information Security plays an important role in Hi-tech computing world. Even though firewall is used to provide security between two different networks, it fails to care about the intranet security (security within a single network). In order to overcome the problem a model called Intrusion Detection System is used. The process of monitoring the events occurring in a computer system or network and analyzing them for sign of intrusions is known as intrusion detection system (IDS). In this experiment, we investigated Decision trees algorithm. To train the soft computing techniques we are using the Benchmark KDD cup Data set.**

Key words— **IDS (Intrusion Detection Systems), Ensemble Approach, Decision Tree Algorithm (DT)**

I. INTRODUCTION

The first lines of defense for computer security are Protection techniques such as user authentication, data encryption, avoiding programming errors and firewalls. If a password is weak and is compromised, user authentication cannot prevent unauthorized use, firewalls are vulnerable to errors in configuration and suspect to one or more possible or undefined security policies. They are generally unable to protect against malicious insider attacks and unsecured modems. Programming error can be avoided, but it makes the system more complex. So these vulnerabilities are left out to make the system simple. Even though computer systems are likely to remain unsecured for the foreseeable future. Therefore, intrusion detection is required as an additional wall for protecting systems despite of these prevention techniques. Intrusion detection is useful not only in detecting successful intrusions, but also in monitoring attempts to break security, which provides important information for timely countermeasures [13]

Intrusion detection an important component of information security technology helps in discovering, determining, and identifying unauthorized use, duplication, alteration, and destruction of information and information systems. Intrusion detection relies on the assumption that information and information systems under attack exhibit several different behavioral patterns. Even though intrusion detection technology is becoming being present everywhere in current network defense; it lacks basic definitions and mathematical understanding. Intrusion detection being subjective; each Intrusion Detection System (IDS) has a different classification and attack labeling mechanisms. It is most common for IDSs to alarm on any set of known attack behaviors. While determining whether a particular activity is normal or malicious, IDS fail to alarm an attack (false negative) or alarm normal activity as malicious (false positive).

The most popular way to detect intrusions has been done by using audit data generated by operating systems and by networks. Since almost all activities are logged on a system, it is possible that a manual inspection of these logs would allow intrusions to be detected. It is important to analyze the audit data even after an attack has occurred, for determining the extent of damage occurred, this analysis helps in attack trace back and also helps in recording the attack patterns for future prevention of such attacks. An intrusion detection system can be used to analyze audit data for such insights. This makes intrusion detection system a valuable real-time detection and prevention tool as well as an analysis tool [11].

Intrusion detection is classified into two types: misuse intrusion detection and anomaly intrusion detection. Misuse intrusion detection uses well-defined patterns of the attack that exploit weaknesses in system and application software to identify the intrusions. These patterns are used to match against user behavior to detect intrusions. Anomaly intrusion detection identifies deviations from the normal usage behavior patterns to identify the intrusion. The normal usage patterns are constructed from the statistical measures of the system features, for example, the CPU and I/O activities by a particular user or program. The behavior of the user is observed and any deviation from the constructed normal behavior is detected as intrusion.

II. INTRUSION DETECTION SYSTEM

The main task of the IDS implementations are detecting intrusions based on audit trails. Most of the IDS are hardware and/or software package or large part of a system. It has four

Manuscript received October 9, 2001. (Write the date on which you submitted your paper for review.) This work was supported in part by the U.S. Department of Commerce under Grant BS123456 (sponsor and financial support acknowledgment goes here). Paper titles should be written in uppercase and lowercase letters, not all uppercase. Avoid writing long formulas with subscripts in the title; short formulas that identify the elements are fine (e.g., "Nd–Fe–B"). Do not write "(Invited)" in the title. Full names of authors are preferred in the author field, but are not required. Put a space between authors' initials.

F. A. Author is with the National Institute of Standards and Technology, Boulder, CO 80305 USA (corresponding author to provide phone: 303-555-5555; fax: 303-555-5555.

S. B. Author, Jr., was with Rice University, Houston, TX 77005 USA. He is now with the Department of Physics, Colorado State University, Fort Collins, CO 80523 USA

T. C. Author is with the Electrical Engineering Department, University of Colorado, Boulder, CO 80309 USA, on leave from the National Research Institute for Metals, Tsukuba, Japan

important components. They are
1. Data collection module.
2. Analysis Module.
3. Storage Module.
4. Response Module.
These set of components built an IDS model.. A generic model of IDS.

Data collection module provides information to the rest of the system to decide whether a particular activity is intrusion or not. Data collection module collects audit trails, user logs, network trails, system calls for the other IDS components to take decisions, without this module the IDS becomes un-functional. An important issue in the data collection module is audit data reduction. Instead of passing the raw data to the analysis module to decide whether a particular activity is malicious or normal, designers implement systems that eliminate audit information believed to be unimportant for intrusion analysis. The goal of audit reduction is to pass only important, reduced or summarized audit trails to the analysis module; they also help in reducing the complexity of the analysis module.

Analysis module takes the inputs (audit trails) form the data collection module. The main goal of this module is to creating classifiers in terms of better IDS performance. The IDS performance includes faster classification, low false alarms and higher accuracies. Several analysis techniques are being proposed ranging from statistical analysis, pattern matching, machine learning, file integrity checkers and artificial immune system methods. Analysis module helps in automated analysis of data by reducing human intervention and speeds up the process of identifying intrusions in real time.

Storage module provides a mechanism to store data collected by data collection and analysis modules in a secure fashion. Data stored might be used for building new signatures, updating user and system profiles and identifying key audit information. Response module can be designed in an active or a proactive mode. In proactive mode, the system does not wait for the response of analysis module. It sets an alarm when an intrusion takes place. Most of the current IDS are designed in proactive mode. Intrusion detection prevention systems (IDPS) not only spot intrusions but also intercept and stop intrusions [10].

III. KDD CUP 1999 DATA SET

The KDD Cup 1999 Intrusion detection contest data is used in our project. The data set contains 24 attack types. These attacks fall into four main categories:

Denial of service (DOS):

In this type of attack an attacker makes some computing or memory resources too busy or too full to handle legitimate requests, or denies legitimate users access to a machine. Examples are Apache2, Back, Land, Mailbomb, SYN Flood, Ping of death, Process table, Smurf.

Remote to user (R2L):

In this type of attack an attacker who does not have an account on a remote machine sends packets to that machine over a network and exploits some vulnerability to gain local access as a user of that machine. Examples are Dictionary, Ftp_write, Guest, Imap, Named, Phf, Sendmail, Xlock.

User to root (U2R):

In this type of attacks an attacker starts out with access to a normal user account on the system and is able to exploit system vulnerabilities to gain root access to the system. Examples are Eject, Loadmodule, Ps, Xterm, Perl, Fdformat.

Probing:

In this type of attacks an attacker scans a network of computers to gather information or find known vulnerabilities. An attacker with a map of machines and services that are available on a network can use this information to look for exploits. Examples are Ipsweep, Mscan, Saint, Satan, Imap. The data set has 41 attributes for each connection record plus one class label. R2L and U2R attacks don’t have any sequential patterns like DOS and Probe because the former attacks have the attacks embedded in the data packets whereas the later attacks have many connections in a short amount of time. Therefore, some features that look for suspicious behavior in the data packets like numbers of failed logins are constructed and these are called content features (w ww.ll.mit.edu).

The 41 attributes are:
duration: length (number of seconds) of the connection.
protocol_type: type of the protocol, e.g. tcp, udp, etc.
service: network service on the destination, e.g., http,telnet, etc.
src_bytes : number of data bytes from source to destination.
dst_bytes: number of data bytes from destination to source.
flag: normal or error status of the connection.
land: 1 if connection is from/to the same host/port; 0 otherwise.
wrong_fragment: number of ``wrong" fragments.
urgent: numb er of urgent packets.
hot: number of ``hot" indicators.
num_failed_logins: number of failed login attempts.
logged in: number of ``compromised" conditions.
root_shell: 1 if root shell is obtained; 0 otherwise.
su_attempted: 1 if ``su root" command attempted; 0 otherwise.
num_root: number of ``root" accesses.
num_file_creations: number of file creation operations
num_shells : number of shell prompts.
num_access_files: number of operations on access control files.
num_outbound_cmds: number of outbound commands in an ftp session.
is_hot_login: 1 if the login belongs to the ``hot" list; 0 otherwise.
is_guest_login: 1 if the login is a ``guest" login; 0 otherwise.
count: number of connections to the same host as the current connection in the past two seconds.
serror_rate: % of connections that have ``SYN" errors.
rerror_rate: % of connections that have ``REJ" errors.
same_srv_rate: % of connections to the same service.
diff_srv_rate: % of connections to different services.
srv_count: number of connections to the same service as the current connection in the past two seconds.
srv_serror_rate: % of connections that have ``SYN" errors.
srv_rerror_rate: % of connections that have ``REJ" errors.
srv_diff_host_rate: % of connections to different hosts.

IV. SOFT COMPUTING TECHNIQUES:

Classifiers are functions that can be tuned according to examples. These examples are known as observations or patterns. In supervised learning, each pattern belongs to a certain predefined class. A class can be seen as a decision that has to be made. All the observations combined with their class labels are known as a data set. When a new observation is received, that observation is classified based on previous experience. A classifier can be trained in various ways; there are mainly statistical and machine learning approaches. Types of base classifiers: SVM (support vector machine), DT (decision tree), fuzzy logic, genetic algorithm, bayes classifier etc.

A. DECISION TREES (DT)

DT induction is one of the classification algorithms in data mining. The classification algorithm is inductively learned to construct a model from the preclassified data set. Inductive learning means making general assumptions from the specific examples in order to use those assumptions to classify unseen data.

The inductively learned model of classification algorithm is known as classifier. Classifier may be viewed as mapping from a set of attributes to a particular class. Data items are defined by the values of their attributes and X is the vector of their values {x1, x2,.., xn}, where the value is either numeric or nominal. Attribute space is defined as the set containing all possible attribute vectors and is denoted by Z. Thus X is an element of Z (XεZ). The set of all classes is denoted by C = {c1, c2, ... , cn}. A classifier assigns a class cεC to every attribute of the vector XεZ. The classifier can be considered as a mapping f, where f: X->C. This classifier is used to classify the unseen data with a class label. A DT classifies the given data item using the values of its attributes.

The DT is initially constructed from a set of preclassified data. Each data item is defined by values of the attributes. The main issue is to select the attributes which best divides the data items into their classes. According to the values of these attributes the data items are partitioned. This process is recursively applied to each partitioned subset of the data items. The process terminates when all the data items in the current subset belongs to the same class.

A DT consists of nodes, leaves and edges. A node of a DT specifies an attribute by which the data is to be partitioned. Each node has a number of edges, which are labeled according to a possible value of edges and a possible value of the attribute in the parent node. An edge connects either two nodes or a node and a leaf. Leaves are labeled with a decision value for categorization of the data. Induction of the DT uses the training data, which is described in terms of the attributes [6] .

C4.5 Algorithm

C4.5 (Learning Sets S, Attributes Sets A, Attributes values V)
Return Decision Tree.
Begin
Load learning sets first, create decision tree root node 'rootNode', add learning set S into root node as its subset.
For rootNode, we compute Entropy (rootNode.subset) first.

8) If Entropy (rootNode.subset) ==0, then rootNode.subset consists of records all with the same value for the categorical attribute, return a leaf node with decision attribute: attribute value.
9) If Entropy (rootNode.subset)! =0, then compute information gain for each attribute left (have not been used in splitting), find attribute A with Maximum (Gain(S, A)). Create child nodes of this rootNode and add to rootNode in the decision tree.

For each child of the rootNode, apply ID3(S, A, V) recursively until reach node that has entropy=0 or reach leaf node. End C4.5.

V. EXPERIMENTS

In our experiments, we have done it in two approaches. First approach is broadly categories of attacks with 5-class classification. The training and testing dataset contains taken from the KDD cup dataset for the five classes. The normal data belongs to class1, probe belongs to class 2, denial of service belongs to class 3, user to super user belongs to class 4, remote to local belongs to class 5. A different randomly selected set of the total data set is used for testing different soft computing techniques. In the second approach, for each attacks we performed the classification.

A. Experiments using Decision Tree (DT):

The data is partitioned into the two classes of ‘‘Normal’’ and ‘‘Attack’’ patterns where Attack is the collection of four classes (DOS, U2R, R2L, PROBING) of attacks. The objective is to separate normal and attack patterns. We repeat this process for all the five classes. First a classifier was constructed using the training data and then testing data was tested with the constructed classifier to classify the data into normal or attack. Table:1 summarizes the results of the test data.

Table: 1 Identification and recognition statistics depending on attack category using DT

	DOS	U2R	R2L	PROBE	NORMAL
DOS	391434	0	0	3	21
U2R	0	45	1	1	5
R2L	0	2	1113	0	11
PROBE	0	0	0	4098	9
NORMAL	1	1	1	0	697269

Table: 2 shows the confusion matrix of C4.5 algorithm classification for the five classes of Attacks

Attack Type	Accuracy
Denial of service	99.996%
User to root	71.153%
Remote to login	97.690%
Probing	99.513%
Normal	99.998%

In the second approach we investigated the C4.5 algorithm for each attacks. The KDD cup dataset has 22 attacks, for each attacks we performed the classification.Table: 3 shows the Identification and recognition depending on attack type category using DT

VI. CONCLUSIONS

In this experiment, we have investigated some new technique for intrusion detection and evaluated their performance based on the benchmark KDD Cup 99 Intrusion data. We have explored Decision Tree (C4.5 Algorithm) as intrusion detection models. Here we have shown the classification for the attacks in two ways. One is for broad categorization of attacks and another way is for individual attacks. In comparison with other approaches the Decision Tree Algorithm permit to design the intrusion detection systems, which have ability to training and working in real time. The experiments have shown the efficiency of DT techniques.

REFERENCES

[1] Denning D.E. An intrusion detection model. IEEE Transactions on Software Engineering 1997:222–8.

[2] Srinivas Mukkamala, Andrew Sung and Ajith Abraham, Cyber Security Challenges: Designing Efficient Intrusion Detection Systems and Antivirus Tools, Department of Computer Science, New Mexico Tech, USA.

[3] Crosbie, M., and Spafford, G. Applying genetic programming to intrusion detection. In Proc.1995 AAAI Symposium on Genetic Programming, pp. 1-8.

[4] Deniz Yuret, Machine Learning lecture on decision tree algorithm, Scribe: Basak Mutlum Fall Term, 2003.

[5] Wei Peng, Juhua Chen and Haiping Zhou , An Implementation of ID3 --- Decision Tree Learning Algorithm, Project of Comp 9417: Machine Learning University of New South Wales, School of Computer Science & Engineering, Sydney, NSW 2032, Australia.

[6] Quinlan JR. C4.5: programs for machine learning. Log Altos,CA: Morgan Kaufmann; 1993.

[7] Mukkamala S, Sung A, Abraham A, Ramos V. intrusion detection systems using adaptive regression splines. In: Seruca I,Filipe J,Hammoudi S,Cordeiro J,editors. Proceedings of the 6th international conference on enterprise information systems,
ICEIS'04, vol. 3, Portugal. 2004b. p. 26–33 [ISBN: 972-8865-00-7].

[8] Friedman, J. H, 1991. Multivariate Adaptive Regression Splines, Annals of Statistics, Vol 19, pp. 1- 141.

[9] Steinberg, D, Colla P. L., Martin K., 1999. MARS User Guide. Salford Systems, San Diego, CA.

[10] Mukkamala S, Sung AH, Abraham A. Intrusion detection using ensemble of soft computing paradigms, third international conference on intelligent systems design and applications, intelligent systems design and applications, advances in soft computing. Germany: Springer; 2003. p. 239–48.

[11] Cannady J. Artificial neural networks for misuse detection. National Information Systems Security Conference 1998.

[12] Joachim's T., 1998. Making Large-Scale SVM Learning Practical.University of Dortmund, LS8-Report, LS VIII-Report.

[13] Sandhya Peddabachigari, Ajith Abraham, Crina Grosan, Johnson Thomas Modeling intrusion detection system using Hybrid intelligent systems: Journal of Network and Computer applications Available at www.sciencedirect.com in the year 2005.

Table: 3 Identification and recognition depending on attack type category using DT

S.No	Attacks	Count	Accuracy
1	Back	2203	99.9546
2	Buffer_overflow	30	93.3333
3	ftp_write	8	50
4	Guess	53	96.2264
5	Imap	12	100
6	ipsweep	9	55.5556
7	Land	21	100
8	Loadmodule	7	85.7143
9	Multihop	7	85.7143
10	neptune	107201	99.9823
11	Nmap	231	98.7013
12	normal	97278	99.9774
13	perl	3	100
14	phf	4	100
15	pod	264	100
16	portsweep	1040	99.3269
17	rootkit	10	30
18	satan	1589	99.4965
19	smurf	280790	100
20	Spy	2	0

F **Saravanan Kumarasamy.** This author became a Member (M) of CSI in 2008.Primary and secondary school education at G.B.H.S. School, Perundurai, Tamilnadu and First bachelor Degree B.E. (India) 2005 in electronics and communication engineering from Anna University, Chennai, India. Received the M.E degree 2008 in computer science from MCET, Anna University, Chennai, India.

From December to till now, he was a Lecturer at the Faculty of Engineering, Erode Sengunthar Engineering College, Erode His current research interests are information security, computer communications, DDoS Attacks and routing architecture

.