ANALYSIS, DESIGN AND IMPLEMENTATION OF IDS USING DATA MINING

B. V. Patel
Computer Technology Department, Shah & Anchor Kutchhi Polytechnic, Mumbai, India

B. B. Meshram
Computer Technology Department, Veermata Jijabai Technological Institute, Mumbai, India

Keywords: Normal Network Traffic, IP, TCP, UDP Packet Format, Fuzzy, Association, Classification, Certainty.

Abstract: To achieve the implementation of intrusion detection system (IDS), we have integrated the Fuzzy Logic with extended Apriori Association Data Mining to extract more abstract patterns at a higher level which look for deviations from stored patterns of normal behaviour of the computer network. Here the various packet formats of TCP, UDP, IP etc are used to study the normal behaviour of the network. Genetic algorithms are used to tune the fuzzy membership functions. The tuned data by genetic algorithms is processed by the modified Apriori algorithm. The association pattern is populated by genetic algorithm for the selection of best population of the network traffic. This best populated data is classified by the C4.5 algorithms to find intrusions. The deployment of IDS is done under the control of secure linux environment and the system is tested in the distributed environment.

1 INTRODUCTION

"Intrusion Detection is the process of monitoring the events occurring in a computer system or network and analyzing them for signs of intrusions. Intrusion is defined as attempts to compromise the confidentiality, integrity, availability, or to bypass security mechanism of a computer or network." While information system in an organization has become a valuable asset of the organization, attempts to compromise it have also increased. Intrusion detection systems add an early warning capability to a company's defences, alerting to the type of suspicious activity that typically occurs before and during an attack. Since we cannot stop an attack, intrusion detection systems should not be considered an alternative to traditional good security practices. There is no substitute for a carefully thought out corporate security policy, backed up by effective security procedures which are carried out by skilled staff using the necessary tools (Meshram, 2004, Kumar, 2005). Instead, intrusion detection systems should be viewed as an additional tool in the continuing battle against hackers and crackers.

However, our work aims to eliminate, as much as possible, the manual and ad-hoc elements from the process of building an intrusion detection system (Meshram, 2004). The central theme of our approach is to describe a data mining framework for adaptively building Intrusion Detection (ID) model.

The rest of the paper is organized as below:

The section 2 presents the literature survey and modified algorithms used in this implementation, section 3 describes data and process modelling of the system and 4th section presents the implementation of the system, finally 5th section summarizes the results.

2 PROPOSED ALGORITHMS USED IN IMPLEMENTATION

The proposed algorithms used in this implementation are modified to meet the needs of network data.

V. Patel B. and Meshram B. (2008).
ANALYSIS, DESIGN AND IMPLEMENTATION OF IDS USING DATA MINING.
In *Proceedings of the Fourth International Conference on Web Information Systems and Technologies*, pages 81-86
DOI: 10.5220/0001516100810086

2.1 Extensions of Apriori Algorithm

The basic Apriori algorithms(Meshram, 2004) do not consider any domain knowledge and as a result they can generate many "irrelevant" (i.e., uninteresting) rules. The above limitations and generalization is achieved by incorporating two modifications to the basic Apriori algorithm. They are: Axis attributes, Reference attributes.

Interestingness Measures based on Attributes: The basic algorithms implicitly measure the interestingness (i.e., relevancy) of patterns by their support and confidence values, without regard to any available prior domain knowledge. That is, assume I is the interestingness measure of a pattern p, then I(p) = f(support(p); confidence(p)) Where f is some ranking function. We attempt to utilize the schema level information about audit records to direct the pattern mining process. Assume IA is a measure on whether a pattern p contains the specified important (i.e. "interesting") attributes, our extended interestingness measure is

Ie(p) = fe(IA(p); f(support(p); confidence(p)))
= fe(IA(p); I(p))

Where fe is a ranking function that first considers the attributes in the pattern, then the support and confidence values.

Using the Axis Attributes (IA): We describe several schema-level characteristics of audit data, in the forms of "what attributes must be considered", that can be used to guide the mining of relevant features. We call the essential attribute(s) as axis attribute(s) when they are used as a form of item constraints in the association rules algorithm. During candidate generation, an item set must contain value(s) of the axis attribute(s). We consider the correlation among non-axis attributes as not interesting. In other words,

IAx (p) = 1 if p contains axis attribute(s)
= 0 otherwise (not interesting attribute).

Using the Reference Attributes (IR): Another interesting characteristic of system audit data is that some attributes can be the references of other attributes. These reference attributes normally carry information about some "subject", and other attributes describe the "actions" that refer to the same "subject". It is important to use the "subject" as a reference when finding such frequent sequential "action" patterns because the "actions" from different "subjects" are normally irrelevant. This kind of sequential pattern can be represented as:

(Subject = X; action = a);
(Subject = X; action = b) → (subject = X; action = c)

Thus subject is simply reference (or a variable).In other words,

IAr (p) = 1 if the item sets of p refer to the same reference attribute value otherwise 0.

Level-wise Approximate Mining: In daily network traffic, some services, for example, gopher, account for very low occurrences. Yet we still need to include their patterns into the network traffic profile, we will then get unnecessarily a very large number of patterns related to the high frequency services.

Here the idea is to first find the episodes related to high frequency axis attribute values. We then iteratively lower the support threshold to find the episodes related to the low frequency axis values by restricting the participation of the "old" axis values that already have output episodes. More specifically, when an episode is generated, it must contain at least one "new" (low frequency) axis value. Then we can infer the more information required in the classification of the network data.

2.2 Integration of Fuzzy Logic with Modified Apriori

Although association rules can be mined from audit data for anomaly intrusion detection, the mined rules are at the data level. Integrating fuzzy logic with association rules allows one to extract more abstract patterns at a higher level.

2.2.1 Probability of Fuzzy Logic Association Rules

Let T = {t_1, t_2, ..., t_n} be the database and ti represents the i^{th} tuple in T. Moreover, we use I = {i_1, i_2, ..., i_m} to represent all attributes appeared in T and ij represents the j^{th} attribute. Since I contain set of items, we call I an itemset. We can retrieve the value of attribute i_k in the j^{th} record simply by t_j [ik]. Besides, each attribute ik will associate with several fuzzy sets. We use F_{jk} = {$f1_{jk}$, $f2_{ik}$, ..., fl_{ik}} to represent set of fuzzy sets associated with ik and fj_{ik} represents the j^{th} fuzzy set in F_{ik}.

Given a database T with attributes I and those fuzzy sets associated with attributes in I, we want to find out some interesting, potentially useful regularities in a guided way. Our proposed fuzzy association rule is in the form: If X is A then Y is B. In the above rule, X = {x_1, x_2, ..., x_p} and Y = {y_1, y_2, ..., y_q} are item sets. X and Y are subsets of I and they are disjoint which means that they share no common attributes. A = {fx_1, fx_2, ..., fx_p} and B = {fy_1, fy_2, ..., fy_q} contain the fuzzy sets associated with the corresponding attributes in X and Y. For example, an attribute x_k in X will have a fuzzy set f_{xk} in A such that f_{xk} ε F_{xk} is satisfied.

We use significance and a certainty factor to determine the satisfiability of itemises and rules.

Significance Factor: To generate fuzzy association rule, we have to find out all large k-item sets which are item sets with significance higher than a user specified threshold.

We use the following formula to calculate the significance factor of <X, A>, i.e. S<X, A>

Significance = (Sum of votes satisfying <X, A>) / (Total number of records in T)

$S_{<X, A>} = \sum t_i \varepsilon T \prod x_j \varepsilon X \{\alpha a_j (t_i [x_j])\} / \text{total } (T)$

where

$$\alpha a_j (t_i [x_j]) = \begin{cases} m_{aj} \varepsilon A(t_i [x_j]) \text{ if } m_{aj} >= w \\ 0 \qquad \text{otherwise.} \end{cases}$$

In the above equation, <X, A> represents the itemset-fuzzy set pair, where X is set of attributes xj and A is the set of fuzzy sets aj.

Certainty Factor: When we obtain a large item set <Z, C>, we want to generate fuzzy association rules of the form, 'If X is A then Y is B.' where X ***C*** Z, Y = Z -X, A ***C*** C and B = C-A. We can calculate the certainty factor as follows:

Certainty = Significance of <Z, C> / Significance of <X, A>

Since the significance factor of an item set is the measure of the degree of support given by records, we use significance to help us estimate the interestingness of the generated fuzzy association rules. The certainty reflects fraction of votes support <X, A> will also support <Z, C>.

2.3 Genetic Algorithm

This algorithm originates from Darwin's Evaluation Theory (Carvalho, 2002). We have used the Genetic Algorithm in distributed environment (Carvalho, 2002).

2.3.1 Applications of Genetic Algorithms in Fuzzy Systems

Without a computer-aided tool (such as GAs), the task of defining membership functions for the fuzzy variables in a target system is usually completed by human experts manually. To ease or release human experts from this tedious work, GAs have been extensively used to derive optimal or sub-optimal membership functions. Integrating fuzzy logic with association rules allow one to extract more abstract patterns of normal behaviour. If deviation is beyond certain threshold, the anomaly is detected as an attempt to intrude network. Genetic algorithms are used to tune the fuzzy membership functions.

2.3.2 Similarity among Fuzzy Association and Fitness Function

The fitness function for the genetic algorithm is based on the similarity of rule sets mined from different sets of audit data. The similarity between two fuzzy association rules and between two fuzzy rule sets is defined as:

Given two association rules, R1: X →Y, c, s, and R2: X'→Y', c', s'; if X=X' and Y=Y', then the similarity between R1 and R2 is:

similarity (R1, R2) = max {0, 1-max [c- c'| /c, |s-s'|/s}

Otherwise, similarity (R1, R2) =0

The similarity between two rule sets S1 and S2 is:

Similarity (S1, S2) = (s/|S1|) * (s/|S2|)

Where s= ∑vR1εS1and vR2 εS2 similarity (R1, R2), and |S1| and |S2| are the total number of rules in S1 and S2, respectively.

Two different sets of audit data were available for testing the fuzzy logic genetic algorithm: a normal data set with no intrusions and "abnormal" data set with different types of intrusions. The abnormal data set is created using tool called nemesis. The normal data (created by TCPdump) set was partitioned into two sets. One partition of the normal audit data is called the reference data and one set is called normal data. The following two fitness functions have been designed and tested (Axelsson, 2000):

F1= Srn / Sra

F2= Srn – Sra

where Srn is the similarity between a reference rule set and the "normal" rule set, Sra is the similarity between the reference rule set and abnormal rule set.

2.4 GA and C4.5 for Intrusion Detection

Intrusion Detection System deals with huge amount of data which contains irrelevant and redundant features causing slow training and testing rate, higher resource consumption as well as poor detection rate. To overcome this problem a combined approach of GA and C4.5 is used.

We have used genetic algorithms for feature extraction and elimination from the association rules.

Here the fitness function is given by the formula:

Fitness = (TP / (TP + FN)) * (TN / (FP + TN)), where TP, FN, TN and FP stand for the number of true positives, false negatives, true negatives and false positives.

Then the classification of the data is done by using C4.5 Algorithms in distributed environment (Carvalho, 2002).

The C4.5 algorithm uses a splitting criterion based on Information Gain Ratio. The idea is to partition the given training set in such a way that the information needed to classify the given example is reduced as much as possible. Formally let s be training set and |s| be number of elements in s, and freq (C_i,s) s number of records that belongs to class I where i={1,2,…,N}.The average information i.e. entropy to identify the given class is:

$\mathrm{Info}(s) =-\sum \mathrm{freq}\ (C_j, s) /|s| * \log_2 (\mathrm{freq}\ (C_j, s) /|s|)$

Hence the amount of information needed to split the set s into N distinct subsets {s_i} in agreement to test (Test_A) result, is:

$\mathrm{Info}_{\mathrm{TestA}}\ (s)= \sum|s_i\,|/|s| *\mathrm{Info}(s_i)$

The gain is computed as:

$\mathrm{gainRatio}(\mathrm{Test}_A)=\mathrm{gain}(\mathrm{Test}_A)/\mathrm{splitInfo}(\mathrm{Test}_A)$

where ,

$\mathrm{gain}(\mathrm{Test}_A)=\mathrm{Info}(s)-\mathrm{Info}_{\mathrm{TestA}}(s)$ and

$\mathrm{splitInfo}(\mathrm{Test}_A) =-\sum |s_i\,|/|s| * \log_2 (|s_i\,|/|s|)$ for I = 1 to n.

3 ANALYSIS AND DESIGN OF THE SYSTEMS

This section introduces the analysis and design of the system.

3.1 Data Model

This section introduces the data structure design (Meshram, 2004) used for authoring system of the IDS. The TCPdump data contains packet data captured by tcpdump as below:

For UDP Datagram. (Timestamp, Packet type, Source address, Source port, Destination address, Destination port, Protocol, Size)

For TCP Datagram. (Timestamp, Packet type, Source address, Source port, Destination address, Destination port, flag, Sequence number, Contained data up to, Number of user data, Acknowledgement, Window size, Option).

The network data captured is pre-processed and fuzzyfied, genetic algorithmic evaluated data is stored in the following data structure.

Fuzzyfier. (attributes: significance_s_a, significance_s_c, Signifiance_z_a, Signifiance_z_c ; **Genetic_Evaluator.** (attributes: crossover_probability mutation, probability, noofgenerations, chromosome; methods: generation_population(), crossover(), mutate(), calculate_fitness()).

The output of these files are given to the extended Apriori algorithms:

Itemset.txt. The **itemset.txt** which is used by training module contains the candidate item sets and its frequencies.

Itemset (Candidate item set, frequencies)

Rules.txt. The **rules.txt** file which is an output of the training module and one of the inputs of detection module contains association rules generated by Apriori algorithm.

Rules.txt (Frequent item set)

Intrusioninfo.txt. The **intrusioninfo.txt** which is generated by detection module contains the information about intruder.

The intruders and alerts are stored in the following file.

Intrusion-Detection (attributes: Connection time, IP address, Port, status, Significance, certainty; methods: generate_alarm(); detection_alarm(); display_results()).

3.2 Process Model

The flow of information of the intrusion detection system is as shown in figure 1.

The Input Data Pre-processing. The network-based approached relies on the tcpdump data as input, which gives per packet information. We used data a 2 GB dataset that we collected over a 24hrs span. We used it as the normal data. This data was pre-processed as grouping records corresponding to one connection. Following this, Content-based, Time-based and Connection-based features were extracted from the data.

The Training Module. IDS is trained using a data set in which the attacks and the attack-free periods are correctly labelled. Both parts of the stream are fed into this module which performs a conventional association rules discovery with the aid of fuzzy logic and genetic algorithms. The output of this module is a profile of rules that depict the behaviour of the network with the possible known attacks.

The Detection Module. In order to detect intrusions in the test data, the Intrusion Detector is invoked, which is implemented using Genetic algorithm and C4-5 algorithm to labels the event as either normal or attacks depending on the certainty theory.

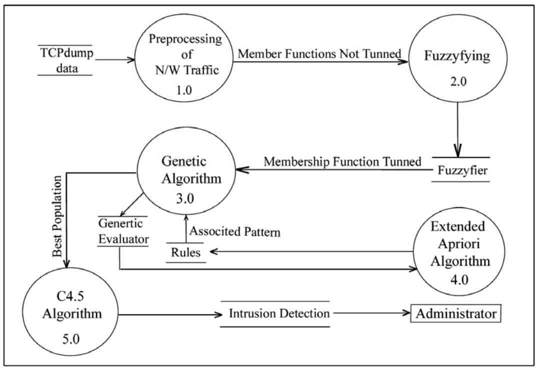


Figure 1: Flow of information of IDS system.

4 IMPLEMENTATION

Tcpdump data is captured and downloaded it in text.txt file by the following command:

```
# tcpdump –i eth0 >> test.txt
```

Algorithms1: Input Pre-Processing ( )

Input: test.txt

>>text.txt file contains the packet information of many records.

Procedure : Pre-Processing ( )

In this module, data in text.txt is processed to generate itemset.txt file. The basic task is to extract extensive set of features for data mining algorithm. This task is perform on the basis of protocol of the packet and frequency of item set. The frequent Item set and their frequency is stored in itemset.txt file.

Output: Itemset.txt

>>itemset.txt file contains frequent Item set and their frequency

Algorithms2: Training Module ( )

Input: itemset.txt of module-1

Procedure : Training Module( ). In this module, data in Itemset.txt is processed to generate rules.txt file. The basic task is to perform Apriori algorithm integrated with fuzzy logic and genetic algorithms on Itemset.txt to generate association rules.

Output: rules.txt

>>rules.txt file contains association rules by which intrusion detection is performed.

Algorithms3: Detection Module ( )

Input: rules.txt of module-2 and netstat.txt (training data)

>> netstat.txt is obtained by capturing packets through tcpdump as below:

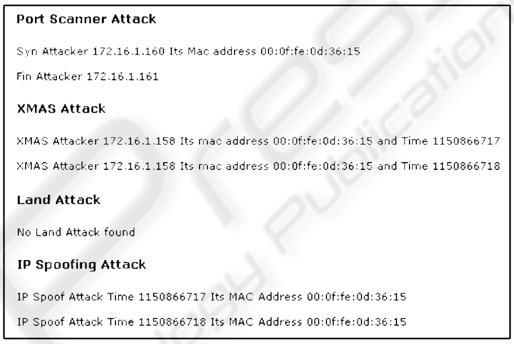
```
# tcpdump –i eth0 >> netstat.txt
```

Procedure: Detection Module ( )

The rules are stored in rules.txt file and processed by the genetic algorithms. In this module, detection process (C4.5 algorithms) is performed on netstat.txt. This task is performed by C4.5 algorithm and produce Intrusion information on network. This intrusion information is stored in intrusioninfo.txt file

Output: intrusioninfo.txt

>> intrusioninfo.txt file contains attack type, connection time, IP address, port and status depending on the data etc. you can read the file and understand the attack. The content of the intrusioninfo.txt are observed in the multimedia authoring graphical interface as in figure 2.



```
Port Scanner Attack

Syn Attacker 172.16.1.160 Its Mac address 00:0f:fe:0d:36:15

Fin Attacker 172.16.1.161

XMAS Attack

XMAS Attacker 172.16.1.158 Its mac address 00:0f:fe:0d:36:15 and Time 1150866717

XMAS Attacker 172.16.1.158 Its mac address 00:0f:fe:0d:36:15 and Time 1150866718

Land Attack

No Land Attack found

IP Spoofing Attack

IP Spoof Attack Time 1150866717 Its MAC Address 00:0f:fe:0d:36:15

IP Spoof Attack Time 1150866718 Its MAC Address 00:0f:fe:0d:36:15
```

Figure 2: Intrusion Detection.

The backend of the system is oracle 9i and the behavioural model and user interface is implemented using Java. The implemented IDS is configured in the distributed Linux environment under the control of role based security and fire wall(Karvande, 2004).

5 CONCLUSIONS

We have demonstrated that the integration of fuzzy logic with extended Apriori algorithm generates more abstract and flexible patterns for anomaly detection.

Fuzzy association rules can be used to implement a network intrusion detection system based upon the assumption that an attack can be identified as burst traffic in audit logs. GA-optimized fuzzy logic i.e., maximizes the similarity between normal association rule sets while minimizing the similarity between a normal and an abnormal association rule set. In addition to tuning membership functions for fuzzy association rules, genetic algorithms is used for feature selection. They can also be used tune other association rule mining parameters such as minimum significance and minimum certainty.

The intrusion detection process is performed by C4.5 algorithm and produce intrusion information for the administrator.

This data mining approach will not only reduce training and testing time but also guarantee high detection rates and low false positive rates among different datasets.

REFERENCES

B. B. Meshram, Alok K. Kumar, 2004. *HyIDS: Hybrid Intrusion Detection System in the Proceedings of National Conference on Research & Practices in Current Areas of IT*, Department of Computer Science & Engineering, Sant Harchand Sing Longowal Central Institute of Engineering & Technology, Longowal, Dist Sangar(Punjab)-148106.

Alok K. Kumar , B. B. Meshram, 2005. NAD: Statistical Network Anomaly Detector. *In International Conference Systemics, Cybernetics and Informatics Icsci – 2005*, Under The Aegis Of Pentagram Research Centre Pvt. Ltd. Venue: Dr. Mcr Hrd Institute Of Andhra Pradesh, Hyderabad.

B. B. Meshram, T.R.Sontakke, 2001. Object Oriented Database schema Design. *In 7Th International Conference on Object Oriented Information Systems*., Calgary, Canada.

Z. Malik, B. B. Meshram, 2004. A Study on Data Mining. *In Proceedings of National Conference on Research & Practices in Current Areas of IT*, Department of Computer Science & Engineering, Sant Harchand Sing Longowal Central Institute of Engineering & Technology, Longowal, Dist Sangar(Punjab)-148106.

S. S. Karvande, B.B.Meshram, 2004. Design And Implementation Of Application Layer Firewall For Secure Internet Access. *In International Conferences on Soft Computing*, Department of Computer Applications, Computer Science & Engineering, Information Technology, Bharath Institute of Higher Education & Research, Chennai, Tamilnadu.

B. V. Patel, B. B. Meshram, 2007. Carpace 1.0 For Multimedia Email Security. In *International Multiconference of Engineers and Computer Scientists,* Hong Kong.

Fan W., Miller M., Stolfo S., Lee W., Chan P, 2001. Using Artificial Anomalies to Detect Unknown and Known Network Intrusions. *In Proceedings of the First IEEE International Conference on Data Mining*CA, http://www.cc.gatech.edu/~wenke/papers/artificial_anomalies.ps.

Axelsson S. 2000. Intrusion Detection Systems: A Taxomomy and Survey. *Technical report No 99-15, Dept. of Computer Engineering, Chalmers University of Technology*, Sweden. http://www.ce.chalmers.se/staff/sax/taxonomy.ps.

Frederick K. K. 2001, Network Intrusion Detection Signatures. http://online.securityfocus.com/infocus/1524.

Marin J., Ragsdale D., Surdu J, 2001. A Hybrid Approach to the Profile Creation and Intrusion Detection. *In Proceedings of the DARPA Information Survivability Conference and Exposition.*

http://www.itoc.usma.edu/Documents/Hybrid_DISCEX_AcceptedCopy.pdf.

Lee W.,2000.A data mining and CIDF based approach for detecting novel and distributed Intrusions. *In Third International Worksho on Recent Advances in Intrusion Detection, RAID*. Toulouse, France. http://www.cc.gatech.edu/~wenke/papers/lee_raid_00.ps.

Elson D., 2000. Intrusion Detection, Theory and Practice. http://online.securityfocus.com/infocus/1203.

NSS Group, 2002. Intrusion Detection Systems http://www.nss.co.uk/ids/edition3/index.htm.

Jones A. K., Sielken R. S., 2000. Computer system intrusion detection: a survey.

http://www.cs.virginia.edu/~jones/IDS-research/Documents/jones-sielken-survey-v11.pdf.

Carvalho, D.R., Freitas, 2002, Genetic Algorithm with sequential with sequential niching for discovering small disjunct rules. *In proceedings of Genetic and Evolutionary Computation Conference.*