ISSN No. 0976-5697

Volume 4, No. 6, May 2013 (Special Issue)

International Journal of Advanced Research in Computer Science

REVIEW ARTICAL

Available Online at www.ijarcs.info

Misuse Detection System Using Various Techniques: A Review

Ms.R.S.Landge*, Mr.A.P.Wadhe
M.E (CSE) App*, M..E (CSE)
G.H.Raisoni College of Engineering & Management,Amravati
rslandge24@gmail.com*, aviwadhe@gmail.com

Abstract:: In the era of information systems and internet there is more concern rising towards information security in day to day life, along with the availability of the vulnerability assessment mechanisms to identifying the electronic attacks. Traditionally intrusion detection systems (IDS) are classified based on the style of detection they are Using. Ther are two main categories of intrusion detection (ID) those are misuse detection and anomaly detection.This paper mainly concerned about misuse detection system (MDS). In misuse detection, the IDS analyzes the information it gathers and compares it to large databases of attack signatures. Essentially, the IDS looks for a specific attack that has already been documented. Like a virus detection system, misuse detection software is only as good as the database of attack signatures that it uses to compare packets against.

Keywords: Intrusion detection system (IDS),Misuse detection system(MDS),pattern matching, Artificial neural network, data mining with fuzzy logic and genetic algorithms.

I. INTRODUCTION

The term `misuse' is herein defined in a broad sense as the use or behavior of a networked environment in any way that is not consistent with the system's expected functionality, as perceived by the provider of the network service. Misuse detection is also sometimes referred to as signature-based detection because alarms are generated based on specific attack signatures. This work focusses on the detection of such misuse events. The misuse is often that of unauthorized access of the system or using the system in an unauthorized way. In this case, the detection of such events is usually referred to as `intrusion detection' and the protection mechanism is called an Intrusion Detection System (IDS).

Intrusion Detection Systems (IDS) aim to detect the actions that attempt to compromise the confidentiality, availability, and integrity of a resource by monitoring the events occurring in computer systems and/or networks.

Examples of these include software engineering flaws in programs that allow cross privilege domain executions, insider abuse and failure of authentication procedures. Intrusion Detection models therefore do not directly overlap with traditional security models [1] which are primarily concerned with modeling information flow in a computer system to ensure that subjects are never able to access unauthorized information, or with modeling access control mechanisms to prevent unauthorized access to objects. Current approaches to detecting intrusions can be broadly classified into two categories: Anomaly Detection and Misuse Detection [2].

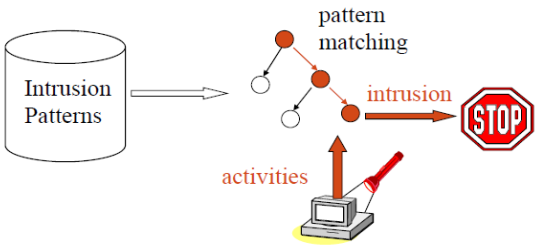

Figure 1. Misuse detection system with pattern matching

Anomaly Detection is based on the premise that intrusive activity often manifests itself as an abnormality. The usual approach here is to devise metrics indicative of intrusive activity, and detect statistically large variances on these metrics. Examples might be an unusually high number of network connections within an interval of time, unusually high CPU activity, or use of peripheral devices not normally used. This approach has been studied extensively and implemented in a large number of systems [3, 4, 5, 6,7, 8]. It attempts to quantify the acceptable behavior and thus identify abnormal behavior as intrusive. The other technique of detecting intrusions, misuse detection, attempts to encode knowledge about attacks as well defined patterns and monitors for the occurrence of these patterns. For example, exploitation of the fingerd and sendmail bugs used in the Internet Worm attack [9] would be in this category. This technique specifically represents knowledge about unacceptable behavior and attempts to detect its occurrence. This paper proposes a review to variation of approaches to misuse detection, by using pattern matching,data mining and artificial neural networks to detect system attacks.

A. Primary Approaches To Misuse Detection:

Misuse detection might be implemented by one the following techniques:

- ***a)*** ***Expert Systerns***, which code knowledge about attacks as if - then implication rules.
- ***b)*** ***Model Based Reasoning Systems***, which combine models of misuse with evidential reasoning to support conclusions about the occurrence of a misuse.
- ***c)*** ***State Transition Analysis***, which represents attacks as a sequence of state transitions of the monitored system [10, 11].
- ***d)*** ***Key Stroke Monitoring***, which uses user key strokes to determine the occurrence of an attack. These methods are summarized in the following sections.

a. Expert Systerns:

An expert system is defined in [12] as a computing system capable of representing and reasoning about some knowledge-rich domain with a view to solving problems and

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

giving advice. Expert system detectors code knowledge about attacks as if- then implication rules. Rules specify the conditions requisite for an attack in their if part. When all the conditions on the left side of a rule are satisfied, the actions on the right side of the rule are performed which may trigger the firing of more rules or conclude the occurrence of an intrusion. The main advantage in formulating if- then implication rules is the separation of control reasoning from the formulation of the problem solution. The primary disadvantage of using expert systems is that working memory elements (the fact base) that match the left sides of productions to determine eligible rules for firing are essentially
sequence-less [13] .

b. Model Based Systems:

This approach was proposed in [14] and is a variation on misuse intrusion detection. It combines models of misuse with evidential reasoning to support conclusions about its occurrence. There is a database of attack scenarios, where each scenario comprises a sequence of behaviors making up the attack. At any moment the system is considering a subset of these attack scenarios as likely ones being experienced by the system. It seeks to verify them by seeking information in the audit trail to substantiate or refute the attack scenario (the anticipator). The anticipator generates the next behavior to be verified in the audit trail, based on the current active models, and passes these behaviors to the planner. The planner determines how the hypothesized behavior will show up in the audit data and translates it into a system dependent audit trail match. This mapping from behavior toactivity must be easily recognized in the audit trail, and must have a high likelihood of appearing in the behavior. The advantage of model based intrusion detection is its basis in a mathematically sound theory of reasoning in the presence of uncertainty. The structuring of the planner provides independence of representation of the underlying audit trail syntax.

c. State Transition Analysis:

In this approach attacks are represented as a sequence of state transitions of the monitored system. States in the attack pattern correspond to system states and have Boolean assertions associated with them that must be satisfied to transit to that state. Successive states are connected by arcs that represent the events/conditions required for changing state. These conditions, or signature actions, are not limited to a single audit trail event, but may be a complex specification of conditions.

d. Keystroke Monitoring:

This technique uses user keystrokes to determine the occurrence of an attack. The primary means is to pattern match for specific keystroke sequences indicative of an attack. The disadvantages of this approach are the general unavailability of user typed keystrokes and the myriad ways of expressing the same attack at the keystroke level.

II. PATTERN MATCHING MODEL FOR MID

There are several benefits to our approach of using a generic model of matching. Pattern matching algorithms are a critical element in many signature based approaches.The term refers to the process of searching for certain patterns, expressedas sequences or tree structures within a body of data. It is most often employed to identify attack signatures in network packets [14].

Weaknesses of the approach include the computational load, as the number of comparisons required to ascertain the presence of a range of signatures within a large sample of traf_c can be large, and the use of _xed signatures, which limits detection to known cases [14]. A a generic model of matching based on Coloured Petri Nets (CPN), a form of Petri net where the arcs contain data and can be used to describe a variety of different system. The system separates the various concerns of a generic misuse detector into components as

A. The Information Layer:

This encapsulates the audit trail and provides a low-level data interface to the monitored computer system.

B. The Signature Layer:

This provides for a system-independent internal representation ofsignatures and a system-independent virtual machine to represent the signature context.

C. The Matching Engine:

This encapsulates the method used to match the patterns. It makes the system independent of any particular choice of matching algorithms. It also allows simple substitution of newer or more powerful mechanisms as they become available. The system uses a table of patterns to model the normal behaviour of such processes, based on audit events, and offers both off-line and real time detection. Kuri et al. [15] present a pattern matching approach based on insertion distance. This model handles an attack as a sequence of letters, and the algorithm detects the text portions everywhere the events of the attack appear, in order, within a window of k other events. This enables the algorithm to quickly _lter out large portions of the text and leave the remaining parts to be examined by another algorithm. The approach was evaluated using audit trails and an attack database, and was successful in addressing some of the problems of speed and complexity of intrusion detection algorithms.

III. ARTIFICIAL NEURAL NETWORKS FOR MDS

Artificial neural networks provide the potential to identify and classify network activity based on limited, incomplete, and nonlinear data sources. Here is an approach to the process of misuse detection that utilizes the analytical strengths of neural networks, and provide the results from our preliminary analysis of this approach. An artificial neural network consists of a collection of processing elements that are highly interconnected and transform a set of inputs to a set of desired outputs. The result of the transformation is determined by the characteristics of the elements and the weights associated with the interconnections among them. By modifying the connections between the nodes thenetwork is able to adapt to the desired outputs [16, 17]. However, the most important advantage of neural networks in misuse detection is the ability of the neural network to "learn" the characteristics of misuse attacks and identify instances that are unlike any which have been observed before by the network. A neural

network might be trained to recognize known suspicious events with a high degree of accuracy. While this would be a very valuable ability, since attackers often emulate the "successes" of others, the network would also gain the ability to apply this knowledge to identify instances of attacks which did not match the characteristics of previous intrusions. However, the most significant disadvantage of applying neural networks to intrusion detection is the "black box" nature of the neural network. Unlike expert systems which have hard-coded rules for the analysis of events, neural networks adapt their analysis of data in response to the training which is conducted on the network. The connection weights and transfer functions of the various network nodes are usually frozen after the network has achieved an acceptable level of success in the identification of events. While the network analysis is achieving a sufficient probability of success, the basis for this level of accuracy is not often known. The "Black Box Problem" has plagued neural networks in a number of applications [18].

A. Neural Network Description:

The first prototype neural network was designed to determine if a neural network was capable of identifying specific events that are indications of misuse. Neural networks had been shown to be capable of identifying TCP/IP network events in [19], but prototype was designed to test the ability of a neural network to identify indications of misuse. The prototype utilized a MLP architecture that consisted of four fully connected layers with nine input nodes and two output nodes. While there are a number of architectures that could be used to address this problem [17] a feed-forward neural network architecture was selected based on the flexibility and applicability of the approach in a variety of problems.

The number of hidden layers, and the number of nodes in the hidden layers, was determined based on the process of trial and error. Each of the hidden nodes and the output node applied a Sigmoid transfer function (**1/(1 + exp (-x))**) to the various connection weights. The neural network was designed to provide an output value of 0.0 and 1.0 in the two output nodes when the analysis indicated no attack and 1.0 and 0.0 in the two output nodes in the event of an attack.

Three levels of preprocessing of the data were conducted to prepare the data for use in the training and testing of the neural network. In the first round of preprocessing nine of the event record data elements were selected from the available set. The nine elements were selected because they are typically present in network data packets and they provide a complete description of the information transmitted by the packet:

- ***a) Protocol ID*** - The protocol associated with the event, (TCP = 0, UDP = 1, ICMP = 2, and Unknown = 3).
- ***b) Source Port*** – The port number of the source.
- ***c) Destination Port*** – The port number of the destination.
- ***d) Source Address*** - The IP address of the source.
- ***e) Destination Address*** - The IP address of the destination.
- ***f) ICMP Type*** – The type of the ICMP packet (Echo Request or Null).
- ***g) ICMP Code –*** The code field from the ICMP packet (None or Null).
- ***h) Raw Data Length*** – The length of the data in the packet.
- ***i) Raw Data*** - The data portion of the packet.

The second part of the preprocessing phrase consisted of converting three of the nine data elements (ICMP Type, ICMP Code and Raw Data) into a standardized numeric representation. The process involved the creation of relational tables for each of the data types and assigning sequential numbers to each unique type of element. This involved creating DISTINCT SELECT queries for each of the three data types and loading those results into tables that assigned a unique integer to each entry. These three tables were then joined to the table that contained the event records. A query was then used to select six of the nine original elements (ProtocolID, Source Port, Destination Port, Source Address, Destination Address, and Raw Data Length) and the unique identifiers which pertain to the remaining three elements (ICMP Type ID, ICMP Code ID, and Raw Data ID). A tenth element (Attack) was assigned to each record based on a determination of whether this event represented part of an attack on a network, (Table 1). This element was used during training as the target output of the neural network for each record.

Table 1: Sample of pre-processed events query

Protocol ID	Source Port	Destination Port	Source Address	Destination Address	ICMP Type ID	ICMP Code ID	Raw Data Length	Data ID	Attack
0	2314	80	1573638018	-1580478590	1	1	401	3758	0
0	1611	6101	801886082	-926167166	1	1	0	2633	1

The third round of data preprocessing involved the conversion of the results of the query into an ASCII comma delimited format that could be used by the neural network (Table 2).

Table 2: Sample of ASCII comma-delimited input strings

0,2314,80,1573638018,-1580478590,1,1,401,3758,0
0,1611,6101,801886082,-926167166,1,1,0,2633,1

The preprocessed data was finally loaded into the DataPro utility provided by Qnet 97.01, (Table 3). Qnet uses this application to load data into the neural network during training and testing.

Table 3: Sample of DataPro input to neural network

Input 1	Input 2	Input 3	Input 4	Input 5	Input 6	Input 7	Input 8	Input 9	Output 1
0	2314	80	1573638018	-1580478590	1	1	401	3758	0
0	1611	6101	801886082	-926167166	1	1	0	2633	1

IV. ID USING DATA MINING ALONG FUZZY LOGIC AND GENETIC ALGORITHMS

Based on fuzzy set theory, fuzzy logic provides a powerful way to categorize a concept in an abstract way by introducing vagueness. On the other hand, data mining methods are capable of extracting patterns automatically from a large amount of data. The integration of fuzzy logic with data mining methods will help to create more abstract patterns at a higher level than at the data level. Decreasing the dependency on data will be helpful for patterns used in intrusion detection.[20]

Although association rules and frequency episodes can be mined from audit data for anomaly intrusion detection, the mined rules or episodes are at the data level. This immediate dependency on data may limit the flexibility of intrusion detection. So, the machine learning component in IIDM will be designed to extract more abstract patterns at a higher level by integrating fuzzy logic with association rules

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

and frequency episodes. Data mining methods have the ability to find new patterns from a large amount of data automatically. Two data mining methods, association rules and frequency episodes, have been proposed to mine audit data to find normal patterns for anomaly intrusion detection. Association rules originate from retail data analysis in business. A piece of sales data, also called basket data, usually records information about a transaction, such as transaction date and transaction items.

A. Architecture:

The Hybrid Fuzzy logic IDS architecture has two modes of operations: rule-generation and detection. When operating in the rule-generation mode, the system processes network data and uses a fuzzy data mining algorithm to generate rules. A subset of the rules produced by the data mining algorithm is used as a model for the input data. The detection mode uses this rule subset for intrusion detection.

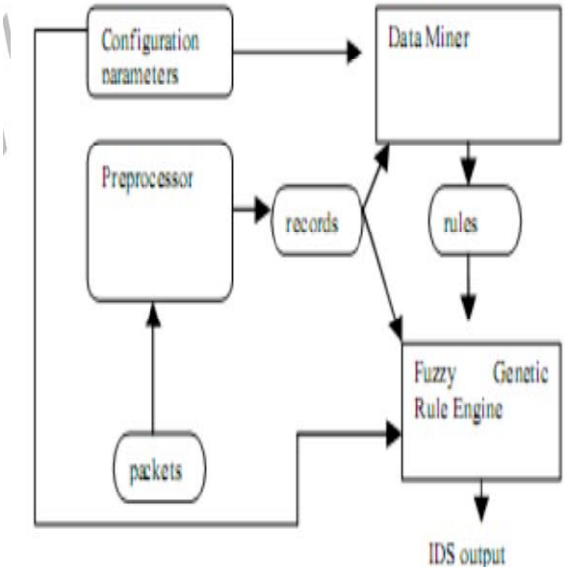

Figure 2: System Architecture

a. Pre-processors:-

The pre-processor is responsible for accepting raw packet data and producing records. This component is used in both modes and is capable of reading packets from the wire or a tcpdump file. The output produced by this component consists of records. Records contain aggregate information for a group of packets. Using records and concentrating only on attributes of interest greatly helps in reducing the amount of information to be used by more computationally intensive components of the architecture. Most of the approaches in the literature [1] differ on how those attributes are selected

b. Configuration Parameters:-

Parameter values stored in the configuration file regulate operation of the Data Miner and Fuzzy Inference Engine. The configuration file associates attributes with a term set and describes functions corresponding to the fuzzy membership functions associated with each term. The structured file identifies the number and names of attributes followed by a description of each attribute.

c. Data Miner

The Data Miner integrates Apriori to produce fuzzy rules. With one pass through the records, the fast and efficient algorithm used by the Data Miner extracts rules with sufficient support and confidence.

d. Algorithm

Input: Measurement from network traffic data and Threshold value for similarity Output:

Detected or null Assumptions:

a) The parameters for network intrusion are assumed which form the bases for the input
b) The existence of trained normal data set (in the experiment conducted, we have assumed the data of one timing is chosen as the normal trained set)

Step1: Identify and collect relevant data from network traffic.

Step2:Convert the quantitative feature of the data in step 1 into fuzzy sets

Step 3: Define membership function for fuzzy variable

Step 4: Apply genetic algorithm to identify the best set of rules.

Step 5: For each of the rules identified in the step 4 do

(a). Apply the fuzzy association rule algorithm to mine the correlation among them
(b). Apply fuzzy frequency algorithm to mine sequential patterns

Step 6: For each test case generate new patterns using the fuzzy association algorithm for same parameters

Step 7: For each new pattern, compare it with normal patterns created by Training data for similarity

Step 8: IF the similarity > the threshold value Then report “Detected” and the pattern.

V. LIMITATION OF MISUSE DETECTION

Current misuse detection systems usually work better than anomaly detection systems for known attacks. The better performance occurs because misuse detection systems take advantage of explicit knowledge of the attacks. The limitation of misuse detection is that it cannot detect novel or unknown attacks. As a result, the computer systems protected solely by misuse detection systems face the risk of being comprised without detecting the attacks. In addition, due to the requirement of explicit representation of attacks, misuse detection requires the nature of the attacks to be well understood. This implies that human experts must work on the analysis and representation of attacks, which is usually time consuming and error prone.

VI. CONCLUSION

Intrusion detection continues to be an active research field. Even after 20 years of research, the intrusion detection community still faces several difficult problems. How to detect unknown patterns of attacks without generating too many false alerts remains an unresolved problem, although recently, several results have shown there is a potential resolution to this problem. The evaluating and benchmarking of IDSs is also an important problem, which, once solved, may provide useful guidance for organizational decision makers and end users. Moreover, reconstructing attack scenarios from intrusion alerts and integration of IDSs will improve both the usability and the performance of IDSs. Many researchers and practitioners are actively addressing these problems. We expect intrusion detection to

CONFERENCE PAPER
“A National Level Conference on Recent Trends in Information Technology and Technical Symposium” On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India

become a practical and effective solution for protecting information systems.

VII. REFERENCES

[1]. Carl E. Landwehr. Formal Models for Computer Security. ACM Computing Surveys, 13(3):247-278, September 1981.

[2]. Stephen E. Smaha. Tools For Misuse Detection. In Proceedings of ISSA '93, Crystal City, VA, April 1993.

[3]. Stephen E. Smaha. Haystack: An Intrusion Detection System. In Fourth Aerospace Computer Security Applications Conference, pages 37-44, Tracor Applied Science Inc., Austin, TX, Dec 1988.

[4]. M. Sebring, E. Shellhouse, M. Hanna, and R. Whitehurst. Expert Systems in Intrusion Detection: A Case Study. In Proceedings ofthe 11th National Computer Security Conference, October 1988.

[5]. G. E. Liepins and H. S. Vaccaro. Anomaly Detection: Purpose and Framework. In Proceedings of the 12th National Computer Security Conference, pages 495-504, October 1989.

[6]. Teresa F. Lunt, R. Jagannathan, Rosanna Lee, Alan Whitehurst, and Sherry Listgarten. Knowledge based Intrusion Detection. In Proceedings of the Annual AI Systems in Government Conference, Washington, DC, March 1989.

[7]. L. T. Heberlein, K. N. Levitt, and B. Mukherjee. A Method To Detect Intrusive Activity in a Networked Environment. In Proceedings ofthe 14th National Computer Security Conference, pages 362-371, October 1991.

[8]. R. Jagannathan, Teresa Lunt, Debra Anderson, Chris Dodd, Fred Gilham, Caveh Jalali, Hal Javitz, Peter Neumann, Ann Tamaru, and Alfonso Valdes. System Design Document: Next-Generation Intrusion Detection Expert System (NIDES). Technical Report A007/A008/A009IAOIl1AOI2/AOI4, SRI International, March 1993.

[9]. Eugene Spafford. The Internet Worm Report. Technical Report 823, Purdue University, February 1990.

[10]. Phillip A. Porras and Richard A. Kemmerer. Penetration State Transition Analysis – A Rule-Based Intrusion Detection Approach. In Eighth Annual Computer Security Applications Conference, pages 220-229. IEEE Computer Society press, IEEE Computer Society press, November 30 - December 41992.

[11]. Koral Ilgun. USTAT: A Real-Time Intrusion Detection System for UNIX. Master's thesis, Computer Science Department, University of California, Santa Barbara, July 1992.

[12]. Peter Jackson. Introduction to Expert Systems. International Computer Science Series. Addison Wesley, 1986.

[13]. Teresa F Lunt. A Survey of Intrusion Detection Techniques. Computers & Security, 12(4):405-418, June 1993.

[14]. Network ICE. Protocol Analysis vs Pattern Matching in Network and Host Intrusion Detection Systems, 2000.

[15]. K. Kuri, G. Navarro, L. M, and L. Haye. A Pattern Matching based Filter for Audit Reduction and Fast Detection of Potential Intrusions. In Proceedings of the InternationalWorkshop on the Recent Advances in Intrusion Detection pages 17.27, Tolouse, France, 2000.

[16]. Fox, Kevin L., Henning, Rhonda R., and Reed, Jonathan H. (1990). A Neural Network Approach towards Intrusion Detection. In Proceedings of the 13th National Computer Security Conference.

[17]. Hammerstrom, Dan. (June, 1993). Neural Networks At Work. IEEE Spectrum. pp. 26-53.

[18]. Fu, L. (1992). A Neural Network Model for Learning Rule-Based Systems. In Proceedings of the International Joint Conference on Neural Networks. pp. (I) 343-348.

[19]. Tan, K.M.C & Collie, B.S. (1997). Detection and Classification of TCP/IP Network Services. In Proceedings of the Computer Security Applications Conference. pp. 99-107..

[20]. Munish Gupta, Manish Saxena, Prof. D.B.Singh, Prof. Rizwan Beg, The International Journal of computer Science and applications (TIJCSA)ISSN -2278—1080, Vol. 1 no.12 february 2013.

CONFERENCE PAPER
"A National Level Conference on Recent Trends in Information Technology and Technical Symposium" On 09th March 2013
Organized by
Dept. of IT, Jawaharlal Darda Inst. Of Eng. & Tech., Yavatmal (MS), India