A Data Mining Based Approach in IDS Design

[1]Rakesh Yadav, [2]Mahesh Malviya

[1,2]Dept. of Computer Science & Engg. jit Borawan, India

Abstract

Security is major issue now in these days in different application level as well as in the network level applications and utilities. This paper is based on a new approach based on process mining. In daily use we use various computer based application and interacted through different processes. Some of the process is well known and they provide support for smart works. But some processes are malicious and interrupting different kinds of applications, in this project we are going to introduce the malicious processes classification for using it over IDS development. For that purpose we make efforts for analysing different processes collected from the server to client's machines.

Keywords

Data Mining, Process Mining, Malicious Processes, Classification

I. Introduction

In this era of technology their various decisions are taken from the machine, this decision making capability of any machine is developed through machine learning concept. Using the machine learning concept we make such algorithms by which machine make intelligences decisions from the previously provided examples and learning sets. During the study of different kinds of data mining based IDS implementation we found that most of the systems uses the data base and they match data in sequential pattern. Due to matching sequentially the time complexity and memory complexity is effected, thus if we create a data structure and using the parameters we navigate this model than we can save time and memory resources by training these models once. Machine learning works in three main steps first data cleaning or pre-processing in this phase data is read from source and converted into a desired form of data which is easily recognized by the algorithm, in the next step algorithm accepts pre-processed data and build data model on the basis of the input training data set. After that in the third step a number of parameters are provided as input and expect the pattern as a classified way. This process in much similar to the finding a formula for a given series which formula satisfy all the similar kind or same kinds of series, thus we can say the data mining techniques take training data as input and arrange them into a structured form for suitable and easy navigation.

In this paper we make an IDS based on processing, an intrusion detection system examines all incoming and outgoing network activities and identifies doubtful patterns that may indicate a network or system attack from someone attempting to break into or compromise a system. In misuse detection, the IDS analyses the information it gathers and compares it to large databases of attack signatures. Fundamentally, the IDS look 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. In anomaly detection, the system administrator defines the baseline, or normal, state of the networks traffic load, breakdown, protocol, and typical packet size. The anomaly detector monitors network segments to compare their state to the normal baseline and look for anomalies.

In this section we provide the general introduction of working domain, in the next section we provide the problem identification of the studying domain, and solution obtained, in addition to here we provide the implementation of our proposed work and results obtained by us. Finally we conclude the complete designed system.

II. Background

IDS are split into two categories: misuse detection systems and anomaly detection systems (Anderson, 1980; Endorf et al., 2004). Misuse detection is used to identify intrusions that match known attack scenarios. However, anomaly detection is an attempt to search for malicious behavior that deviates from established normal patterns. In this paper our interesting is in anomaly detection.

Misuse Detection:

Characteristics: use patterns of well-known attacks (signatures) to identify intrusions, any match with signatures is reported as a possible attack

Drawbacks:

- False negatives
- Unable to detect new attacks
- Need signatures update
- Known attacks has to be hand-coded
- Overwhelming security analysts

Anomaly Detection:

Characteristics: use deviation from normal usage patterns to identify intrusions; any significant deviations from the expected behaviour are reported as possible attacks

Drawbacks:

- False positives.
- Selecting the right set of system features to be measured is ad hoc and based on experience
- Has to study sequential interrelation between transactions
- Overwhelming security analysts

Intrusion detection is the process of monitoring and analysing the data and events occurring in a computer and/or network system in order to detect attacks, vulnerabilities and other security problems [16]. IDS can be classified according to data sources into: host-based detection and network-based detection. In host-based detection, data files and OS processes of the host are directly monitored to determine exactly which host resources are the targets of a particular attack. In contrast, network-based detection systems monitor network traffic data using a set of sensors attached to the network to capture any malicious activities. Networks security problems can vary widely and can affect different security requirements including authentication, integrity, authorization, and availability. Intruders can cause different types of attacks such as Denial of Services (DoS), scan, compromises, and worms and viruses [17-18]. In this paper, we emphasize on network-based intrusion detection which is discussed in the next sub-section. The primary assumption in intrusion detection is that user and program activities can be monitored and modelled [16-17]. A set of processes represent the framework of intrusion detection, first, data files or network traffic are monitored and analysed by the system, next, abnormal activities are detected, finally, the system raises an alarm based on the severity of the attack [16]. Fig. 1 below shows a traditional framework for ID. In order for IDS to be successful, a system is needed to satisfy a set of requirements. IDS should be able to detect a wide variety of intrusions including known and unknown attacks. This implies that the system needs to adapt to new attacks and malicious behaviours. IDS are also required to detect intrusions in timely fashion, i.e., the system may need to respond to intrusions in real-time. This may represent a challenge since analyzing intrusions is a time consuming process that may delay system response. IDS are required to be accurate in a sense that minimizes both false negative and false positive errors. Finally, IDS should present analysis in simple, easy-to understand format in order to help analysts get an insight of intrusion detection results [16].

III. Proposed Work

During the study we found that some of the authors are provide the methods and techniques using the process analysis the misuse activity over system audit trials. Mining the audit trail transections they make security architectures. Thus in this paper we follow the same concept to mining the processes and their patterns to recognize the malicious processes access patterns. in the proposed system we work with the security of internal network and analyse the processes running on the different network machines. for that purpose we propose an agent based IDS development scheme, where all the network computers read their running processes and send them to the server end, server machine contains a multithreaded program that respond all the connected machines. And collect all the system processes and save them over a data base where all the incoming processes are stored.The server machine contains a decision mining algorithm that is trained using previously identified legitimate and malicious data patterns. this decision mining algorithm build a data model using the pre-classified data and using this data model upcoming data is analyzed.

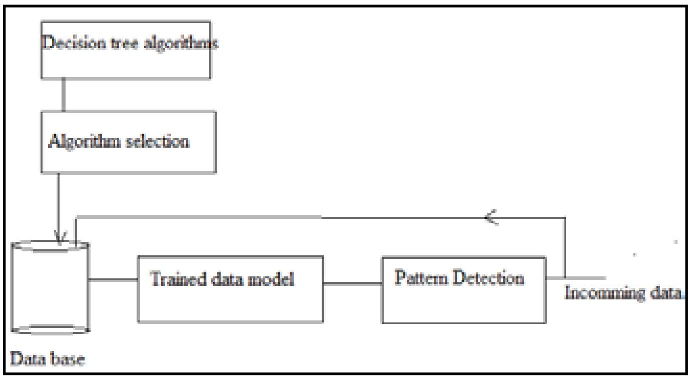

Fig. 1: Basic Server Analysis Flow

IV. Implementation

The proposed system is implemented through the visual studio dot net framework; this framework provides supports for multiple languages and a rich class library. Additionally the rich IDE contains various tools that support the implementation and design of any application using programmer friendly environment.

In this section includes the classes written, libraries used functions and methods which is used. These classes are some important classes that are used methods and functions.

Table 1: User Defined Classes

S. No.	Class Name	Description
1	ID3	This class contains different methods and functions that are used to implement ID3 decision tree
2	C4.5	This class contains all the methods and functions that are help to implement C4.5 algorithm
3	Server	This class interacted through the machines that active in internal network and help to fetch data from the machines
4	client	This class read data from the local machine and arrange them in string formatted to send it over the server system.
5	crossvalidation	This class is used to evaluate the performance parameters such as accuracy and error rate
6	frmMain	This is MDI form that contains all the project documents into a single window

Table 2: System Class Library

S No.	Class name	Description
1	System.Data. SqlClient	That is system library used to work with SQL server database
2	System.Runtime. Remoting.Channels. Http	This enables the channels for the network utility
3	System.Diagnostics	The System.Diagnostics namespace provides classes that allow you to interact with system processes, event logs, and performance counters.
4	System.Drawing	The System. Drawing namespace provides access to GDI+ basic graphics functionality. More advanced functionality is provided in the System.Drawing. Drawing2D, System. Drawing. Imaging, and System.Drawing.Text namespaces.
5	System.Linq	The System.Linq namespace provides classes and interfaces that support queries that use Language-Integrated Query (LINQ).

Table 3: Methods and Signature

S. No.	Method	Signature
1	ComputeEntropy (Instances data)	This function accepts the instance data and compute entropy of the supplied data
2	SplitData	That is decision tree utility and used to split supplied input data
3	BuildClassifier	This function is used to build the classifier from the selected data and algorithm
4	TotalSeconds	This function is used evaluate the time required to build the model and search any object in tree
5	EvaluateModel	This function classify data and measure the accuracy of the selected data model
6	GetCurrentProcess()	This function is used to collect the system running process
7	PeakWorkingSet64	This function is used to calculate the memory consumed by any selected processes

V. Results

As we know that the main ingredient of any decision tree is input data set. If the training data is much suitable then the model returns much better pattern detection. And the error rate is low. Here we measure the performance of our proposed system under different performance parameters.

Here the supplied data contains different processes running on the system and separated with the semi column, which is produces as the input to the system.

Accuracy

search accuracy of any decision tree algorithm is denoted using total number of instances supplied to evaluate and total correctly classified values; this parameter is evaluated in terms of %.

Table 4: Accuracy of C 4.5 and ID3 in %.

S. No.	C 4.5	ID3
1	78.27	72.33
2	82.18	78.21
3	78.32	83.87
4	72.37	87.23
5	83.29	87.02
6	87.27	82.23
7	78.32	73.77

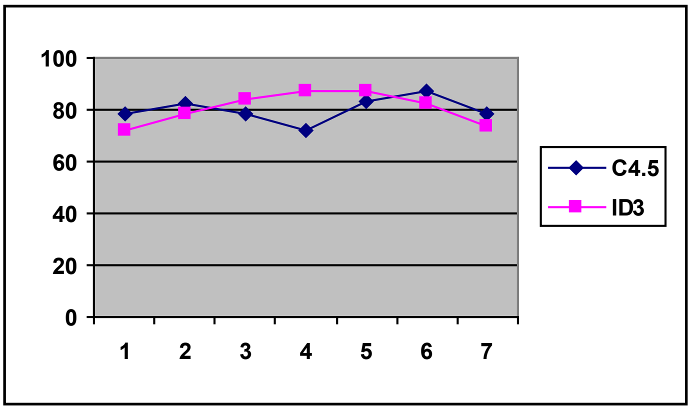

Fig. 2: Accuracy of C 4.5 and ID3 in %

Memory uses: memory uses is measured in terms of KB that is an amount of memory required to execute our system smoothly.

Table 4: Memory used by C 4.5 and ID3 in kb

S. No.	C4.5	ID3
1	37783	34887
2	37253	41980
3	38194	37387
4	38718	37826
5	32723	38232
6	38298	38373
7	38343	39827

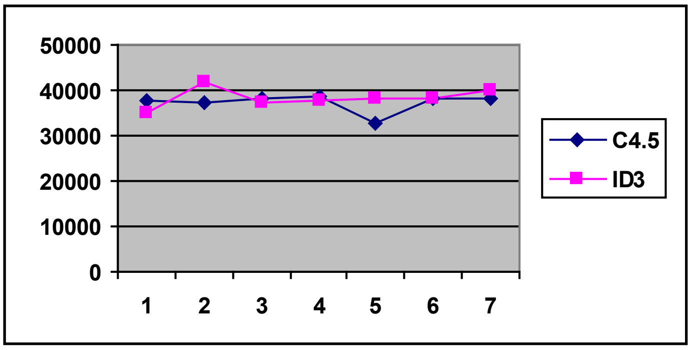

Fig. 3: Memory Used by C 4.5 and ID3 in kb.

Model build time: the time required to build model is known as model build time.

Table 5: Model Build Time of of C 4.5 and ID3

S. No.	C4.5	ID3
1	2.62	3.83
2	7.83	7.37
3	13.78	12.38
4	18.73	23.87
5	23.28	28.21
6	28.38	28.28
7	23.82	22.23

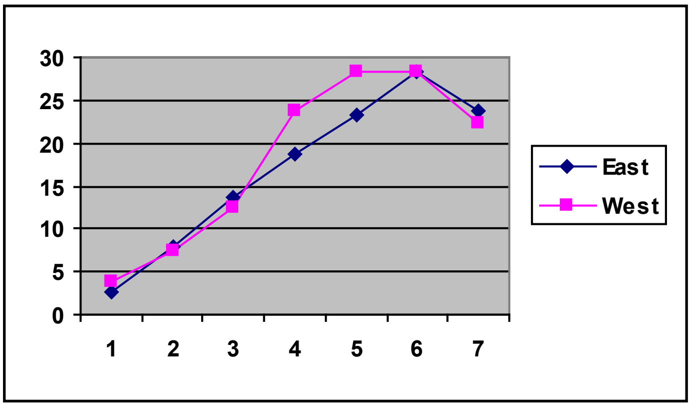

Fig. 4: Model Build Time of of C 4.5 and ID3

Due to performance analysis we found that the build time of the ID3 is better than the C4.5 additionally it consumes a constant amount of memory resources thus decision tree ID3 is better performing for our application. And for process analysis here we use the ID3 algorithm.

After implementation we found that the manually analysis of each pattern is much harder than the machine thus here we make changes over the proposed model and use a separate data base for containing the malicious processes by which we find the and analyse the malicious patterns and update the training data sets.

After examine the results using the different network architectures we found that our implemented model produces better performance analysis than other systems.

VI. Conclusion and Future Work

In this paper we propose a data mining based IDS implementation for misuse analysis of the processes, here we propose an agent based scheme for malicious process detection and the implementation of the system is performed using visual studio IDE, after implementation of the desired system we calculate the performance analysis over the different performance analysis parameters, that leads to implement system using the ID3 algorithm. after implementation we found that the updating of malicious process over the training data set manually is typical and complex task thus to improve the administrators efforts and improvement over performance of the classification we add a separate data base which contains different malicious processes, by using this data we update the training set of the system.

in this paper we keep focus to provide the processes mining and data mining based IDS implementation which is performed well but due to the small collection of malicious processes the performance of the IDS is varied over different systems and network parameters, in future we make a large collection of malicious processes and more data mining algorithms that provide much accurate results for classification of patterns.

References

[1] Process Mining and Security: Detecting Anomalous Process Executions and Checking Process Conformance, This is a preliminary version. The final version will be published in Electronic Notes in Theoretical Computer Science, [Online] Available: http://www.elsevier.nl/locate/entcs

[2] "Network Intrusion Detection Using Data Mining and Network Behaviour Analysis", International Journal of Computer Science & Information Technology (IJCSIT) Vol. 3, No. 6, Dec 2011.

[3] A Data Mining Framework for Building Intrusion Detection Models, This research is supported in part by grants from DARPA (F3060296-1-0311) and NSF(IRI-96-32225 and CDA-96-25374).

[4] Intrusion Detection Using Neural Networks and Support Vector Machines, 2002 IEEE.

[5] A new approach to intrusion detection using Artificial Neural Networks and fuzzy clustering, 0957-4174/$ - see front matter 2010 Elsevier Ltd. All rights reserved

[6] MAD-IDS: Novel Intrusion Detection System using Mobile Agents and Data Mining Approaches, ImenBrahmi, Sadok Ben Yahia, and Pascal Poncelet, Faculty of Sciences of Tunis, Tunisia sadok.benyahia@fst.rnu.tn

[7] ADAM:A Testbed for Exploring the Use of Data Mining in Intrusion Detection, SIGMOD Record, Vol. 30, No. 4, December 2001.

[8] Multi Agent Based Approach For Network Intrusion Detection Using Data Mining Concept, Vol. 3, No. 3, March 2012 Journal of Global Research in Computer Science RESEARCH PAPER Available Online at www.jgrcs.info

[9] Distributed Intrusion Detection System Using P2P Agent Mining Scheme, © 2012 Afr J Comp & ICT – All Rights Reserved www.ajocict.net © African Journal of Computing & ICT March, 2012.

[10] Model for Intrusion Detection System with Data Mining, International Journal of Advanced Research in Computer Engineering & Technology Vol. 1, Issue 4, June 2012

[11] Obtaining an Optimal MAS Configuration for Agent-Enhanced Mining Using Constraint Optimization, Chayapol Moemeng, Can Wang, Longbing Cao Quantum Computing and Intelligent Systems, Faulty of Engineering and Information Technology, University of Technology, Sydney.

[12] Neural Visualization of, Network Traffic Data for Intrusion Detection, EMILIO CORCHADO, and ÁLVARO HERRERO Departamen to de Informáticay Automática, Universidad de Salamanca Plaza de la Merced s/n, 37008, Salamanca, Spain escorchado@usal.es

[13] Learning Classifiers for Misuse and Anomaly Detection Using a Bag of System Calls Representation, Supported by NSF grant IIS 0219699

[14] CONDOR: A Hybrid IDS to Offer Improved Intrusion Detection, [Online] Available: http://www.shura.shu.ac.uk, Sheffield Hallam University Research Archive

[15] Improving the Performance Efficiency of an IDS by Exploiting Temporal Locality in Network Traffic, Govind Sreekar Shenoy, Jordi Tubella and Antonio Gonz´alez, Department of Computer Architecture, Universitat Polit`e cnicade Catalunya, Barcelona, Spain.

[16] A Semantic Approach to Host-based Intrusion Detection Systems Using Contiguous and Discontiguous System Call Patterns, Manuscript submitted on December 5, 2012, IEEE

Rakesh Yadav graduated from Department of Information Technology at Jawaharlal Institute of Technology, Borawan and M.E Scholar at JIT, Borawan and published his research work in 2 international (IJERT,IJCSMR) journals & gave presentations in conferences of the institutions in Indore like Truba College, Maharaja Ranjitsingh College, Oriental University, Indore etc and he conquered his major research work in Software Engineering from last Two years.

Mahesh Malviya is a Professor of Computer Science & Engineering, JIT,Borawan and published many Research Paper in prestigious Conference, Jounals.He gave Guidance to many M.E Scholars. His major research work in Software Engineering.