Intrusion Detection and Protection System by Using Data Mining

Virendra Chauhan[1], Romit Deshpande[2], Abhijeet Patil[3] and Amit Rana[4]

[1,2,3]Student, Dept. of Computer Engineering, Dilkap Research Institute And management Studies of Engineering and Technology, Neral, Maharashtra, India

[4]Asst. Professor Priya Deshpande, Dept. of Computer Engineering, Dilkap Research Institute And management studies of Engineering and Technology, Neral, Maharashtra, India

---***---

Abstract Currently, most computer systems use user IDs and passwords as the login patterns to authenticate users. However, many people share their login patterns with coworkers and request these coworkers to assist co-tasks, thereby making the pattern as one of the weakest points of computer security. Insider attackers, the valid users of a system who attack the system internally, are hard to detect since most intrusion detection systems and firewalls identify and isolate malicious behaviors launched from the outside world of the system only. In addition, some studies claimed that analyzing system calls (SCs) generated by commands can identify these commands, with which to accurately detect attacks, and attack patterns are the features of an attack. Therefore, in this paper, a security system, named the Internal Intrusion Detection and Protection System (IIDPS), is proposed to detect insider attacks at SC level by using data mining and forensic techniques. The IIDPS creates users' personal profiles to keep track of users' usage habits as their forensic features and determines whether a valid login user is the account holder or not by comparing his/her current computer usage behaviors with the patterns collected in the account holder's personal profile. The experimental results demonstrate that the IIDPS's user identification accuracy is 94.29%, whereas the response time is less than 0.45 s, implying that it can prevent a protected system from insider attacks effectively and efficiently.

1. INTRODUCTION

In the past decades, computer systems have been widely employed to provide users with easier and more convenient lives. However, when people exploit powerful capabilities and processing power of computer systems, security has been one of the serious problems in the computer domain since attackers very usually try to penetrate computer systems and behave maliciously, e.g., stealing critical data of a company, making the systems out of work or even destroying the systems.

Generally, among all well-known attacks such as pharming attack, distributed denial-of-service (DDoS), eavesdropping attack, and spear-phishing attack [1], [2], insider attack is one of the most difficult ones to be detected because firewalls and intrusion detection systems (IDSs) usually defend against outside attacks. To authenticate users, currently, most systems check user ID and password as a login pattern. However, attackers may install Trojans to pilfer victims' login patterns or issue a large scale of trials with the assistance of a dictionary to acquire users' passwords. When successful, they may then log in to the system, access users' private files, or modify or destroy system settings. Fortunately, most current host-based security systems [3] and network-based IDSs [4], [5] can discover a known intrusion in a real-time manner. However, it is very difficult to identify who the attacker is because attack packets are often issued with forged IPs or attackers may enter a system with valid login patterns. Although OS-level system calls (SCs) are much more helpful in detecting attackers and identifying users [6], processing a large volume of SCs, mining malicious behaviors from them, and identifying possible attackers for an intrusion are still engineering challenges.

Therefore, in this paper, we propose a security system, named Internal Intrusion Detection and Protection System (IIDPS), which detects malicious behaviors launched toward a system at SC level. The IIDPS uses data mining and forensic profiling techniques to mine system call patterns (SC-patterns) defined as the longest system call sequence (SC-sequence) that has repeatedly appeared several times in a user's log file for the user. The user's forensic features, defined as an SC-pattern frequently appearing in a user's submitted SC-sequences but rarely being used by other users, are retrieved from the user's Application usage history.

2. Literature Survey

We introduce model-based autonomic security management (ASM) approach to estimate, detect and identify security attacks along with planning a sequence of actions to effectively protect the networked computing system. In the proposed approach, sensors collect system and network parameters and send the data to the forecasters and the intrusion detection systems (IDSes). A multi-objective controller selects the optimal protection method to recover the system based on the signature of attacks. The proposed approach is demonstrated on several case studies including Denial of Service (DoS) attacks, SQL Injection attacks and memory exhaustion attacks. In this paper we have referred Autonomic Computing, Self-Protection, intrusion detection systems, Denial of Service (DoS) attacks, SQL Injection attacks and memory exhaustion attacks.[1]

This model is proposed for such an attack based on network traffic flow. In addition, a distributed mechanism for detecting such attacks is also defined. Specific network topology-based patterns are defined to model normal network traffic flow, and to facilitate differentiation between legitimate traffic packets and anomalous attack traffic packets. The performance of the proposed attack detection scheme is evaluated through simulation experiments, in terms of the size of the sensor resource set required for participation in the detection process for achieving a desired level of attack detection accuracy. In this paper we have referred Attack detection, Network traffic, packets, topology.[2]

We propose a novel approach to limiting pollution attacks by rapidly identifying malicious nodes. Our scheme can fully satisfy the requirements of live streaming systems, and achieves much higher efficiency than previous schemes. Each node in our scheme only needs to perform several hash computations for an incoming block, incurring very small computational latency. The space overhead added to each block is only 20 bytes. The verification information given to each node is independent of the streaming content and thus does not need to be redistributed. The simulation results based on real PP Live channel overlays show that the process of identifying malicious nodes only takes a few seconds even in the presence of a large number of malicious nodes. In this paper we referred Malicious nodes identification, Network coding, Peer-to-peer streaming.[3]

We design a practical trace back framework to identify active compromised mobiles in the mobile Internet environment in this letter. In the proposed framework, we creatively use the IMEI number of mobile hardware as unique marks for the trace back purpose. Two-layer trace back tables are designed to collect global attack information and identify local attacking bots, respectively. Our analysis and simulation demonstrate that the proposed trace back method is effective and feasible, and it can identify every possible attacking mobile in the current mobile Internet environment with single packet marking. In this paper we have referred Trace back, mobile Internet, and attack source.[4]

The Intrusion Detection and Identification System (IDIS), which builds a profile for each user in an intranet to keep track of his/her usage habits as forensic features. In this way the IDIS can identify who the underlying user in the intranet is by comparing the user's current inputs with the features collected in the profiles established for all users. User habits are extracted from their usage histories by using data mining techniques. When an attack is discovered, the IDIS switches the user's inputs to a honey pot not only to isolate the user from the underlying system, but also to collect many more attack features by using the honey pot to enrich attack patterns which will improve performance of future detection. In this paper we have referred Forensic Features, Intrusion Detection, Data Mining, Identifying Users.[5]

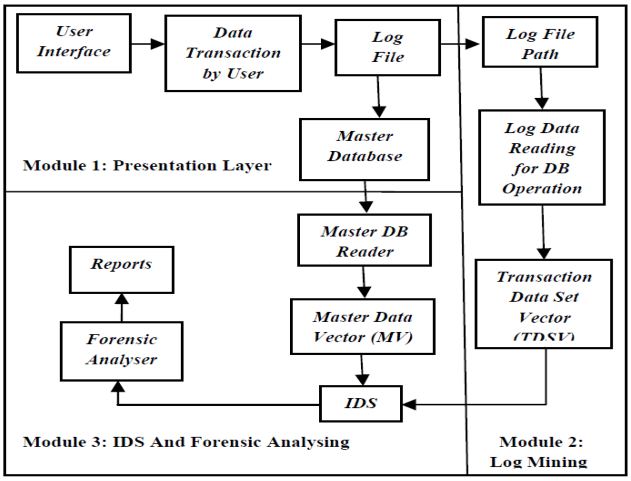
We proposed A new logging-based IP trace back approach using data mining techniques IP Trace back is a way to search for sources of damage to the network or host computer. IP Trace back method consists of reactive and proactive methods, and the proactive method induces a serious storage overhead. However, a system capable of solving these problems through cluster-based mass storage, digestible packets and hierarchical collections was designed. It not only performs trace back but also communicates with analysis data of other security systems by using the logging methods. In this paper we have referred IP Trace back, logging-based approach, data mining.[6]

We analyze the possibility of using data stream mining for enhancing the security of AMI through an intrusion detection system (IDS), which is a second line of defense after the primary security methods of encryption, authentication, authorization, etc. We propose a realistic and reliable IDS architecture for the whole AMI system, which consists of individual IDSs for three different levels of AMI's components: smart meter, data concentrator, and AMI head end. We also explore the performances of various existing state-of-the-art data stream mining algorithms on a publicly available IDS data set, namely, the KDD Cup 1999 data set. Then, we conduct a feasibility analysis of using these existing data stream mining algorithms, which exhibit varying levels of accuracies, memory requirements, and running times, for the distinct IDSs at AMI's three different components. In this paper we have referred Advanced metering infrastructure (AMI), data stream mining, intrusion detection system (IDS), smart grid (SG).[7]

We design a method of integration between HTTP GET flooding among DDOS attacks and Map Reduce processing for a fast attack detection in cloud computing environment. This method is possible to ensure the availability of the target system for accurate and reliable detection based on HTTP GET flooding. In experiments, the processing time for performance evaluation compares a pattern detection of attack features with the Snort detection. In this paper we have referred DDoS Attack, HTTP GET Flooding Attack, Web Security, Map Reduce.[8]

3. Proposed System

System Architecture



The proposed database intrusion detection system consists of log mining mechanism and an intrusion detection mechanism. In this we are using vector concept for detection of intrusion. Vector is array list with extended properties which follows dynamic and automatic addition of data at run time. So it reduces the computations. In this we are mining log file for comparison purpose to detect intrusion. Initially, system copies the contents from log file into temporary file as no one can perform operations on log file directly. Then with original database the comparison is carried out. And intrusion is detected if any and report is generated which gives field where the intrusion is occurred and also gives date and time.

So the system follows:

1. To allow Users to perform transactions.
2. Provides the facility to read log file and collect all transaction data.
3. Provides a facility to collect all infected data from Master DB.
4. Provides a facility to detect tamper detection.
5. Provides a facility to perform forensic analysis on tampered data.

As a result our system works faster with better performance.

4. Methodology

Consists of three modules:-

1. Presentation Layer: - This module is for the most part identified with User Interface (UI). In this client cooperate with framework through ASPX pages. So in this client gives the info to the framework and this information go through the log document and put away into the expert database.
2. Log Mining: - This module is identified with log document filtering. In this as of now created log record is perused by the framework. So by checking the substance for each database exchange, it makes the exchange information set vector (TDSV) for each exchange.
3. IDS and Forensic Analysis: - In this module, the framework will identify the interruption if happened and will create the report for interruption. In this, framework will produce the expert information set vector (MV) for every exchange in the expert database. Utilizing MV and TDSV it will recognize the interruption happened in the expert database and will produce the report where the really the interruption happened, when and by whom.

5. Feasibility Studies

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study activity is to determine whether it would be financially and technically feasible to develop the product.

The feasibility study activity involves the analysis of the problem and collection of all relevant information relating to the product such as the different data items which would be input to the system, the processing required to be carried out on these data, the output data required to be produced by the system as well as various constraints on the behavior of the system.

5.1 Technical Feasibility

This is concerned with specifying equipment and software that will successfully satisfy the user requirement. The technical needs of the system may vary considerably, but might include: The facility to produce outputs in a given time, Response time under certain conditions, Ability to process a certain volume of transaction at a particular speed, Facility to communicate data to distant locations etc.

In examining technical feasibility, configuration of the system is given more importance than the actual make of hardware. The configuration should give the complete picture about the system's requirements: Is project technically feasible? Is it within state of the art? Can defects be reduced to a level of matching the application's need? All such questions are studied in technical feasibility.

5.2 Economic Feasibility

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as Cost / Benefit analysis, the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs. If benefits outweigh costs, a decision is taken to design and implement the system. Otherwise, further justification or alternative in the proposed system will have to be made if it is to have a chance of being approved. Also is project financially feasible? Can development be completed at the cost of organization, its client, or the market can afford? Such questions are studied.

5.3 Operational Feasibility

Operational feasibility is dependent on human resources available for the project and involves projecting whether the system will be used if it is developed and implemented. Operational feasibility is a measure of how well a proposed system solves the problems, and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase of system development. The essential questions that help in testing the operational feasibility of a system include the following:

Does current mode of operation provide adequate throughput and response time? Does current mode provide end users and managers with timely, pertinent, accurate and useful formatted information? Does current mode of operation provide cost-effective information services to the business? Could there be a reduction in cost and or an increase in benefits? Does current mode of operation offer effective controls to protect against fraud and to guarantee accuracy and security of data and information? Does current mode of operation make maximum use of available resources, including people, time, and flow of forms? Does current mode of operation provide reliable services? Are the services flexible and expandable? Are the current work practices and procedures adequate to support the new system? etc.

6. Cost Analysis

Software cost estimation is the process of predicting the effort required to develop a software system. Many estimation models have been proposed over the last 30 years. Most cost estimation models attempt to generate an effort estimate, which can then be converted into the project duration and cost. Although effort and cost are closely related, they are not necessarily related by a simple transformation function. Effort is often measured in person months of the programmers, analysts and project managers. This effort estimate can be converted into a dollar cost figure by calculating an average salary per unit time of the staff involved, and then multiplying this by the estimated effort required. We are using COCOMO model for cost analysis.

The Constructive Cost Model (COCOMO) is an algorithmic software cost estimation model developed by Barry W. Boehm. The model uses a basic regression formula with parameters that are derived from historical project data and current project characteristics. There are three types of COCOMO model, we will be using basic model for cost estimation. Basic COCOMO computes software development effort (and cost) as a function of program size. Program size is expressed in estimated thousands of source lines of code (SLOC). COCOMO applies to three classes of software projects. Class of our project is Organic projects in which "small" teams with "good" experience working with "less than rigid" requirements.

The basic COCOMO equations take the form

Effort Applied (E) = $a_b(KLOC)^{b_b}$ [man-months]

Development Time (D) = $c_b(\text{Effort Applied})^{d_b}$ [months]

People required (P) = Effort Applied / Development Time [count]

where, KLOC is the estimated number of delivered lines (expressed in thousands) of code for project. The coefficients a_b, b_b, c_b and d_b are given in the following table:

Software project	a_b	b_b	c_b	d_b
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

Constant values for COCOMO Model

Basic COCOMO is good for quick estimate of software costs. However it does not account for differences in hardware.

7. System Requirements

HARDWARE REQUIREMENTS:

- 1 GB RAM.
- 200 GB HDD.
- Intel 1.66 GHz Processor Pentium 4

SOFTWARE REQUIREMENTS:

- Windows XP, Windows 7,8
- Visual Studio 2013
- MS SQL Server 2008 R2

8. Conclusion and Future Scope

In this paper, an approach that utilizes data mining and forensic techniques to identify the representative SC-patterns for a user is proposed. The time that a frequent SC- pattern appears in the user‘s log file is counted, the most frequently used SC-patterns are filtered out, and then a user‘s profile is established. By identifying a user‘s SC-patterns as his/her application usage habits from the user‘s current input SCs, the IIDPS resists suspected intruders.

9. REFERENCES

1] Q. Wang, L. Vu, K. Nahrstedt, and H. Khurana, “MIS: Malicious nodes identification scheme in network-coding-based peer-to-peer streaming,” in Proc. IEEE INFOCOM, San Diego, CA, USA, pp. 1–5.

2] S. Yu, K. Sood, and Y. Xiang, “An effective and feasible trace back scheme in mobile internet environment,” IEEE Commun.Lett., vol. 18, no. 11, pp. 1911–1914, Nov. 2014.

3] F. Y. Leu, K.W. Hu, and F. C. Jiang “Intrusion detection and identification system using data mining and forensic techniques,” Adv. Inf. Comput. Security, vol. 4752, pp. 137–152.

4] H. S. Kang and S. R. Kim, “A new logging-based IP trace back approach using data mining techniques,” J. Internet Serv. Inf. Security, vol. 3, no. 3/4, pp. 72–80, Nov. 2013.

5] M. A. Faisal, Z. Aung, J. R. Williams, and A. Sanchez, “Data-streambased intrusion detection system for advanced metering infrastructure in smart grid: A feasibility study,” IEEE Syst. J., vol. 9, no. 1, pp. 1–14.

6] J. Choi, C. Choi, B. Ko, D. Choi, and P. Kim, “Detectingweb based DDoS attack using MapReduce operations incloud computing environment”, J. Internet Serv. Inf.Security, vol. 3, no. 3/4, pp. 28–37.

7] Q. Chen, S. Abdelwahed, and A. Erradi, “A model-basedapproach to self-protection in computing system",inProc. ACM Cloud Autonomic Comput. Conf., Miami, FL, USA, pp. 1–10.

8] Z. A. Baig, "Pattern recognition for detecting distributednode exhaustion attacks in wireless sensor networks",Comput. Commun., vol. 34, no. 3, Mar.

9] Z. Shan, X. Wang, T. Chiueh, and X. Meng, "Safe sideeffects commitment for OS-level virtualization", in Proc.ACM Int. Conf. Autonomic Comput., Karlsruhe,Germany, pp. 111–120