International Journal of Information Technology and Knowledge Management
July-December 2010, Volume 2, No. 2, pp. 651-656

Database Intrusion Prevention Cum Detection System with Appropriate Response

Alka Jaiswal[1] & Sweta Jain[2]

Web based applications such as search engines, web mail, shopping carts and portal system are extensively used nowadays. Such technological advancement had not only led enterprises to develop it for being proficient but at the present they are heavily dependent on it. The attackers knowing increase in availability of such services are trying to search weaknesses in the system to gain access and perform malicious activities. Most intrusions are committed from within the organization by employees. That's why defending database against both internal and external attacks is becoming more vital. Database Intrusion Detection System can be deployed to detect potential violations in database security and to minimize the risk of attacks. In this paper, we have explored about the various vulnerabilities to database, the different types of attacks and the existing intrusion detection techniques for database system. The architecture of Database Intrusion Prevention cum Detection System with appropriate Response has also been proposed. The proposed architecture uses Genetic Algorithm for intrusion detection.

Keywords: Intrusion Prevention, Anomaly Detection, Misuse Detection, Genetic Algorithm.

1. Introduction

Intrusion literally means interrupting or interfering in others work. In a better way it can be defined as any set of actions that attempts to compromise integrity, confidentiality and availability of resource. Intrusion detection is a security technology that attempts to identify either individual who is trying to break into system and misuse information without authorization and\or those who have legitimate access to the resource but are taking undue advantage of their rights [1].

The job of Intrusion Detection System (IDS) is to dynamically monitor the events occurring in a system and alert when any suspicious activity occurs so that defensive action can be taken to prevent or minimize damage. In general, the main goal of IDS is to detect malicious transactions before they are being committed and then dropping and rolling them back. If the malicious transactions have been committed and have caused damages, then locating the damaged parts and repairing them on time will be much more problematic. Intrusion detection systems serve three essential security functions: they monitor, detect and respond to unauthorized activity.

Security in Database System

Database security is vital nowadays as database systems contain valuable information. Database security refers to protecting database from malicious attacks or accidental changes. Organizations maintain databases that contain list of customer relevant information like name, address, SSN, credit card information, medical records, payroll information, trade secrets and employee records or any financial transaction information or private documents. This information is also available over network. This has led to the deployment of database driven web applications in enterprise and on the internet. Databases serve as groundwork for business systems such as web servers and Enterprise Resource Planning applications [2]. Database intrusion is a major threat to any organization storing above mentioned valuable and confidential data in databases. The intrusive activities are going on increasing more and more as the number of database servers connected to the Internet are increasing rapidly. The extensive use of database systems makes it decisive to detect any intrusion or intrusion attempts made at data base level. There are some major reasons that motivated the development of ID systems at application layer. Firstly, actions malicious for a database application may not be essentially malicious for the network or the operating system. Secondly, ID systems designed for networks and operating systems are not adequate to protect databases against insider threats, which is an important issue when dealing with privacy [3].

Database systems are targeted either by direct access from internet or by SQL injection in web application. Direct access here means abusing vulnerabilities by guessing passwords or by buffer overflow. SQL injection is a form of attack on a database-driven web site in which the attacker executes unauthorized and well crafted SQL commands by taking advantage of insecure code on a system connected to the Internet, bypassing the firewall. As firewall is behind a database, it does not mean that it is not needed to worry about the database being attacked. There are several other

[1,2]Department of Computer Science & Engineering, MANIT, Bhopal, INDIA

Email: [1]alka_jais@yahoo.co.in, [2]shweta_j82@yahoo.co.in,

forms of attack that can be made through the firewall. The most common of these attacks is SQL Injection which is not an attack directly on the database but is caused by the way in which web applications are developed. The most commonly used SQL injection technique involves access either via login page or via URL [4]. They can be mitigated by adopting suitable safeguards, for example, by adopting defensive programming techniques and by using SQL prepare statements.

The rest of paper is organized as follows. This section further describes about the threats and type of attacks to the database. Detection techniques along with their advantages and disadvantages are mentioned in Section 2. Section 3 refers to the previous database related intrusion detection systems. Finally, Section 4 specifies about the proposed work.

Basic Threats: Vulnerability means a weakness or fault in a system or protection mechanism that exposes information to attack or damage. Vulnerabilities to database system can be classified as [5]:

- Vendor Bugs: Buffer overflows and other programming errors resulting after the execution of allowable commands. Downloading and applying patches usually fix vendor bugs.
- Poor Architecture: Results due to improper designing. This problem is the hardest to fix because it requires a revise and major modifications by the vendor. For example, architecture utilizing weak form of encryption.
- Misconfigurations: It is caused by not properly locking down databases. Many of the configurations options of databases can be set in a way that compromises security. Some of these parameters are set insecurely by default. Most are not a problem unless one unsuspectingly changes the configuration.
- Incorrect Usage: Incorrect usage refers to building programs using developer tools in ways that can be used to break into a system. An example of incorrect usage is SQL injection. The attacker can execute arbitrary SQL commands on the backend database server through the web application.

Types of Attacks: Attacks can be used to disclose information, to sidestep authentication mechanisms, to alter the database, and to execute arbitrary code, in certain instances, on the database server itself.

On the basis of relationship between intruder and victim, attacks to the database can be classified as:

- Insider: An authorized user, who can be from own enterprise's employees or their business partners or customers, misuses his privilege or performs unauthorized access. They have privileges to access the application or system but misuse it and are usually harder to defend.
- Outsider: An unauthorized user coming from outside, frequently via the Internet who tries to gain access to system. They do not have proper rights to access the system and can be defended using strong security mechanisms.

A more generic classification of attack is presented in [6]:

- Attempted Break Ins: When an unauthorized user tries to gain access to a computer system. Most often detected by typical behavior profiles or violations of security policies.
- Masquerader (Internal) Attacks: When an authorized user pretends to be as another user. These attacks are also called internal because they are caused by already authorized users. It is also detected by a typical behavior profiles or violations of any security policies.
- Penetration Attack: Usually detected by monitoring for specific patterns of activity like when a user attempts to directly violate the system's security policy.
- Leakage: Moving potentially sensitive data from the system. Mostly detected by a typical usage of I/O resources.
- Denial of Service: Denying, by making the resources unavailable to other users. Often detected by a typical usage of system resources like denying other users, the use of system resources by making them unavailable.
- Malicious Use: It involves various attacks such as file deletion, viruses etc. Often detected by typical behavior profiles, violations of security policies, or use of special privileges.

2. DETECTION TECHNIQUES

There are two common approaches in database IDSs to detect intrusions when focusing primarily on audit data and access pattern: misuse-based intrusion detection and anomaly-based intrusion detection.

Misuse-Based Intrusion Detection System/ Knowledge Based detection:

Such systems use well-known attack patterns or signatures as the basis for detection. It tries to identify activities matching a signature stored in a database. Whenever a match is found an alarm is triggered.

Advantage:

- False alarm rate is very low. Any action if found unclear is not allowed. Hence, their accuracy is very high.
- Easy creation of attack signature databases.

Disadvantage:

- Created attack signatures may not cover all attacks as new attacks are hard to forecast.
- Updation of signature database is difficult [7, 8].

The creation of the well-known attack patterns or signatures is a tedious, manual process that requires detailed knowledge of each software exploit that is supposed to be captured. Simplistic signatures tend to generate large numbers of false positives and too specific ones cause false negatives. The user activity is compared against a repository of signatures that define characteristics of an intrusion.

The purpose of attack signatures is to describe the essential features of attacks. For a signature to be good, signature must be narrow enough to capture precisely the characteristic aspects of exploit, it attempts to address and at the same time, it should be flexible enough to capture variations of the attack. Failing to generate a good signature can cause either large amounts of false positives or false negatives.

Anomaly-Based Intrusion Detection System/ Behavior Based Detection:

These systems use user profile as the basis for detection. When it detects any sufficient deviation of the recent activity from the normal profile of activities or expected behavior of user then it considers such an activity as intrusion. Then, immediately it generates an alarm to take appropriate action.

Advantages:

- It can detect attempts that try to exploit new and unexpected vulnerabilities.
- Anomalies are recognized without getting inside the causes and characteristics.
- Ability to detect abuse of user privileges.

Disadvantages:

- It may have very high false alarm rate.
- The broad knowledge of expected user behavior is required which makes it difficult to implement.
- User behaviors can vary with time, thereby requiring a constant update of the normal behavior profile database [7, 8, 9].

But intrusive activity does not always coincide with anomalous activity. There are four possibilities, each with a non-zero probability:

1. False Negatives: These are intrusive but not anomalous. That is, the activity is intrusive but because it is not anomalous we fail to detect it. These are called false negatives because the intrusion detection system falsely reports absence of intrusions.
2. False Positives: These are not intrusive but anomalous. That is, the activity is not intrusive, but because it is anomalous, we report it as intrusive. These are called false positives because the intrusion detection system falsely reports intrusions.
3. True Negatives: These are not intrusive and not anomalous. The activity is not intrusive and is not reported as intrusive.
4. True Positives: These are intrusive and anomalous. The activity is intrusive and is reported as such because it is also anomalous.

3. EXISTING IDS IN DBMS

In this section, some of existing database intrusion detection systems as proposed in the literature is described in brief:

In 1999 a Misuse Detection System for Database System (DEMIDS) has been proposed by Chung et al [10]. It is a misuse-detection system, especially for relational database systems. It uses audit data log to derive profiles describing typical behavior of users in DBMS. The profiles derived are then used to detect misuse behavior. In this, the concept of distance measure has been introduced to search frequent item sets that describes the working scope of users. A data mining algorithm called Frequent Item set Profiler has been used to discover the frequent item sets from an audit session.

In 2000, a method which has used time signatures is presented by Lee et al. for discovering database intrusions [11]. It involves tagging the time signature to data items. A security alarm is raised when a transaction attempts to write a temporal data object that has already been updated within a certain period.

In 2002, another similar work, Detecting Intrusion in Databases through Fingerprinting Transactions (DIDAFIT) was proposed by Low et al. [12]. A system is developed using misuse detection approach (signature-based) to perform database intrusion detection at the application level. SQL statements are summarized as regular expressions which are considered to be 'fingerprints' for legitimate transactions. This system works by fingerprinting access patterns of legitimate database transactions, and using them to identify illegitimate accesses.

In 2003, yet another relevant approach towards a database specific intrusion detection mechanism is given by Hu and Panda [13]. It is capable of finding data dependency relationships among transactions and uses this information to find hidden anomalies in the database's log. The dependencies are determined by using the read, pre-write, and post-write sets of data items which are generated by the static semantic analyzer.

Peng Liu introduced an Intrusion Tolerant Database System in [14] which can operate through attacks in such a way that the system can continue delivering essential services in the face of attacks. It extends traditional secure system to be able to survive attacks.

Elisa Bertino et al. introduced intrusion detection in RBAC-administered Databases [15]. The approach used is based on the well-known role-based access control (RBAC) model. In this approach, authorizations are specified with respect to roles and not with respect to individual users. It deals with building and maintaining role profiles representing accurate and consistent user behavior; and then using these profiles for the intrusion detection. An attempt by one user-role to execute a query associated with another role indicates anomalous behavior and a possible attempt at masquerade.

In 2005, Ke Chen et al. presented an intrusion detection model for a database system based on digital immunity in [16]. It provides an additional layer of defense against DBMS misuse, especially malicious transactions. Such a intrusion detection model can produce antibodies through immune analysis, and then send them out to all the modules respectively to enhance resistance of the whole database system. This speeds up the detection of malicious transaction attacks and improves its accuracy without causing performance degradation.

In 2006, Rietta proposed an application layer intrusion detection system, which should take the form of a proxy server and employ an anomaly detection model which can be used in combination with the existing methods to give the database server a way to mitigate the SQL injection risk. The detection model is based on specific characteristics of SQL and the transaction history of a particular user and application [17].

Ashish Kamra et al. presented mechanisms for database intrusion detection and response in [18]. They tried to develop advanced security solutions for protecting the data residing in a DBMS. The strategy was to develop an Intrusion Detection (ID) mechanism, implemented within the database server that is capable of detecting anomalous user requests to a DBMS. The key idea is to learn profiles of users and applications interacting with a database and database requests that deviates from these profiles are then termed as anomalous.

In 2008, Aziah Asmawi et al. proposed a (SIIMDS) SQL Injection and Insider Misuse Detection System [3]. They tried to tackle both kinds of intrusions from internal and external threats in order to provide a high level of security to database system.

In 2009, Gongxing Wu and Yimin Huang proposed a new Intrusion Detection System for database systems [19]. It involves the characteristics of both misuse detection and anomaly detection. It applies the Apriori algorithm based on Trie tree to the database IDS.

Sunu Mathew et al. proposed a Data-Centric Approach to Insider Attack Detection in Database Systems [20] in 2009. In this, the database access pattern of user is profiled by looking at exactly what the user accesses.

A database intrusion detection system architecture based on the database communication content detection mechanism was proposed by Yawei Zhang et al. in 2009 [21]. They build a practical DIDS based on the method of pattern recognition, which could not only detect user defined database intrusions, but also monitor and audit communication contents between clients and databases.

4. PROPOSED ARCHITECTURE FOR DATABASE INTRUSION DETECTION SYSTEM

An Intrusion Detection System should involve some preventive measures at the beginning in order to put a stop to the entrance of intruders into system. Then detection method should be used, to identify if any intruder had successfully bypassed the preventive step. If any intrusion has been detected then an appropriate response should be taken to defense from the attacker. In brief it can be said to have: Intrusion Prevention + Intrusion Detection + Appropriate Response in a database system to get rid of intruders. It will be better to include role based access and use data mining technique for intrusion detection in the IDS for producing more fine results.

In a role based access system, roles are created for various job function and the permissions to perform certain operations are assigned to specific roles. Through the role assignment, the users acquire the permissions to perform particular system functions. In an ID system having role based access, it is easy to grab intruders as any individual holding a specific role if behaves differently from the normal behavior of the role. A benefit of role-based systems that apply to groups of users instead of individual users is that such a system provides greater flexibility and is easier to manage to support large and dynamic organizations. With data mining it will be easy to correlate data related to alarms with mined audit data, thereby considerably reducing the rate of false alarms. Data that results from errors or that represent unusual activities should be examined carefully because they may carry important information.

Speed, accuracy and adaptability are the common problems in IDS. The extensive amount of data that intrusion detection systems need to monitor in order to observe the entire situation causes speed problem. To handle this situation, the most important portion of information should be extracted in order to provide efficient detection of attacks. The adaptation and accuracy issues of the intrusion detection can be solved by incorporating learning algorithms. In case of intrusion detection, learning means discovering patterns of normal behavior or pattern of attacks. In this way intrusion detection can combine the advantages of both signature-based and anomaly-based IDS. The block diagram of a hybrid Database IDS is shown in figure.

From the figure, it is clear that initially the user sends service request via web based application, to the application server. The application server issues request to the database. Then, user can log to database. At this level, preventive measures can be taken by using exact password entry log ins in role based authentication and analyzing the requested https. As the user logs into the database, the database session gets started and the SQL statements that are received from the application are passed to the detection module. Detection phase is divided into two modules. Firstly, the statements are passed to the Misuse Detection Module where they are matched against a set of SQL injection signatures. If matched, then result is passed to response module to take appropriate action. And if not matched, the statements are forwarded to the Anomaly Detection Module where the user activities are compared with the normal profiles to determine if there is any significant deviation. Genetic algorithm will be used for intrusion detection. Genetic algorithms (GA) are search algorithms based on the principles of natural selection and genetics.

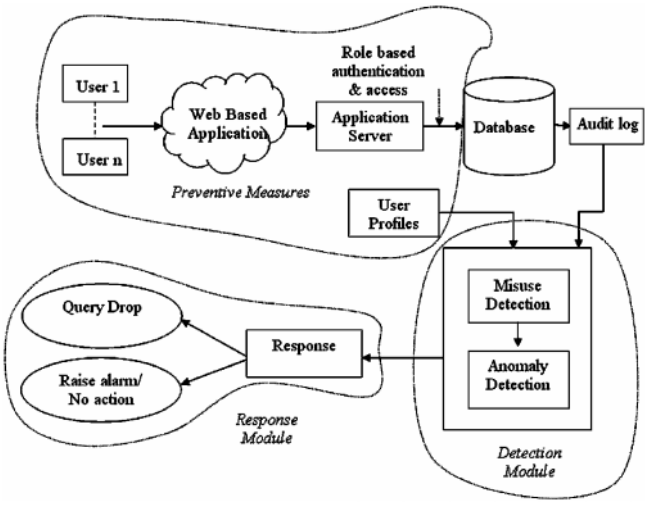

Fig: Database Intrusion Prevention cum Detection System with Appropriate Response

Genetic algorithm will be employed in the detection module, to find exact or approximate solutions to optimization and search problems [22]. The features like number of attempts taken to log in, basic data access operations: select, insert, update, delete and clauses used like having, equal to etc extracted from the audit log will be considered in the genetic algorithm. GA operates on a population of potential solutions applying the principle of the survival of the fittest to produce better and better approximations to the solution of the problem that GA is trying to solve. Now after detecting intrusions, appropriate actions can be provided as response by using response module. The response may be either dropping the query or raising alarm to alert the system administrator, if any activity seems to be malicious. Suppose in case of emergency, if a user is performing some operations under someone else role then no actions should be taken.

5. CONCLUSION AND FUTURE WORK

The great demand of web facing application providing faster access to information has made databases more vulnerable. There is a need to prevent such systems from unauthorized access to the database structure, data values and privileges. We would be implementing intrusion detection cum prevention model for database. We have considered some detection and prevention techniques and added response module to present such a model.

REFERENCES

[1] Y. Xiao, X. Shen, D.-Z. Du, " Wireless Network Security", Springer.

[2] Andriy Furmanyuk, Mykola Karpinskyy, Bohdan Borowik, "Modern Approaches to the Database Protection", IEEE, 2007.

[3] Aziah Asmawi, Zailani Mohamed Sidek, Shukor Abd Razak "System Architecture for SQL Injection and Insider Misuse Detection System for DBMS", IEEE, 2008.

[4] Sharma P., (2005), "SQL Injection Techniques & Countermeasures", CERT-In White Paper CIWP-2005-0, http://www.certin.org.in/knowledgebase/whitepapers/ciwp-2005-06.pdf

[5] Protecting Databases, White Paper, APPLICATION SECURITY, INC. http://www.appsecinc.com

[6] S. Axelsson. Research in Intrusion Detection Systems: A Survey: in Technical Report 98-17 (revised in 1999) Chalmers University of Technology, 1999.

[7] Denning, D. E., "An Intrusion-Detection Model," IEEE Transactions on Software Engineering, SE-13, No. 2, pp. 222-232, 1987.

[8] S. Kumar, E. H. Spafford, "An Application of Pattern Matching in Intrusion Detection", Technical Report CSD-TR-94-013, Purdue University, 1994.

[9] A. K. Ghosh, "Learning Program Behavior Profiles for Intrusion Detection", Proceedings of the 1st USENIX Workshop on Intrusion Detection and Network Monitoring, 1999.

[10] Chung, C., Gertz M., and Levitt, K. DEMIDS: A Misuse Detection System for Database Systems. In Third Annual IFIP TC-11 WG 11.5 Working Conference on Integrity and Internal Control in Information Systems, Kluwer Academic Publishers, Pages 159-178, November 1999.

[11] V.C.S. Lee, J.A. Stankovic, S.H. Son, "Intrusion Detection in Real-time Database Systems Via Time Signatures", Real Time Technology and Application Symposium, Pp. 124, (2000).

[12] S. Y. Lee, W. L. Low, and P. Y. Wong, "Learning Fingerprints for a Database Intrusion Detection System," in Proc. of the 7th European Symposium on Research in Computer Security (ESORICS'02), Pp. 264–280, 2002.

[13] Y. Hu and B. Panda, "Identification of Malicious Transactions in Database Systems," in Proc. of the 7th International Database Engineering and Applications Symposium, pp. 329–335, 2003.

[14] P. Liu, "Architectures for Intrusion Tolerant Database Systems," in Proc. of the 18th Annual Computer Security Applications Conference (ACSAC '02, p. 311), 2002.

[15] E. Bertino, A. Kamra, E. Terzi and A. Vakali, "Intrusion Detection in RBAC-administered Databases", 21st Annual Computer Security Applications Conference, 2005.

[16] Ke Chen, Gang Chen, and Jinxiang Dong, "An Immunity-Based Intrusion Detection Solution for Database Systems", Springer-Verlag Berlin Heidelberg 2005, LNCS 3739, pp. 773 – 778, 2005.

[17] Frank S. Rietta, "Application Layer Intrusion Detection for SQL Injection", ACM SE'06 March 10-12, 2006.

[18] Ashish Kamra, Elisa Bertino, Guy Lebanon, "Mechanisms for Database Intrusion Detection and Response", ACM, 2008.

[19] Gongxing Wu and Yimin Huang, "Design of a New Intrusion Detection System based on Database", IEEE, 2009.

[20] Sunu Mathew, Michalis Petropoulos, Hung Ngo and Shambhu Upadhyaya "A Data-Centric Approach to Insider Attack Detection in Database Systems", 2009.

[21] Yawei Zhang and Xiaojun Ye, Feng Xie and Yong Peng, "A Practical Database Intrusion Detection System Framework", IEEE Ninth International Conference on Computer and Information Technology, 2009.

[22] Zorana Bankovic, José M. Moya, Álvaro Araujo, Slobodan Bojanic and Octavio Nieto-Taladriz, "A Genetic Algorithm-based Solution for Intrusion Detection", Dynamic Publishers Inc, Journal of Information Assurance and Security 4, 2009.