Volume 5, Issue 11, November 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Database Security: A Survey

Mohd. Suhel Khan, Neha Mehra
Department of Computer Science,
TIT College, RGPV, Bhopal,
M.P. India

Abstract*— Information is the most valuable asset for organizations. Web database is combination of database and web technology. Web database is placed on the Internet, there are many security problems. The secrecy and the integrity are two important demands of security system. Data security is a major issue in any web-based application. Real world web databases have information that needs to be securely stored and accessed. In this survey paper we look at the various current researches being done to solve these issues, the current trends in securing, ensuring privacy and availability of these data on web database..

***Keywords*— Database security, Web database, Fine grained access control, Intrusion detection**

I. INTRODUCTION

Information is the most valuable asset for organizations. The information disclosure from such databases may have very serious impact on organization business. Damage and misuse of data a effect not only a single user or application, but may have disastrous consequences on the entire organization. The recent rapid proliferation of Web-based applications[1] and information systems has further increased the risk exposure of databases and, thus, data protection is today more crucial than ever. It is also important to appreciate that data needs to be protected not only from external threats, but also from insider threats It is important to properly handle network and web database security issues including authentication, denial of service, and fine-grained access control. So new access control [2] approaches for databases and especially for web databases[3] have become a dire necessity. Web database is a combined production with database technology and web technology. Data security is a major issue in any web-based application. Real world web databases have information that needs to be securely stored and accessed. Web applications are becoming increasingly commonplace and the database can be easily accessible. Many applications are developed with loosely-typed scripting languages and make use of a single database user with full permissions, a so-called administrator user. But with the development of web systems, the number of attacks[4] on databases increased and it has become clear that their access control mechanism is inadequate for web-based systems.

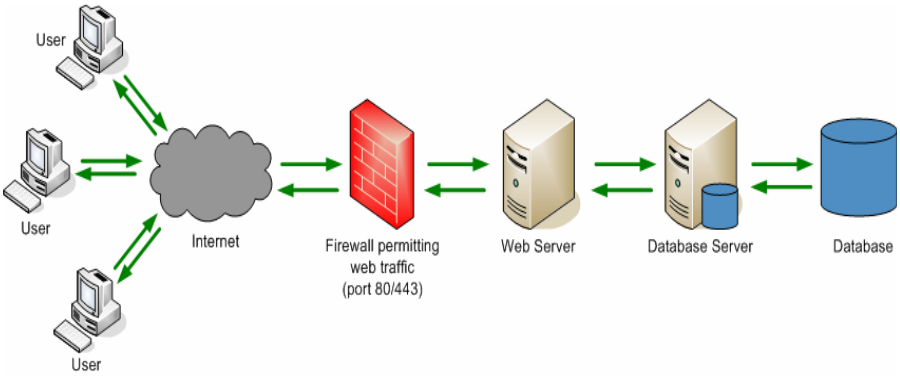
Security breaches are typically categorized as unauthorized data observation, incorrect data modification , and data unavailability . Unauthorized data observation results in the disclosure of information to users not entitled to gain access to such information. All organizations, ranging from commercial organizations to social organizations, in a variety of domains such as healthcare and homeland protection, may suffer heavy losses from both financial and human points of view as a consequence of unauthorized data observation. Incorrect modifications of data, either intentional or unintentional, result in an incorrect database state. Any use of incorrect data may result in heavy losses for the organization. When data is unavailable, information crucial for the proper functioning of the organization is not readily available when needed.

Thus, a complete solution to data security must meet the following three requirements: 1) secrecy or confidentiality refers to the protection of data against unauthorized disclosure, 2) integrity refers to the prevention of unauthorized and improper data modification, and 3) availability refers to the prevention and recovery from hardware and software errors and from malicious data access denials making the database system unavailable. These three requirements arise in practically all application environments.

II. BACKGROUND

A. Web Application Organization

A basic understanding of web application architecture[5] is essential before a discussion on database security within web applications can take place. A high level view of a web application consists of 5 primary parts: the user, the firewall, the web server, the database server, and the actual database. A high level diagram can be observed in figure 1.



The user of a web application is responsible for the manipulation and insertion of data across the internet and into the web application. HTML pages are manipulated by the user and the data is submitted via an HTML request into the web application. Data specific to the user is submitted within this request through the use of HTML forms. After travelling across the internet, the request sent by the client's browser is first encountered by the web application's firewall. Assuming the request is legitimate according to the rules of the firewall, the request is passed on to the web server for processing.

The primary job of the web server is to dynamically generate and send static HTML pages in response to client requests. When a request is permitted into the web application by the firewall, it is parsed by the web server to determine what type of processing must occur. If a non-dynamic HTML page has been requested, the page is sent back to the client and the transaction completes. A page with dynamic components, such as PHP or ASP code, however, requires further processing. These pages are generated by the web server to create a "customized" static HTML page which is in turn sent back to the client. The dynamic portions of these pages are generated based off user-specific data submitted via HTML forms within the request. Dynamic portions of these pages allow for the creation of web pages containing real-time data, and are the backbone of any data-driven web application. The majority of this dynamic content is stored within databases and must be requested from one of the most important components of a web application, the database server.

The job of the database server is to accept requests for data from various components of the web application and retrieve this data from the database. The database itself is managed directly by the database management system, or DBMS, and is not directly accessible. Requests must be sent to the database server, a component of the DBMS, which retrieves and delivers data from the database.

B. Security Concepts

Security is a major concern in the application of data mining techniques to datasets containing personal, sensitive, or confidential information. Some information contained in such datasets is highly confidential. Data is one of the most important corporate assets of companies, governments and research institutions. In order to preserve data privacy, we assume that no one except the data owner or authorized users have the right to access the original data.

Basic security concepts require that web based system be able to identify and control critical aspects such as:

Who the authorized users are (identification and authentication)?

What users should have access to (object access controls)?

What types of operations users can perform on those objects (also part of object access control)?

What types of activities have occurred (e.g., the ability to maintain accountability via auditing)?

Extended security[6] concepts further address issues such as data and system integrity, reliability and availability, further conditional access controls (such as for special business rules), and assurance that all the above are operating properly and consistently.

C. Intrusion Detection System

Lot of existing intrusion Detection Systems (IDSs)[7] examines the network packets individually within both the web server and the database system. However, there is very little work being performed on multitier Anomaly Detection (AD) systems that generate models of network behavior for both web and database network interactions. In such multitier architectures, the back-end database server is often protected behind a firewall while the web servers are remotely accessible over the Internet. Unfortunately, though they are protected from direct remote attacks, the back-end systems are susceptible to attacks that use web requests as a means to exploit the back end. In order to protect multitier web services, an efficient system called as Intrusion detection systems is needed to detect known attacks by matching misused traffic patterns or signatures. IDS is mostly used to perform security monitoring of the network infrastructure.

There are two types of network IDS:

1. Anomaly detection
2. Misuse detection.

An alert is generated when an attack is detected. This alert is used to describe the type of attack and the entities that are involved in it(e.g.-hosts, processes, users). IDS can perform focused analysis of the audit data and they are used to produce incorrect or wrong detections. The actions that are taken in a given environment are dynamically monitored by IDS. An IDS also decides that whether these actions are permissible in the given environment.

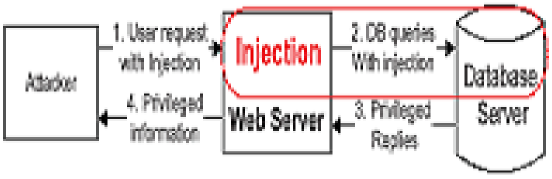
There are following three measures to evaluate efficiency of Intrusion Detection System:

1. Accuracy – Inaccuracy occurs when an IDS signals that an abnormal action is taken in the given environment.
2. Performance – The performance of the system describes the quality of that system. If the performance of IDS is poor then real time detection is not possible.
3. Completeness – When IDS fails to detect an attack then incompleteness occurs. This is very difficult to evaluate because it is impossible to have a global knowledge about all the attacks.

III. LITERATURE SURVEY

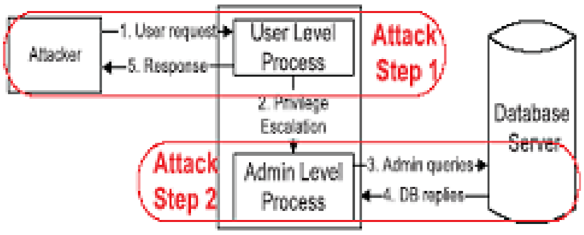
A. Types of attacks on multitier web application

I. Injection attack


Attacker
1. User request with Injection
Injection
2. DB queries With injection
Database Server
4. Privileged information
Web Server
3. Privileged Replies

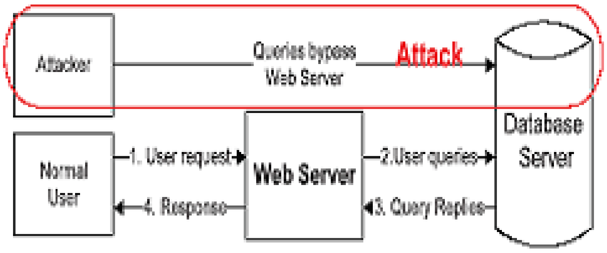
In this type of attack, an attacker can use existing exposure in the web server logic to inject the data or string content which contains the achievements and then use the web server to control these achievements to attack the backend database. The expected structure for the given web server request to the database server would not be able to take by the controlled contents. The SQL injection attack changes the structure of SQL queries and it generates SQL queries in different structure, even if the injected data were to go through web server side. This may be detected as a deviation from the SQL query structure which follows such type of web request. Fig. shows SQL injection attack.

II. Privilege escalation attack



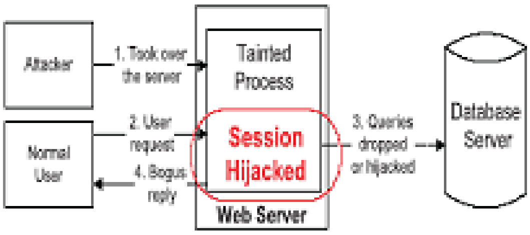
Suppose that the website is used by both regular users and administrators. Regular users will trigger a web request with the set of SQL queries while an administrator will trigger a web request with the set of admin level queries. Suppose that an attacker logs into the web server as a normal user, changes or upgrades his/her details and tries to obtain an administrators data by triggering an admin queries. This type of attack can never been detected by IDS, either it is web server IDS or database IDS, because both the requests and queries are permissible. But according to our mapping model, a database query doesn't match the request and therefore we can detect these types of attacks. Fig shows how regular user may use admin queries to obtain privileged information.

III. Direct DB attack



An attacker can bypass the web server or firewalls and connect directly to the database. An attacker can submit these queries from the web server without sending web request. Web server IDS could not detect anything without matching web request for these queries. The database IDS could not detect these database queries if these are within the set of allowed queries. This type of attack can be detected using our container architecture technology since we cannot match any web request with these queries. Fig.6 shows the scenario of injection attack in which attacker bypasses the web server to directly query the database.

IV. Hijack future session attack



This type of attack is mainly happened at web server side. An attacker takes over the web server and hijacks all the permissible user sessions to launch attacks. An attacker can listen, send spoofed replies and drop user request by hijacking the sessions of other users. We can say that a man-in-the-middle attack, a Denial-of-Service attack or a Replay attack are the categories of hijack session attack. Fig.4 states that a web server can harm all the Hijack future sessions by not generating any database queries for normal user requests. According to the mapping model, for detection of abnormal situations, the web request should generate some database queries (e.g.-Deterministic Mapping). An IDS cannot detect such kind of attacks whether it is web server IDS or database IDS. Our container architecture will provide facility to detect these kinds of attacks. As each user's web requests are separated into individual container, an attacker can never break into other users session[1].

B. Related work

Fine-grained access control was first introduced as a part of the access control system in INGRES by Stonebraker and Wong (1974), which was implemented by query modification technology. The basic idea of query modification is that before being proc-essed, user queries are transparently modified to en-sure that users can access on ly what they are authorized to access (Bertino et al. , 2005; Wang et al. , 2007). Views are used to specify and store access permission for users. When a user submits a query, DBMS first finds all views whose attributes include the attributes of the issued query, and then add the predicates of these views to the predicates of the original query to form a new modified query, which will be carried out.

Recently, work on the policy for preserving privacy has boosted the research of FGAC (Agrawal et al. , 2002; Bertino et al. , 2005). Bertino et al. (2005) presented a privacy preserving access control model for relational databases, which needs a basis of FGAC in relational databases. Nevertheless, they did not describe how to implement the model. LeFevre et al. (2004) proposed a practical approach to incorporating privacy policy enforcement into an existing application and database environment where the implementation of FGAC at cell level was provided.

Chaudhuri et al. (2007) also extended SQL language to support fine-grained authorization by predicated grants. Not only the column- and cell-level authorizations, but also the authorizations for function/procedure execution were supported. Moreover, they designed query defined user groups and authorization groups to simplify the administration of authorizations.

Wang et al. (2007) proposed a correctness criterion of FGAC for databases, which contains three properties: secure, sound, and maximum. They argued that any algorithm used to implement FGAC must be sound and secure, and should strive to be maximum. They also pointed out that no algorithm exists that is both sound and secure. Then, they proposed an algorithm that is sound and secure. In this paper, we do not consider these aspects.

The concept of database response policies was first introduced by [15]. The current paper is a major extension of our previous work. The policy matching algorithms in the current paper take into account arbitrary predicates while the scheme in only considers equality predicates. Also, the JTAM policy administration model presented in this paper is a novel contribution.

An algorithm for event-matching based on the concept of subscription trees is described in context of the GRYPHON project [16]. The algorithm pre-processes the set of subscriptions to build a subscription tree such that each node of the tree is an elementary test on an event attribute. The leaves of the subscription tree are the actual subscriptions. The matching algorithm walks through the subscription tree to find the set of matching subscriptions. Since no analysis of the preprocessing algorithm is provided, it is not clear if the order according to which subscriptions are chosen affects the size of the subscription tree. Also, the scheme is formulated only for elementary predicates, and it has been optimized only for the equality predicates. However, for the policy matching problem, we need to consider arbitrary predicates.

IV. CONCLUSIONS

Many security methods have been devised for protecting the database. Different security models have been developed based on different security aspects of database. These security methods are useful only when the database management system is designed for protecting the database. With the recent growth of web application with database at its backend Secure Database Management System is more essential than only a Secure Database. Hence this paper enlightens on the Vulnerabilities, Threats and Security Methods in Database Management System with the help of survey conducted on the field of secure database. In this survey paper we looked at the various current researches being done to solve these issues, the current trends in securing, ensuring privacy and availability of these data on web database.

REFERENCES

[1] Ashish Kamra, Elisa Bertino, "Design and Implementation of an Intrusion Response System for Relational System for Relational Database", IEEE Transaction and Data Engineering, Vol 23, pg 875-888, 2011

[2] Maria Vanina Martinez, Cristian Molinaro, John Grant, and V. S. Subrahmanian, Customized Policies for Handling Partial Information in Relational Databases, IEEE Transaction and Data Engineering, pg 1-18, 2012

[3] Zhu Yangqing, Yu Hui, Li Hua, Zeng Lianming,Design of a new web database security model, IEEE, 2009, 292-297

[4] Jie SHI, Hong ZHU, A fine-grained access control model for relational databases, Journal of Zhejiang University-SCIENCE C, Springer, 2010, p 575-586

[5] Qing Zhao, Shihong Qin, Study on security of web based database, IEEE, 2008, 902-910

[6] Xueyong Zhu, J. William Atwood, A Web Database Security Model Using the Host Identity Protocol, IEEE 2007

[7] Elisa Bertino, Ravi Sandhu ,Database Security—Concepts, Approaches, and Challenges, IEEE 2005, p 2-19

[8] Leon Pan, A Unified Network Security and Fine-Grained Database Access Control Model, Second International Symposium on Electronic Commerce and Security, IEEE 2009, p 265-269

[9] Zhou Wen, A new web accessing database module basing in security of information computer security, 2008, 63-66

[10] Lianzhong Liu, Qiang Huang, A framework for database auditing, IEEE, 2009, 982-988

[11] Alex Roichman, Ehud Gudes, Fine-grained Access Control to Web Databases, SACMAT, IEEE, 2007, P 31-40

[12] Marty Humphrey, Sang-Min Park, Jun Feng, Norm Beekwilder, Glenn Wasson, Jason Hogg, Brian LaMacchia, and Blair Dillaway ,Fine-Grained Access Control for GridFTP using SecPAL, International Conference on Grid Computing, IEEE/ACM, 2007, p1-9

[13] A. Kamra and E. Bertino, and R.V. Nehme, Responding to Anomalous Database Requests, Secure Data Management, pp 50-66, Springer, 2008

[14] A. Kamra and E. Bertino, Design and Implementation of SAACS: A State-Aware Access Control System, Proc. Ann. Computer Security Applications Conf (ACSAC) , 2009

[15] A. Kamra, E. Bertino, and R.V. Nehme, “Responding to Anomalous Database Requests,” Secure Data Management, pp. 50-66, Springer, 2008.

[16] M.K. Aguilera, R.E. Strom, D.C. Sturman, M. Astley, and T.D. Chandra, “Matching Events in a Content-Based Subscription System,” Proc. Symp. Principles of Distributed Computing (PODC), pp. 53-61, 1999..