G.Swapna, R.Pavani Srivatsav / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 6, November- December 2012, pp.432-435

Securing Web Applications By Analyzing The Logs Of The Database Server Or Web Server

G.SWAPNA
Asso.Professor (IT),
SVITS, Mahabubnagar
Mahabubnagar, Andhra Pradesh,

R.PAVANI SRIVATSAV
Asst.Professor (IT)
SVITS, Mahabubnagar
Mahabubnagar, Andhra Pradesh,

Abstract

SQL injection attacks [1] are one of the major security threats for web applications. In fact, the open web application security project (OWASP), an international organization of web developers, has placed SQLIAs among the top ten attacks that a web application can have. SQL injection is a technique for maliciously exploiting applications that use client-supplied data in SQL statements. SQL injection is now one of the most common attacks in the Internet. Simply go to Yahoo! or Google and search for "SQL injection" and we can find tones of related documents. Although the awareness of SQL injection is rising, still many people do not have very concrete ideas on how to prevent SQL injection attack. This paper is not going to tell you what is SQL injection, nor going to tell you the latest techniques in SQL injection attacks, but more important, how to prevent SQL injection correctly and in a more integrated approach. Using SQLIAs, an attacker may be able to read, modify, or even delete database information. Solutions to avoid these attacks are, placing a powerful Network SQL Injection Intrusion Detection Systems (IDS), but insider or internal employer of the organizations can easily bypass the security. So, Network Intrusion Detection Systems cannot fully protect the databases from the attacks, The Propose System can detect the attacks that are from Internet and Insider Attacks, by analyzing the logs of the database server or web server log details, and using limited number of signatures it is going to avoid the internal and external attacks.

Keywords: access log parsing; detection system; sql injection attacks;

I. Introduction

An SQL injection attack targets a Web application that uses database services. These types of applications accept user input, such as form fields, and then include this input in database requests, typically SQL statements [6]. In SQL injection, the attacker can submit the user input that result in a different Database request than it is executed by the application programmer. That is, the intended user input is embedded as part of a larger SQL statement, results in an SQL statement of a different form than originally intended.

Today's tendency in computer security shows an increased amount of work being done in database Security research. The reason behind such an increase is because in traditional security mechanisms such as the use of firewall are no longer effective in today's database security research. This is because, in web-based application scenario, business partners and customers must have access to data including the organization's Employee. Because of that reason the data cannot simply be hidden behind a firewall. Web applications that are connected to the Internet and accessing the database as a backend, make the DBMS more vulnerable to attacks.

Almost all of the organizations use databases and web application to maintain their information. Security of these systems became crucial. Internet threats like SQL Injection attacks on database through web applications are more. The Propose System can detect the attacks that are from Internet and Insider Attacks, by analyzing the logs of the database server or web server log details and it is also avoid the cross site scripting attacks.

The remainder of this paper is organized as follows. Section 2 describes the type of Intrusion Detection System suitable for databases as a solution for the database security problems. Section 3 presents related work on sql injection attacks Section 4 presents the proposed system and algorithm for the proposed IDS system. Finally section 5 concludes the paper.

II IDS

Intrusion Detection System (IDS) [2] is defined as the process of wisely monitoring the actions occurring in a computer system or network, analyzing them for signs of violations of the security policy. The main aim of Intrusion Detection Systems (IDS) is to guard the availability, confidentiality and integrity of critical networked information systems. Intrusion Detection Systems (IDS) are defined by both the method used to detect attacks and the placement of the IDS on the network.

Intrusion Detection System or IDS is software, hardware or combination of both used to detect intruder activity. Snort is an open source IDS available to the general public. An IDS may have different capabilities depending upon how complex and complicated the components are. IDS appliances that are a combination of hardware and software are available from many companies.

A. Types of intrusion detection system

Network IDS or NIDS

NIDS are intrusion detection systems that capture data packets traveling on the network media (cables, wireless) and match them to a database of signatures. Depending upon whether a packet is matched with an intruder signature, an alert is generated or the packet is logged to a file or database. One major use of Snort is as a NIDS.

Host IDS or HIDS

Host-based intrusion detection systems or HIDS are installed as agents on a host. These intrusion detection systems can look into system and application log files to detect any intruder activity. Some of these systems are reactive, meaning that they inform we only when something has happened. Some HIDS are proactive;

Signatures

Signature is the pattern that we look for inside a data packet. A signature is used to detect one or multiple types of attacks. For example, the presence of "scripts/iisadmin" in a packet going to were web server may indicate an intruder activity. Signatures may be present in different parts of a data packet depending upon the nature of the attack. For example, we can find signatures in the IP header, transport layer header (TCP or UDP header) and/or application layer header or payload.

Alerts

Alerts are any sort of user notification of an intruder activity. When an IDS detects an intruder, it has to inform security administrator about this using alerts.

Logs

The log messages are usually saved in file. Usually all web server's records all processed requests into special file. This file is named as access log file. The presence of such information allows analyzing the access log files and allows to trace the sources of visitors of our site

Intuitively, an *SQL Injection Attack (SQLIA)* occurs when an attacker changes the intended effect of an SQL query by inserting new SQL keywords or operators into the query, According to Sharma (2005), there are Two major SQL Injection techniques: i) *access through login page* and ii) *access through URL*. In first technique users are bypass the login forms in which users are authenticated by using passwords. This kind of technique can be performed by the attackers using: *'or' condition*, *'having' clause*, *multiple queries* and *extended stored procedure*. The second technique can be performed by the attackers through: *manipulating the query string in URL* and *using the 'SELECT' and UNION statements.*

This kind of attack represents a serious

Normal usage:

SELECT * FROM scenarios WHERE username='greg' AND password='secret';

Malicious Usage

SELECT * FROM scenarios WHERE username='greg';--'AND password = 'anything';

If the username and password as provided by the User are used, the query to be submitted to the Database takes the form;

If the user were to enter ['OR 1=1 --] and [ ]

Instead of [greg] and [secret], the query would take the form

SELECT rollno FROM students WHERE username='' OR 1=1 -- 'AND pswd= ' '

Now the query checks for the conditional equation of [1=1] or an blank password, then a legal row has been found in the students table. The first ['] quote is used to terminate the string and the characters [--] mark the beginning of a SQL comment, and anything further than is ignored. The query as interpreted by the database now has a tautology and is always fulfilled. Thus an attacker can bypass all authentication modules gaining unrestricted access to critical information on the server.

III Related work on SQL injection attacks

Many researchers on web server security or database security revolve about access policies, roles, administration procedures, physical security, and security models [7]. But, in recent years much effort has been invested in developing methods for detecting intrusions to database. Here the main idea is analyzing the logs of the database server based on transactions that arrive to the database to detect the intrusions.

A misuse detection system to find hidden anomalies in the database log, this approach towards a database specific intrusion detection mechanism [3] is by Hu and Panda (2003). They proposed a mechanism that is capable of finding data dependency relationships among transactions and use this information to find hidden anomalies in the database log.

Another relevant approach towards a real-time intrusion detection mechanism based on the profile of user roles has been present by Bertino[4] *et al.* (2005). This approach is based on mining SQLqueries stored in database audit log files. The result of the mining process is used to form profiles

that are capable to model normal database access behavior and identify intruders.

Another similar work is a Misuse Detection System for Database System (DEMIDS) [5] has been proposed by Chung *et al.* (1999) is a misuse-detection system, customized for relational database systems. It uses audit data log to derive profiles describing distinctive behavior of users in DBMS. Chung introduces the notion of distance measure and frequent item sets to capture the working scopes of users using a data mining algorithm.

SQL Injection attack allows the attackers to fully compromise web database via web application. An SQL injection attack (SQLIA) is a type of attack on web applications that exploits the fact that input provided by web clients is directly included in the dynamically generated SQL statements. SQLIA is one of the foremost threats to web applications. Cross site scripting attack allows the attackers to modify the web content. All these are because of design flaws in the web applications. Almost all of the organizations use databases and web application to maintain their information. Security of these systems became crucial. Internet threats like SQL Injection attacks on database through web applications are more, various techniques has been proposed to avoid this attacks , such as the use of stored procedures, prohibiting display of database server error messages and use of escape sequences for sanitizing user inputs are employed as a quick fix solution. Unfortunately, even these security measures are also inadequate against highly sophisticated SQL injection attacks. As a solution, better protections such as Intrusion Detection System (IDS) have been proposed.

IV Proposed IDS and algorithm for proposed IDS system.

The propose system can detect the attacks that are from internet and insider Attacks, by analyzing the logs of the database server or web server log details.

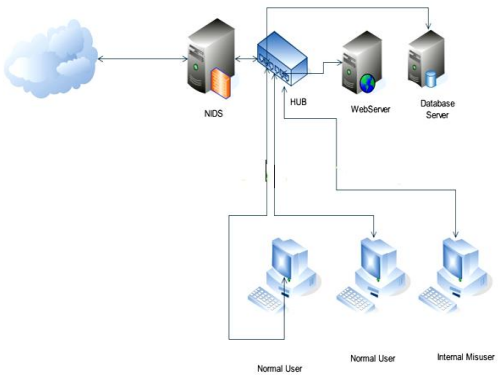


Figure 1. Network Intrusion Detection Setup Diagram

Network Intrusion Detection Systems can monitor the packets that flow through the network, but insider or internal employer of the organizations can easily bypass the security. So, Network Intrusion Detection Systems cannot fully protect the databases from the attacks. The Propose System can detect the attacks that are from Internet and Insider Attacks, by analyzing the logs of the database server or web server log details.

The proposed system architecture gives brief idea about how the detection system avoids these attacks.

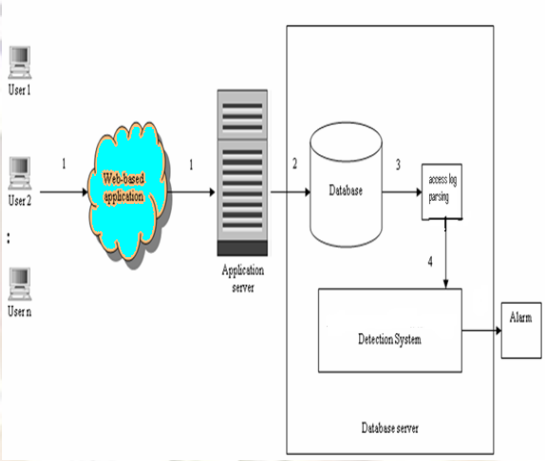


Figure 2 . Proposed System Architecture.

It is implemented in three modules

1. WEB SERVER ACCESS LOG FILES
2. ACCESS LOG PARSING
3. DETECTION SYSTEM

A. web server access log files

Usually all web server's requests are recorded in a special file. This file is named as access log file. The presence of such information allows analyzing the access log files and allows to trace the sources of visitors of our site.

B. Access log parsing

Access log parsing is performed on that access log, access log parsing in this we are extracting and storing the information, so we can answer interesting questions about what our site's visitors have been doing.

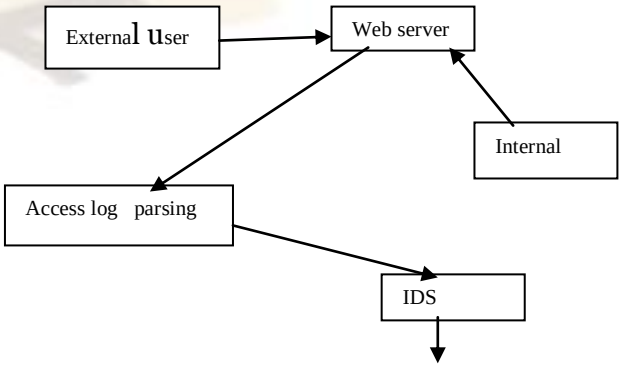


Figure 3. Access Log Parsing.

We are going for log analysis based detection, whatever the request from internal or external users will touch the web applications, the web server stores those details in the log file. Parsing modules identifies all the fields from the log file, and finds the GET request based on the request the IDS will try to identify the SQL injection attacks.

Example

192.290.192.30 - - [03/Dec/2009:19:00:11 +0530]"GET/index.php?login=;select%20*%20from %20users HTTP/1.1"400312 "-" "-

C. Detection system

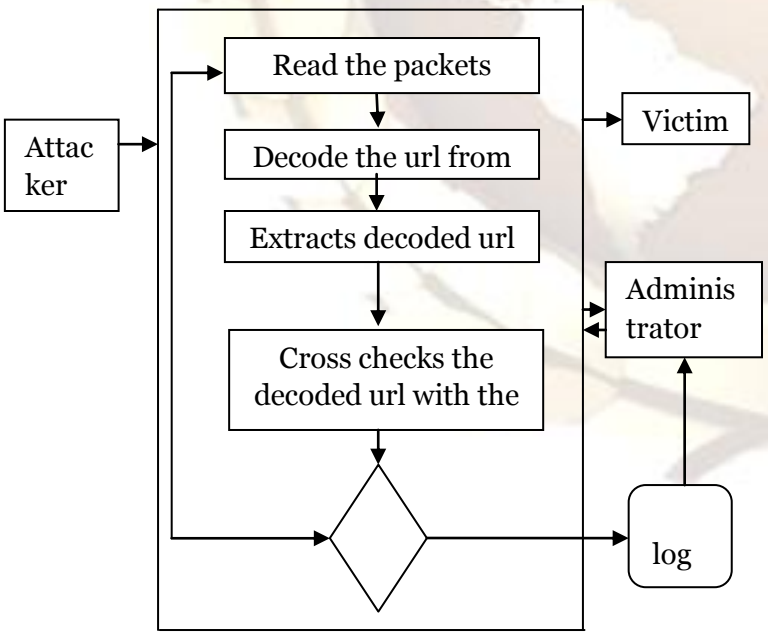
After sending access log information to the IDS it cross matches with signatures, if the signature is match with the pattern it displays the attack otherwise it allows the normal user to access the database. Detection logic cross matching signatures with request lines.

Signature: set of rules pertaining to a typical intrusion activity simple example rule: any ICMP packet > 10,000 bytes

D. Detection Algorithm

1. Read the packets from the log file
2. Parsing the packet in the access log file
3. Parsing the packet in the access log
4. Url_decoded=url_decode(url) (Step check)
5. Check the url_decoded with the next available regular expression.
    1. Generate log
    2. Else repeat the step check with the next available regular expression.

Detection Process



V. Conclusion

This paper presented a sketch on the sql injection attacks and the intrusion from the both internal and external against the database system. Most of the researchers focused on detecting external attacks only. The proposed system detects the attacks that are from both internal and external attacks. In that by using access log parsing we going for log analysis based detection, the work of the detection logic system is, it cross matches with signatures request lines.

References

[1] Advanced SQL Injection In SQL Server Applications, Chris Anley,An NGSSoftware Insight Security Research(NISR) Publication © 2002 Next Generation Security software Ltd. http://www.ngssoftware.com.

[2]Intrusion Detection; Hot Based & Network-Based IDS. Harley kozoshko.

[3] Hu, Y., Panda, B., (2003), "Identification of malicious transactions in database systems". In Proceedings of the. International Database Engineering and Applications Symposium (IDEAS).

[4] Bertino E., Kamra A., Terzi E. and Vakali A., (2005), "Intrusion Detection in RBAC Administered Databases", Proceedings of Annual Computer Security Applications Conference (ACSAC).

[5] Chung C., Gertz M., Levitt K., (1999), "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.

[6] spett. k. (2005) sql injection; Are your web applications vulnerable/ available t URL http:// www.spidynamics.com/papers/sql injectionwhitepaper.pdf.

[7]. Castano, S., Fugini,M.G.,Martella, G. and Samarati, P.,(1995), "Database security". Wokingham, England: Addison-Wesley Publishing Company.