AN ANALYSIS FRAMEWORK FOR SECURITY IN WEB APPLICATIONS

GORE DINKAR ARJUNRAO

Research Scholar, Dept. of Computer Science
CMJ University, Shillong, Meghalaya

ABSTRACT

SQL Injection attacks are one of the gravest threats for web applications. SQL injection attacks pose a serious security threat to Web applications: they allow attackers to obtain unrestricted access to the databases underlying the applications and to the potentially sensitive information these databases contain. Although researchers and practitioners have proposed various methods to address the SQL injection problem, current approaches either fail to address the full scope of the problem or have limitations that prevent their use and adoption. Many researchers and practitioners are familiar with only a subset of the wide range of techniques available to attackers who are trying to take advantage of SQL injection vulnerabilities. As a consequence, many solutions proposed in the literature address only some of the issues related to SQL injection. To address this problem, we present 'Evading Input Filters' to avoid SQL injection attacks. We also present and analyze existing detection and prevention techniques against SQL injection attacks. For each technique, we discuss its strengths and weaknesses in addressing the entire range of SQL injection attacks.

INTRODUCTION

WEB APPLICATIONS

In recent years, web applications have grown dramatically popular, with organizations converting legacy mainframe and database systems into dynamic web applications using technologies such as PHP, Ajax, JavaScript, JSP, Java, ASP, ASP.NET etc., These applications expose customer information, financial data and other sensitive and confidential data over the Internet and Intranet. With the accessibility of such critical data, web application security testing also becomes paramount. Ensuring that web applications are secure is a critical need for companies today.

As businesses have becomes dependent upon Web applications, they are getting difficult to secure. Web Security has become a prime concern these days due to the financial damages caused by malicious hacking attacks.

Many organizations have deployed sophisticated security mechanisms, such as firewalls or Intrusion Detection Systems (IDS), to help protect their information assets and to quickly identify potential attacks. While these mechanisms are important, they are not foolproof.

A firewall cannot protect against what is allowed through – such as online applications and allowed services. While an IDS can detect potential intrusions, it can detect only what it has been programmed to identify, and it will not be effective at all if the company does not monitor or respond to the alerts. Moreover, firewalls and intrusion detection systems must be continuously updated or they risk losing their effectiveness at preventing or detecting attacks.

The purpose of information protection is to protect an organization's valuable resources, such as information, hardware, and software. Through the selection and application of appropriate safeguards, security helps the organization meet its business objectives or mission by protecting its physical and financial resources, reputation, legal position, employees, and other tangible and intangible assets. We examine the elements of web applications security, roles and responsibilities, and common threats. We also examine the administrative controls, policies and procedures, and risk analysis.

WEB ATTACKS

Applications such as Content Management Systems (CMS), Wikis, Portals, Bulletin Boards, and discussion forums are being used by small and large organizations. All web frameworks (PHP, .NET, J2EE, Ruby on Rails, ColdFusion, Perl, etc) and all types of web applications are at risk from web application security defects, ranging from insufficient validation through to application logic errors.

Web applications commonly face a unique set of vulnerabilities due to their access by browsers, their integration with databases, and the high exposure of related web servers.

The modern web server setup commonly presents multiple applications running on one host and available via a single port, creating a large surface area for attack.[21]

Every week hundreds of vulnerabilities are being reported in these web applications, and are being actively exploited. The number of attempted attacks every day for some of the large web hosting forms range from hundreds of thousands to even millions. The most common web application attacks are:

- SQL Injection
- Code Injection
- Remote Code-Inclusion
- Cross-Site Scripting(CSS)
- Cross-Site Request Forgeries (CSRF)
- Directory Traversal

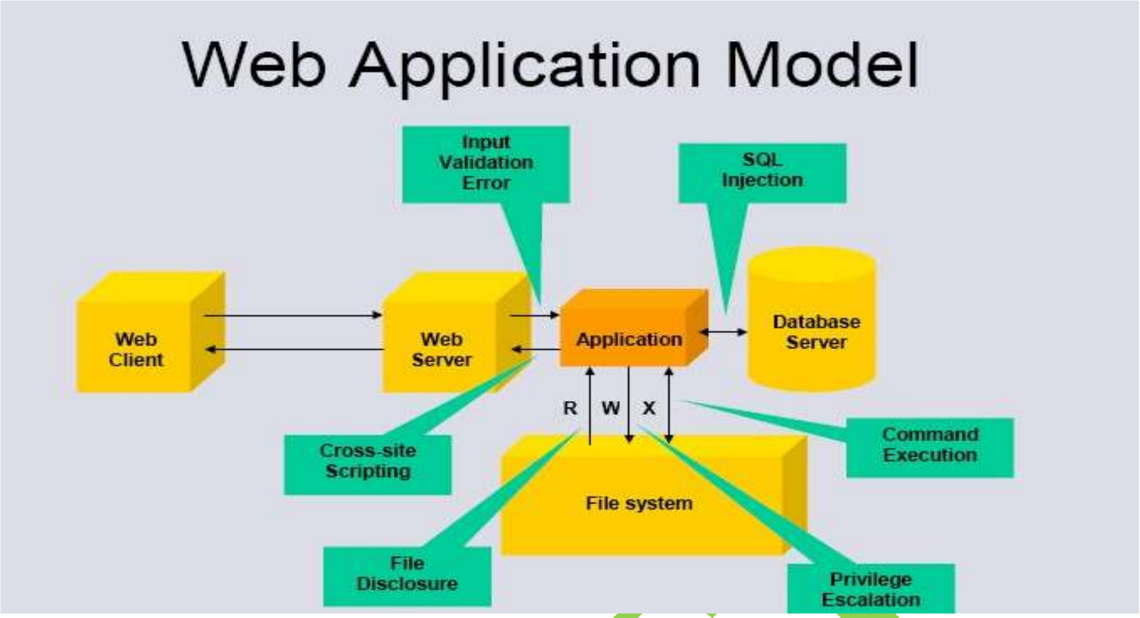

Figure 1.1 Web Application Model

REVIEW OF LITERATURE

Gaurav Kumar Tak, Gaurav Ojha (2012) have executed the methodology on an online examination system and recorded and analyzed all the SQL strings over a significant period of time. These attacks were carried out by some students of universities who provided technical support during analysis part of the methodology. The experimental results provide the complete scenario of the problem and accuracy of above steps. Our system indicated that the attacks were detected with 98.51 % accuracy. Popular Romil Rawat, Shailendra Kumar Shrivastav (2012) will use SVM (Support Vector Machine) for classification and prediction of SQL-Injection attack. In our propose algorithm, SQL-Injection attack detection accuracy is (96.47%) and which is the highest among the existing SQL-Injection detection Techniques. The term SVM is typically used to describe classification with support vector methods and support vector regression is used to describe regression with support vector methods. SVM is a useful technique for data classification. Our concept provides a secure application, based classification of original and suspicious query strings using SVM. Here dataset of different size is used for training and classification .different parameters like Accuracy, detection time ,training time ,TPR,TNR,FPR,FNR and the graphical description shows the performance of our system. Our system shows the best performance result in accuracy which is 96.47% and best among the existing systems.

S. Ramachandran, A. Ramachandran (2012) the vulnerabilities inherent in the Cloud systems should be addressed so they can be eliminated before exploited by malicious software or hackers. Our approach plays a major role in detecting and managing vulnerabilities present in the Cloud infrastructure. Implementation of this methodology proves to be cost effective and saves analyzing time. However this technology is still in its initial stages of development, as it suffers from threats and vulnerabilities that prevent the users from trusting it. Various malicious activities from illegal users have threatened this technology such as data misuse, inflexible access control and limited monitoring. By this approach it can optimize time and cost, such that the vulnerabilities can be eliminated before exploited by malicious software or hackers, in the network component. The complexity of modern enterprises, their reliance on technology, and the heightened interconnectivity among organizations are rapidly evolving developments that create widespread opportunities for theft, fraud, and other forms of exploitation by offenders both outside and inside an organization. Internal and external perpetrators can exploit traditional vulnerabilities in seconds. It is envisioned that using the vulnerability scanning as process on a regular basis, it will response to problems identified will alleviate these risks.

Imperva's Web Application Attack Report (2012) have identified and investigated malicious traffic containing the following technical attacks: Remote File Inclusion (RFI), SQL Injection (SQLi), Local File Inclusion (LFI), Cross Site Scripting (XSS) and Directory Traversal (DT). Cross Site Scripting and Directory Traversal are the most prevalent classical attack types. It also investigated two types of Business Logic attacks: Email Extraction and Comment Spamming. Comment Spamming injects malicious links into comment fields to defraud consumers and alter search engine results. Email Extraction simply catalogs email addresses for building spam lists. These Business Logic attacks accounted for 14% of the analyzed malicious traffic. Email Extraction traffic was more prevalent than Comment Spamming. A full anatomy of BLAs is described in this report. Web applications face attacks that are becoming more diverse, more technically sophisticated and more difficult to detect and block. Obviously, security counter-measures must keep up with the threats to prevent damages and losses to the business and its customers.

Indrani Balasundaram, E. Ramaraj (2011) an authentication scheme for preventing SQL Injection attacks using Advance Encryption Standard (PSQLIA-AES). Encrypted user name and password are used to improve the authentication process with minimum overhead. It have implemented and tested the proposed scheme with three different ways, 1) evaluated with different key sizes (128, 256, and 512), 2) compared the processing overhead (time needed for encryption of user name and password) of proposed scheme with existing related schemes SQLIPA (Shaukat Ali, Azhar Rauf, and Huma Javed (2009)) and AQE-PSQLIA, and 3) compared the processing overhead of the proposed scheme by using different number of users. The proposed scheme is more efficient, it needs 3.144ms for encryption or decryption and this can be negligible.

K. Ahmad, J. Shekhar and K. P. Yadav (2011) proposed an approach coalesce techniques, that were based on a filter, cryptographic hash-function, linear probing technique, customized error message and POST method. The filter is used to detect malformed SQL queries whereas the hash function is used to match hash values of usernames and passwords against stored hash values. A linear probing technique is used to address the data collision and debug error message problems because it is able to stop the reconnaissance progress of threat agents.

Shikhar Jain & Alwyn R. Pais (2011) use the static-dynamic model analysis to prevent SQL injection attack. Static model define query structure at compile time while dynamic model define query structure at the run time. This is completely automated approach which is performed after complete development of the application. Approach is to extract query structure from the programs using static analysis and at the runtime capture the dynamic query and validate it against the static query model. SQL injection attack will add tokens to the user input and hence change the query structure. If the dynamic query contains malicious code than it will not match the static query model then it will be rejected. The SQLIA prevention has four steps. The method presented to prevent SQL injection attack is completely automated. It is developed to prevent SQL injection attacks on applications developed in .Net language. Presented approach is based on the assumption that the program contains structure of the queries. It uses validation of static and dynamic query model for the prevention. Prototype model of .Net string analyzer is developed for static analysis phase which generates the regular grammar for the strings present in the program. Tool can be used to prevent SQL injection attacks on existing applications as well as new applications. It doesn't depend on developer to prevent SQL injection attack, hence saves development time. Tool is tested on the sample web applications, results show that it is able to prevent and report all the SQL injection attacks performed on the web applications.

Selvamani Kadirvelu, Kannan Arputharaj (2011) introduces an intelligent dynamic query intent evaluation technique to learn and predict the intent of the SQL queries provided by users and to compare the identified query structure with the query structure which has been generated with user input in order to detect possible attacks by unethical users automatically. This type of evaluation is helpful in reducing the need for the user to have more consciousness when SQL queries are written. This work has been fully automated by implementing intelligent validation techniques in order to minimize user intervention. The main advantage of this system is that it applies the decision tree classification algorithm which is enhanced with temporal rules to find the unethical users intelligently at the query execution points where the database manager of the system can be informed of the new possible query execution points with an intent for attacks, and thereby preventing the SQL injection attacks.

Atefeh Tajpour, Suhaimi Ibrahim, Maslin Masrom (2011) SQL injection is a type of attack which the attacker adds Structured Query Language code to a web form input box to gain access or make changes to data. SQL injection vulnerability allows an attacker to flow commands directly to a web application's underlying database and destroy functionality or confidentiality. Researchers have proposed different tools to detect and prevent this vulnerability. It presents SQL injection attack types and also current techniques which can detect or prevent these attacks. Finally it evaluate these techniques.

RESEARCH METHODOLOGY

To reach the final goal, a new application will be developed to demonstrate the abilities of the proposed solution. To achieve the final goal, following steps were taken as the part of research procedure:

- **Questionnaires:**

The questionnaire are prepared for IT Professionals, Programmers, Devlopers. The questionnaires are contain both open ended as well as multiple choice questions. Ten percent of the IT orgnizations would be randomly selected, as to sample atleast 234 people for above experiment. Respondent were divided into three categories good, average and poor according to the awareness level about increasing threat “SQL Injection Attacks”.

- **Data Analysis:**

Secondary data is collected from Business Journals, Business Newspapers, Books, Magazines and Publications. Primary data is collected by personally interviewing the General Public, IT professionals, Programmers, Devlopers in relation to impact of SQL Injection Attacks with the help of questionnaires. After collection of data from primary and secondary sources, researcher has coded the data for different categories in SPSS and analyse the statistical measures such as average, percentages, coefficients of correlation and regression to provide meaningful analysis. Based on the data collected appropriate statistical tools was applied for the analysis of data.

- **Model:**

After collection and analysis of data, the available models are explored. Then the new model is proposed. After the testing and implementation of the model, for its interoperability, reusability, flexibility and openness, robustness and reliability and collaborative learning features are done. The helping tools used for proposed model are PHP and MySql.

CONCLUSION

SQL injection is a common technique for attackers using SQL queries to attack on Web-based applications. These attacks reshape SQL queries and thus alter the behavior of the program for the benefit of the hacker. To minimize the potential damage of a successful SQL injection attack, one should minimize the privileges assigned to every database account in their environment. Do not assign DBA or admin type access rights to one’s application accounts. It is always recommended to prevent attacks before the processing of the user’s (attacker's) request. Input validation can be used to detect unauthorized input before it is passed to the SQL query. We also recommend to adopt the given below additional defenses, in order to provide security in depth.

- Least Privilege
- Finding Second-Order Vulnerabilities

REFERENCES

Gaurav Kumar Tak, Gaurav Ojha,” Advanced Query-based Multi-tier Approach towards

Detection and Prevention of Web Attacks", International Conference on Recent Advances and Future Trends in Information Technology (iRAFIT2012), Proceedings published in International Journal of Computer Applications® (IJCA), 2012.

Romil Rawat, Shailendra Kumar Shrivastav," SQL injection attack Detection using SVM", International Journal of Computer Applications (0975 – 8887) Volume 42– No.13, March 2012.

S. Ramachandran, A. Ramachandran,"Rapid and Proactive Approach on Exploration of Vulnerabilities in Cloud based Operating Systems", International Journal of Computer Applications (0975 – 8887) Volume 42– No.3, March 2012.

Imperva's Web Application Attack Report, Hacker Intelligence Initiative Overview. Edition #2, 2012.

Shikhar Jain & Alwyn R. Pais," Model Based Approach to Prevent SQL Injection Attacks on .NET Applications", International Journal of Computer Science & Informatics, Volume-I, Issue-II, 2011

Indrani Balasundaram, E. Ramaraj," An Authentication Mechanism to prevent SQL Injection Attacks", International Journal of Computer Applications (0975 –8887) Volume 19– No.1, April 2011.

Atefeh Tajpour, Suhaimi Ibrahim, Maslin Masrom," SQL Injection Detection and Prevention Techniques", International Journal of Advancements in Computing Technology Volume 3, Number 7, August 2011.

Selvamani Kadirvelu, Kannan Arputharaj," ISQL-IDPS: Intelligent SQL- Injection Detection and Prevention System", European Journal of Scientific Research ISSN 1450-216X Vol.51 No.2, pp.222-231, 2011.

K. Ahmad, J. Shekhar and K. P. Yadav," Coalesce Techniques to Secure Web Applications and Databases against SQL Injection Attacks", electronic Journal of Computer Science and Information Technology (eJCSIT), Vol. 3, No. 1, pp. 26-30, 2011.

Jon Oltsik," The Network Application Security Architecture Requirement ", ESG White Paper March 2011.

P. Naresh Kumar, Yugandhar. G and K. Nageswara Rao," THE IMAGE LEVEL TAINTING: A NEW APPROACH FOR REVENTING SQL INJECTION ATTACKS", International Journal of Engineering Science and Technology (IJEST), Vol. 3 No. 7, pp. 5622-5628 July 2011.