Evaluating the Security Flaws in Web Applications

Prabhdeep Kaur
Department of CSE
GNDU, Amritsar
Punjab, India

Harkamal Kaur
Department of CSE
GNDU, Amritsar
Punjab, India

ABSTRACT
Web security is an important area of research. This work has focused on web securing schemes. The primary concentration is to interpret the way to handle the SQL Injections. It is one of the many web attack methods used by hackers to steal data from industries. It is one of the most usual technique used in present era for application layer attack It is the category of attack that takes the benefit of. Improper coding of your web applications that allows hacker to inject SQL commands into say a login form to allow them to gain access to the data held within your database. So in this work we have reviewed different research on the SQL injections.

Keywords
SQL injections, Hackers, Web security, Crackers

1. INTRODUCTION
Nowadays web applications have become increasingly close to people's life. Nevertheless, the attacks on the web programs vulnerabilities pose a serious risk to user's privacy security and property safety. [1]Common vulnerabilities include SQL Injection, Cross-site Scripting, Malicious File Execution and Cross-site Request Forgery.

1.1 Security
Security [1] is a crucial aspect of Web programs .These types of applications by definition permits users access to a central resource — the Web server — and by it, to some other sources such as database servers. Through interpreting and applying proper security measures, one can guard their own resources as well as provide a protected environment in which users feel comfortable working with the web application. Security work demands the knowledge of the primary concepts of security. Authentication confirms that users who are using are empowered. It is the procedure of allowing or refusing access to resources for specific users.

1.1.1. Web Application Security in IIS and ASP.NET
Security for Web application starts with the Web server (IIS). As a Windows-based service, IIS is fully incorporated with the Windows protection..Similar to other processes, to access a file, IIS requires proper verification. When the Web applications work, they do so under ASP.NET, which posses its own security features. These come into play when program requires resources accessibility. Several techniques are provided by IIS and ASP.NET[1] for establishing authentication like anonymous access, basic and digest authentication, Windows Integrated Security, Certificate Authentication

1.1.2. SQL Injection
An SQL injection [2] is a kind of injection vulnerability in which the attacker tries to inject irrelevant malicious data into the input fields of a program, which, when worked upon by the application program, causes that data to be run as a piece of code by the back end SQL server, thereby giving unwanted outcomes which the designer of the program did not predict. The backend server can be any SQL server (My SQL, MSSQL, ORACLE, and POSTGRESS, to name a few).

2. LITRATURE REVIEW
The web application security [3] has become the primary discussion for security managing persons, because attacks on application are rising constantly posing new risks for organizations. Several trends have emerged about the attacks launched against web programs. The International security standard implementation is to minimize the security failures and to mitigate their consequences. A individual unprotected question declaration can outcome in compromising the protection of the program, details or data resource server. Designers must be well-organized enough to apply the protection techniques to every web accessible procedure and operate. .Each dynamic query must be protected. It is apparent that protection of protection is becoming more challenging because the possible strike technological innovations are becoming progressively innovative.

Procedures of SQL injection vulnerability Detection [4] was proposed based on hidden web creeping and apply a discovering system with the purpose of raising the website protection and enhancing the SQL injection weaknesses detecting ability of checking device of web. Authentication is combined with the crawler model, and finds SQL Injection weakness by simulating web attacking and analysing the data response. Main objective of analyzing the web attack detection mechanisms is to enhance the ability of web checker and raise the web page protection of crawler model. On the basis of techniques, enforcement of SQL injection vulnerability detection system was done and the experiments shows well expected outcomes.

Among much vulnerability in the Web Application programs, SQL injection as well as Cross-site scripting [5] have been the most dominant class of web vulnerabilities. For these weaknesses, the static analysis tools ASPWC is used to detect XSS attacks and SQL injection vulnerabilities based on spoil analysis, it tracks various kinds of external feedback, control circulation graph is constructed in accordance with data flow analysis of the relevant details, taint data propagate to various kinds of functions of vulnerability, and detects the attacks of XSS or SQL Injection in web application's source code. Security vulnerability within web application is usually reported. Time price, individual price and fallibility deplete its extensive use. Thus automatic program code examination sources for the detection of weaknesses in programs create a key means. The strategy depending on taint tracing to identify malicious input details and ensure no outcome taint details could be used to derive weaknesses for most spoil details.

There are many web vulnerabilities [6] in the web applications and thus an automated vulnerability scanner was provided that check for the injection attacks applied a program that

computerized scanned the injection strike weak points. Proposed techniques for checking Web program protection were detecting weaknesses depending on injection factor, exactly obtain the details of injection factor, and using black box testing to analysis what potential weaknesses, tackled vulnerable injection factor. According to identification depend on injection factor, it is clearly known where the bug is, decrease the debug time and improve efficiency.

A survey [7] on different techniques to prevent SQL Injection and XSS attacks and there is remedy to identify and prevent against the malicious attacks over the developer's Web Application written in programming languages like PHP, ASP.NET and JSP also created an API in local terminology through which transactions and interactions are sent to IDS Server through Inter Server Communication Mechanism. The suggested program was designed identify and prevent Vulnerabilities for the web programs designed in .NET and JSP by which the intrusions are recognized and prevented via IDS Server with the API designed in local terminology. The analysis is performed by exploration the data resource log which is a real-time log monitored by IDS from various Web 2.0 programs, the offline log which was acquired from the .NET and JSP programs tested in LAN is also incorporated with this log dynamically. The analysis review gives the pattern of timestamp which can be used to further enhance the security features provided by the IDS Server.

The web intrusion prevention techniques [8] are well-known for protecting web programs against usual attacks, like XSS and SQL injection, but a consistent methodology to evaluate and benchmark such techniques is not available. For this, present the idea of a benchmarking test bed, which automatically performs the evaluation in a consistent and reproducible way. By enabling the benchmarks to draw from a corpus of installable segments which can be depend on actual security vulnerabilities, members of the protection community can consistently maintain and enhance the benchmark, the benchmark, allow it to be altered as menaces and defences evolve.

A new test method called SMART [9] was proposed, which automatically tests SQL injections in web programs. SMART analyzes the SQL queries produced by web programs and uses a structure matching validation procedure to figure out to whether SQL Injection vulnerabilities exist or not in the web application. Comprehensive tests display that SMART is efficient in finding SQL Injection attacks. trying it out the web application programme with SMART,SMART tests each input parameter of web program, matches the SQL queries produced by both original HTTP request and injected HTTP request, and decides whether it has SQL injection vulnerability or not .

The SQL injection is the well-known attack [10] against the web programs, has become a serious protection risk..The traditional techniques are insufficient to analyse the SQL injection weak points in the web programs. In comparison to the well-known attacks as SQL injection or XSS, transmission analysing sources for the web services specific attacks do not exist. For this, first this was the motivation for developing the first automated computerised sources for web services against the particular attacks are created which is known as WS-Attacker. It gives an overview of design decisions and offer assessment of four Web Services frameworks and their resistance against WS-Addressing spoofing. It is showed that design decisions, which enabled to construct a general framework extensible with Web Service particular attack plug-ins.

SQL injection can be used for unauthorized accessibility to a database to penetrate the program illegally, alter the data source or even remove it. For a hacker to alter a data source details such as field and table names are needed. So, recommend a solution to the above issue by avoiding it using an encryption security algorithm based on randomization [11]. Main aim is to offer enhance the security by developing a tool which prevents unlawful accessibility to the data source. SQL injection is one most essential web protection risk hat needs attention so as to enhance security for the users and their details. This deals with an application specific randomized security encryption algorithm to identify it and prevent it.

IPAAS is a novel strategy for preventing the exploitation of XSS and SQL injection vulnerabilities based on computerized automated data type detection of input parameters. IPAAS (**I**nput **Pa**rameter **A**nalysis **S**ystem)[12] automatically and transparently augments otherwise vulnerable web data source integration environments with input validations that outcome in important and tangible security improvements for real systems. Web programs are well-known weak points such as XSS (cross-site scripting) and SQL (Structured Query Language) injection are, unfortunately, still prevalent. Current mitigation techniques for XSS and SQL injection vulnerabilities mainly concentrate on some aspect of computerised automated outcome sanitization. It identifies automated input validation as an effective alternative to output sanitization for avoiding XSS and SQL injection vulnerabilities in legacy programs, or where developers choose to use vulnerable legacy languages and framework.

The web solutions inherited various security issues of Web programs and also imparted some new issues. Bad input approval is the major cause of data theft at the level of application. For implementation of a model to validate an input for services of web an appropriate method should be selected which is helpful in preventing SQL injections and cross-site scripting by using predetermined design for valid input stipulation. The proposed model called WSIVM (Web Services Input Validation Model) [13] give emphasis on data entry validation and allows acceptance of valid entries only, enabling only legitimate entries to be sanctioned as it is in conformity with the with the white-list strategy ,in which only predetermined values are approved and all others are regarded invalid.

The cyclomatic complexity [14] and lines of code etc features of the static code have been proven as useful in finding flaws in the modules of software. First of all information and sink types should be categorized that may be responsible for security attacks. Secondly, to prevent security risks, sanitization techniques that are used commonly on inputs, should be categorized .Then static code features should be proposed for each tender sink according to which each categorization scheme can be characterize. Then data mining applications helps in finding web application vulnerabilities.

2.1 Gaps in literature

After conducting the literature survey we have found that the most of existing researchers has neglected the role of cracker in the web based technologies. Crackers, also known as "malicious hackers" and "black-hat hackers", are dissimilar because while they have similar attainment as a cyberpunk , they apply their great power to devote acts of crime like broadcasting of viruses, stealing of personal data and overpowering other's personal computers with help of their own skills to do other criminal acts. Evidently, the actions of majority of the crackers are considered illegal. So it is required to save the web queries

from crackers. So problem of cracker is found to be critical issue in Web security.

3. CONCLUSION AND FUTURE WORK

This work has focused on finding the gaps in existing research on the SQL injections. It is found that many researchers are currently working on improving the security techniques. An SQL injection attack targets interactive web applications that employ services of database. Applications of these types take input from user, such as fields of a form, and then admit this input in database calls or requests, usually SQL statements. In case of SQL injection vulnerability, the attacker gives user input of that kind which results in a different database request than was intended by the developer of the application. It means the reading and understanding of the user input as part of a larger SQL statement, results in a dissimilar SQL statement from the results which were intended originally. It is found that the existing researchers have not focused on the crackers who can also crack the queries. So in near future we will develop a new technique which can handle both hackers and crackers to provide the secure web to users.

4. REFERENCES

[1] "Introduction to Web Application Security" [online available]: http://msdn.microsoft.com

[2] "What is an SQL Injection? SQL Injections: An Introduction"[online available]: http://resources.infosecinstitute.com

[3] Madan, Sushila. "Security Standards Perspective to Fortify Web Database Applications From Code Injection Attacks." In Intelligent Systems, Modelling and Simulation (ISMS), 2010 International Conference on, pp. 226-230. IEEE, 2010.

[4] Wang, Xin, Luhua Wang, Gengyu Wei, Dongmei Zhang, and Yixian Yang. "Hidden web crawling for SQL injection detection." In Broadband Network and Multimedia Technology (IC-BNMT), 2010 3rd IEEE International Conference on, pp. 14-18. IEEE, 2010.

[5] Zhang, Xin-hua, and Zhi-jian Wang. "A static analysis tool for detecting web application injection vulnerabilities for ASP program." In e-Business and Information System Security (EBISS), 2010 2nd International Conference on, pp. 1-5. IEEE, 2010.

[6] Chen, Jan-Min, and Chia-Lun Wu. "An automated vulnerability scanner for injection attack based on injection point." In Computer Symposium (ICS), 2010 International, pp. 113-118. IEEE, 2010.

[7] Priyadarshini, R., D. Jagadiswaree, A. Fareedha, and M. Janarthanan. "A cross platform intrusion detection system using inter server communication technique." In Recent Trends in Information Technology (ICRTIT), 2011 International Conference on, pp. 1259-1264. IEEE, 2011.

[8] Stuckman, Jeff, and James Purtilo. "A testbed for the evaluation of web intrusion prevention systems." In Security Measurements and Metrics (Metrisec), 2011 Third International Workshop on, pp. 66-75. IEEE, 2011.

[9] Wu, Haiyan, and Guozhu Gao. "Test SQL injection vulnerabilities in web applications based on structure matching." In Computer Science and Network Technology (ICCSNT), 2011 International Conference on, vol. 2, pp. 935-938. IEEE, 2011.

[10] Mainka, Christian, Juraj Somorovsky, and Jorg Schwenk. "Penetration testing tool for web services security." In Services (SERVICES), 2012 IEEE Eighth World Congress on, pp. 163-170. IEEE, 2012.

[11] Avireddy, Srinivas, Varalakshmi Perumal, Narayan Gowraj, Ram Srivatsa Kannan, Prashanth Thinakaran, Sundaravadanam Ganapthi, Jashwant Raj Gunasekaran, and Sruthi Prabhu. "Random4: An Application Specific Randomized Encryption Algorithm to prevent SQL injection." In Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, pp. 1327-1333. IEEE, 2012.

[12] Scholte, Theodoor, William Robertson, Davide Balzarotti, and Engin Kirda. "Preventing Input Validation Vulnerabilities in Web Applications through Automated Type Analysis." In Computer Software and Applications Conference (COMPSAC), 2012 IEEE 36th Annual, pp. 233-243. IEEE, 2012.

[13] Brinhosa, Rafael Bosse, Carla Merkle Westphall, and Carlos Becker Westphall. "Proposal and development of the web services input validation model." In Network Operations and Management Symposium (NOMS), 2012 IEEE, pp. 643-646. IEEE, 2012.

[14] Shar, Lwin Khin, and Hee Beng Kuan Tan. "Mining input sanitization patterns for predicting SQL injection and cross site scripting vulnerabilities." In Proceedings of the 2012 International Conference on Software Engineering, pp. 1293-1296. IEEE Press, 2012.

.