ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 6, December 2019

An Overview of Web Application Security

Peng Gao, Zhihua Li,Yafei Chen

Abstract— The application and development of various dynamic Web technologies have aggravated the problem that Web applications are difficult to prevent. Based on the typical attack penetration mode of Web application, the research progress of common injection vulnerability detection in Web application is summarized; Introduces the common infiltration methods of Web attack, puts forward the common weak points and introduces the relevant protective strategies; and finally gives the research prospect.

Index: **Web security; Web infiltration; Security protection.**

I. INTRODUCTION

With the rapid development of the Internet, all walks of life have paid more and more attention to security issues, especially in the fields of e-commerce and online banking, which are extremely demanding on security. However, due to the openness and interactivity of the Internet itself, the network is destined to be a high-risk area full of Trojan horse viruses, phishing scams, and attack damage. In modern times, all Internet companies pay more attention to network security, and the cost of investment is also more and more high. The firewall products and software protection products launched by various vendors are also in many categories, but they are still inevitably subjected to various attacks, such as CSDN password leakage incidents, foreign Yahoo account leakage incidents, etc. These incidents have caused the attacked company to lose a lot of information, causing its reputation to suffer losses and letting users lose confidence. Almost every website today has a history of being attacked; the difference is whether these attacks have successfully penetrated into the server.

II. WEB SECURITY STATUS

A. The Importance of Web Security

With the improvement of people's security awareness, the security costs of various companies have also increased year after year, but hackers have repeatedly succeeded, for the following reasons:

Manuscript received: 22 November 2019
Manuscript received in revised form: 15 December 2019
Manuscript accepted: 05 January 2020
Manuscript Available online: 10 January 2020

(1) The defender (Web service provider) must defend against all vulnerabilities, and the hacker only needs to find the weakest link, and once it succeeds, it is likely to cause the defender to lose all. Some vulnerability defenders are hard to find, such as system software vulnerabilities. Sometimes, the measures taken to ensure security are a breakthrough for hackers, such as a firewall device with vulnerabilities.

(2) The defender is in the light, the hacker is in the dark. The defender's IP address, server information and so on are largely public, while the hacker's information defender is almost impossible to know, and before the attack, the hacker has 100% initiative. This feature determines the success of DDoS (flood attack).These hackers often make it difficult to find the source by forging IP, proxy, and decentralized zombie remote control, and the cost of finding it is high. When the harm is not serious, few victims will go after it. The weak technical force of law enforcement agencies also makes these hackers even more unscrupulous. The cost of hackers is mostly due to time cost and bandwidth cost.

(3) The cost of defenders is high and the cost of hackers is low. To successfully defend against a Website, defenders must spend a lot of effort, use limited tools to find vulnerabilities and fill them, and hackers can attack at any time, try any and even illegal tools, and attack almost every Website. The same is true, greatly reducing their skill costs.

(4) Hackers tend to be more professional. The existing small Websites are equipped with very few special security personnel, most of which are programmers and engineers. What many programmers and engineers do is to add special security measures after the completion of the program. Hackers tend to specialize in infiltration attacks.

(5) Some social engineering issues are also biased towards hackers. For example, adding a firewall affects the business function of the Website, the leader requests to down or simplify the downgrade, and for example, a Website system user uses a short, easy to guess password, or transmits the password as a plain text on the network, or a certain call a We Chat message had lied to modify the password and so on. Therefore, security personnel should

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 6, December 2019

pay attention not only to coding and technical problems, but also to these non-technical problems.

B. Research Status at Home and Abroad

SQL injection attack is one of the common methods used by hackers to attack databases. It is also the main way of information leakage. In order to improve the accuracy of SQL injection penetration testing, the current SQL injection penetration testing field mainly reduces the false negatives or false positives by improving the input point discovery ability of Web applications or improving the vulnerability analysis reaction. For example, in the research on enhancing the ability of crawling methods to collect information in penetration testing, the literature [1] proposed a SQL injection vulnerability testing method based on model analysis to improve crawling, and increased the integrity of penetration testing information collection. Literature [2] proposes an improved crawler technology that automatically generates valid input and populates Web application forms to discover more Web application input points and improve penetration test coverage. Alenezi [3] proposed a new method of infiltration test information collection based on source code static analysis to solve the problem that when testing complex Web applications, some types of input points of Web applications cannot be found by ordinary crawling methods.

For the research of the analysis phase of the penetration test, Kim [4] compares the normal and post-attack SQL query tree structure in the Web application through data mining of the database log, and proposes an improved Web application response analysis method to detect the SQL injection vulnerability. Jang [5] proposed a response analysis method based on effective recognition of the size of SQL query results. This method determines SQL injection vulnerability by identifying different responses to the size of SQL command query results sent to the backend database for execution, so as to overcome the problem of missing information caused by unclear return information size after Web applications are attacked. In addition, the literature [6] models the behavior of SQL commands based on token graphs, and uses SVM machine learning to compare the normal behavior of Web applications with their post-attack responses to determine SQL injection security vulnerabilities.

The above research work mainly focuses on the two aspects of the thoroughness of information collection and the correctness of vulnerability analysis, and the lack of research on the effectiveness of SQL injection attack generation. For example, related research such as [3] takes attack generation as an external factor, focusing only on how to make its referenced attack input reach the attackable input point effectively, without analyzing the diversity or sufficiency of the cited attack input itself.

In the attack generation research of SQL injection penetration test, the literature [7] proposed a random enumeration method to generate attack input. This kind of random enumeration penetration test method is difficult to fully test the Web application, determine whether its defense measures are sufficient, and it is difficult to trigger SQL injection vulnerabilities hidden behind inadequate defense measures, which may cause false negatives or errors in SQL injection vulnerabilities and reduce the accuracy of penetration testing. Literature [8] proposed to instantiate SQL injection test data based on the combination avoidance coverage criteria, but did not study the effectiveness of attack generation including the diversity or sufficiency of penetration test cases. Literature [9] proposed a second-order SQL injection vulnerability testing tool based on static analysis, but could not accurately locate the intermediate storage location of the pollution data in the storage phase and could not judge whether the pollution data in the trigger phase was effectively filtered before it reached the dangerous function. Therefore, there is The problem of high false positive rate and false negative rate. Literature [10] proposed a second-order SQL injection vulnerability detection method based on dynamic and static combination analysis, but this method only considers the case where the stain information is spliced in the storage stage and the complete SQL code is written in the same statement. High false negative rate. In recent years, some models have proposed a model-based penetration test method [11] to model the relevant elements of the penetration test, to model the specification or to guide the penetration test process. For example, [12] proposes a model-driven penetration testing framework that combines the Web application development process with penetration testing activities to achieve secure knowledge sharing. In addition, some experts have also studied the

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 6, December 2019

attack model based SQL injection attack modeling method [13,14]. However, the model methods proposed in these studies still do not study how to model the SQL injection penetration test attack method and enhance the effectiveness of attack generation to determine the SQL injection vulnerability and its defense adequacy. Attack methods are explicitly stated to improve the accuracy of SQL injection penetration testing.

III. COMMON ATTACK PENETRATION METHODS FOR THE WEB

Denial of service attacks. DDoS attack is a very common and harmful attack. It is commonly known as flood attack. The name is DDoS: Distributed Denial of Service. It is an upgraded version of the normal denial of service (DoS) attack. The basic principle is that the server is embarrassed by sending a large number of requests or data packets to the server. Ordinary denial of service attacks are usually initiated by only one computer, but DDoS is a large number of computers (commonly known as "broilers") that are held or controlled by hackers and simultaneously attack the server. There are many protective devices for DDoS attacks on the market. But this is only an order of magnitude. Once the attack volume exceeds one value, these protective devices are too much to eat. DDoS attacks are very effective, and there is no one-size-fits-all solution. Even some world-class Websites will fall under the DDoS attack. Ordinary DoS attacks are easier to guard against, and the power of DDoS attacks can increase as the number of "broilers" increases, but the cost is also increasing. It requires a long time, high technical means or a lot of attacks. The money to prepare the "broiler chicken".

Virus Trojan attack. Some hackers with programming experience will develop some Trojan horses, infecting Websites related to the Website (such as administrators or edited computers, servers, etc.) by mail, downloading, and bundling software. Some worms are very harmful, such as the "Panda Burning" virus that ravaged the Internet in 2007. For example, in 2017, "WannaCry" swept the world and wanted to cry and blackmail the virus. There are also some worms with no files, only in the computer memory, very hidden. And some of the lesser-known virus Trojans, such as the Trojan horse, may cause the online game player equipment to be stolen, so that you can cry without tears, or let people's QQ be stolen, QQ-related mailboxes, important information in the micro-cloud Theft, the consequences are very serious.

System/software vulnerability attacks. Attack server systems or software with system or software vulnerabilities that have been announced but not patched, or vulnerabilities that have not yet been discovered. In this case, it was originally to prevent the hacker's hardware and software, which turned into a hacker's entry point.

Social engineering attacks. Use various deceptive and unethical means to obtain the account and password of the Website back-end system. Such attacks are often not technically preventable, and they require system users to work together and comply with security regulations.

Program vulnerability attack. Take advantage of the code writer's inconsistency and launch an attack on a site. Among them, SQL injection attacks are the most popular and the most harmful. SQL injection attack is one of the common methods used by hackers to attack databases. It is also the main way of information leakage. SQL injection attacks are one of the most important tasks in Web application security research because of the hazards, types, mutations, and hidden attacks.Through sql injection, an attack on a dynamic page, and even lead to a riot. Obtain user privacy data, etc. through injection scripting attacks and cross-site scripting attacks. Hackers bypass system authentication or snoop and destroy databases by entering malicious SQL statements in Web forms. For example, the login interface requires a user name and password. The program will enter the user name and password, which is usually queried in the database with the following SQL statement:

String sql = select * from user_table where username = ' " + user Name + " ' and password = ' " + password + " ' ;

If the hacker enters 'or 1 =1-- in the text box after the username, the above SQL statement becomes: SELECT * FROM user_table WHERE username = " or 1 =1 -- and password = "; = 1 Yongzheng, and and password = "by being commented out, then this statement will always execute correctly, and the hacker can easily fool the system and obtain legal identity.

IV. COMMON WEAK POINTS

In the process of hacking attacks, hackers will start their attacks from the weakest point they find, because it is the easiest to succeed. In all kinds of production environments

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 6, December 2019

(including code, operation and maintenance, and various aspects of use), there are always some common weak points that are easily overlooked or scorned:

Weak passwords and weak authentication. The non-technical back-office users (editors) of the Website are often ridiculous, using passwords such as birthdays, names, phone numbers, short passwords, etc., leaving a security risk to the background.

Open and protected against the background. Some Websites openly expose the portal of the management backend, which is equivalent to giving the hacker a good entrance. Some Websites have no restrictions on password input in the background, and some directly indicate "password error" (should prompt "account or password error", this dangerous practice even exists on some large e-commerce Websites).

Unpatched systems and applications. Both the server and the back-end user's computer should be patched and upgraded in time to fill the gap. Otherwise it is easy to be infected by a virus Trojan.

The code program does not fully validate the user input to output or store it. The danger of this behavior is unquestionable and will lead to various security issues.

The server opens up unused ports and features. The idle function is often not safe, which will lead to more opportunities for hackers to invade.

Unencrypted data transfer. Some sensitive and important information is transmitted on the Internet in a clear-text manner. Once intercepted by a hacker, it is likely to cause the system to be compromised. Clearly save important information. For example, until 2011, the famous technology Website CSDN actually used plaintext to save user passwords without any encryption!

Buffer overflow. The contents of the buffer copy exceed its own capacity and cause an overflow.

Security and functional conflicts. Security and functionality are often conflicting. Adding various security measures will inevitably lead to the consumption of certain system resources, and the functionality will inevitably be affected more or less. As I said before, some bosses who don't pay attention to security issues can also lead to security risks.

V. TARGETED SECURITY PROTECTION

For the common vulnerabilities described above, we can carry out some security measures in a targeted manner:

Develop and adhere to strict and rigorous security measures. All system background users must comply with these rules so that many unnecessary social engineering problems can be avoided. At the same time, it is necessary to prevent the attacks or leaks of secrets by some paid employees or commercial spies.

Hide all relevant information in the background as much as possible, and increase the defense of the background entrance. Some practical practices can be quite effective, such as limiting the number of password errors; simply prompting "account or password error", don't prompt "account not exists" or "password error", which adds little user experience, but Undoubtedly very good for hackers; you can make the management background into a separate second-level domain name, and do not parse on the public network, but configure HOSTS files on each user's computer, manually resolve these second-level domain names, and the server only You need to bind these domain names.

Give different accounts an authorization difference. Some technically insensitive editors can assign editing accounts and stipulate smaller permissions, while ordinary technicians can assign ordinary administrator accounts, and supervisors and specialized security personnel can assign super administrator accounts.

Separation of privileges. For example, all IIS requests are handled by an external process running under a low-privilege account, while important processes in the system itself are processed by a more privileged local account that does not participate in any HTTP response. Another example is Apache, which starts the main server process httpd as the root process, and then starts many httpd child processes running with lower-privileged nobody accounts.

High-intensity encryption of some sensitive information. Information such as passwords, amounts, etc. can be transmitted using SSL (Secure Sockets Layer) and its successor TLS (Secure Transport Layer Protocol) technology.

Multi-layer encryption of information such as user passwords into the library. If the conditions permit (mainly

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 6, December 2019

external supervision conditions), you can use the multi-layer + mixed encryption method to encrypt the password before entering the library. This way, even if the hacker gets the database and sees the encrypted password, it is difficult to crack.

It must be assumed that the external world is not secure, only the necessary ports are opened, only the necessary services are opened, and only the privileges necessary for the authorized account are given. Windows and Linux systems use the "netstat" command to view the current port. Some unused ports can be turned off. IIS, Tomcat, Apache, Nginx and other server software like SMTP are not required to be turned off, and then turned on when needed. The permissions for reading, modifying, etc. of files are only assigned to the required authorization level. If necessary, you can turn off file upload permissions in the background of the Website.

Install system and software patches in a timely manner. The server is best to download auxiliary software like 360 to update the system and server software in a timely manner.

Full user input verification. Coders must always keep in mind that once a user enters a part, it must be fully verified and processed. Also add error handling mechanism, try to avoid exposing the program code.

Do not mix code and data. From a pessimistic perspective, HTML itself is an example of a mix of code and data. HTML code files will mix content and various JS code. The concealment of hyperlinks often makes people inadvertently attack, and because of the mechanism of reshaping character shape such as Escape code, HTML itself will become more confusing and lead to scripts. Attacks can be multiplied. However, the programming language running on the server side can separate the code from the data, and the object-oriented encapsulation function can better hide the data.

Enable logging. Windows and Linux systems have system logs, such as IIS, Tomcat, Apache, Nginx and other mainstream server software, as well as MSSQL, MySQL, Oracle and other databases have log systems, you should enable these logs, and see what happens at any time. It should be noted that you need to set the automatic save time of the log, or override the condition to prevent the log file from filling up the disk space, which will affect the server performance and normal log records.

Adequate errors, crash response mechanisms and emergency drills. Although we are confident that we have done a lot of protection, the computer system is an extremely complicated system. In addition, there is almost no solution like DDoS. Computer security problems are often only a matter of time. Therefore, when the program goes wrong, we should have an emergency plan to respond quickly and process.

VI. CONCLUSION

The security problem of Web applications has become one of the main problems of Web applications. How to determine whether the Web application meets the security needs, how to strengthen the security management of Web application scientifically, and how to test and evaluate the security of Web application scientifically are important issues faced by all Web application systems.

Aiming at common injection vulnerability, it is an important research direction for Web application security vulnerability detection to establish a comprehensive and rapid discovery mechanism of injection points in Web application system; In addition, through the effective combination of various technologies and models, automatic vulnerability analysis can be explored to form a targeted and highly automated Web application injection vulnerability detection system; Finally, in the case of limited time and resource conditions, integrating multiple technologies to alleviate the contradiction between the accuracy of vulnerability analysis and resource consumption is still one of the technical difficulties to be solved in the analysis and detection of embedded vulnerabilities in Web application systems.

REFERENCES

[1] M. I. P. Salas, E. Martins, “A Black-Box Approach to Detect Vulnerabilities in Web Services Using Penetration Testing,” IEEE Latin America Transactions, vol. 13(3), pp. 707-712, 2015.

[2] J. M. Chen, C. L. Wu, “An automated vulnerability scanner for injection attack based on injection point,” In: Computer Symposium. IEEE, pp. 113-118, 2010.

[3] M. Alenezi, Y. Javed: Open source Web application security: A static analysis approach,” In: International Conference on Engineering & Mis. IEEE. pp. 1-5, 2016.

ISSN: 2277-3754
ISO 9001:2008 Certified
International Journal of Engineering and Innovative Technology (IJEIT)
Volume 9, Issue 6, December 2019

[4] M. Y. Kim, H. L. Dong, "Data-mining based SQL injection attack detection using internal query trees," Expert Systems with Applications. vol. 41(11), pp.5416-5430, 2014.

[5] Y. S. Jang, J. Y. Choi, "Detecting SQL injection attacks using query result size," Computers & Security. vol. 44(2), pp. 104-118, 2014.

[6] D. Kar, S. Panigrahi, S Sundararajan, "Detecting SQL injection attacks using graph of tokens and SVM," Computers & Security. Pp.206-225, 2016.

[7] A. Kieyzun, P. J. Guo, K. Jayaraman, " Automatic creation of SQL Injection and cross-site scripting attacks," In: IEEE, International Conference on Software Engineering. IEEE. pp. 199-209, 2009.

[8] H. C. Huang, Z. K. Zhang, H. W. Cheng, "Web Application Security : Threats, Countermeasures, and Pitfalls," Computer. Vol. 50(6), pp. 81—5, 2017.

[9] J. Dahse, T. Holz, "Static Detection of Second-Order Vulnerabilities in Web Applications," 2014.

[10] L. Yan, X. Li, R. Feng, "Detection Method of the Second-Order SQL Injection in Web Applications Structured Object-Oriented Formal Language and Method," Springer International Publishing, 2013.

[11] A. Marback, H. Do, K. He, "A threat model- based approach to security testing," Software Practice & Experience. vol. 43(2), pp. 241-258, 2013.

[12] P. Xiong, L. Peyton, "A model-driven penetration test framework for Web applications," In: Eighth International Conference on Privacy Security and Trust. IEEE. pp. 173-180, 2012.

[13] N. Kaur, P. Kaur, "Modeling a SQL injection attack," In: International Conference on Computing for Sustainable Global Development," IEEE, 2016.

[14] D. BYERS, N. SHAHMEHRI, "Unified modeling of attacks, vulnerabilities," In: ICSE Workshop on Software Engineering for Secure Systems (SESS). pp. 36-42, 2010.

AUTHOR BIOGRAPHY

Peng Gao is a master student in Hebei University Of Engineering, major in computer technology，research area is internet security.

Zhi-Hua Li is a doctor, associate professor, graduated from Hebei University of Technology, majoring in electrical engineering. Have long engaged in theory and technology of Internet of things, wireless sensor network.

Ya-Fei Chen is a master student in Hebei University of Engineering, major in computer technology, research area is internet security.