Research on Attack-defense Technology Based on Web Server Side

He Gao[1, a], Yijie Shi[1, b], Yan Gao[2, a] and Qiuyu Zhang[2, b]

[1]State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, China;

[2]State Grid Xingtai Electric Power Supply Company, Xingtai 054001, China.

[a]gao_he@bupt.edu.cn, [b] yijieshi2000@bupt.edu.cn, [c]gaoyaniqy@163.com, [d] uud001@163.com

Keywords: Web server side, Attack-defense technology, Sql injection, File upload vulnerability.

Abstract. With the development of web 2.0 technology, all kinds of services based on web have appeared many new features. The invasion methods and attack technologies aimed at web service side are becoming more and more diverse, which threaten the security of network seriously. Therefore it is necessary to research these attack technologies deeply. The most harmful vulnerabilities for the web server side are sql injection, file load vulnerability, software of server parsing vulnerability and so on. Through the research of these technologies of attacking, we can simulate attack scenarios and implement security protection against these vulnerabilities.

Introduction

With the rapid development of Internet and the growing maturity of web technology, a variety of web application platforms have been well spread and widely applied, and web service has dramatically changed the way, with which people work and communicate. But the current web security is not optimistic, viruses and attacking methods against the web application are emerging, which have entered into an advanced stage with its faster attacking speed, higher degree of automation and more concealed attacking. Because of the frequent massive data leakage and network attacking events, web security issues are also increasingly prominent. At present there are two major categories of attacking mode for web: one is the script against the client-side, another is attacking against the server application services.

According to the open web application security project (OWASP) top ten web security risks, the security attacks against web server applications occupy more than sixty percent and sql injection attacks on the server side rank in the top ten web security risks [1]. In recent years, common server software such as IIS or apache has been pointed out the parsing vulnerability [2], which means a malicious attacker can easily get the web administrator privilege and even obtain the highest authority of the system according to these parsing vulnerabilities, and the consequences will be very serious. Contents of the website will be rewritten while data leakage will occur and the web application service will be paralyzed. In the year of 2013 and 2014, there are two major network security vulnerabilities against the web server: one is against the web framework of Struts2 and the other is against OPENSSL which is named "heart bleed bug". The security of network has been pushed to the forefront of public media because of the emergence of these two vulnerabilities, and there are many websites affected by the two vulnerabilities. With the increasing popularity of network security, people pay more and more attentions to the security of web server side, but there are still many details that we should follow closely. In section 2, we review common attack methods on web server side, and in section 3 we give some solutions to the protection for web server side.

Common Attack Methods on Web Server Side

Web server side security is more important than the client security. Security vulnerabilities in the server side will make a disastrous event, so people usually pay more attentions on the web server side security. There are many attacking methods on the web server side, while the most common and harmful methods include: sql injection, file upload vulnerability, attack based on cookie or session, and so on.

Sql Injection and Its Variants.

Sql injection attack is one of the most common and harmful attack methods against the web applications. Attackers can construct a special statement as a parameter and put it to the web application. Then attackers can execute malicious operations by constructing the corresponding sql statements [3]. The main reason for this vulnerability is that the script written by the programmer does not filter special characters entered by the user, and then the malicious sql command can be executed.

In the era of low security awareness, an attacker using a conventional sql injection vulnerability can cause a great harm to the target website. However, with the development of the times, more and more websites will use WAF (Web Application Firewall) to protect their security, and WAF can put an end to the vast majority of sql injection vulnerabilities. But some skilled hackers can still gain control of the website by constructing a variation of sql injection statement. Preparations for attacking the target website by sql injection from attackers are shown in Fig. 1.

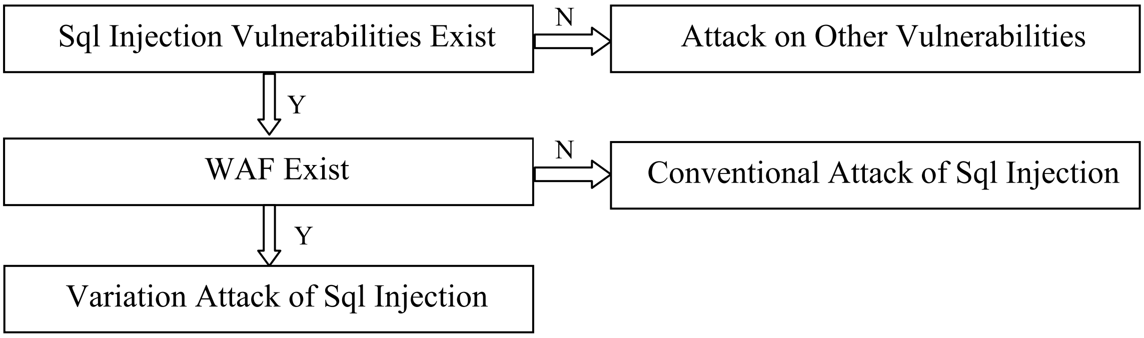
Sql Injection Vulnerabilities Exist
N
Attack on Other Vulnerabilities
Y
WAF Exist
N
Conventional Attack of Sql Injection
Y
Variation Attack of Sql Injection

Fig. 1 The preparatory work of SQL injection

Conventional Attack of Sql Injection.

We can get the type and version of database first and create different injection statements according to the specific function of different databases [4]. The test environment is based on a combination of php5 and mysql5 under the operating system of Windows 2003 Sp2. The corresponding statements are constructed as shown in Table 1.

Table 1 The Corresponding statements

Number	Attack Statements	Attack Results
1	and ord(mid(version(),1,1))>51	Get the type and version of database
2	order by n	Determine the number of fields
3	union select 1, database(),user()	Get the database name and user
4	union select 1,2,table_name from information_schema.tables where table_schema=dbname limit n-1,n	Get the corresponding table name
5	union select 1,2,column_name from information_schema.columns where table_name=tname limit n-1,n	Get the corresponding filed name
6	union select 1, root_name,root_pass from dbname.tname	Get admin account and password

Variation Attack of Sql Injection.

First of all, the famous penetration testing system kali has some WAF detection tools. We can use the tools to determine whether there is a WAF behind the target website and obtain the WAF type, and then we can construct the corresponding variation sql injection statement to attack the target site [5].

The methods bypassing WAF mainly include: mixed case, a variety of combination of coding, the use of inline comments, the use of equivalent uncommon function, the use of special symbols and so on. There are many different bypass methods that we should combine according to the different WAF. For example, we can construct the injection statement like this: / * ! U % 6eIon *//*!S e%6cEct* /1, @ @ user,3; /* */`.

File Upload Vulnerability.

The function of file upload is a normal business need [6]. If the processing logic of the file in the server side is not enough safe after uploading the file, it will lead to serious consequences. The definition of file upload vulnerability is that users upload an executable script file, and obtain the ability to execute the command in the server from that script file [7]. If the server side has that vulnerability, attackers can upload the web trojan directly and obtain the web permissions, and even get the highest system authority.

The code shown in Fig. 2 can lead to the occurrence of a file upload vulnerability.

```
<?php
if(isset($_POST["form"])){
      $uploadfile = "upfiles/".$_FILES['file']['filename'];
      move_uploaded_file($_FILES['file']['tmp_name'], $uploadfile);
      }
?>
```

Fig. 2 The vulnerability code

Session Hijacking.

The communication is connected via session between the server side and client, and the process is continuous or intermittent [8]. The server will establish a session for the user after the client browser connecting to the server, and each user's session is independent and maintained by the server.

Session hijacking occurs when an attacker uses a variety of means to obtain the session id from the specified user. The attacker will disguise as the identity of the user to visit the website when he gets the value of that id, and then obtains the corresponding permissions. There are many ways to get the user's session id from attackers, and the common methods include: 1) Enumeration method: attackers will generate a dictionary based on certain rules for the session id, and then continue to enumerate until success is achieved. 2) Man-in-the-Middle Attack: attackers get the session id through the network sniffer or intercepted manner. 3) Calculation generation: if session id is not randomly generated, it can be generated through a certain algorithm.

The attack steps for session hijacking are shown in Fig. 3.

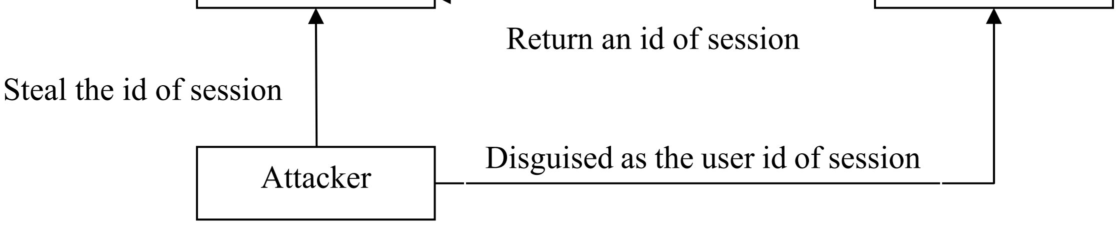

Fig. 3 Session hijacking step

Common Protection Methods for Server Side

Security attack and defense are not the opposite events, and the technologies of attack and defense are growing up in the process of competing with each other. If we don't know how to attack, we would not know how to defend [9]. It will help the defenders protecting their safety of website through the study of the web application vulnerabilities and attack means. Good safety protection scheme should be a system from design to operation, and not just security coding or testing. Any security schemes can't guarantee complete success.

Protective Measures for Sql Injection.

The attack of sql injection accesses the server through the port of 80. There is no difference with normal accessing to the web page, and the traditional firewall can't stop this attack [10]. At present, the common defensive methods are making security reinforcement to the web application and deploying web application firewall (WAF).

Making security reinforcement to the web application is one of the most popular methods for protecting from the attacking of sql injection. Firstly, the parameters from the users will be filtered and escaped for some keywords. Secondly, we should secure the source code, middleware and database, and improve the rationality of their application deployment. Lastly, the security logs of the web should be stored under relative safety directory, and the logs will record the time and method of attack from the attacker.

Web application firewall is a product which works in the application layer and provides security for web applications through specific security policies. WAF can effectively eliminate the vast majority of sql injection attacks. There are lots of keywords for sql injection in the rule library, which can effectively block the malicious requests combined with regular expressions and determine whether the parameters are passed by the malicious codes. For some variation attacks of sql injection, we can prevent the attacker from malicious access by updating the WAF rule or limiting the length of the passed parameters.

Protective Measures for File Upload Vulnerability.

The harm of file upload vulnerability is very serious, and attacker can upload malicious script to obtain the server management authority when the vulnerability exists in the server side. The key to prevent file upload vulnerability is to ensure that the uploaded files from the user could not be parsed into executable script [11].

Using Whitelist to Set up the Upload Types.

We should use a whitelist instead of a blacklist in the checking of file types. It will have a significant vulnerability risk when we use the blacklist and some types of files are not included in the blacklist. However, whitelist only allows specific file types to be uploaded, and other types will be banned.

Set the Saving Directory of Uploading File to Unenforceable.

If the server software is unable to resolve the file below that directory, the malicious script will not executed even if it has uploaded to the server, and the server will not be destroyed.

Protective Measures for Session Hijacking.

The attacker can disguise as the user and execute many illegal commands when the user's session is hijacked by the attacker [12]. The mainly effective solutions include: a) encrypting the data: for example, replace the protocol HTTP with HTTPS. b) allowing the value of session id to be set up only from cookie, which property is true, and not from the way via the resetting URL. c) giving session an additional value for creating the value of time, and as the server side timeouts, it will destroy the session id and generate a new session. To some extent, it will prevent the session hijacking from the attacker.

Conclusion

With the development of the information age, more and more services of web are provided. Although network security based on Web has received a lot of attention, attack and protection based on web server side is still a persistent and complex problem. The course of the study will encounter various difficulties because of the different situations. In this paper, we have provided a feasible scheme for attack and protection based on web server side and we hope that can provide reference for the research of web security.

Acknowledgements

This work is supported by NSFC (Grant Nos. 61300181, 61502044), the Fundamental Research Funds for the Central Universities (Grant No. 2015RC23).

References

[1]. 2013 TOP 10 List on https://www.owasp.org/index.php/Top_10_2013-Top_10

[2]. Wu Hanqing. White Hat Hacker Teach Web Security. Publishing House of Electronics Industry, 2014, p. 180-190.

[3]. Puspendra Kumar. A survey on SQL injection attacks, detection and prevention techniques. Computing Communication & Networking Technologies, Coimbatore, 2012, p. 130-132.

[4]. Wang Yun, Guo Waiping, Chen Chenghuan. Research on SQL injection attacks and guard method in web project. Computer Engineering and Design. Vol. 31 (2010) No.5, p. 150-152.

[5]. Song Chaochen, Huang Junqiang, Wu Qiong. SQL Injection Bypass Technology and Defense Mechanism. Information Security and Communications Privacy. Vol. 368 (2015) No.2, p. 110-112.

[6]. Xu Wenting, Xiao Qiang. Develope Security Web Application Based on PHP. Information Security and Technology. Vol. 51 (2015) No.7, p. 59-61.

[7]. Cheng Maohua. Based on PHP Security Vulnerabilities Web Attack Prevention Research. Information Security and Technology. Vol. 4 (2013) No.5, p. 53-55.

[8]. Yang Fengfan, Liu Jiayong, Tang Dianhua. Research of HTTPS Session Hijacking Based on Script Injection. Netinfo Security. Vol. 168 (2015) No.3, p. 59-63.

[9]. Pu Shi. Research on Web Security Penetration Testing (Master, XiDian University, China 2010). p. 20-28.

[10]. Ming Yu, Ling Jie. SQL Injection Vulnerability Detection Based on Sequence Value Comparison of Webpage DOM Tree. Computer Engineering and Design. Vol. 36 (2015) No.2, p. 350-354.

[11]. Jiang Yuyan, On The Techniques of Hackers Attacking and Defending Computer Network. Computer Applications and Software. Vol. 20 (2003) No.3, p. 56-58.

[12]. Xu Bowen, Liu Chunhui, Cao Weihua, Lu Xiaoming. Research on the Attacks and Defense of Web Real-time Session Hijacking. Network Security Technology & Application. Vol. 51 (2014) No.2, p. 60-62.