The Business Logistics Security Research on Web Application

Kangyu Tan1*, Yongjie Xie1
1 Network Engineering Department, Neusoft Institute Guangdong, Foshan, Guangdong, 528225, China
*Corresponding author’s e-mail: tankangyu@nuit.edu.cn

Abstract. The rich functions of web applications have played an important role in the rapid development of the Internet. Different web applications have different functions. Compared with traditional XSS vulnerabilities, SQL injection vulnerabilities, and command execution vulnerabilities, the security issues involved in different server functions of Web applications vary widely, which leads to the absence of general solutions for specific security vulnerabilities in business logic process. After analyzing the security of the business logic process, it can be found that when the various security points of the business logic process are used by the attacker, and the behaviour of the attacker does not have any obvious difference from the normal user, which leads to the inability to achieve effective protection by directly using various types of scripts. Therefore, it is of practical significance to research a penetration detection idea of a logical vulnerability.

1. Introduction

1.1 Research background and significance
Logging of logical vulnerabilities has always been a “lasting” topic in security testing. Compared with traditional security vulnerabilities such as SQL injection and XSS vulnerabilities, attackers are now more inclined to take advantage of application security issues in the business logic layer. Such attacks often bring great harm to system. In the face of this kind of attack, the traditional security measures can hardly get any effective defence. Logical vulnerabilities mean that some logic branches cannot process properly or errors occurred during process due to the lack of strict logic or using complex logic. Among them, the account system, transaction system, and business processing are often prone to problems.

Web applications are usually done by programmer. However, due to some of them lacking of security awareness, business logic vulnerabilities in the functional modules responsible for them cause the entire application to be in danger, and even the vulnerability is attacked. The risk of malicious exploitation can have very serious consequences. In order to ensure the overall security of web applications, we investigate the attack and prevention methods of some common web application business logic vulnerabilities. Therefore, it is of great practical significance to study the penetration detection idea of a logical vulnerability.

1.2. Present situation
By observing academic researchers' research on the logical vulnerabilities of Web applications, it is difficult to find logical vulnerabilities by using automated detection tools, and observing common
vulnerability scanning tools at home and abroad can be seen that these tools have little effect on the
detection of logical vulnerabilities in Web applications.

In the 2018 cybersecurity situation report, the overall attack trend of the website showed an
increasing trend this year, with an average daily attack volume exceeding 800 million times,
accompanied by large fluctuations. In August 2018, the peak of the attack occurred, and the peak
reached 4.9 billion times a day. In the past year, hackers used more than 9 kinds of Web attacks, the
two most important ones: scanners and backdoors, as shown in Figure 1-1. The logical vulnerabilities
include other types of attacks and special attacks[1].

2. The basic knowledge of Web applications

2.1. Development and security issues of web applications.
Web application security is also called Web security. The Internet itself is secure. Since people began
to study security, the Internet has become unsafe. The essence of Web security is the issue of trust.
Normally handling malicious input from users leads to security problems. Safety is the principle of
wooden barrels. The short board determines how much water the barrel can hold. Similarly, assuming
that 99% of the problems are handled, 1% of the remaining will be the short board that causes safety
problems.

2.1.1. Development of Web applications, In the 1960s. GML (Universal Markup Language) and later
SGML (Standard Generalized Markup Language) appeared.
In the 1990s, HTML (Hypertext Markup Language) and browsers appeared.
In 2004, XMLHttpRequest (asynchronous request) appeared.

2.1.2. Development of Web Application Security. In the Web 1.0 era, more attention is paid to
server-side script security issues, such as SQL injection.
In the Web2.0 era, the outbreak of the Samy worm in 2005 shocked the world. In addition, attacks
such as XSS and CSRF have become more powerful. The idea of Web attack has also shifted from the
server side to the client side, turning to browsers and users.

2.2. The basis of penetration testing technology for Web applications.
Penetration testing is a legitimate, authorized system attack behaviour designed to assess the security
of a system based on a structured procedure for step-by-step penetration testing. “Planning and
Preparing, Investigating, Discovering, Analyzing and Evaluating, Actively Invading, Summarizing,
Writing Reports” Broadly speaking, there are three types of penetration testing: zero knowledge,
partial knowledge, and full knowledge, namely black boxes, gray boxes, and white boxes. The
difference between each type of penetration testing is the amount of information the evaluator gives
before the test begins. There are a large number of user interaction functions in the Web system. As
the name implies, users can interact with the server to implement specific functions. At present,
attacks against logical vulnerabilities (business processes) are emerging one after another, so it is more
necessary to attract the attention of developers and security operations personnel.

2.2.1. HTTP protocol overview. HTTP is an object-oriented protocol, located at the application layer,
which is ideal for Internet applications due to its simplicity and speed. With HTTP, users can access
different application systems by using a browser, avoiding the inconvenience of a large number of
client operations. At the same time, this way of requesting by the client and processing by the server
according to the user's request is also very suitable for large-scale application development.

The main features of the HTTP protocol can be summarized as follows:
• The HTTP protocol is simple enough to be summarized as "user initiated request, server response,
  new request re-initiated", each request is an independent behaviour, which reflects the stateless nature
  of HTTP.
The HTTP protocol supports the B/S mode and works as long as there is a browser. The user is simple to use and easy to operate it. In a sense, an APP can also be viewed as a browser for a particular content.

The HTTP protocol is flexible and can be used for data transmission, video playback, interaction, etc., so it is suitable for a fast iterative Internet application environment.

For Web security itself, HTTP is the transmission method of the application layer. At present, a large number of security problems are brought about by HTTP applications, but HTTP itself does not have very good protection measures. For example, a door lock is not safe. The first solution is the safety of the door lock. However, there is not much direct protection against the corridor (the HTTP protocol) that the door lock relies on.

2.2.2. HTTPS protocol security. Strictly speaking, HTTPS is not a alone protocol, but an HTTP protocol that works on the SSL protocol. SSL is a secure protocol that provides security and data integrity for network communications. Its subsequent specification protocol TLS extends the original SSL protocol. Currently, the HTTP protocol uses TLS to implement the transmission encryption process. Commonly, the HTTPS protocol is that the HTTP relies on the SSL protocol to achieve reliable data transmission. This is also a measure to effectively protect data security.

The main features of the HTTPS protocol are summarized as follows:

- HTTPS does not change the characteristics of the HTTP protocol itself, but uses SSL/TLS technology for encryption to ensure security during transmission.

- HTTPS technology can effectively protect user information from being leaked, avoid obtaining current user behaviour by online behaviour devices or proxy devices, and can effectively avoid TCP hijacking from operator networks.

- HTTPS mainly protects the security during transmission. If Burp suite is used at the client, the hijacking of Web access can still be achieved through proxy technology, so it will not effectively improve the security of the server.

HTTPS focuses on the security issues in the transmission process, which can be used to ensure the security of the data transmission of the client, and does not directly improve the security of the Web site. The problem of web security still needs to be solved from the functional layer, to find out the keys of solutions and let the problem be solved effectively.

2.2.3. Encoding and encryption. Characters are a general term for various words and symbols, including different country characters, punctuation marks, graphic symbols, numbers, and so on. There are a large number of different languages in the world, and the expression or format used in each language is different. In web applications, you must consider using some encoding to represent the text and format of the language. There are two types of encoding: encoding for characters and encoding for transmission. The standard method of encryption is to encrypt the parameters submitted by the user (such as passwords and other sensitive information), and then avoid the parameters being hijacked during transmission, resulting in user data loss. When the data is transferred to the web server, the server will decrypt and process the parameters. The encryption for process is also divided into two types: the server does not need to know the content of the plaintext, and the server needs to know the content of the plaintext.

3. The analysis of web application business logic security vulnerabilities
Web logic vulnerabilities are a new type of vulnerability that has emerged in recent years, unlike traditional SQL injection, cross-site scripting attacks, and file inclusion vulnerabilities. This kind of vulnerability is a mistake in the logic of human thinking. Generally, it tampers with data by using business processes and HTTP/HTTPS requests. After finding a key point, it usually does not need to construct a malicious request to complete the attack. It is easy to circumvent various security measures. And there is no fixed pattern for attacking logical vulnerabilities, so it is difficult to detect them using conventional vulnerability detection tools.
The user management function module is a functional component that directly relates to the core information and benefits of the user, and its security affects the normal development of the user or even the Web service. Therefore, the user's basic management function module is taken as the research object, and the security situation of common user registration, user login and business development process is discussed item by item.

3.1. Penetration testing of business logic vulnerabilities

3.1.1. Common business logic vulnerability classification. OWASP divides logical vulnerabilities into ten categories: business consistency security, verification code breakthrough, business authorization security, business process out of order, business interface invocation, aging bypass testing, identity authentication security, retrieving password vulnerabilities, business data tampering, users Enter legality[3].

3.1.2. Web application logic vulnerability analysis. The password reset vulnerability is a vulnerability that exploits the password recovery service. Because users inevitably forget passwords, many websites have designed a mechanism to help users get back their passwords. This mechanism generally authenticates users by means of sending verification codes for user mobile phone or sending verification link for user email. There are several reasons for this type of vulnerability[4]:

- The response packet containing the SMS request contains a verification code
- When resetting the password, it is not verified whether the phone or mailbox is associated with the user to be reset.
- Verification code is always valid
- No defensive blast attack

To sum up the above reasons, the breakthroughs of such vulnerabilities are as follows: force cracking authentication credential, authentication credential echo, authentication credential reuse and rebinding, user identity authentication, server-side authentication, local authentication, password recovery process bypassed etc. Therefore, when designing the password reset mechanism, we must consider whether there will be vulnerabilities due to the above reasons. There are three main repair methods, and the recommendations are enabled:

- Limit the times of sending verification code, such as 5 times, and then the verification code will be invalid regardless of the right or wrong.
- Limit the verification code submission frequency, control for more than 1 second
- Limit the valid time of the verification code, such as the limit time is 5 minutes

Exceed Authority Access is a common business logic vulnerability in web applications. Because of its wide range and great harm, it is ranked second among the top ten security risks of Web applications by OWASP. Exceed Authority Access is divided into vertical override access and level override access. Vertical override is an override between users of different levels, such as an ordinary user performing administrator user's rights. Level override is an override operation between users of the same level. The reason for this type of vulnerability is

- The server does not isolate the user's data, such as the A user can access the B user's order.
- Background business logic incorrectly trusts user information from the foreground

Since the front-end parameters are user-controllable, there is an opportunity to modify such permission information. If an unauthorized access vulnerability occurs in a web application, it may lead to user sensitive information leakage or malicious modification or deletion of user information. Fixing such a vulnerability usually requires starting with a business unit. Any sensitive data about the user is taken from the database and verified if the user's Session is their own. In summary, don't trust any critical information from the user side.

Business data tampering vulnerability, Business data tampering often leads to economic losses directly to the business, and the most frequently falsified is the transaction amount and quantity. This
kind of vulnerability belongs to one of the top ten security risks of Web applications in OWASP statistics, and the scope of influence is quite wide. There are several reasons for this type of vulnerability:

- Order data packet is transmitted in plain text, unencrypted, and there are no restrictions and secondary verifications on the amount and quantity. For example, it must be a positive number, etc.
- The payment amount is not checked against the third-party payment platform. The amount exceeds the specified size. No manual review, etc.
- Refund and other sensitive operations on the server side did not verify the amount, quantity, etc.

To sum up the above reasons, the way to prevent business data from tampering with vulnerabilities is to perform secondary verification or encryption on the data to prevent tampering.

4. Summary

Business logic type vulnerabilities are different from other vulnerabilities. This is directly related to the business logic design of current web applications, and is directly related to the design ideas and implementation methods of each business in terms of business logic. Therefore, the mining and protection of business logic vulnerabilities must be analyzed from a business perspective. After perfecting the defects of the original business logic vulnerability, it is clear that which protection measures are effective. Since the code writing methods of different types of business systems are different at the time of design, it is necessary to re-combine the business logic to correct the security risks.

References

[1] Know Chuangyu Cloud Security China (2019) Annual Report | Know Chuangyu Cloud Security 2018 Network Security Situation Report. https://www.yunaq.com/news/5e6a1c2c796db42f25c77b7a/?ad_code=BG&source=baidu&utm_medium=SEM&e_keywordid=113785250907&e_keywordid2=113785250907

[2] Cai Jingjing, Zhang Zhaoxin, Lin Tianxiang. (2018) The Basics of Web Security Protection Guide. China Machine Press Publishing, Beijing.

[3] Xue Nanfeng. (2018) Research on logic vulnerability detection technology based on penetration test. Shanxi: master’s degree thesis of University of Electronic Science and Technology, 7-11.

[4] Feng Dan. (2017) Research on Web Application Business Logic Vulnerability Detection Technology, Zhejiang: master’s degree thesis of Zhejiang University of Commerce and Industry, 13-15.