Web Application Security Threats and Protection Technical Analysis

Ying Wu[1,a]

[1]Shanghai University of Political Science and Law, Shanghai, 201701, China
[a]wuying@shupl.edu.cn;

Yu Sheng[1,b]

[1]Shanghai University of Political Science and Law, Shanghai, 201701, China
[b]shengyu@shupl.edu.cn

Abstract : With development of Internet technology, Web technology has been increasingly widely used..More and more Web-based application systems are deployed on the Internet to provide a wide range of services, due to the openness of the Internet itself, it always faced with potential malicious attacks. Security issues have become more prominent. In this context, the study was conducted and Status Web application security protection technology has great practical significance. In the paper, we will illustrate the issues from current Web security threats and the problems faced by Web applications, and server and client from two aspects of today's Web application security threats faced by systematically analyzed, and for common security threats facing their respective proposed the safety program and protection recommendations.

Keywords: Web application Security network Security threats

1. Introduction

Traditional network security system has been unable to ward off hackers of today. Today, cyber criminals have not only their eyes locked on the well-known large companies, each vulnerable sites are likely to be targets of attack.
Therefore, protection of Web application security has become an important research topic.

2. Current Web Problems

2.1 Users do not understand the Web technology

Perhaps one of the most prominent features of the browser is the most users know nothing about technology. Of course, playing from the date of birth of the computer, computer white entertainment have been a very, very innocuous question. But since the Web increasingly deepened people's lives, due to the very low threshold, so we ran into a new situation: the majority of users on how to stay safe online is completely no idea.
For a long time, the engineers in the development of common software, generally will not take into account the level of the user's computer. In most cases doing so really no big problem; for example, the value of a text input box quite right, hardly any impact on the security of the entire system. If the user's operating problems, and that he probably could not handle the software, it would be an excellent self-correcting mechanism of.
But this law on the Web browser was not working. And other complex software is different, even if the user even text editors are used not to, but the use of a browser is completely no problem. But at the same time, but only for computer technology and public key infrastructure (Public-Key Infrastructure, PKI) such technical terms quite understand, are likely to safely use the browser. Needless to say, nowadays there are numerous popular Web applications, the target population, the vast majority do not meet this requirement.

2.2 It is difficult to isolate the Web operating environment

Isolation of applications and application-related data between completely unrelated very weak. In the past 15 years, the traditional model of PC era in the application layer data objects (documents), bound code (applications) user layer between the operating system kernel and is very clear, by the operating system kernel is responsible for all cross-application communications, hardware, input / output (I / O) and through a configurable security policies restrict application of cross-border actions. These boundaries have been well researched, but are also available to create the actual security policy very helpful. In a text editor to open the file, it is almost impossible to steal your e-mail, unless the implementation is very unfortunate indeed have a serious defect, causing all of the isolation layer are completely ineffective.

2.3 Browser, lack of a unified security mechanism

Web did not a common overall security model. In this regard, we did not expect to have a grand vision of world peace so resolved, said here are just some of the regular common and flexible security paradigm set, if not for all situations, but it can solve the vast majority of the relevant safety logic on the line. For example, in the UNIX system, the user / group permissions mode rwx way the case is such

a highly unified security model. But in the browser field what is it?

In the browser field, "same origin policy" mechanism can be kind of core security paradigm, but in fact a lot of problems this itself is a cross-domain interaction mechanism is only in a small subset of it. Even if only discuss same-origin policy, there are still at least seven kinds of usage scenarios, which makes security boundary between different applications will be slightly different. There are also several mechanisms, they and homology model has nothing to do, but controlling for other key actors in the browser.

2.4 cross-browser synergistic interaction problems

When multiple browsers when attempting to interact with each other where there is a series of hard to blame on the specific code segment but very serious vulnerability. You can not ferret out which specific product is the culprit: they are nothing but the diligence to complete the task it, the only problem is that not all browsers define a common specification should comply with them. For example, a browser that, according to its security mode, the URL for a particular program or to the external storage / read certain types of data on the hard drive is safe. Of such assumptions, almost always there is some recognition of the browser completely, and expect other browsers will act according to their own rules. And each manufacturer also wants his hand was as long as possible, so often did not inform the user in case the user does not give permission, forced to use their own browser to open the page.

Another closely related issue is that even though on the surface it seems the browser's security mechanism is very similar, in fact they are not compatible, this rarely occurs before the Web appeared. If the various browser security model is different, then a strip Web application development standard for one browser may be reasonable, but for another it may be completely appropriate and would be misleading. In fact, even some very basic tasks, such as opening a plain text file provided by the user in some browsers can not safely be achieved. These issues application developers are often unaware of, unless they happen to be using this affected browsers - and even then, often need to wait until they stepped on a mine-awareness has been.

2.5 client and server-side boundaries blur

Web of origin in full compliance with the regular "client - server" architecture, but functional boundaries client and server-side response was quickly blurred. JavaScript culprit is technology, which in the browser (that is, "Client"), the application logic HTTP proxy server implementation, doing so with two very attractive reasons. First of all, this approach makes the user interface more sensitive response, because every tiny change in state UI and server do not need to be synchronized. Secondly, greatly reduces the server CPU and memory requirements, since the equivalent of each separate computer through participation around the world, reducing the amount of computation. But security issues need to be responsible for the client, which is obviously unrealistic.

In the traditional "client - server" model where there are clear and specific uses of API, regardless of the client it can be very easy to evaluate the behavior of the server, and vice versa. Further, in each component, the fan can easily isolate smaller functional groups, to determine what action will be within this range. But a new model Web, plus the common Web application API is vague and temporary, so those previously analyzed by means of rational inference security of a system has been completely impossible.

3. Web Application security threats

Many Web applications vulnerable to attack by servers, applications and internally developed codes of. These attacks directly bypass the perimeter firewall security because port 80 or 443 (SSL, Secure Socket Layer) must be open to allow the application to work. Web application security presence illegal input, Broken access control, failure of the accounts and thread management, cross-site scripting, buffer overflows, injection attacks, exception error handling, unsafe storage, denial of service attacks, insecure configuration management issues. Chief among them is the following four attacks:

(1) Injection attacks. Injection vulnerabilities, such as SQL, OS, and LDAP injection. Most notably SQL injection attacks, SQL injection attacks are the main way cleverly constructed SQL statements submitted content and pages combine injection attacks. The more common techniques have use annotation symbols, identities (eg 1 = 1), a joint union statement query, insert or update statement insert or modify data. In addition you can also use some of the built-in functions supporting attacks, such as the use phpinfo function to display basic information, char function to avoid the single quotes and so on.

(2) DOS, DDOS attacks. DOS is a "denial of service" (Denial of Service) acronym, it refers to the deliberate attacks on network protocols of defects or directly affected by resource depletion targeted by brutal means, the aim is to make the target computer or network can not provide normal services, and even system crashes. Early DoS attacks require considerable bandwidth resources to achieve, and the individual units of the "intruders" are often no such conditions. But then the attacker invented distributed attacks, namely the use of software tools to set a lot of network bandwidth requests simultaneously launching a large number of attacks on the same target, which is DDOS (Distributed Denial Of Service) attack. In short, DDOS attacks are by the "invaders" centralized control, DOS attack launched by a group of collection, very difficult to resist.

DOS, DDOS attack online business, resulting in service paralysis, seriously affecting the availability of the system.

(3) Cross-site attacks (XSS attacks): When the application receives the data contains not credible, in the absence of proper validation and escaping, it will be sent to a web browser, which will have a cross-site scripting attacks. An attacker using a web application for lack of user input filtering, input can be displayed on the page impact on other users of HTML code to steal user data, using the user identity or some action on your visitors viruses or redirect the user to a malicious Web site.

(4) website linked to horse: Website linked to horse refers to a Trojan horse program which is then uploaded to a Web site with a Trojan horse give birth to a network builder, then uploaded to the space inside, plus the code so that the Trojan runs when you turn the page! As a site linked to the horse spread, its purpose is to Trojan downloaded to the user's local, and further implementation, after the Trojan implemented, it means there will be more Trojan is downloaded, further execution into a vicious cycle, so that the user's computer attack and control.

Website linked to horse cause the user image is destroyed: the attacker through the normal page (usually the homepage) into a piece of code, Internet users open the page, the code is executed, then download and run a Trojan horse server-side program, and then control the Internet's host.

4. Web application security measures

For the above-mentioned Web application security risks faced improve Web security, establish and improve Web protection system, here for protection from server and client two aspects protection

4.1 Web Server

(1) To the Web server service or daemon configuration allows it to normal operation with minimal privileges. Thus, even if the attacker control of the Web server, they can only get permission to run the software corresponding to the user accounts. Thus, the attacking computer programs or other software on the network viable is extremely limited.

(2) Install the latest security patches and always pay attention to the latest developments in vulnerability.

(3) Remove the default installation example and avoid similar examples.

(4) By removing unwanted applications, security configuration other network services on the same computer, make sure that the operating system has the latest security patches to ensure the security of the computer hosting Web server.

(5) Ensure that permission is only required to separate directory script execution run.

4.2 Web client protection

1. How to make Web applications more secure?

Specific implementation from the following aspects

(1) query parameterization: There are many attacks against Web applications can be traced to successfully steal passwords SQL injection attacks. Business, government, social networking sites have become the victims of this attack, which makes it a common problem. While many people think the problem is manufacturers issue, but fundamentally it belongs to the developer of programming problems.

Web form comment box, data field, or to allow free data entry form regions, particularly open string input, will cause this loophole. SQL injection attacks can even non visible Web elements (such as the HTTP header values) to pass. Simply insert malicious SQL code can sometimes lead to the entire database there may have been stolen, remove or tamper with, and even be used maliciously run the operating system command to destroy the enterprise database.

To prevent SQL injection, developers must prevent part of the non-creditable input is parsed into SQL commands. The best way to prevent SQL injection is through the use of parameter queries programming techniques.

(2) To ensure the safety of storing passwords: Obviously, if the SQL injection can be used to steal passwords, we need to save the user password security. Save password worst way of course is to use plain text; however, encryption is not much better. The reason is that the encryption is reversible, there is a reason for that or any other MD5 hash algorithms are problematic. Today's hackers can access the powerful but not very expensive computing resources, which allows them to create even buy "Rainbow table", which can be deciphered in real time the general strength of the password. Today, hackers are no longer even use rainbow tables in favor of high-performance home computers to perform high-speed dictionary attacks. Password salt technology is a help to password hash table attacks and delete duplicate data encoding technology, but the technology just is not enough.

Use of limited resources, the attacker can generate GPU cracking program for storing passwords, it can perform 25 billion times per second password attempts.

GPU to store passwords and prevent cracking programs and similar resources reveal the password, we recommend combination of three main techniques: the use of one-way algorithm, salt, intends to take advantage of the slow algorithm. There are two good algorithm, SCRYPT and PBKDF2 in this form can be used to securely store passwords.

(3) output encoded XSS defense: Cross-site scripting attacks (XSS) have a more appropriate name, namely JavaScript hijacking, which can be used for session hijacking, sabotage sites, network scanning, destruction CSRF defense, loading site to redirect or fishing, remotely hosted scripts, data theft and keystroke recording, etc., but the attacker is also increasingly frequent use of XSS.

Output coding is a key programming techniques for preventing XSS, this technology executed at the output. When you build the user interface, such codes can be dynamically added to the final moments before the execution HTML untrusted data.

First rule is to deny all, that is not to put untrusted data to an HTML document, except for special circumstances. The

most important thing is, do not accept the JavaScript code from an untrusted source.

(4) In order to make effective content security policy, embedded into HTML you need to remove all the JavaScript, and deploy a separate external JavaScript files. From this point of view, if we can understand the contents of the security policy of the browser detected the HTML document embedded in JavaScript (for example, a hacker attempts to insert such a script), the operation will be rejected. This approach can really lock the browser, preventing many forms of XSS attacks.

5. Conclusion

WAF more features, including the ability to deliver safe, cache-based application acceleration, hanging horse inspection, anti-DDOS attacks, in line with PCIDSS of anti-leak requirements so that this is not only a bell attack protection, while meeting the customer experience and protection of highly confidential data integration expertise. This paper analyzes the only part of the technical principles of WAF, but did not deny the value of the IPS, after all, both have great differences in the deployment scenarios and functions.

References

[1] Heith E. Strassberg waiting Ang translated firewall technology Daquan Beijing: Mechanical Industry Press, 2003.3

[2] Pan Zhixiang, edited hacker attack and defense Programming parsing Beijing: Mechanical Industry Press, 2003.6

[3] Zhouhai Gang An intrusion detection system framework mobile agents, University of Electronic Science and Technology Based on Volume 32, No. 6 December 2003