International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

Web Security Problem and Web Service Attack

Simanshu Chaubey[1], Jaya Singh[2]

[1]Ewing Christian College, Allahabad, Uttar Pradesh, India
[2]Ewing Christian College, Allahabad, Uttar Pradesh, India

Abstract: ***In the present day's the security of web pages is performed so, we need to provide the security to the web pages which are vulnerable to such type of attacks. SQL injection attack, XML injection attack and so many attacks which are discussed. This research paper is focus on various aspect of the web security. SQL injection exploits security vulnerability in application software. SQL injection is most known as an attack vector through public facing websites, but can be used to attack SQL database in a verity of ways.***

Keywords: Creation of web page, Web service vulnerability, Web service attacks, Web service attack modification, Denial of Service attacks, SQL injection.

1. Introduction

Web security group is a part of ECC laboratory. In the current day, many applications and web pages depend upon web services to seamlessly conversation informationamong one another. So the main purpose of making the research paper is making the web page and how to secure our created web page.

This research project focus on various aspects of web Application security

2. Research Approach

This paper performs aninclusive analysis of current studies on web service security.

3. Paper Inclusion Standard

Papers with the following standards were included:

- Papers that describe answer to address the web service security problematic.
- Papers that use challenging methods for vulnerability discovery in web services.
- Papers that address web service attack such as SQL Injection, Spoofing and Denial of Service.

a) What are the web service security problems that are talked in the research papers?

There are many attacks on web services

1) SQL Injection Attack

SQL injection attacks are very common in a web service environment. Most of the web services have incorrectly coded chunks that fail to filter non-validated user participations. These inject and insert them as a restriction in a SQL statement trying to run non-administrative commands

2) XML Injection Attack

When any service fails to validate malicious. XML injection vulnerability arises. The injection of malicious XML content into any service can alter the operational logic.

When any service miscarries to validate malicious.XML injection vulnerability rises. The injection of Malicious XML content into any service can change the working logic.

b) Website Security Vulnerabilities?

The most common security vulnerabilities you must defendactually against.

1) Cross-Site Scripting (XSS)

XSS target an application's users by injecting code, usually a client-side script such as JavaScript, into a web application's output. The concept of XSS is to encouragement client-site script of a web application to complete in the manner desired by the attacker. XSS allow attackers to execute scripts in the victim's browser which can takeover user session, treat websites, or redirect the user to malicious sites. The XSS vulnerabilities are divided into imitated and persistentbuilt on how the site returns the injected scripts to a browser.

- A reflected XSS weakness occurs when user satisfied that is passed to the server is returned immediately and original for display in the browser. Any scripts in the original user contented will be run when the new page is loaded.
- A persistent XSS vulnerability occurs when the malicious script is stored on the website and then later redisplayed unmodified for other users to execute accidentally.

For example, a discussion board that accepts commentaries that contain original HTML could store a malicious script from an attacker. When the comments are displayed, the script is executed and can send to the attacker the information required to access the user's account. This sort of attack is very popular and powerful, because the attacker might not even must any direct promotion with the victims.

2) SQL injection

SQL Injection is a type of web application security vulnerability in which an attacker challenges to use application code to access or corrupt database content. If successful, this permits the attacker to create, read, update, alter, or delete data stored in the back-end database. SQL injection is one of the most leading types of web application security vulnerabilities.

3) Cross-Site Request Forgery (CSRF)

CSRF attacks allow a malicious attack where a user is tricked into presentation an action he or she didn'tmean to do. A third party website will send a request to a web application that a user is already authenticated against (e.g. their bank). The attacker can then access functionality via

the object's already authenticated like social media, in browser email clients, online banking, and web interfaces for network devices. Don't get caught with your guard down. Practice harmless website security measure and always be ready to protect yourself, and your company's future, from. The best method to tell if your website or server is vulnerable is to conduct regular security audits.

4) Other Threats

Other common attacks/vulnerabilities include:

a) **CLICKJAKING: -** In this attack, malicious user takeovers clicks meant for a visible top-level site and paths them to anunknown page beneath. This methodstrength be used, for example, to display a legitimate bank site but capture the login identifications into an invisible <iframe>controlled by the attacker. Click jacking could also be used to get the user to click a button on a visible site, but in doing so actually accidentally click a wholly different button. As a defense, your site can prevent itself from being fixed in an iframe in another site by location the proper HTTP headers.

b) **Denial Of Service (DOS):-** DOS is usually achieved by submerging a target site with bogus requests so that access to a site is troubled for legitimate users. The requests may simply be many, or they may individually consume large amounts of resource (e.g., slow reads or uploading of large files). DOS emplacements usually work by identifying and blocking "evil" traffic while permittingreal messages through. These earthworks are typically located before or in the web server (they are not part of the web application himself).

c) **Directory Traversal** (File and discovery). In this attack, a malicious user efforts to access parts of the web server file system that they should not be able to access. This vulnerability occurs when the user is able to pass filenames that comprise file system direction finding characters (for example, /.../). The answer is to clean input earlier using it.

d) **File Inclusion: -** In this attack, a user is able to specify a "chance" file for display or presentation in data passed to the server. When loaded, this file strengthremains executed on the web server or the client-side (leading to an XSS attack). The solution is to sterilize input before using it.

e) **Command Injection**: - Commandjab attacks allow a malicious user to execute chance system commands on the host operating system. The solution is to perfect user input beforehand it might be used in system calls.

Simple Ways to Improve your Website Security

1) Keep your software up-to date.
2) Enforce a strong password policy.
3) Encrypt your login pages.
4) Use a Secure Host.
5) Keep your website clean.
6) Backup your data.
7) Scan your website for vulnerabilities.
8) Hire a Security Expert.

C. How to protect your website from any attack

1. SQL Injection

SQL injection attacks are when an attacker uses a web form field or URL parameter to handle or to gain access your database. When you use standard handle SQL it is easy to accidentally insert rogue code into your query that could be used to change tables, get info and delete data. You can easily prevent this by always using parameterized inquired, most web languages have this features and it is easy to tool.

Consider this query
"SELECT*FROM table WHERE column="+parameter+";"
If an attacker changed the URL parameter to pass in 'or '1'='1 this will cause the query to look like this:
"SELECT * FROM table WHERE column=" OR '1'= '1';"
Since '1' is equal to '1' this will allow the attacker to add an additional query to the end of the SQL statement which will also be completed.

2. Protect against XSS attacks

Cross-site scripting (XSS) attacks inject malicious JavaScript into your sheets, which then runs in the browser of your operators, and can alteration page content, or steal info to send back to the attacker. For example, if you show comments on a page without authentication, then an attacker might commentscover script tags and JavaScript, which could run in every other browser and steal their login info, allowing the attack to income control the account of every user who viewed the observation. It is to be defensive that user cannot insert active JavaScript content into your pages.

Another powerful tool in the XSS protector's toolbox is CSP (content security policy). CSP is a goal your server can reoccurrence which expresses the browser to limit how and what JavaScript is executed in the page, for example to cancel running of any scripts not hosted on your domain, cancel online JavaScript, or disable evil().

3. Use HTTPS

HTTPS is a protocol used to provide security over the internet. And it assurances that user is interactive with server they expect, and that nobody else can disturb or change the content they are sighted transit. If you have anything that your users might want private, it's highly sensible to use only HTTPS to deliver it. That of course means credit card and login pages (and the URLs they submit to) but typically far more of your site too. A login form will often set a cookie. Example, which is sent with every other request to your site that a logged-in user makes, and is used to authenticate these request. An attacker theft this would be able to perfectly copy a user and tale over their login session. To downfall these kinds of attacks, you must always want to use HTTPS for your entire site.

4. Get Website Security Tools

Once you think you have done all you can then it's time to test your website security. The most effective way of doing this is via the use of some website security tools, often referred to as diffusion testing or pen testing for short.

Some free tools:

a) **NETSPARKER**: - good for testing SQL Injection and XSS.
b) **OPENVAS:** Claims to be the greatestliberal open source security scanner. Good for testing known Vulnerabilities, currently scans over 25000. But it can be problematic to setup and requirements an OpenVas server to be installed which only runs on *nix.
c) **SECURITYHEADERS.IO:-** a tool to fast report which security headers stated above (such as CSP and HST) a domain has allowed and properly configured.
d) **XENOTIX XSS EXPLOIT FRAMEWORK:** - A tool from OWASP (open web application security project) that contain a huge selection of XSS attack sexample, which you can run to fast confirm whether your site's weak in chrome, Firefox and IE

4. Conclusion

The main objective of this paper is to provide the security of our web page. This research paper gives detail study about security of web page, how to find the vulnerability and how to shelter our webpage from the attackers.

Web page has become the primary way in which key info is impeccably exchanged between applications. This make web service security as an important component and web service attack a serious threat to the integrity and availability of data. We have judiciously analyzed on web service attacks. Most addresses attacks are Denial-of- service attacks followed by xml injection attack.Methods to deal with bouts predominately focus on attack detection measures.

Since web service attacks difficultwould be complete as part of every development. This will guarantee extraguard as well as subordinate attacks on web services.

References

[1] www.google.com.
[2] www.defenseCode.com/
[3] https://www.researchgate.net>publication
[4] www.williamstallings.com
[5] http://www.wikepedia.com
[6] William Stallings, cryptography and network security.
[7] Kshitij pathak "SQL injection attacks: techniques and prevention mechanism"