International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN : 2456-3307

A Study on Common Web Based Hacking and Preventive Measure

Surajit Sarma
DOEACC B Level, M.Sc(IT), Krishna Kanta Handiqui State Open University, Guwahati, Assam, India

ABSTRACT

Internet has emerged as new trend in today's world. Organizations are adapting web based technology to interact with user thorough online. Web based technologies are easily accessible and is available anytime from anywhere. With the increase in use of web based technology the internet related crimes are also increasing. Web based hacking is one such threat that has affected many websites of well know organizations. In this paper I have tried to threw light on the areas of web site that are more often attacked by hackers and the measures that we can take to prevent it.
Keywords: Internet Security, Hacking, Website Development

I. INTRODUCTION

In this 21st Century whole world is converting to digital world. From a small business to a large business web based technology has become a common need for every organization. Web technology has become one of the most prevalent technologies for information and service delivery over internet. [1]

A web application is an application that works on client server model. Many web applications used database in their back-end that contain sensitive information. Increase in use the Web based applications have given rise to many security threat. Due to poor coding of website they are more prone to security attacks.

A web application is a complex structure that is composed of number of components and technologies such as HTTP protocol, PHP, CGI, ASP, web browser, web server, database such as MySQL, MSSQL, PostgreSQL, SQLite.

As per report of the Web Application Security Consortium, review has been done upon 49% of web application and it was found that they contain vulnerabilities of high risk level out of them not more than 13% of websites can compromised completely automatically. [2]

II. COMMON WEB APPLICATION THREATS

Web applications are mostly hosted in public server which can be easily accessible to the user through internet. Due to easily accessible of web application by everyone it has make it vulnerable to attack. The common web application threats have been discussed below:

1. *SQL Injection*: SQL Injection is a most common web hacking technique. In this technique a malicious code in sql statement are injected through web page input to destroy our database. SQL Injection mostly occurs when there is a situation where the user has to passed some sort of data as input to our web application such as login password or username, the hacker may take the advantage by passing malicious SQL statement which our system will run unknowingly.

2. *Denial of Service Attacks*: In this type of attack an attacker flooded the network with excessive message such that the legitimate user is prevented form access the service. There are two most common form of DOS attack –flooding and crashing service.

In flooding service, the network is flooded with unwanted traffic causing the network to slow down.

Crashing service basically exploit the vulnerabilities to cause the target system or service to be crashed. In this type of DOS attack taking the advantage of bug in the system the attacker sends malicious inputs that causes severe damage to the system causing system crash.

3.*Cross Site Scripting XSS*: This is the type of attack in which the attacker injects client side script in the web pages that is being viewed by the user. Through this the attacker gain access to our cookies, session IDs, password, private message etc. All the information of a user could be access and displayed by an attacker through this attack.

4. *Cookie/Session Poisoning*: Cookie/Session poisoning is another great threat to our web based application. In this technique an attacker achieves impersonation and breach of privacy by manipulating the session cookies. Tokens are not generated in a secure manner that is why an attacker can forge these cookies and thus impersonate a valid client and gain his personal information and perform action on the behalf of the victim.

5. *Form/Parameter Tampering*: This is a type of attack in which the attacker modifies the parameters that are being exchange between the client and the server. The attacker mainly targets the hidden field and cookies in order to modify application data, such as user credentials and permissions, price and quantity of products etc that is being passed by client to the server. This attack can be performed by a malicious user who wants to exploit the application for their own benefit, or an attacker who wishes to attack a third-person using a Man-in-the-middle attack.

6. *Code Injection:* In this type of attack the hacker injects codes written in different programming language such as PHP, PYTHON etc to the server which are executed in the server and reveals sensitive information. The system which is poorly programmed, lack behind proper input/output data validation are prone to such attack.

7. *Defacement:* Website defacement is a kind of attack where the hacker changes the visual appearance of the site by breaking into the webserver and replacing the hosted website with their own.
As per [10] Website defacement is usually done using SQL injections to log on the administrator's account. The usual targets for defacement are government organizations and religious websites. These acts are usually perpetrated by activists (or hacktivists) working against the principles and ideals of the sponsoring organization.

III. PRECAUTION

Precaution that we can take to minimize the website hacking are discussed as below:

1. *Protection from SQL Injection*: The best way to protect our web application from SQL Injection is to use SQL parameter. This are the values that are added to SQL query at the time of execution in a controlled manner. The SQL engines checks each parameter to ensure that it is correct for its column and are treated literally, and not as a part of the SQL to be executed. [4]

2. *Protection from Denial of Service Attack*: There is no effective ways to prevent the DOS attack but by taking certain steps we can reduce the chance of being a victim. We must install anti-virus in our system and regular update it. By installing and configuring the firewall we can restrict the traffic coming in and going out of our computer. By following good security practice while distributing e-mail.

3. *Protection from Cross Site Scripting XSS:* Prevent the creation of attributes by converting the untrusted user input to HTML entities. Avoid passing user data into id, class or name parameter. Be very cautious in handling through DOM event handler.

Passed encoded data through URL. Use proper URL library to run user data. JavaScript protocol handler must be avoided. The use of ampersand as it may lead to parameter pollution issues should be avoided. Prevent line breaks.

We should have enforced the use of safer functions whenever applicable and should be very careful the data that is allowed to be printed.

4. *Protection from Cookie/Session Poisoning:* Although it is difficult to completely prevent a web system from Cookie/Session Poisoning certain steps could be proof effective to reduce such threat. By encrypting the content of cookies we can prevent it from such attack. We should regularly clear stored

cookies from our browser. It is advisable to regularly do virus and malware scanning to keep the browser free from malicious scripts which can hack sensitive information from the cookies. We can also give time out for a cookie after sometime.

5. *Protection from Form Tampering/Parameter Tampering:* Access control check should be done to ensure that the authorized user is requesting the services. Before sending any data to database the check should be done whether the user is genuine by matching the user with existing information in the database. During render time the data must be stored in the session and while submitting the data should be checked with the data stored in the session. Customer validation on the input should be done on both client and server side. Client side validation never guaranteed a valid input will be passed to the system.

6. *Protect from Code Injection:* Code Injection could be prevented by following proper design model during web application developmental stage. All the parameters should be treated as data rather than executable code. Sanitization and validation should be used to implement the codes.

7. *Protect from Defacement:* Increase the web security by adopting security policies. Install a web application firewall. Scheduled security test have to be done on our web system. Encrypted data should be passed in the server. Proper validation of input output should be done. Prevent the system from SQL injection, XSS attack. File upload should be properly validating. HTTPS is a protocol used to provide security over the Internet. Test the system by using website security tools to ensure overall security.

IV. CONCLUSION

Today world is converting to digital world. Internet has become a battleground with the rapid growth in the use of web sites and applications. Day to day business are now relying on the internet to take information from customer. Financial organization, universities, Government organization everyone has started to favor online systems.

In this article some common web application threat and their preventive measure has been outline. By utilizing the preventive measure in developing the web application can reduce the chances of vulnerable attack on our website

VI REFERENCES

[1]. Xiaowei Li and Yuan Xue (2011), "A Survey on Web Application Security", Vanderbilt University

[2]. Web Application Security Statistics,http://project.webappsec.org/w/page/13246989/WebApplicationSecurityStatistics.

[3]. Shenam Chugh, Dr. Kamal Dhanda(2015)," Denial of Service Attacks", International Journal of Advanced Research in Computer Science and Software Engineering, ISSN: 2277 128X

[4]. W3SCHOOL,"SQL INJECTION", http://www.w3school.com/sql/sql_injection.asp

[5]. US-CERT,"Security Tip(ST04-015)",http://www.us-cert.gov/ncas/tips/ST04-015.

[6]. Detectify," What is Cross-site Scripting and how can you fix it?", https://blog.detectify.com/2015/12/16/what-is-cross-site-scripting-and-how-can-you-fix-it/

[7]. Amit Klein(2002)," Hacking Web Applications Using Cookie Poisoning",Sanctum

[8]. OWASP," Web Parameter Tampering", https://www.owasp.org/index.php/Web_Parameter_Tampering

[9]. HDiv, " Parameter Tampering (OWASP Top 10 - A4 Insecure Direct Object Reference) "https://hdivsecurity.com/owasp-insecure-direct-object-reference

[10]. Techopedia," Defacement", https://www.techopedia.com/definition/4870/defacement

[11]. Cyberpedia (2017)," WHAT IS A DENIAL OF SERVICE ATTACK (DoS)?", https://www.paloaltonetworks.com/cyberpedia/what-is-a-denial-of-service-attack-dos