Security Vulnerabilities of Websites and Challenges in Combating these Threats

Dhananjay*
Priya Khandelwal**
Kavita Srivastava***

Abstract

The public use of Internet started in 1990s. Since then, billions of websites have been developed. Also the technology has caused development of websites easier and less costly. It has enabled people to make their online presence quickly and easily through the use of websites. In recent years a number of Open Source CMS (Content Management Systems) have developed which enabled creation of websites in minutes. This large number of adoption of website by people also led to the growth of unskilled website administrators and developers. As a result almost 75% of websites are found to be infected with malware.

Google reported in March 2015 that around 17 million websites either have installed malicious software or trying to steal information. This number is increased to 50 million in March 2016. Google blocks nearly 20000 websites per week for malware and phishing. Most of these blocked websites are found to be implemented with WordPress, Joomla, and Magento.

This paper addresses various security vulnerabilities found in websites implemented with different technologies, methods of combating these vulnerabilities and research and development in this direction.

Keywords:

Introduction

Security is one of the critical phases of quality of any software or any application. Security testing of web applications attempts to figure out various vulnerabilities, attacks, threats, viruses etc related to the respective application. Security testing should attempt to consider as many as potential attacks as possible.

Increase in usage of web applications has opened the doors for hackers around the world for penetrating these applications. Hackers and attacker try to find out loop holes in coding of web applications to harm them in a number of ways such as applying Denial of Service (DoS) attack, spreading malware, illegal redirection to another website access or posting malicious content by gaining access into the application.

In order to prevent such attacks, the most effective method is to develop web applications by applying good and secure coding skills. Most of the web applications which suffer from security vulnerabilities have common coding problems such as improper input field validations, wrong or no session management, poor configuration settings in web applications as well as the web server which runs these applications.

We can organize the threats to web applications in a number of classes like Inadequate Authentication, Cross-Site Scripting, SQL Injection and so on. In the next sections all these web security vulnerability classes are elaborated.

Security Issues in Websites

In this section we discuss the classification of website security vulnerabilities.

Dhananjay*
BCA, IV, IITM
Priya Khandelwal**
BCA, IV, IITM
Kavita Srivastava***
Associate Professor, IITM

(1) Poor Access Grant and Lack of Sufficient Authorization

Authorization it is a process where a requester is allowed to perform an authorized action or to receive a service. Often a web application grants the access of some of its features to specified users only. The web application verifies the credentials of users trying to access these features through a Login page. This type of vulnerability exists in an application if users can access these features without verification through certain links or tabs and access other users' accounts also.

(2) Poorly Implemented Functionality

This kind of vulnerability exists in a website due to its own code which results in harmful consequences such as password leak, consuming large amount of resources and giving access to administrative features. The security breaches may lead to the disclosure of any confidential or sensitive data from any web application.

(3) Inadequate Exception and Error Handling Mechanisms

The error messages and exception handling code should return only limited amount of information which prevents an attacker to identify a place for SQL Injection. For Instance consider the following code.

…catch(Exception e) {Console.WriteLine(e.Message);}
If it is an SQL exception, this code cn display information related too database.

(4) Brute Force Attack

This is the process of trial and error in order to guess users' credentials such as user name, password, security questions for the purpose of hacking a user's account.

(5) Data/Information Leak

This kind of security breache may lead to the disclosure of any confidential or sensitive data from any web application. This vulnerability exists in web applications as a result of improper use of technology for developing application. It can cause revealing of developer's comments, source code, etc. It can give enough information to hacker for exploiting the system.

(6) Inadequate Authentication

Authentication this involves confirming the identity of an entity/person claiming that it is a trusted one. Sometimes a developer doesn't provide a link for administrative access. Yet administrative access is provided through another folder on the server. If a hacker identifies its path it becomes very easy to exploit the application.

(7) Spoofing

This is an attack where an attacker tries to masquerades another program or user by falsifying the content/data. Hacker injects malicious piece of code to replace the original content.

(8) Cross-Site Scripting

This type of attack is possible when a website containing input fields accepts scripts as well and leads to the phishing attack. The script gets stored in the database and executed every time the page is attacked. For example, <script>alert(message)</script>. Message could be a cookie also. When any user visits the page and application searches for username or password, the script will be executed.

(9) Denial of Service Attack

This kind of attack prevents normal users to access a website. The attacker attempts to access database server and performs SQL injections on it so that database becomes inaccessible. The attacker may also try to gain access as normal user with wrong password. After few attempts the user is locked out. The attacker may also gain access to web server and sends specially crafted requests so that web server is crashed.

(10) SQL Injection

It is an attack where any malicious script/code is inserted into an instance of SQL server/database for execution which eventually will try to fetch any database information.

(11) Poor Session Management

If an attacker can predict a unique value that identifies a particular user or session (session hijacking) he can use it to enter in the system as a genuine user. This problem also occurs when logout activity just redirects

the user to home page without termination of current session. The old session IDs can be used for authorization.

(12) Application Configuration Settings

Certain configuration settings exist in a web application by default such as debug settings, permissions, hardcoded user names, passwords and admin account information. An attacker may use this information to obtain unauthorized access.

(13) Cross site request forgery [6,7]:

It is a vulnerability which includes exploitation of a website by transmitting unauthorized commands from a user that a website trusts. Thus it exploits the trust of a website which it has on its user browser.

(14) Xml injection [1]:

It is an attack where an attacker tries to inject xml code with aim of modifying the xml structure thus violating the integrity of the application.

(15) Malicious file execution [3]:

Web applications are often vulnerable to malicious file execution and it usually occurs the code execution occurs from a non trusted source.

(16) Cookie cloning [11]:

Where an attacker after cloning the user/browser cookies tries to change the user files or data or may even harm the injected code.

(17) Xpath injection [3]:

It occurs when ever a website uses the information provided by the user so as to construct an xml query for xml data.

(18) Cookie sniffing [11]:

It is a session hijacking vulnerability with the aim of intercepting the unencrypted cookies from web applications.

(19) Cookie manipulation [5]:

Here an attacker tries to manipulate or change the content of the cookies and thus can cause any harm to the data or he may even change the data.

(20) Sidejacking [11]:

It is a hacking vulnerability where an attacker tries to capture all the cookies and may even get access to the user mailboxes etc.

(21) Social vulnerability (hacking), session hijacking [4, 5, 10, 11]:

It is a popular hijacking mechanism where an attacker gains unauthorized access to the information. xviii. Mis-configuration [24]: in appropriate or inadequate configuration of the web application may even lead to the security breaches.

(22) Absence of secure network infrastructure [9]:

Absence of any intrusion detection or protection system or failover systems etc may even lead to violation of the security breaches.

(23) Off the shelf components [9, 11]:

These components are purchased from third party vendors so there occurs a suspicion about their security aspect.

(24) Firewall intrusion detection system [8, 9,10]:

A firewall builds a secured wall between the outside/ external network and the internal network which is kept to be trusted.

(25) Path traversal [3]:

It is a vulnerability where malicious untrusted input causes non desirable changes to the path.

(26) Command injection [3]:

It is the injection of any input value which is usually embedded into the command to be executed.

(27) Parameter manipulation [5]:

It is similar to XSS where an invader inserts malicious code/script into the web application.

(28) LDAP injection [3]:

It is similar to SQL and Xpath injection where queries are being targeted to LDAP server.

(29) Bad code or fault in implementation [2]:

Iimproper coding or fault in the implementation of the web application may even lead to the violation of the security of the web application.

(30) Clickjacking [6]:

It is an attack where a user's click may be hijacked so that the user would be directed to some other link which may contain some malicious code.

(31) Content injection [8, 6]:

It is a vulnerability where an attacker loads some static content that may be some false content into the web page.

(32) File injection [8]:

It refers to the inclusion of any unintended file and is a typical vulnerability often found in web applications. Example: remote file inclusion.

Challenges faced by security testing of web applications

One of the concerns of security testing of web applications is the development of automated tools for testing the security of web applications [3]. Increase in the usage of Rich Internet Applications (RIAs) also poses a challenge for security testing of web application. This is due to the fact that the crawling techniques which are used for exploration of the web applications used for earlier web applications do not fulfil the requirements for RIAs [3]. RIAs being more users friendly and responsive due to the usage of AJAX technologies. Another challenge could be the usage of unintended invalid inputs which may result in security attacks [1]. And these security breaches may lead to extensive damage to the integrity of the data. While working the mutants, one should be sincere enough to incorporate them as injecting && (and) instead of || (or) or any such other modification may lead to fault injection which could result in a security vulnerability as vulnerabilities do not take semantics into consideration [1]. This may even pose a challenge to the security testing of any such web application. Usage of insecure cryptographic storage may even pose a challenge to the web application security testing [1]. Security testing of web applications may face repudiation attacks where any receiver is not able to prove that the data received came from a specific sender or from any other unintended source [1]. Also the web development languages which we use may lack in enforcing the security policy which may even violate the integrity and confidentiality of the web application [11]. This may even pose a security risk. At times it is also possible that an invader is able to launder more information than intended, in such a case again this may lead to the set back to the integrity of the data which could be another challenge for a security tester.

Conclusion

In this paper we have describes various kinds of security vulnerabilities that may exist in a website if proper consideration is not taken during development. A website developer must employ all possible measures to combat any known threats during the whole development cycle of a website from its design, implementation to testing. If any security loop hole remains undetected hackers can use it for exploiting the system.

References

1. An Approach Dedicated for Web Service Security Testing, S´ebastienSalva, Patrice Laurencot and IssamRabhi. 2010 Fifth International Conference on Software Engineering Advances.
2. Security Testing of Web Applications: a Search Based Approach for Cross-Site Scripting Vulnerabilities, Andrea Avancini, Mariano Ceccato , 2011- 11th IEEE International Working Conference on Source Code Analysis and Manipulation.
3. SUPPORTING SECURITY TESTERS IN DISCOVERING INJECTION FLAWS. Sven T¨urpe, Andreas Poller, Jan Trukenm¨uller, J¨urgenRepp and Christian Bornmann, Fraunhofer-Institute for Secure Information Technology SIT, Rheinstrasse 75,64295 Darmstadt, Germany, 2008 IEEE,Testing: Academic & Industrial Conference - Practice and Research Techniques.
4. A Database Security Testing Scheme of Web Application, Yang Haixia ,Business College of Shanxi University, Nan Zhihong, Scholl of Information Management,Shanxi University of Finance &Economics,china. Proceedings of 2009 4th International Conference on Computer Science & Education.

5. Mapping software faults with web security vulnerabilities. Jose Fonseca and Marco Vieira. International conference on Dependable Systems &Networks : Anchorage, Alaska,june 2008 IEEE.

6. D-WAV: A Web Application Vulnerabilities Detection Tool Using Characteristics of Web Forms. Lijiu Zhang, Qing Gu, Shushen Peng, Xiang Chen, Haigang Zhao, Daoxu Chen State Key Laboratory of Novel Software Technology, Department of Computer Science and Technology, Nanjing University. 2010 Fifth International Conference on Software Engineering Advances.

7. Enhancing web page security with security style sheets Terri Oda and Anil Somayaji (2011) IEEE.

8. Security Testing of Web Applications: a Search Based Approach for Cross-Site Scripting Vulnerabilities, Andrea Avancini, Mariano Ceccato , 2011- 11th IEEE International Working Conference on Source Code Analysis and Manipulation.

9. Assessing and Comparing Security of Web Servers. Naaliel Mendes, AfonsoAraújoNeto, JoãoDurães, Marco Vieira, and Henrique Madeira CISUC, University of Coimbra. 2008 14th IEEE Pacific Rim International Symposium on Dependable Computing.

10. Firewall Security: Policies, Testing and Performance Evaluation. Michael R. Lyu and Lorrien K. Y. Lau. Department of computer science and engineering. The Chinese University of Hong kong, Shatin, HK. 2000 IEEE.

11. Top 10 Free Web-Mail Security Test Using Session Hijacking Preecha Noiumkar, Thawatchai Chomsiri, Mahasarakham University, Mahasarakham, Thailand. Third 2008 International Conference on Convergence and Hybrid Information Technology. Development of Security Engineering Curricula at US Universities.Mary Lynn Garcia, Sandia National Laboratories.1998 IEEE.