Security For Three-Tiered Web Application

Wisal H. Abdulsalam
Dept. of Computer Science/College of Education for Pure Science (Ibn Al-Haitham)/ University of Baghdad

Received in: ١٦ December 2014, Accepted in: 14 March 2015

Abstract

Web application protection lies on two levels: the first is the responsibility of the server management, and the second is the responsibility of the programmer of the site (this is the scope of the research).

This research suggests developing a secure web application site based on three-tier architecture (client, server, and database). The security of this system described as follows: using multilevel access by authorization, which means allowing access to pages depending on authorized level; password encrypted using Message Digest Five (MD5) and salt. Secure Socket Layer (SSL) protocol authentication used. Writing PHP code according to set of rules to hide source code to ensure that it cannot be stolen, verification of input before it is sent to database, and update scripts periodically to close gaps in the site. Using 2Checkout company (2CO), which is trusted international electronic money transfer to allow customers pay money in a secure manner.

Keywords: MD5, PHP, SSL, three-tiered, Web application, 2CO.

1. Introduction

Internet users use web applications for a wide spectrum of tasks [1]. When the web application consists of a combination of web technology and database then it is called web-database application, this combination allows users to search and browse some database contents on the web. Many web applications' database consist of three layers: the backend layer is a database server, which contains a database, and DBMS, in the frontend layer is the client browser, and at the middle layer most of the applications, usually developed by a web server-side scripting language such as PHP [2].

2. Aim of the Research

To develop a secure web application site based on three-tier architecture.

3. Security Mechanisms

Several mechanisms are used for security requirements. Encryption is one of the primary mechanisms used, which is a process of converting information to the incomprehensible blades to prevent non-licensed access to information to anyone except by those who have decryption key, which consists of a collection of letters, numbers and symbols. The newer types include also graphics. Encryption methods exist use one key or two keys or combine the two previously [3].

Hyper Text Transfer Protocol (HTTP) delivered webpages without encryption or protection of data transmitted between web browsers and web servers. To eliminate eavesdropping several technologies are developed to protect sensitive data in transit. Secure Socket Layer (SSL) is the protocol that became standard for securing web communications [4].

HTTPS is the web protocol that uses SSL to encrypt HTTP and it uses worldwide for securing web communications [5].

4. SSL Protocol

To secure client-server communications over the internet Netscape designed SSL. Two types of cryptography are used within its session: symmetric and asymmetric. Its main capability is to encrypt messages, for example, any time client order from most large vendors such as Amazon.com merchandise, SSL will automatically encrypt the order in client computer browser before sent over the internet. It is a completely new layer of protocol, which works below high-level application protocols and above the Transmission Control Protocol (TCP) [٣, ٤].

5. System Architecture

The system built a web application site for selling e-products using PHP and MySQL constructed from three tiers as shown in Figure No. (1), these tires are:

- Client tier: user interface or frontend, which runs on the user's computer.
- Application server tier: running PHP that handles all application operations runs on the application server. It is responsible for accessing the database tier to make different functions such as add, retrieve, modify, delete data, and send the results to the devices in the client tier. It resides in the middle tier.
- Database tier: it represents the backend running in MySQL, which stores the data for web application site required by the middle tier. It stores information about the different types of users' accounts, categories of e-products, e-product, shopping cart, orders, and buying history for authenticated users. Only the administrator has a direct access to this tier.

The site appears to the client containing four levels of accessing content. These levels are:

1- Visitor: allows user to browse the offers of e-products, access to the shopping cart, uses the search, gets support, and registering in the system.
2- Customer: This role is extended from the previous to possibility to access the special page, buy products, retrieve/change password, browse history of shopping, modify profile, and add money in account for buying items. When customer decides to add money, then he/she will be redirected to 2 Checkout (2CO) company to add it in a secure way.
3- Employee: the employee must first login through individual Uniform Resource Locator (URL) address to access the special page. This role entitles:
    - Supervisor: Responsible for follow-up staff, check the progress of work assigned to them, discuss with them to achieve the desired goals, and sending a notice to the manager for the course of things.
    - Technical employee: This has the ability to solve problems such as error messages and payment problems.
    - Support officer: This has the ability to interact effectively with unauthorized and customers' inquiries.
4- Manager: access to manager main page is possible through individual URL address and access to it requires first logging in.

This role entitles managing categories, products, orders, customers, employees, emails, and statistics.

The responsibilities of server tier are to authorize users. SSL authentication is used to authenticate client web browser to application server and that server authenticates itself to the client and to the database server and database server authenticate itself to application server. The manager, employee, and customer need to enter email address and password, before they can enter to their special pages. The system takes user's supplied email address and performs a database checkup to determine if this account exists. If it is, it needs to check the user password. It runs MD5which is a one-way hash function, produces 128-bit output from an input of arbitrary length and salt over this password to generate an encrypted password, and compares it to the returned one from the database. If the two are equal, then the user supplied corrects password and it can log him in. When logins correctly, a welcome message at the top of the special page appears with the date and time of the last end login. This is important because it may give a chance to discover whether someone else accesses account or not.

6. Proposed System Security

The security of this system can be describe as follows:

1- User authorization: it is used to determine the level of the user (visitor, customer, employee or manager). In addition, the direct access to database tier is allowed only to the administrator of the web application site.
2- Using SHA-1 or MD5 alone as algorithm to encrypted passwords produces hash value with a fixed value that stored in database, which means if another user entered same password then the hash value will be the same, so it can cracked. These algorithms have proven unsafe and should not be used alone.

   The solution for this problem in this proposed system is by using salt with MD5for encryption passwords transfer from client to server and then stored in MySQL database to protect them from stolen as shown below:

```
<?php
function createsecurepass($password){
$encryptpassword = md5($password);
$salt = 'wisal80';
return md5($encryptpassword.$salt);
}
?>
```

3- The code is written according to set of rules to hide it, and to ensure that it cannot be stolen as described below:

- Hide mistakes messages from users so as not to be able to figure out a way of writing code by turning off the code in the php.ini file: display_errors, and instead of using log file writing these messages to my own file to ensure that no one can see it.
- Protecting the code from the injections of SQL. This is done by calling a function to make data safe for use in a query: mysql_ real_escape_string. Because it is used to filter data passed to an SQL statement as shown in the example below:

```
<? php
$db = mysql_connect ('localhost', 'wisaladmin', '226308910kwh');
mysql_select_db('e_products', $db);
$productCost =
mysql_real_escape_string ($_POST ['e_product_cost'], $db);
$Result = mysql_query ("INSERT INTO e_product (e_product_cost)
VALUES ('$productCost'' ;}
?>
```

- Verification of input before sent to web application database. Fields required to fill by the visitor to register(for example) are checked using JavaScript to ensure that all required fields are completed, but this use with HTML code open the possibility to inject Cross-Site Scripting (XSS) ,to close; the code written as the example shown below:

```
$attributive = htmlentities($_POST[' Attributive']);
$sql = "INSERT INTO e-product (e_product_attributive) VALUES
('$attributive ')";
```

- To prevent people from seeing things they are not supposed to see, which means prevent them from direct access to the files of the root, it is important to make the index file looks like below:

```
<?php
require_once '../librarydocument/database.php';
require_once 'functions.php';
?>
```

Which means calling for implementation files inside another file without using directly, and taking into account that the folder (librarydocument) can be accessed to it via a user name and a password.

- Update scripts periodically to close gaps in the site such as changing session identification number by using a built-in functions called session regenerate id(), and sets cookie_secure = 1 in php.ini configuration file.

4- SSL authentication is used to authenticate all parties: Clients' web browsers to application server and that server authenticates itself to the client and to the database server and database server authenticates itself to application server.
5- Using 2CO: which is a trusted international electronic money transfer company. It can be an essential assuring step. The main reason for choosing it in addition to using SSL certificates is the variety of payment methods it provides and it can deliver money to many countries including Iraq, while other famous sites like PayPal do not include Iraq within their country list.
6- We testing the web application site to verify its security online and find the possibility of attacking by entering the proposed web application URL address www.ihjpas.com through Netsparker, which is quick start scanning procedure testing application in seconds and it is very easy to use. The URL address of it is **https://www.netsparker.com/netsparker/**. The reason of choosing it is because it gives a demo ration free test through the following link http://php.testsparker.com/ provides different security check group such as reflected XSS vulnerabilities, SQL injection, and test for access vulnerabilities. When a test is complete, it displays the solutions besides the possible issues. After using it, the result appeared no XSS vulnerabilities, the proposed web application site is good, and there is no warning from attacking by SQL injection.

7. Conclusions

1- The three-tier architecture of the proposed system plays the basic role of database security because the client does not have a direct access to the database server connect to it across the middle application server.
2- Enhance maintainability for example it's easy to modify database tier without affecting other tires.
3- The code behind page provided the greater security support for web application. To ensure that the PHP code cannot be stolen, the source code is protected and hidden.
4- Using a trusted and affordable international electronic money transfer company to complete the payment transaction in a secure way is an essential step to assuring parties, the customers and the seller. In this research, 2CO company was chosen.

References

1- David, L. and Hugh, E., (2004), Web Database application with PHP and MySQL, 2nd edition, ISBN-10: 0596005431, O'Reilly Media Inc..
2- Stefan J.; Ilia P.; Chrisitian M. and Udo M., (2004), Guide to web application and platform architecture, 1st edition, ISBN: 3642056687, Springer, Softcover Rprint of Hardcover.
3-Efraim T., (2003), Introduction to information technology, 2nd edition, John Wiley and Sons Inc.
4- Joseph, S.; Tim, S. and Simon, J., (2005), Overview of SSL VPN: understanding, evaluating and planning secure web-based remote access to private networks, 1st edition, Packt Publishing.
5- http://www.ehow.com/how_7664027_tls-ssl-tutorial.html for how SSL work information, (2011).

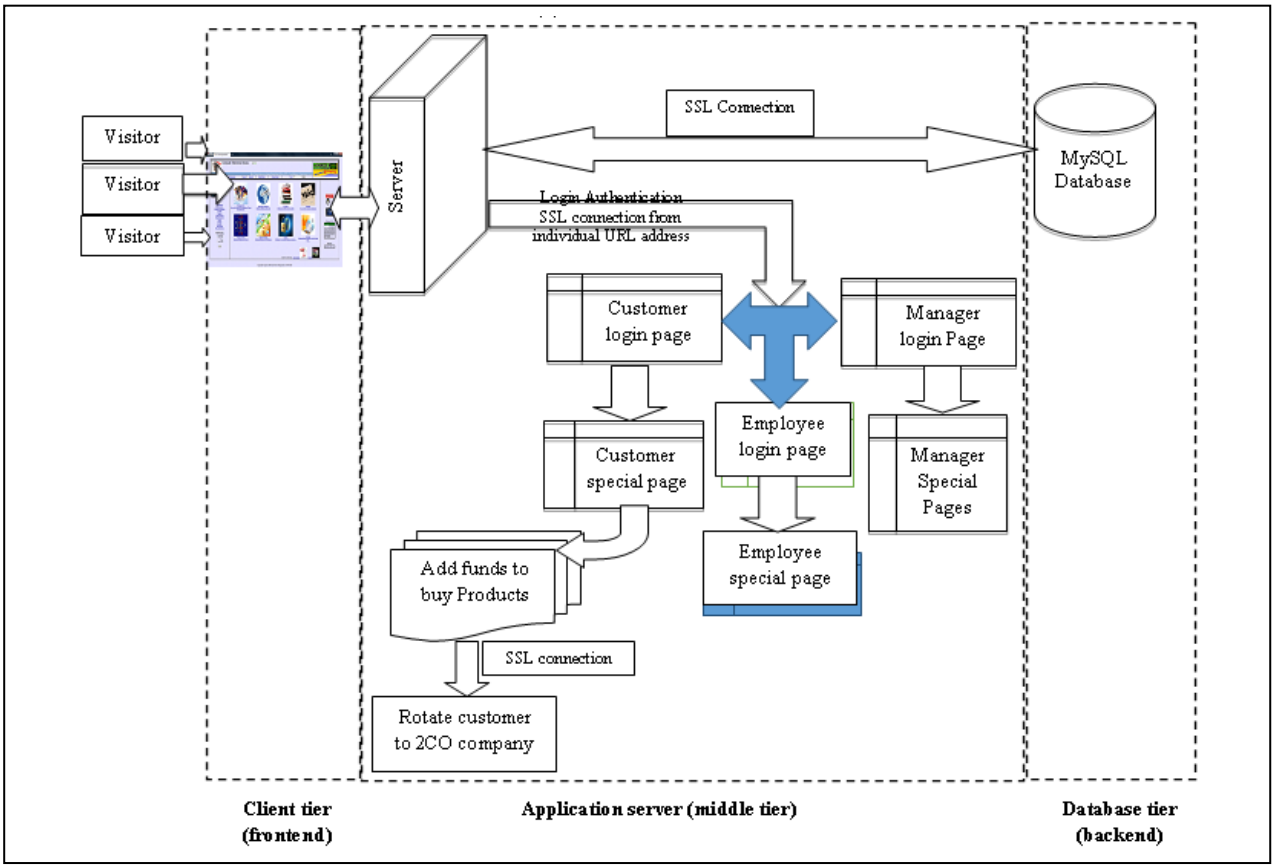

Figure No. (1): System structure

الأمن لتطبيق الويب من ثلاثة مستويات

وصال هاشم عبد السلام

قسم علوم الحاسبات/كلية التربية للعلوم الصرفة(ابن الهيثم)/جامعة بغداد

استلم البحث في:14 كانون الاول 2014،قبل البحث في : ١٤ نيسان 2015

الخلاصة

حماية تطبيقات الويب تقع على مستويين: الأول من مسؤولية إدارة الخادم، والثاني من مسؤولية مبرمج الموقع (وهذا هو نطاق البحث) .

هذا البحث يقترح تطوير موقع تطبيق ويب آمن على أساس الهندسة المعمارية من ثلاث طبقات (العميل، والخادم، وقاعدة البيانات). وصف أمن هذا النظام على النحو الآتي: استعمال صلاحية الوصول متعدد المستويات، وهو ما يعني السماح بالوصول إلى الصفحات اعتمادا على مستوى الصلاحية، كلمة المرور مشفرة باستعمال (MD5) والملح. بروتوكول طبقة المقابس الآمنة (SSL) تم استعماله للمصادقة. كتابة التعليمات البرمجية PHP وفقا لمجموعة من القواعد لإخفاء الكود المصدري للتأكد من أنه لا يمكن سرقته والتحقق من المدخلات قبل إرسالها إلى قاعدة البيانات، وتحديث البرامج النصية بشكل دوري لإغلاق الثغرات في الموقع. استعمال شركة (2CO)، الدولية الموثوقة لتحويل الأموال الكترونيا للسماح للعملاء بدفع المال بطريقة آمنة.

الكلمات المفتاحية: MD5، PHP، SSL، ثلاث مستويات، تطبيق ويب، 2CO.