SECURED WEBSITE

Audu Jonathan Adoga[1] and Dennis Dogara Yakubu[2]

[1]Department of Computer Science, Nasarawa State Polytechnic Lafia, Nasarawa State- Nigeria.
[2]Department of Science Laboratory Technology, Nasarawa State Polytechnic Lafia, Nasarawa State- Nigeria.

ABSTRACTS: *Since the invention of the internet by Tim Berners-Lee, web development has continued to witness monumental and revolutionary changes. The changes are either in form of increase in functionalities or introduction of versions of languages and software that are less prone to attacks. But, as more functionalities and new versions are being introduced, they come with unforeseen flaws that are later discovered by the inventors themselves or professionals in the field. These problems bring about the need for a model to secure system that individuals and organization could learn from. This study used a research methodology called design and creation research. The work started by trying to study how HTML, CSS, JavaScript, PHP and SQL could work together to produce a dynamic web application, starting from their histories. The study also expatiates on so many types of PHP vulnerabilities that exist on the internet and possible ways of preventing them through recommendations from relevant professional bodies such as OWASP, PHP The Right Way, and CVE. A model secure system was developed with the recommended security features.*

KEYWORDS: Secured Website, Web App, Web Vulnerability, PHP, HTML, CSS, SQL

INTRODUCTION

Subject of the Research

This research work was conducted on the topic "Secured Website”. A secured system is a system that is devoid of flaws, security holes, while vulnerability is defined as a flaw that makes a software system to work contrary to the purpose that it was designed for and could be exploited by fraudulent users (OIS, 2004).

This research uses HTML5, CSS3, JavaScript, PHP and MySQL, a perfect blend for the development of a secured system. This system claims to be secured as at the time of development.

PHP is the most common server-side scripting language and over 70% of web servers deploy this but the challenge is this: other competitors such as ASPX or JSP developers enjoy in-built security programs in such languages but, PHP is left with a lot of laws (OWASP, 2013). This problem triggers a research in this direction.

Aims and Objectives of the Research

The aims or this research work are:

1. To study and understand languages such as Hypertext Markup Language 5 (HTML5), Cascading Style Sheet 3 (CSS3), JavaScript, PHP and MySQL.
2. To learn the possibility of making the languages in (1) above to work together for the purpose of producing a reliable and dependable site.
3. To learn about vulnerabilities that is related to PHP or PHP site. Though it may or may not be purely PHP as we know that PHP without other compatible languages may not give desired result especially at the front end.
4. To learn about the right PHP coding method_ security conscious coding method, and apply the knowledge gained to this research.
5. To develop a site that greatly depends on server-side validation input. All aspects of validation will be server-side except if it becomes very necessary to include client-side validation.

Research Motivation

This research is motivated by the fact that, over the years, software vulnerabilities have increased at an alarming rate. (CERT. 2009) said that, 1090 Vulnerabilities were discovered in the year 2000 but after few years (2006), 8,064 new flaws were discovered, (CERT. 2000) further explain that from 1995 to 2008, 44,074 flaws were reported.

Software vulnerabilities have led to series of tragedies: account is constantly being attacked due to broken authentication and session management, privileges are being taken over from the real owners, forgery of victims' identity and redirection of users to wrong sites is on the increase (OWASP. 2013).

The Significance of the Study

The core significance of this study is to create awareness in the aspect of security conscious coding to especially the beginning programmers who in most cases are curious to write codes and make them run without considering the security implications of such software.

This awareness will be created through exemplary coding that shows acceptable practices. "PHP is a terrific language for the rapid development of dynamic websites" (Dickson, 2005). It is also a user-friendly scripting language since

a data such as 124 could be used as an integer and also as a string. Another good aspect of PHP is that, one must not declare variables but the user friendliness of the language also comes with possibilities of making the beginning programmers to write codes that have so many security flaws (Dickson. 2005),

THE CONTEXT OF THIS RESEARCH

Some Research Work on PHP Security Flaws

Several research works have been conducted on PHP security: a research was also conducted to detect vulnerabilities in web applications using static analysis tool called Pixy. The tool was able to discover 15 unknown PHP flaws from three web applications but they were all cross-side scripting vulnerabilities (Javanovic, Kruegel and Kirda, 2006).

In another research conducted by (Vieira, Antunes and Madeira, 2009), 300 web services were evaluated for flaws using four recognized scanner security holes were discovered but these tools also had limitations. There were differences in the flaws detected and furthermore, the number of false-positive was high-40% and 35% in two situations, some scanners were also observed to have low coverage-2 scanners had less than 20%,

Moreover, (Yoshioka, Washizaki and Maruyama, 2008) conducted a security patterns survey, the study sheds light on the use of pattern at every stage of software development but failed to produce a working model that could be used as reference point during development of a secured site.

Why is this Research Being Conducted?

Most of the research work on PHP security may not be very reliable as they are either survey kind of research or a tool(s) that detect one or few vulnerabilities and sometimes with high false positive. These research works have failed to provide a reliable working model and in some cases such research only use tools to detect flaws. Most research work failed to provide preventive e approach to security threat. This is why the research focuses on preventive measure by using professionally acceptable coding style.

LITERATURE REVIEW

Hypertext Mark-Up Language and the Web

According to Longman (1998), Hypertext Mark-Up Language (HTML), the Language of the web that almost everybody uses today either directly or indirectly, is the result of the work of one person. Tim Berners-Lee wrote the prototype in 1992. He continued the work after 1992 and invited the web in 1989 with HTML as its language. The World Wide Web started at the European Laboratory for Particle Physics (CERN), a place that no one could think that such invention could be possible.

Moreover, before this invention, there was a similar invention by an Apple Computer programmer (Bill Atkinson) in the late 1980s called HyperCard which drew the attention of many but had limitation. The hypertext links could only jump from one document to another on the same computer. The links would not work globally so, it became a matter of concern.

Tim Berners-Lcc's HTML was Based on SGML (Standard Generalized Mark –Up Language).

Berners-Lees invention of HTML was a good idea to follow because it was based on SGML an internationally accepted standard for making text into various formats such as paragraphs heading and title. Hypertext links were also compatible with most of the systems such as Macintoshes personal computers and UNIX system that were connected to the Internet at that time. Another advantage was being browser independent. A System of protocol called hypertext Transfer Protocol was also developed by him for easy retrieval of hypertext documents through hypertext links.

HTML+ and Mosaic

The popularity of this invention continued to increase and by 1992. a browser called mosaic was developed at the University of Illinois in a research centre called National Centre [or Supercomputer Applications (NCSA) and by 1993 in the month of April. the first version was released to Sun Microsystems' workstation.

The Image Tag

There was a debate in a www-talk group the insertion of images in HTML documents. Notable amongst the members were Dan Connolly. Tim Berners-Lee. Dave Ragget and Marc Andreessen. The image tag by Mosaic team was presented by Marc Andreessen and it was implemented in HTML, but with the coming of HTML4, the OBJECT tag has the qualities of replacing the IMG tag. Today, the web has grown to the level of using HTML 5.

Hypertext processor (PHP)

PHP 2.0

According to (PHP, 2013), PHP was originally called PHP/FI Rasmus Lerdorf created it in 1994. Lerdorf, used C programming language to develop binaries of Common Gateway Interface. The set of binaries was used by Lerdorf to track visitation to his internet resume. He gave the suit of script name that is commonly called PHP Tools.

In 1996, there was a complete rewrite of the whole program that led to an evolution of a readopted name PHP/FI. This was more of a programming language than a mere set of tools that was previously used because; there were built in capabilities [or databases, cookies and functions that were defined by users. By 1997 PHP/FI has moved out of beta stage and continued to enjoy great popularity with PHP. 2.0 version but was full of limitations due to its one-man major developer nature and minor contributors. **PHP** development has undergone series of changes up to **PH**P 5.0, one of the most stable versions.

MySQL

MySQL and the MySQL Server

Based on a historical account by MySQL (2013), MySQL is the most popular among the open source Database Management System (DBMS) All the benefits derived from MySQL is as a result of philanthropic work of Oracle Corporation. Anything from simple database to a complex one could be done through MySQL. For one to add data or delete data or process data or restructure data in an acceptance format, the importance or a relational database management system such a cannot be over emphasized.

The SQL part of the MySQL means "Structured Query, Language" and the responsibility of defining the language is saddled on ANSI/ISO standard that in existence since 1986.

In addition, one could download and use the MySQL softy free. The modification of the code by programmers is also allowed. This is why it is called open source software. MySQL server is very reliable and fast software that could be adjusted to take the advantage of the computer’s CPU to increase its speed. It can work effectively in a network environment and in a highly demanding condition. MySQL software is client server software that is compatible with a wide range of application at both back and front end. It is very compatible with so many applications programming interfaces (APls). MySQL is pronounced “My Ess Que Ell”, officially. Though, it is allowed to be pronounced in some other ways

Web Application Attacks

Some Web Vulnerabilities that have occurred

According to CVE (2013), ext/xml/xrnl.c that existed in PHP versions lower than 5.3.27 allowed an attacker to remotely cause denial of service (DoS) or other effects that were not crystal clear. Denial of service is a situation where one tries to make a machine or resources that belong to a network inaccessible to users. In this case, the target device is so much saturated with irrelevant requests that it cannot respond to genuine ones. The targets are usually the gateways for card payments and servers that are meant for banks. DoS is a gross violation of the internet policies and illegal in the constitutions of the nations. CVE (2013) reported that cross-site vulnerabilities were discovered in MiniBB he version that came before 3.0,1. The file bb_admin.php was very vulnerable because an arbitrary web script could be injected by fraudulent users. This exploit does not need any form of authentication therefore, making this a very subtle attack.

Ragan (2012) said that, PHP vulnerability was by accident made known to the public. The flaws were a code execution flaw. This caused fears due to possibility of attacks on vulnerable websites on a large scale by the attackers. The flaw was said to have been in existence since 2004 but could not be discovered until recently. While the group that first created awareness or this flaw (Eindbazcn) were waiting for a patch before the release of the bug, details of the bug were made known to Reddit. This led to the disclosure of what Eindbazen discovered (Ragan. 2012). As a result of this, the development group of PHP encouraged users to upgrade to PHP 5.4.3 as a remedy.

When inputs are not properly sanitized and validated, an attacker could gain access to confidential parts of a database. This could later result to greater exposure or users' data and could result to serious problems if not properly checked. The case of "MDAC RDS vulnerability" that affected IIS 4.0 in the month of July, 1999 is a good example (Security TechCentcr. 1999). The MDAC (Microsoft Data Access Component) of the server was vulnerable because it could allow illegal user to gain access to a site hosted by the server.

PHP: The Right Ways

OWASP (2013) in her security tip titled "PHP Security Cheat Sheet" gives vital information concerning PEP security. With PHP being the most popular open source scripting language. There is great need to pay attention to the right developing PHP applications. The core PHP is considerably safe but other components such as plugins and libraries are not secure, hence the need for proper coding.

Injection

OWASP (2013) in her top ten project said that injection is a situation where system, especially at the user side send data that cannot be trusted to an interpreter vulnerability in form of injection are from queries, commands from operating systems and XML, parsers. Security 'holes related to injections are not easy to discover through testing but by close examination of the code. Security holes of this nature are exploited by the attackers through the use of fuzzers and scanners.

Storing Password

Storing passwords as clear text could be very dangerous, this is because, confidentiality of your data may be compromised in a situation where a hacker is able to gain access to your site. To take care of this problem, so many password hashing algorithms exist that could be used to prevent unwanted users from gaining access to the database. This can be achieved through the use of functions such as MD5, sha256, Crypt and Berypt functions.

SUMMARY AND CONCLUSION

This research work "Development of a Secure Online Admission System" was successfully executed. This could be seen clearly by relating the aims of the research work with the result of the test plan that was carried out.

Nobody can claim to have been able stop crime completely in a society but what every society does is to reduce it to the minimum. A research in computer security is not an easy one at all considering the numerous areas of exploit that an attacker may try to penetrate a particular system. Security of a system is a daily routine. When a security hole is discovered in an application, a patch may be immediate or it may delay so, developers and other similar professionals must always be alert in other to cope with the security challenges that exist on the internet.

Moreover, professional site like that of OWASP, PHP The Right Way and CVE ought to be visited always considering the fact that web development is very dynamic. These are professional security organizations with members all over the world that are working tirelessly and have contributed greatly to PHP security. PHP security is being approach by these professionals in the aspect of creating consciousness amongst members on what the attackers do and good coding methods that could prevent such attacks. They also create awareness to the entire public on system security issues.

Considering the difficulties of studying so many software development languages before this model site was developed, this research agitates for an all-in-one language that could be used to take care of both the front end and the back-end development. This will make software development easier. Moreover, there are deprecations in all these languages again for an update. This research thinks, it is more tedious than studying the proposed all-in-one language for an update.

REFERENCES

CERT, (2009). CERT Statistics (Historical). Retrieved June 21,2013, from http://www.cert.org/stats/cert stats.html/

CVE, (2013). Common Vulnerabilities and Exposures. retrieved July 11, 2013, form http://cve.mitre.org/cgi-bin/covename.cgi?name=cve2013-4113

CVE, (2013). CVE-2013-Retrieved July 12, 2013, from http://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2013-5020

CVE, (2013). CVE-Retrieved July 12, 2013, from http://cve.mitre.org/cgi-bin/cvename.cgi?name=cve-2013-1862

Dickson, P., (2005). Top Seven PHP Security Blunders: Site point. Retrieved June 9, 2013, from http://www.sitepoint.com/php-security-blunders/

Jovanovic, N., Kruegel, C., & Kirda, E., (2006). Pixzy: A Static Analysis Tool for Detecting Web Application Vulnerabilities. Security and Privacy. 2006 IEEE Symposium 6pp-263 Berkeley/Oakland, CA. Retrieved June 8, 2013,from http://ieeexplore.ieee.org/ xpl/login.jsp?tp=&arnumber=1624016&url=http%3A%2F%2Fieeexplore.ieeexplore.ieee.org%2fxpls%Fabs-all.jsp%3Farnumber%3D1624016

Lockhart, J. & Jordan, K., and Sturgeon, P., (2013). PHP The Right Way. Retrieved June 29, 2013, form http://www.phptherightway.com/

Lockhart, J., (2013). PHP The Rureight. Retrieved July 5, 2013, from http://www.phptherightway.com

Longman, A.W. (1998). A History of HTML. Retrieved August 25, 2009, from http://www.w3.org/people/Ragget/book4/cho2.html

MySQL, (2013). History of MYSQL. Retrieved August 5, 2013, from http://dev.mysql.com/doc/refman/5.0/en/history.html

MySQL, (2013). MYSQL Development History. Retrieved August 5, 2013, from http://dev.mysql.com/doc/refman/5.0/en/development-history.html

MySQL, (2013). The Main Features of MYSQL. Retrieved August 5, 2013 from http://dev.mysql.com/doc/refman/5,0/en/features.html

MySQL, (2013). Upgrading MySQL. Retrieved August 5, 2013, form http://dev.mysql.com/doc/refman/5.0/en/upgrading.html

MySQL, (2013). What is MySQL? . Retrieved August 5, 2013 from http://dev.mysql.com/doc/refman/5.0/en/what-is-mysql.html

MySQL, (2013). What is New in MYSQL 5.0 Retrieved August 5, 2013, from http://dev.mysql.com/doc/refman/5.0/en/mysql-nutshell.nutshell.html

MySQL, (2013). What is New in MYSQL 5.1 Retrieved August 5, 2013, from http://dev.mysql.com/doc/refman/5.1/mysql-nutshell.nutshell.html

MySQL, (2013). What is New in MYSQL 5.5 Retrieved August 5, 2013, from http://dev.mysql.com/doc/refman/5.5/en/mysql-nutshell.nutshell.html

MySQL, (2013). What is New in MYSQL 5.6 Retrieved August 5, 2013, from http://dev.mysql.com/doc/refman/5.6/en/mysql-nutshell.nutshell.html

Nixon, R., (2012). PHP, MYSQL, JAVAScript & CSS. Sebastopol: O'Reilly, Inc.

Oates, B.J., (2006). Research Information Systems and Computing. London: SAGE Publications

OIS, (2004). Guidelines for Security Vulnerability and Response: Organization for OWASP, (2013). PHP Security Cheat Sheet. Retrieved July 12, 2013 from https://www.owasp.org/index.php/php-Security_Cheat_Sheet

PHP, (2013). History of PHP. Retrieved August 5, 2013 form http://php.net/manual/en/history.php

Ragan, S., (2012). Official Fix for PHP Flaw Easily Bypassed, Researchers say, Security Bulletin ms99-019. Retrieved June 23, 2013 from http://www.securityweek.com/official-fix-php-flaw-easily-bypassed-reseachers-say

Security TechCenter, (1999). Patch Available for "malformed HTR request" vulnerability: Microsoft Security Bulletin ms99-019. Retrieved June 23, 2013 from http://technet.microsoft.com/en-us/security/bulletin/ms99-019

Yoshioka, N., Washizaki, H., and Maruyama, K., (2008). A Survey on Security Patters: Progress in Informatics, No.5, pp35-47, (2008) PDF file special issue: The Future of Software Engineering for Security and Privacy. Retrieved June 23, 2013 form http://scholar.google.com/scholar?start=20&q=php+security+flaws&hl=en&as-sdt=0.5