UNCOVER SECURITY LEAKS IN PHP WEBSITE

Abrar A. Alsulaiman  
King Saud University  
Email: abr.alsulaiman@gmail.com

Eyas El-Qawasmeh  
King Saud University  
Email: eyasa@usa.net

Abstract – This paper re-visits the security issues that are related to websites. In particular, it tries to highlights the possible attacks for website from inside. The word inside means that we are doing investigation of some leaks or bugs in the developments or some seeded programs that violates the principles of security. A program was written for this purpose and we run it on sdiwc.net/emailer website. Many studies that show more than one fourth of PHP website has security problems. The result of the study shows that there must be awareness to the decision makes for the dangers that their website face. Our developed tool and our suggested solution will contribute to this issue. The authors suggest implementing this tool in PHP website as a security lock.

Keyword – PHP, websites, threats, backdoor, security.

I. INTRODUCTION

The amount of websites has increased rapidly during the last few years. While websites consisted mostly of static HTML files in the last decade, more and more web applications with dynamic contents appear as a result of easy to learn scripting languages such as PHP and other new technologies [1]. Currently, all website are developed either using ASP.NET or PHP. In fact, PHP is the most popular language for open source web applications and more secure than ASP [2]. Actually, 30% of all vulnerabilities found in computer software that PHP was related [3]. "Vulnerability is a hole or a weakness in the application, which can be a design flaw or an implementation bug that allows hacker to cause harm to an application" [4]. Unfortunately, more than 600 new kinds of vulnerabilities have been established each year [5]. And during 2010, the average website had 230 serious vulnerabilities [6].

Since the hackers only need a web browser to access web applications, web application vulnerabilities have greater impact than vulnerabilities in other applications and software, and normal web application use involves sensitive information, such as keys for secure sessions [7].

The paper target websites written in PHP since is it more difficult to develop and targeting the security of the websites. In particular, the paper tries to check the security for the website from inside the organization that has the website. This will avoids threats from many sources such as x-employees.

A program is writing for this purpose. It takes an input the source code of any websites written in PHP, and generates a report of the holes in the system.

The organization of this paper is as follows. Chapter 2 introduces the types of attacks. Chapter 3 discusses the current related work and Security testing. Chapter 4 presents the suggested approach to un-cover threats, business analysis of our tool, performance results, and discussion. Chapter 5 is the future work and conclusions and finally, Chapter 6 lists the references and appendixes.

II. TYPES OF ATTACKS

Following are the most common types of vulnerabilities. Some of these attacks are simple while others are sophisticated. In Figure 1 shows the evolution of attack sophistication versus the necessary technical knowledge of the intruders throughout the years.
A. **SQL injection:** "is an attack method used by hackers to retrieve, manipulate, fabricate or delete information in organizations’ relational databases through web applications" [8]. According to OWASP (Open Web Application Security Paper), the most common two vulnerability types in 2010 are SQL Injection and cross-site scripting (XSS) [9].

B. **Cross site scripting:** Cross-site scripting (XSS) has been the number one web application vulnerability for many years. It is a type of vulnerability that allows attackers to inject unauthorized code into a web page, which is interpreted and executed by the user’s web browser [10].

In 2010, 64% of websites had at least one Information Leakage vulnerability, which overtook Cross-Site Scripting as the most prevalent vulnerability by a few tenths of a percent [11].

C. **Password guessing:** "A password guessing attack occurs when an unauthorized user tries to log on repeatedly to a computer or network by guessing usernames and passwords. Many password guessing programs that attempt to break passwords are available on the Internet. Password guessing might use one of the following two approaches. They are brute force or dictionary approach. Each one has advantages and disadvantages" [12].

D. **Backdoor:** "A back door is a means of access to a computer program that bypasses security mechanisms. A programmer may sometimes install a back door so that the program can be accessed for troubleshooting or other purposes. However, attackers often use back doors that they detect or install themselves, as part of an exploit. In some cases, a worm is designed to take advantage of a back door created by an earlier attack" [13].
E. **Password cracking:** "is the process of recovering passwords from data that has been stored in or transmitted by a computer system. A common approach is to repeatedly try guesses for the password. The purpose of password cracking might be to help a user recover a forgotten password to gain unauthorized access to a system, or as a preventive measure by system administrators to check for easily crackable passwords" [14].

F. **Self-replication code:** "is any behavior of a dynamical system that yields construction of an identical copy of that dynamical system." [15]. "Self-replication is the common feature of most malicious codes, allowing them to maximize their impact. This approach is an extension of the earlier developed method for detecting previously unknown viruses in script based computer codes" [16].

G. **Hijacking session:** "is a common form of attack against websites. Hackers using this attack are able to take advantage of poorly configured websites to literally hijack a user's session and take over their identity. Session hijacking, also known as TCP session hijacking, is a method of taking over a Web user session by surreptitiously obtaining the session ID and masquerading as the authorized user" [17].

Many web applications are vulnerable to session hijacking attacks due to the insecure use of cookies for session management. For this reason, the most recommended defense against to this threat is to completely replace HTTP with HTTPS [18].

H. **Sniffers:** "a program that tracks a person straight to their IP and computer" [19]. "In common industry usage, a sniffer (with lower case "s") is a program that monitors and analyzes network traffic, detecting bottlenecks and problems. Using this information, a network manager can keep traffic flowing efficiently" [20].

I. **Denial-of-service attack (DoS attack) or distributed denial-of-service attack (DDoS attack):** "is an attempt to make a computer resource unavailable to its intended users. Although the means to carry out, motivations for, and targets of a DoS attack may vary. It is generally consists of the concerted efforts of a person, or multiple people to prevent an Internet site or service from functioning efficiently or at all, temporarily or indefinitely." [21].

J. **Spoofing attack:** "is a situation in which one person or program successfully masquerades as another by falsifying data and thereby gaining an illegitimate advantage", [22].

"Wireless networks are vulnerable to spoofing attacks, which allow many other forms of attacks on the networks. Spoofing attacks are a serious threat as they represent a form of identity compromise and can facilitate a variety of traffic injection attacks, such as evil twin access point attacks" [23].

K. **Stealth techniques:** "The "stealth" techniques tries to gather information about target sites while avoiding detection by using techniques that might be overlooked by intrusion detection systems and system administrators" [24].

L. **Multi-Staged network attacks:** "in which machine A penetrates and “takes over” machine B, which then does the same to machine C, etc." [25]. "Multi-stage attacks can be orchestrated to strike highly protected targets, to coordinate waves of scripted exploits and/or to conceal the true origin of an attack" [26].

M. **Sweepers:** "Virus Sweeper is a fake spyware remover, rogue anti-spyware application. Likely, it’s just a rename of
another rogue anti-spyware application called Virus Doctor. Virus Sweeper could be downloaded and installed manually from certain malicious websites; however in most cases Virus Sweeper enters the system without user permission and knowledge from various noxious websites" [27].

N. Automated scans: "As far as web-based applications are concerned, there are a number of methods and security evaluation techniques that can be used to uncover information about an application that has a security context. An automated scanner makes use of one or more discovery techniques to request data and scans each page returned by the web server and attempts to categories or identify relative information" [28].

O. Exploiting known vulnerabilities: "Network devices can be discovered and profiled in much the same way as other types of systems. Attackers usually start with port scanning. After they identify open ports, they use banner grabbing and enumeration to detect device types and to determine operating system and application versions. Armed with this information, an attacker can attack known vulnerabilities that may not be updated with security patches" [29].

"One of the most urgent security problems facing administrators of networked computer systems today is the threat of remote attacks on their systems over the Internet, based on vulnerabilities in their currently running software. Particularly damaging have been self-propagating attacks, or "worms", which exploit one or more vulnerabilities to take control of a host, then use that host to find and attack other hosts with the same vulnerability" [30].

In addition to the previous mentioned attacks, there are other uncommon types of attacks. They are: Auto Coordinated attacks, GUI attacks, burglaries attacks, disabling audits attacks and www attacks.

III. CURRENT RELATED WORKS

In this section, we will review the current related work of security vulnerabilities in web sites. The researcher Maureen Doyle performs an empirical investigation of the evolution of vulnerabilities in the most widely used open source PHP web applications. The mentioned researcher found that the security of open source web applications improved from 2006 to 2010 [31]. However, there still a lot of risk that still exists.

There are few tools that try to detect vulnerabilities mainly related to the static analysis that is required to detect vulnerabilities or to detect potential injection locations. These tools are interesting to analyze vulnerability detection on PHP web sites. They are:

- **Pixy**
  Nenad Jovanovic and his colleagues were developed Pixy, the first open source tool aimed to detection of XSS vulnerabilities, other vulnerabilities such as SQL injection and command injection using taint analysis [32]. Pixy is a Java program takes a PHP4 program as input [33], and it is run by specifying one file where vulnerabilities will be searched, then presented in a summary in the terminal. Also, a DOT file can be produced in figure 2, which can be visualized using the dot application from Graphviz that represents the taint path that causes the vulnerability. Pixy uses flow-sensitive; inter procedural and context-sensitive analysis [34]. Also, Pixy it's free and can scan online or download a free version and it's available on web site:
  http://pixybox.seclab.tuwien.ac.at/pixy/webinterface.php
RIPS

Johannes Dahse was developing RISP. RIPS is a tool written in PHP to find vulnerabilities in PHP applications using static code analysis. The goal of RIPS is to build a new approach of this written in PHP itself using the build-in tokenizer functions [35].

RIPS it's requires setting up a local web server in order to use it. Also, it can be controlled completely using a practical web interface that allows scanning files for vulnerabilities while customizing the verbosity level, the vulnerabilities to analyze, and even the code style in which results are presented [36]. In addition to that, RIPS is open source and can anyone to download the program from http://rips-scanner.sourceforge.net/.

It is introduced which automates the process of identifying potential security flaws in PHP source code. The result of the analysis can easily be reviewed by the penetration tester in its context without reviewing the whole source code again [37].

Acunetix

Acunetix Web Vulnerability Scanner (WVS) is an automated web application security testing tool that scans web applications by checking for vulnerabilities like SQL Injections, Cross site scripting, open ports, and other exploitable hacking vulnerabilities. In general, Acunetix WVS scans any website or web application that is accessible via a web browser and uses the HTTP/HTTPS protocol.

Acunetix is typically consists of two phases:

1. Crawling – the Crawler automatically crawls and analyzes the entire website by following all the links on the site and in the robots.txt file and sitemap.xml (if available) the website and then builds a site structure.

2. Scanning – Acunetix WVS launches a series of web vulnerability checks against the website or web application – in effect, emulating a hacker. The results of a scan are displayed in the Alert Node tree and include comprehensive details on all the vulnerabilities found within the website.

Suggested approach

We have written a program that implements the following algorithm:

Algorithm Detect_vulnerabilities (All the URL code)

Begin

Step 0: Call Function Check_For_Open_Ports( );
 //This function aimed to detect if there is any opening port without closing it that is used by hackers to get some data.

Step 1: Call function Check_For_Hidden_Access_To_FTP( );
 //This function aimed to detect if there any hidden access to FTP files that is used by hackers to get some data.

Step 2: Call function Check_For_Hidden_pages( );
 //This function aimed to detect if there any of any sub-domain that is not public and is not listed on the website map.

Step 3: Call function Check_For_SQL_Injection( );
 // Checks for the 3 possible types of SQL injection.

Step 4: Call function Check_For_Cross-Site_Scripting( );
 //This function aimed to detect if there any XSS.
Step 5: Call function 
Check_For_Backdoor( );
//This function aimed to detect any backdoor used by hacker to access the website.

Step 6: Call function 
Check_For_PHP’s_Execution_Functions( );
//This function aimed to detect if there any execution functions command in PHP codes.

Step 7: Call function 
Check_For_Robots_Pages( );
//This function aimed to detect if there "robots.txt" file in website.

End

The listed algorithm has a lot of details in each step. However, due to limitations in space, we are not able to list them.

IV. PERFORMANCE ANALYSIS

We have run our system on www.sdiwc.net/emailer website that is national and we got the following results:

<table>
<thead>
<tr>
<th>Website Name</th>
<th>Rank*</th>
<th>Pixy</th>
<th>RIPS</th>
<th>Suggested Approach</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.sdiwc.net/emailer">www.sdiwc.net/emailer</a></td>
<td>2,210,652</td>
<td>XSS found</td>
<td>XSS, SQL injection, and other vulnerability found.</td>
<td>Open port, hidden pages, and existing system function.</td>
</tr>
</tbody>
</table>

*alexa.com/

We have tried to get another PHP source code. However, no organization able to offered their source code. We have approached many friend and people, and the response was negative. We hope that we can get some in the future so that we can do comparison with other organizations. The results show in Appendix A.

V. DISCUSSION

Pixy has not any updates since 2007 and there are many requirements to run Pixy like a database management system, the Java environment, Perl programming language and PHP. While RIPS has continuous updates and the latest in March 2012. In addition, RIPS doesn’t need any requirement to run it only Mozilla Firefox and a local web server parsing PHP files.

Pixy’s main advantage lies in its static analysis [38] and the disadvantage of Pixy is it scan every single entry file, this is means we must scan all the files one after another until finished from all files. This method is time consuming when we have to scan the entire website. For this reason, we just test seven files in Pixy while we scan a whole files in RIPS. Also we can show taint path that causes the vulnerability using a DOT files. Both Pixy reports and a dot file shows in Appendix A.

Both Pixy and RIPS used a static source code analysis. It widely used for a variety of goals such as syntax highlighting, type checking, optimization as well as bug and security finding [39].

Acunetix is a commercial program and there is a free version that only scans Cross Site Scripting, it's available on the web site: http://www.acunetix.com/vulnerability-scanner/download.htm. In addition, the free version not works well and there is no output because the scan is abort!

CONCLUSION

In conclusion, 30% of all vulnerabilities found in computer software were PHP related [40] and as well known the static analysis tools are great for helping programmers to understand the source code they are working on, and to find potential problems. In addition, there are few tools that try to detect vulnerabilities through the use of static analysis even though the impact of its usage is great in detecting vulnerabilities or potential injection locations.
For this reason, we have created a static analysis tool in java programming language that scans PHP website folders and generate a report. Our tools found some of the vulnerabilities where no existing tool (to our knowledge) were able to detect and some other vulnerabilities that can be detected by a few existing tools.

In this paper and due to time constraint, we shed light on business analysis. However, we focused our efforts in the implementation and programming side to reach tangible results.

VI. REFERENCES


