TOPIC

VULNERABILITY AND THEIR POTENTIAL IMPACT

By

CYBER SQUAD

Introduction

Vulnerabilities in computer systems, software, and networks are weaknesses exploited by attackers to compromise information integrity, confidentiality, or availability, with potential impacts varying depending on the nature and context.

The potential impact of vulnerabilities can vary widely depending on the nature of the vulnerability and the context in which it is exploited. Here are some common types of vulnerabilities :

~Vulnerabilities
~Network Vulnerabilities
~Human-Related Vulnerabilities
~Hardware Vulnerabilities
~Web Application Vulnerabilities
~IoT Device Vulnerabilities

This report is based on the web application vulnerability assessment conducted on http://testphp.vulnweb.com/*

Findings

After a close evaluation, we found that the website is at a high-risk level.

CVSS - Common Vulnerability Scoring System

CVE - Common Vulnerabilities and Exposures

VULNERABILITIES FOUND FOR SERVER-SIDE SOFTWARE

CVSS	CVE	Summary	Exploit	Affected software
7.5	CVE-2017-8923	The zend_string_extend function in Zend/zend_string.h in PHP through 7.1.5 does not prevent changes to string objects that result in a negative length, which allows remote attackers to cause a denial of service (application crash) or possibly have unspecified other impact by leveraging a script's use of .= with a long string.	N/A	php 5.6.40
7.5	CVE-2019-9641	An issue was discovered in the EXIF component in PHP before 7.1.27, 7.2.x before 7.2.16, and 7.3.x before 7.3.3. There is an uninitialized read in exif_process_IFD_in_TIFF.	N/A	php 5.6.40
6.8	CVE-2015-9253	An issue was discovered in PHP 7.3.x before 7.3.0alpha3, 7.2.x before 7.2.8, and before 7.1.20. The php-fpm master process restarts a child process in an endless loop when using program execution functions (e.g., passthru, exec, shell_exec, or system) with a non-blocking STDIN stream, causing this master process to consume 100% of the CPU, and consume disk space with a large volume of error logs, as demonstrated by an attack by a customer of a shared-hosting facility.	N/A	php 5.6.40
6.5	CVE-2022-31629	In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the vulnerability enables network and same-site attackers to set a standard insecure cookie in the victim's browser which is treated as a `__Host-` or `__Secure-` cookie by PHP applications.	N/A	php 5.6.40
5.8	CVE-2017-7272	PHP through 7.1.11 enables potential SSRF in applications that accept an fsockopen or pfsockopen hostname argument with an expectation that the port number is constrained. Because a :port syntax is recognized, fsockopen will use the port number that is specified in the hostname argument, instead of the port number in the second argument of the function.	N/A	php 5.6.40

These vulnerabilities expose the affected applications to the risk of unauthorized access to confidential data and possibly denial of service attacks. An attacker could search for an appropriate exploit (or create one himself) for any of these vulnerabilities and use it to attack the system.

The recommendation is to upgrade the affected software to the latest version in order to eliminate the risk of these vulnerabilities.

The URL we used here is http://testphp.vulnweb.com Here the communication is made over unsecured, unencrypted HTTP.

COMMUNICATION IS NOT SECURE

Communication is made over unsecure , unencrypted HTTP.The communication between the web browser and the server is done using the HTTP protocol, which transmits data unencrypted over the network. Thus, an attacker who manages to intercept the communication at the network level is able to read and modify the data transmitted (including passwords, secret tokens, credit card information, and other sensitive data).

The recommendation is to reconfigure the web server to use HTTPS - which encrypts the communication between the web browser and the server.

INSECURE CLIENT ACCESS POLICY

The crossdomain.xml file controls the access of externally hosted Flash scripts to this website. The external websites that are permitted to read content from this website via Flash are specified in the XML tag <allow-access-from> . If the value of this tag is too permissive (ex. wildcard), it means that any Flash script from an external website could access content from this website, including confidential information of users.

The clientaccesspolicy.xml file specifies that other websites can read content from this website - which is normally denied by the Same Origin Policy. If the allowed domains are too permissive (ex. wildcard) then any external website will be able to read content (including sensitive information) from this website.

Flash is not supported anymore and this poses a risk only if the user's clients use older browsers, making them vulnerable to their information being accessed by a malicious external Flash script .

The recommendation is carefully reviewing the content of the policy file and permitting access only for legitimate domains.

MISSING SECURITY HEADER: X-CONTENT-TYPE-OPTIONS

Response header do not include the X-Content -Type -OptionsHTTP securely header.

The HTTP header X-Content-Type-Options is addressed to the Internet Explorer browser and prevents it from reinterpreting the content of a web page (MIME-sniffing) and thus overriding the value of the Content-Type header). Lack of this header could lead to attacks such as Cross-Site Scripting or phishing.

The recommendation is setting the X-Content-Type-Options header such as X-Content-Type-Options: nosniff .

MISSING SECURITY HEADER : REFERENCE -POLICY

Response headers do not include the Referrer-Policy HTTP security header as well as the <meta> tag with the name 'referrer' is not present in the response.

The Referrer-Policy HTTP header controls how much referrer information the browser will send with each request originating from the current web application.

For instance, if a user visits the web page "http://example.com/pricing/" and it clicks on a link from that page, e.g."https://www.google.com", the browser will send to Google the full originating URL in the Referer header, assuming the Referrer-policy header is not set. The originating URL could be considered sensitive information and it could be used for user tracking.

The Referrer-Policy header should be configured on the server side to avoid user tracking and inadvertent information leakage. The value no-referrer of this header instructs the browser to omit the Referer header entirely.

MISSING SECURITY HEADER :X-FRAME -OPTIONS

The response headers do not include the HTTP X-Frame-Otions security header.

Because the X-Frame-Options header is not sent by the server, an attacker could embed this website into an iframe of a third-party website. By manipulating the display attributes of the iframe, the attacker could trick the user into performing mouse clicks in the application, thus performing activities without user consent (ex: deleting the user, subscribing to a newsletter, etc). This is called a Clickjacking attack.
https://owasp.org/www-community/attacks/Clickjacking

The recommendation is to add the X-Frame-Options HTTP header with the values DENY or SAMEORIGIN to every page that you want to be protected against Clickjacking attacks.

MISSING SECURITY HEADER :CONTENT -SECURITY -POLICY

The Content-Security-Policy (CSP) header activates a protection mechanism implemented in web browsers which prevents exploitation of Cross-Site Scripting vulnerabilities (XSS). If the target application is vulnerable to XSS, lack of this header makes it easily exploitable by attackers.

The recommendation is to configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

SERVER SOFTWARE AND TECHNOLOGY FOUND

Software / Version	Category
Ubuntu	Operating systems
PHP 5.6.40	Programming languages
DreamWeaver	Editors
Nginx 1.19.0	Web servers, Reverse proxies

An attacker could use this information to mount specific attacks against the identified software type and version.

The recommendation is to eliminate the information which permits the identification of software platform, technology, server, and operating system: HTTP server headers, HTML meta information, etc.

SECURITY.TXT FILE IS MISSING

Missing: http://testphp.vulnweb.com/.well-known/security.txt

We have detected that the server is missing the security.txt file. There is no particular risk in not creating a valid Security.txt file for your server. However, this file is important because it offers a designated channel for reporting vulnerabilities and security issues.

The recommendation is implement the security.txt file according to the standard, in order to allow researchers or users to report any security issues they find, improving the defensive mechanisms of your server.

SOME POSITIVE SIDES WE FOUND

~The website is accessible.

~Nothing was found for robots.txt file.

~Nothing was found for use of untrusted certificates.

~Nothing was found for enabled HTTP debug methods.

~Nothing was found for directory listing.

~Nothing was found for missing HTTP header - Strict-Transport-Security.

~Nothing was found for domain too loose set for cookies.

~Nothing was found for HttpOnly flag of cookie.

~Nothing was found for Secure flag of cookie.

~Nothing was found for unsafe HTTP header Content Security Policy.

OVERVIEW OF SOLUTIONS

Based on the vulnerabilities and security issues found, the following recommendations are suggested:

1. Upgrade affected software to the latest version to eliminate the risks of vulnerabilities such as CVE-2017-8923, CVE-2019-9641, CVE-2022-31629, CVE-2017-7272, and CVE-2015-9253.

2. Enable HTTPS to encrypt communication between the web browser and server to prevent interception and modification of sensitive data.

3. Carefully review the crossdomain.xml and clientaccesspolicy.xml files to permit access only for legitimate domains.

4. Set the X-Content-Type-Options header to nosniff to prevent MIME-sniffing and potential attacks such as Cross-Site Scripting or phishing.

5. Configure the Referrer-Policy header to avoid user tracking and inadvertent information leakage. The value no-referrer should be used to omit the Referer header entirely.

6. Add the X-Frame-Options HTTP header with the values DENY or SAMEORIGIN to protect against Clickjacking attacks.

7. Configure the Content-Security-Policy header to apply specific policies needed by the application to prevent exploitation of Cross-Site Scripting vulnerabilities (XSS).

8. Eliminate information that permits the identification of software platforms, technology, servers, and operating systems to prevent specific attacks against identified software types and versions.

9. Implement the security.txt file according to the standard to allow researchers or users to report any security issues they find, improving the defensive mechanisms of the server.

10. Review the website's accessibility, robots.txt file, use of untrusted certificates, enabled HTTP debug methods, directory listing, missing HTTP header - Strict-Transport-Security, domain too loose set for cookies, HttpOnly flag of cookie, Secure flag of cookie, and unsafe HTTP header Content Security Policy.