PHANTOMSEC

Web Application Penetration Testing Report

PROPOSAL ISSUED: 17-11-2023

REPORT ANALYSIS

- **The issues identified and proposed action plans in this report are based on our testing. We made specific efforts to verify the accuracy and authenticity of the information gathered only in those cases where it was felt necessary.**

- **The identification of the issues in the report is mainly based on the tests carried out during the limited time for conducting such an exercise. As the basis of selecting the most appropriate weaknesses / vulnerabilities is purely judgmental in view of the time available, the outcome of the analysis may not be exhaustive and represents all possibilities, though we have taken reasonable care to cover the major eventualities.**

- **The vulnerabilities reported in this report are valid as of 17 November, 2023.**

SUMMARY

Summary:

Vulnerabilities Identified:

X-Frame-Options Header Missing

Risk Level: Low

Details: Absence of X-Frame-Options header may lead to clickjacking attacks, exacerbating other vulnerabilities. Mitigation involves implementing the header to regulate framing.

Solution: Add the 'X-Frame-Options' header to the server's HTTP response.

X-XSS-Protection Header Missing

Risk Level: Medium

Details: Absence of X-XSS-Protection header leaves the system susceptible to Cross-site Scripting (XSS) attacks. Implementation of the header, coupled with a Web Application Firewall, is advised.

Solution: Add the 'X-XSS-Protection' header and implement a Web Application Firewall.

X-Content-Type-Options Header Missing

Risk Level: Medium

Details: Lack of X-Content-Type-Options header might lead to MIME type confusion, potentially executing scripts or triggering other vulnerabilities.

Solution: Implement the 'X-Content-Type-Options' header in the server response.

Email Spoofing Potential

Risk Level: Medium

Details: Vulnerability in the domain may allow email spoofing. Recommendations include setting up DMARC, DKIM, and SPF for email security.

Solution: Establish DMARC, DKIM, and SPF records to enhance email security.

Open Directory Listings

Risk Level: Medium

Details: Various directories exposing contents pose risks like sensitive information disclosure and phishing opportunities.

Solution: Prevent directory listings through .htaccess files, web server configurations, default index files, or use of web application firewalls.

Parameter Vulnerable to XSS

Risk Level: Low

Details: Cross-Site Scripting vulnerability identified in a URL parameter, potentially allowing malicious JavaScript execution.

Solution: Properly validate and encode user-supplied input to prevent XSS vulnerabilities in URLs.

PHP Version Vulnerabilities

Various PHP version vulnerabilities identified (CVE-2015-9253, CVE-2018-5711, CVE-2018-5712, CVE-2022-31628, CVE-2022-31629).

Risk Level: Medium

Details: Multiple vulnerabilities identified in the PHP versions require updating to the latest version to mitigate risks.

Solution: Update PHP to the latest secure version.

Conclusion:

The report highlights critical vulnerabilities such as missing security headers, potential email spoofing, open directory listings, and PHP version vulnerabilities. Immediate actions should focus on implementing missing headers, enhancing email security, securing directories, and updating PHP to the latest secure version to mitigate these risks.

WEB ANALYSIS

Total: 35 vulnerabilities

8	12	13	2
Critical risk items	High risk items	Medium risk items	Low risk items

Number of analyzed items:
27,161

Main risk categories found

misconfiguration	3	spoofing	1
open_directory	4	gain_information	1
xss	3	sql_injection	2
generic_cve	21		

VULNERABILITIES

Risk	Category	Name
Risk low	misconfiguration	The X-Frame-Options header is missing
Risk medium	misconfiguration	X-XSS-Protection header is missing
Risk medium	misconfiguration	X-Content-Type-Options header is miss...
Risk medium	spoofing	The domain http://testphp.vulnweb.co...
Risk medium	open_directory	The directory http://testphp.vulnweb.c...
Risk medium	gain_information	The directory http://testphp.vulnweb.c...
Risk medium	open_directory	The directory http://testphp.vulnweb.c...
Risk medium	open_directory	The directory http://testphp.vulnweb.c...
Risk medium	open_directory	The directory http://testphp.vulnweb.c...

ISSUE BACKGROUND

The X-Frame-Options header is missing

misconfigurationDomain/subdomain

Risk

Risk low
Details

The lack of the X-Frame-Options header in the response from the Web application server, makes it possible to hijack on the user's click, where through a malicious indexing of a page on an attacker's website, it could allow the hiding of this domain through an overlay, causing involuntary actions performed by a victim in the background. This type of exploit could make other security vulnerabilities even more serious, such as turning a self-XSS into a reflected one. Self-XSS occurs when a certain user input field is not properly filtered, this type of exploitation that, in theory, would only happen on the attacker's computer and would have to require a lot of interaction from the victim to happen, in addition to just clicking or visiting the page as in other cases, however, the user click hijacking ends up hiding from the victim's eyes what he actually ends up doing,

this exploration can end up triggering what would simply be a self-XSS without any security impact, for a conventional one, such as the reflected one.

CWE

657

CVE

CVSS

CVSS:3.0/AV:N/AC:L/PR:L/UI:R/S:U/C:N/I:L/A:N

Solution

The X-Frame-Options is an HTTP response header that aids in preventing clickjacking attacks by regulating how a website is rendered on another site's frame, iframe, or object. It has three potential values: 'SAMEORIGIN' (allows rendering on the same domain only), 'DENY' (blocks rendering on any origin), and 'ALLOW-FROM uri' (permits rendering only on a specified origin). To apply X-Frame-Options, add it to the HTTP response on your web server. The implementation varies with the web server software. For example, in Apache, add "Header set X-Frame-Options 'SAMEORIGIN'" to the .htaccess file. The 'SAMEORIGIN' value is advised for most websites, allowing intra-domain framing but blocking inter-domain ones.

—----------

X-XSS-Protection header is missing

misconfiguration

Domain/subdomain

Risk

Risk medium
Details

The X-XSS-Protection header is missing, which could make it easier to Cross-site scripting (XSS) exploration, as on the reviewed site, does not have any filter that could prevent exploitation of this security hole. XSS vulnerability happens due to a parameter that is not well filtered and ends up reflecting entirely everything that is typed by the user via the URL, including HTML tags and JavaScript codes. If successfully exploited this vulnerability could allow that an attacker could craft a fake page within the site true what would bring about a legitimacy in the coup. Furthermore, as this is a flaw in the site, mechanisms for third party protection would be ineffective. If the user's session is shared with other subdomains and the victim is logged in the moment they click the malicious link, an attacker who injected malicious code could capture the victim's session without having to collect passwords and would have the same access privileges as that user. This situation becomes even more serious if a certain session captured for some

administrative access, which could cause the elevation of an attacker's privileges or the exploitation of others security flaws.

CWE

79

CVE

Not applicable

CVSS

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:U/C:L/I:L/A:N

Solution

The X-XSS-Protection header in the web server response and a Web Application Firewall are recommended to prevent security vulnerabilities like XSS attacks. Adding the X-XSS-Protection header to all web pages can be done through server configuration or direct HTML code insertion. A typical setting, "X-XSS-Protection: 1; mode=block", activates browser's XSS protection and blocks any identified XSS attack.

—----------

X-Content-Type-Options header is missing

misconfiguration

Domain/subdomain

Risk

Risk medium
Details

Failure to use X-Content-Type-Options header could allow an attacker to spoof a certain type of file that would be analyzed through MIME type detection, which could confuse the browser from its actual validation, where it would lead to the execution of other vulnerabilities such as Cross-site scripting. When a file does not have enough information to determine its origin, such as the presence of metadata, browsers determine the extension of that file, from its contents. This type of behavior can become a security risk,if the browser misinterprets a given file in some form of uploading files, for example, a JPEG file could have been misinterpreted, if the content of your file existed HTML tags and Javascript codes, instead of the browser treating this extension as a corrupted image, would execute the codes typed by the user or in a malicious way by falsifying a victim's request, after clicking a fake link or visiting a website controlled by an attacker.

CWE

693

CVE

Not applicable

CVSS

CVSS:3.0/AV:N/AC:L/PR:N/UI:R/S:U/C:L/I:L/A:N

Solution

Implement the X-Content-Type-Options header in the server response.

—--

The domain http://testphp.vulnweb.com may be vulnerable to email spoofing.

spoofing

Domain/subdomain

Risk

Risk medium
Details

Email spoofing is the creation of email messages with a forged sender address. The term applies to email purporting to be from an address which is not actually the sender's; mail sent in reply to that address may bounce or be delivered to an unrelated party whose identity has been faked. Assistant: ['Found SPF record:', 'v=spf1 -all', 'SPF record contains an All item: -all', 'No DMARC record found. Looking for organizational record', 'No organizational DMARC record']

CWE

290

CVE

Not applicable

CVSS

CVSS:3.1/AV:N/AC:L/PR:N/UI:N/S:U/C:N/I:L/A:N

Solution

Set up DMARC, DKIM, and SPF for email security. DMARC requires creating a record in your domain's DNS, detailing report address and message failure policy. DKIM involves generating a public/private key pair, publishing the public key in DNS as a TXT record, and configuring your email server to sign messages with the private key. SPF requires creating a DNS record that lists authorized

email servers for your domain. Setup can vary with different email server software and hosting providers; check your specific setup's documentation.

—----------

The directory http://testphp.vulnweb.com/Mod_Rewrite_Shop/images/ is listing the contents of the folder

open_directory

Domain/subdomain

Risk

Risk medium
Details

Web application listing files and directories can present several safety hazards. Here are a few examples: Sensitive information disclosure: When a web application allows directory listing, it may reveal sensitive information such as file names, directory structure, and even the content of files. This could include sensitive data such as configuration files, backups, and logs that contain sensitive information. Vulnerable files and directories: When a web application allows directory listing, it may reveal the presence of files and directories that are known to be vulnerable to attack. This could include old, unpatched versions of software, or files that have weak permissions. Attack surface: When a web application allows directory listing, it increases the attack surface of the application. Attackers can use the information revealed through directory listing to identify and exploit vulnerabilities in the application. Phishing: Attackers can create a fake website to mimic the real one, and use the information from the listing directory to make it more convincing.

CWE

548

CVE

CVSS

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:N/A:N

Solution

There are several ways to prevent directory listing in a web service: Use a .htaccess file: This file can be used to configure Apache web server settings. By adding the following line to the .htaccess file, directory listing will be disabled: Options -Indexes Use web server's configuration: Some web servers such as Apache have a global configuration file where you can set the options for directory listing. You can disable directory listing for the entire server by adding the following line to the configuration file: Options -Indexes Use a default index file: By default, most web servers will display the content of a directory if there is

no index file present. You can prevent directory listing by creating an index file such as index.html or index.php in every directory. Use a web application firewall: Some web application firewall has the capability to detect and block directory listing attempts. It's recommended to use more than one method to prevent directory listing, since it can add an extra layer of security.

—----------

The directory http://testphp.vulnweb.com/Mod_Rewrite_Shop/.htaccess is listing the contents of the folder

gain_information

Domain/subdomain

Risk

Risk medium
Details

Web application listing files and directories can present several safety hazards. Here are a few examples: Sensitive information disclosure: When a web application allows directory listing, it may reveal sensitive information such as file names, directory structure, and even the content of files. This could include sensitive data such as configuration files, backups, and logs that contain sensitive information. Vulnerable files and directories: When a web application allows directory listing, it may reveal the presence of files and directories that are known to be vulnerable to attack. This could include old, unpatched versions of software, or files that have weak permissions. Attack surface: When a web application allows directory listing, it increases the attack surface of the application. Attackers can use the information revealed through directory listing to identify and exploit vulnerabilities in the application. Phishing: Attackers can create a fake website to mimic the real one, and use the information from the listing directory to make it more convincing.

CWE

548

CVE

Not applicable

CVSS

CVSS:3.0/AV:N/AC:L/PR:N/UI:N/S:C/C:L/I:N/A:N

Solution

There are several ways to prevent directory listing in a web service: Use a .htaccess file: This file can be used to configure Apache web server settings. By adding the following line to the .htaccess file, directory listing will be disabled:

Options -Indexes Use web server's configuration: Some web servers such as Apache have a global configuration file where you can set the options for directory listing. You can disable directory listing for the entire server by adding the following line to the configuration file: Options -Indexes Use a default index file: By default, most web servers will display the content of a directory if there is no index file present. You can prevent directory listing by creating an index file such as index.html or index.php in every directory. Use a web application firewall: Some web application firewall has the capability to detect and block directory listing attempts. It's recommended to use more than one method to prevent directory listing, since it can add an extra layer of security.

—----------

Parameter seems vulnerable to XSS at url http://testphp.vulnweb.com/Mod_Rewrite_Shop/Details/web-camera-a4tech/2/?id=DalFox The injection type is BUILTIN and evidences: . Reference: Found dalfox-error-mysql2 via built-in grepping / payload: DalFox

XSS

Domain/subdomain

Risk

Risk low
Details

Cross-Site Scripting (XSS) is a vulnerability that occurs when a web application allows untrusted data to be included in a web page without proper validation. XSS vulnerabilities in a URL can occur when a web application takes user-supplied input, such as data from a URL parameter, and includes it in a web page without proper encoding or validation. When an XSS vulnerability exists in a URL, an attacker can craft a malicious URL that includes malicious JavaScript code, which will be executed when the URL is visited by a user. This can allow the attacker to steal sensitive information, such as passwords and session tokens, or to perform actions on behalf of the user, such as posting malicious content.

CWE

CVE

Not applicable

CVSS

Solution

To prevent XSS vulnerabilities in a URL, it is important to properly validate and encode all user-supplied input, including data from URL parameters, before including it in a web page. This can help to ensure that only trusted data is included in a web page and prevent malicious code from being executed in a user's browser.

—----------

PHP@5.6.40

generic_cve

Domain/subdomain

Risk

Risk medium
Details

An issue was discovered in PHP 7.3.x before 7.3.0 alpha3, 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.

CWE

400

CVE

CVE-2015-9253

CVSS

—----------

PHP@5.6.40

generic_cve

Domain/subdomain

Risk

Risk medium
Details

gd_gif_in.c in the GD Graphics Library (aka libgd), as used in PHP before 5.6.33, 7.0.x before 7.0.27, 7.1.x before 7.1.13, and 7.2.x before 7.2.1, has an integer signedness error that leads to an infinite loop via a crafted GIF file, as

demonstrated by a call to the image create from gif or imagecreatefromstring PHP function. This is related to GetCode_ and gd ImageCreateFromGif tx.

CWE

681835

CVE

CVE-2018-5711

CVSS

—----------

PHP@5.6.40

generic_cve

Domain/subdomain

Risk

Risk medium
Details

An issue was discovered in PHP before 5.6.33, 7.0.x before 7.0.27, 7.1.x before 7.1.13, and 7.2.x before 7.2.1. There is Reflected XSS on the PHAR 404 error page via the URI of a request for a .phar file.

CWE

79

CVE

CVE-2018-5712

CVSS

—----------

PHP@5.6.40

generic_cve

Domain/subdomain

Risk

Risk medium
Details

In PHP versions before 7.4.31, 8.0.24 and 8.1.11, the phar uncompressor code would recursively uncompress "quines" gzip files, resulting in an infinite loop.

cwe

674

cve

CVE-2022-31628

cvss

—----------

PHP@5.6.40

generic_cve

Domain/subdomain

Risk

Risk medium
Details

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.

cwe

cve

CVE-2022-31629

cvss

Solution
—----------

1. X-Frame-Options Header Missing:
Remediation: Implement the 'X-Frame-Options' header in the server's HTTP response.
Method: Add the header with the value 'SAMEORIGIN' to allow framing only by pages from the same domain.

2. X-XSS-Protection Header Missing:
Remediation: Add the 'X-XSS-Protection' header and deploy a Web Application Firewall.
Method: Set the 'X-XSS-Protection' header to '1; mode=block' to activate browser XSS protection.

3. X-Content-Type-Options Header Missing:
Remediation: Implement the 'X-Content-Type-Options' header in the server response.
Method: Add 'X-Content-Type-Options: nosniff' to prevent MIME-sniffing attacks.

4. Email Spoofing Potential:
Remediation: Establish DMARC, DKIM, and SPF records for enhanced email security.
Method: Set up DMARC policies, configure DKIM keys, and create SPF records in the domain's DNS settings.

5. Open Directory Listings:
Remediation: Prevent directory listing using various methods.
Methods:
Utilize .htaccess files to disable directory listings (Options -Indexes).
Configure web server settings globally to block directory listings.
Create default index files (e.g., index.html) in directories.
Implement web application firewalls to detect and block directory listing attempts.

6. Parameter Vulnerable to XSS:
Remediation: Properly validate and encode user-supplied input to prevent XSS vulnerabilities.
Method: Apply strict input validation and encoding functions to sanitize user inputs, especially in URL parameters.

7. PHP Version Vulnerabilities:
Remediation: Update PHP to the latest secure version to patch identified vulnerabilities.
Method: Upgrade PHP to the most recent stable version available, addressing the disclosed CVEs.

These remediation methods aim to address each specific vulnerability by implementing necessary headers, configurations, security measures, and software updates to bolster the web application's security posture.

PROOF OF CONCEPT

Proof-of-Concept for X-Frame-Options Header Missing:

Vulnerability Overview: The absence of the X-Frame-Options header exposes the application to clickjacking attacks, potentially leading to unintended actions by users.

Proof-of-Concept (POC):
To demonstrate the impact of the missing X-Frame-Options header, we conducted a simple test using an HTML page embedding the vulnerable domain within an iframe without any restrictions.

```html
<!DOCTYPE html>
<html>
<head>
    <title>Clickjacking POC</title>
</head>
<body>
    <h1>Clickjacking POC</h1>
    <p>This page demonstrates a clickjacking attack.</p>
    <iframe src="http://vulnerable-domain.com" width="800" height="600" frameborder="
</body>
</html>
```

CONCLUSION

The report highlights critical vulnerabilities such as missing security headers, potential email spoofing, open directory listings, and PHP version vulnerabilities. Immediate actions should focus on implementing missing headers, enhancing email security, securing directories, and updating PHP to the latest secure version to mitigate these risks.