True Inspect Powered by Invicti
by True Positives

Vulnerability Scan Report

Basic Information

Report Delivered:	Jan 25 2024
Requested By:	DK
Company Name:	ACME

True Inspect Threat Level 4

One or more critical-severity type vulnerabilities have been discovered by the scanner. A malicious user can exploit these vulnerabilities and compromise the backend database and/or deface your website.

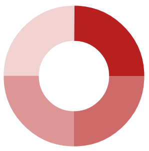
Scan Detail

Target	http://php.testinvicti.com
Scan Type	Full Web and Network Scan
Start Time	Apr 2, 2024, 2:46:46 PM GMT
Scan Duration	32 minutes
Requests	35500
Average Response Time	1ms
Maximum Response Time	29999ms
Application Build	vnull
Authentication Profile	-

4	32	28	26	19
Critical	High	Medium	Low	Informational

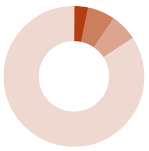
Severity	Vulnerabilities	Instances
Critical	4	4
High	18	32
Medium	23	28
Low	21	26
Informational	14	19
Total	80	109

Critical Severity



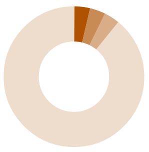
		Instances
🟥	Code Evaluation (PHP)	1
🟥	Command Injection	1
🟥	Server-Side Template Injection	1
🟥	Others	1

High Severity



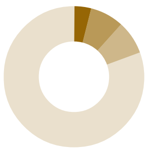
		Instances
🟧	Apache 2.2.14 mod_isapi Dangling Pointer	1
🟧	Apache HTTP Server < 2.4.49 Multiple Vuln...	2
🟧	Apache HTTP Server <= 2.4.51 Buffer Over...	2
🟧	Others	27

Medium Severity



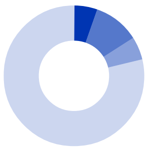
		Instances
🟧	Apache 2.x version older than 2.2.9	1
🟧	Apache httpd remote denial of service	1
🟧	Apache httpOnly cookie disclosure	1
🟧	Others	25

Low Severity



		Instances
🟫	Apache 2.x version older than 2.2.10	1
🟫	Apache HTTP Server < 2.4.54 Multiple Vuln...	2
🟫	Apache HTTP Server 'httpOnly' Cookie Inf...	2
🟫	Others	21

Informational



		Instances
🟦	Apache HTTP Server Detection Consolidati...	1
🟦	CGI Scanning Consolidation	2
🟦	Content Security Policy (CSP) Not Impleme...	1
🟦	Others	15

Impacts

SEVERITY	IMPACT	
Critical	1	Code Evaluation (PHP)
Critical	1	Command Injection
Critical	1	Server-Side Template Injection
Critical	1	SQL Injection
High	1	[Possible] Backup Source Code Detected
High	1	Apache 2.2.14 mod_isapi Dangling Pointer
High	2	Apache HTTP Server 'mod_auth_digest' Multiple Vulnerabilities (Windows)
High	2	Apache HTTP Server < 2.4.49 Multiple Vulnerabilities - Windows
High	2	Apache HTTP Server <= 2.4.51 Buffer Overflow Vulnerability - Windows
High	2	Apache HTTP Server <= 2.4.52 Multiple Vulnerabilities - Windows
High	2	Apache HTTP Server End of Life (EOL) Detection (Windows)
High	1	Apache HTTP Server Multiple Vulnerabilities June17 (Windows)
High	6	Cross-site Scripting
High	1	Directory traversal
High	1	Local File Inclusion
High	2	PHP '_php_stream_scandir()' Buffer Overflow Vulnerability (Windows)
High	2	PHP 'type confusion' Denial of Service Vulnerability (Windows)
High	2	PHP < 5.6.29, 7.0.x < 7.0.14 DoS Vulnerability - Windows
High	2	PHP Denial of Service Vulnerability - 02 - Aug16 (Windows)
High	1	Security vulnerability in MySQL/MariaDB sql/password.c
High	1	SVN Detected
High	1	User controllable script source
Medium	1	Apache 2.x version older than 2.2.9
Medium	2	Apache HTTP Server < 2.4.48 NULL Pointer Dereference Vulnerability - Windows
Medium	2	Apache HTTP Server Man-in-the-Middle Attack Vulnerability - July16 (Windows)
Medium	1	Apache httpd remote denial of service

SEVERITY	IMPACT	
Medium	1	Apache httpOnly cookie disclosure
Medium	1	Directory listings
Medium	1	HTTP parameter pollution
Medium	1	Insecure crossdomain.xml policy
Medium	1	Insecure HTTP Usage
Medium	1	Microsoft Access Database File Detected
Medium	1	Password transmitted over HTTP
Medium	2	PHP 'gdImageScaleTwoPass()' Multiple Denial of Service Vulnerabilities (Windows)
Medium	1	PHP 'phar/tar.c' Heap Buffer Overflow Vulnerability (Windows)
Medium	1	PHP 'socket_connect()' Buffer Overflow Vulnerability (Windows)
Medium	1	PHP 'timelib_meridian' Heap Based Buffer Overflow Vulnerability (Windows)
Medium	2	PHP 'WDDX Deserialization' Denial of Service Vulnerability - (Windows)
Medium	2	PHP Denial of Service Vulnerability Jul17 (Windows)
Medium	1	PHP hangs on parsing particular strings as floating point number
Medium	1	PHP register_globals Is Enabled
Medium	1	PHP session.use_only_cookies Is Disabled
Medium	1	PHPinfo pages
Medium	1	Source code disclosures
Medium	1	SSL/TLS Not Implemented
Low	1	[Possible] Internal IP Address Disclosure
Low	1	Apache 2.x version older than 2.2.10
Low	2	Apache HTTP Server 'httpOnly' Cookie Information Disclosure Vulnerability
Low	1	Apache HTTP Server 'mod_dav_svn' Denial of Service Vulnerability (Windows)
Low	2	Apache HTTP Server < 2.4.54 Multiple Vulnerabilities - Windows
Low	1	Apache HTTP Server Scoreboard Security Bypass Vulnerability (Windows)
Low	1	Apache mod_negotiation filename bruteforcing
Low	2	HTTP Debugging Methods (TRACE/TRACK) Enabled
Low	1	Insecure Frame (External)
Low	2	PHP 'tsrm_win32.c' Denial Of Service Vulnerability (Windows)

SEVERITY	IMPACT	
Low	1	PHP allow_url_fopen Is Enabled
Low	1	PHP allow_url_include Is Enabled
Low	2	PHP Denial of Service Vulnerability - 01 - Jul16 (Windows)
Low	1	PHP display_errors Is Enabled
Low	1	PHP EXIF Header Denial of Service Vulnerability (Windows)
Low	1	PHP open_basedir Is Not Configured
Low	1	Possible sensitive directories
Low	1	Possible virtual host found
Low	1	Programming Error Messages
Low	1	TRACE/TRACK Method Detected
Low	1	Version Disclosure (PHP)
Informational	1	[Possible] Internal Path Disclosure (Windows)
Informational	1	Apache HTTP Server Detection Consolidation
Informational	2	CGI Scanning Consolidation
Informational	1	Content Security Policy (CSP) Not Implemented
Informational	1	Error page web server version disclosure
Informational	1	Generic Email Address Disclosure
Informational	1	Hostname Determination Reporting
Informational	2	HTTP Server Banner Enumeration
Informational	1	OS Detection Consolidation and Reporting
Informational	1	Permissions-Policy header not implemented
Informational	2	PHP < 7.2.33, 7.3 < 7.3.21, 7.4 < 7.4.9 DoS Vulnerability - August20 (Windows)
Informational	2	PHP Detection (HTTP)

Code Evaluation (PHP)

This script is vulnerable to PHP code injection.

PHP code injection is a vulnerability that allows an attacker to inject custom code into the server side scripting engine. This vulnerability occurs when an attacker can control all or part of an input string that is fed into an eval() function call. Eval will execute the argument as code.

Impact

An attacker can execute any PHP code on your server.

http://php.testinvicti.com/hello.php

URL encoded GET input **name** was set to **;assert(base64_decode('cHJpbnQobWQ1KDMxMzM3KSk7'));**

Possible execution result:

```
6f3249aa304055d63828af3bfab778f6
```

Request

```
GET /hello.php?name=;assert(base64_decode('cHJpbnQobWQ1KDMxMzM3KSk7')); HTTP/1.1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36
Host: php.testinvicti.com
Connection: Keep-alive
```

Recommendation

Your script should properly sanitize user input.

References

Dynamic Evaluation Vulnerabilities in PHP applications
https://seclists.org/fulldisclosure/2006/May/35

OWASP PHP Top 5
https://www.owasp.org/index.php/PHP_Top_5

Command Injection

This script is possibly vulnerable to code execution attacks.

Code injection vulnerabilities occur where the output or content served from a Web application can be manipulated in such a way that it triggers server-side code execution. In some poorly written Web applications that allow users to modify server-side files (such as by posting to a message board or guestbook) it is sometimes possible to inject code in the scripting language of the application itself.

Impact

A malicious user may execute arbitrary system commands with the permissions of the web server.

http://php.testinvicti.com/nslookup.php Verified

URL encoded POST input **param** was set to **echo zsxnvo$()\ gjulzs\nz^xyu||a #' &echo zsxnvo$()\ gjulzs\nz^xyu||a #|" &echo zsxnvo$()\ gjulzs\nz^xyu||a #**

Possible execution result:

```
zsxnvo$()\ gjulzs\nzxyu
```

Proof of Exploit

DNS lookup - hitswtxxyrkie53af7.bxss.me

```
DNS IP: 3.228.172.252
DNS TYPE: CNAME
DNS QUERY: hitswtxxyrkie53af7.bxss.me
```

Request

```
POST /nslookup.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Content-Length: 185
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36
Host: php.testinvicti.com
Connection: Keep-alive

param=echo%20zsxnvo%24()%5C%20gjulzs%5Cnz%5Exyu%7C%7Ca%20%23'%20%26echo%20zsxnvo%24()%5C%20gjulzs%5Cnz%5Exyu%7C%7Ca%20%23%7C"%
20%26echo%20zsxnvo%24()%5C%20gjulzs%5Cnz%5Exyu%7C%7Ca%20%23
```

Recommendation

Your script should filter metacharacters from user input.

References

Security Focus - Penetration Testing for Web Applications (Part Two)
https://www.symantec.com/connect/articles/penetration-testing-web-applications-part-two

OWASP PHP Top 5
https://www.owasp.org/index.php/PHP_Top_5

Server-Side Template Injection

This script is possibly vulnerable to Server-side template injection attacks.

Server-side template injection occurs when user-controlled input is embedded into a server-side template, allowing users to inject template directives. This allows an attacker to inject malicious template directives and possibly execute arbitrary code on the affected server.

Impact

An attacker may inject malicious template directives and possibly execute arbitrary code on the affected server.

http://php.testinvicti.com/artist.php Verified

URL encoded GET input **id** was set to **dfb{{98991*97996}}xca**.
The response contained the result of the evaluated expression: **dfb9700722036xca**
Templating engine: **Twig/Jinja2/Unknown**

Proof of Exploit
DNS lookup - hitcqpfcrsaxmf49c2.bxss.me

```
DNS IP: 3.228.172.160
DNS TYPE: CNAME
DNS QUERY: hitcqpfcrsaxmf49c2.bxss.me
```

Request

```
GET /artist.php?id=dfb{{98991*97996}}xca HTTP/1.1
Referer: http://php.testinvicti.com/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36
Host: php.testinvicti.com
Connection: Keep-alive
```

Recommendation

Templates should not be created from user-controlled input. User input should be passed to the template using template parameters.

References

Server-Side Template Injection
https://portswigger.net/blog/server-side-template-injection

SQL Injection

SQL injection (SQLi) refers to an injection attack wherein an attacker can execute malicious SQL statements that control a web application's database server.

Impact

An attacker can use SQL injection to bypass a web application's authentication and authorization mechanisms and retrieve the contents of an entire database. SQLi can also be used to add, modify and delete records in a database, affecting data integrity. Under the right circumstances, SQLi can also be used by an attacker to execute OS commands, which may then be used to escalate an attack even further.

http://php.testinvicti.com/artist.php Verified

URL encoded GET input **id** was set to **-1 OR 3*2*1=6 AND 000663=000663**

Tests performed:

- -1 OR 2+663-663-1=0+0+0+1 => **TRUE**
- -1 OR 3+663-663-1=0+0+0+1 => **FALSE**
- -1 OR 3*2<(0+5+663-663) => **FALSE**
- -1 OR 3*2>(0+5+663-663) => **FALSE**
- -1 OR 2+1-1+1=1 AND 000663=000663 => **FALSE**
- -1 OR 3*2=5 AND 000663=000663 => **FALSE**
- -1 OR 3*2=6 AND 000663=000663 => **TRUE**
- -1 OR 3*2*0=6 AND 000663=000663 => **FALSE**
- -1 OR 3*2*1=6 AND 000663=000663 => **TRUE**

Original value: **test**

Proof of Exploit

SQL query - SELECT database()

```
sqlibench
```

Request

```
GET /artist.php?id=-1%20OR%203*2*1=6%20AND%20000663=000663 HTTP/1.1
X-Requested-With: XMLHttpRequest
Referer: http://php.testinvicti.com/
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36
Host: php.testinvicti.com
Connection: Keep-alive
```

Recommendation

Use parameterized queries when dealing with SQL queries that contain user input. Parameterized queries allow the database to understand which parts of the SQL query should be considered as user input, therefore solving SQL injection.

References

SQL Injection (SQLi) - Acunetix
https://www.acunetix.com/websitesecurity/sql-injection/

Types of SQL Injection (SQLi) - Acunetix
https://www.acunetix.com/websitesecurity/sql-injection2/

Prevent SQL injection vulnerabilities in PHP applications and fix them - Acunetix
https://www.acunetix.com/blog/articles/prevent-sql-injection-vulnerabilities-in-php-applications/

SQL Injection - OWASP
https://www.owasp.org/index.php/SQL_Injection

Bobby Tables: A guide to preventing SQL injection
https://bobby-tables.com/

SQL Injection Cheet Sheets - Pentestmonkey
http://pentestmonkey.net/category/cheat-sheet/sql-injection

[Possible] Backup Source Code Detected

A possible backup file was found on your web-server. These files are usually created by developers to backup their work.

Impact

Backup files can contain script sources, configuration files or other sensitive information that may help an malicious user to prepare more advanced attacks.

http://php.testinvicti.com/process.bak Confidence: 80%

This file was found using the pattern **${fileName}.bak**.
Original filename: **process.php**
Pattern found:

```
<?php
require("auth.php");
ini_set("display_errors","0");

//global configuration area
$globals["title"] = "Invicti Test Web Site - PHP";
function EndsWith($FullStr, $EndStr)
{
// Get the length of the end string
$StrLen = strlen($EndStr);
// Look at the end of FullStr for the substring the size of EndStr
$FullStrEnd = substr($FullStr, strlen($FullStr) - $StrLen);
// If it matches, it does end with EndStr
return $FullStrEnd == $EndStr;
}
?>

<?php include "Internals/header.php"?>
<body>
<div id="wrapper">
<?php include "Internals/upmenu.php"?>
<?php
$file = $_REQUEST["file"];
if(EndsWith($file,".nsp"))
include $_REQUEST["file"];
?>
<!-- end #page -->
</div>

<?php include "Internals/footer.php"?>
<!-- end #footer -->
</body>
</html>
```

Request

```
GET /process.bak HTTP/1.1
Range: bytes=0-99999
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Encoding: gzip,deflate,br
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36
Host: php.testinvicti.com
Connection: Keep-alive
```

Recommendation

Remove the file(s) if they are not required on your website. As an additional step, it is recommended to implement a security policy within your organization to disallow creation of backup files in directories accessible from the web.

References

Testing for Old, Backup and Unreferenced Files (OWASP-CM-006)
https://www.owasp.org/index.php/Review_Old,_Backup_and_Unreferenced_Files_for_Sensitive_Information_(OTG-CONFIG-004)

Security Tips for Server Configuration
https://httpd.apache.org/docs/2.4/misc/security_tips.html

Protecting Confidential Documents at Your Site
http://www.w3.org/Security/Faq/wwwsf5.html

Apache 2.2.14 mod_isapi Dangling Pointer

This alert was generated using only banner information. It may be a false positive.

By sending a specially crafted request followed by a reset packet it is possible to trigger a vulnerability in Apache mod_isapi that will unload the target ISAPI module from memory. However function pointers still remain in memory and are called when published ISAPI functions are referenced. This results in a dangling pointer vulnerability.

Affected Apache versions (up to 2.2.14 on Windows platform).

Impact

Successful exploitation results in the execution of arbitrary code with SYSTEM privileges.

http://php.testinvicti.com/

Version detected: Apache/2.2.8 .

Recommendation

Upgrade Apache to the latest version.

References

Apache 2.2.14 mod_isapi Dangling Pointer
https://web.archive.org/web/20210328112429/http://www.senseofsecurity.com.au/advisories/SOS-10-002

Apache homepage
http://httpd.apache.org

Apache HTTP Server 'mod_auth_digest' Multiple Vulnerabilities (Windows)

Apache HTTP Server is prone to multiple vulnerabilities.

Impact

Successful exploitation will allow remote attackers to cause the target service to crash. A remote user can obtain potentially sensitive information as well on the target system.

107.20.213.223:443/tcp

Installed version: 2.2.8 Fixed version: 2.2.34 Installation path / port: 443/tcp

107.20.213.223:80/tcp

Installed version: 2.2.8 Fixed version: 2.2.34 Installation path / port: 80/tcp

Recommendation

Update to Apache HTTP Server 2.2.34 or 2.4.27 or later.

Description

Family: Web Servers

Insight:

The flaw exists due to error in Apache 'mod_auth_digest' which does not properly initialize memory used to process 'Digest' type HTTP Authorization headers.

Affected:

Apache HTTP Server 2.2.x before 2.2.34 and 2.4.x before 2.4.27.

Apache HTTP Server < 2.4.49 Multiple Vulnerabilities - Windows

Apache HTTP Server is prone to multiple vulnerabilities.

Impact

107.20.213.223:443/tcp

Installed version: 2.2.8 Fixed version: 2.4.49 Installation path / port: 443/tcp

107.20.213.223:80/tcp

Installed version: 2.2.8 Fixed version: 2.4.49 Installation path / port: 80/tcp

Recommendation

Update to version 2.4.49 or later.

Description

Family: Web Servers

Insight:

The following vulnerabilities exist:

CVE-2021-34798: NULL pointer dereference in httpd core

CVE-2021-39275: ap_escape_quotes buffer overflow

CVE-2021-40438: mod_proxy SSRF

Affected:

Apache HTTP Server version 2.4.48 and prior.

Apache HTTP Server <= 2.4.51 Buffer Overflow Vulnerability - Windows

Apache HTTP Server is prone to a buffer overflow vulnerability.

Impact

107.20.213.223:443/tcp