Pentest-Tools.com

Website Vulnerability Scanner Report

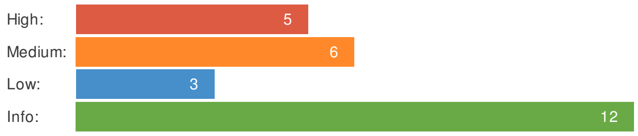
✓ **http://demo.pentest-tools.com/webapp/**

Summary

Overall risk level:

High

Risk ratings:



Scan information:

Start time:	2017-09-13 15:27:25
Finish time:	2017-09-13 15:27:50
Scan duration:	25.0 seconds
Tests performed:	26/26
Scan status:	Finished

Findings

Potentially sensitive files found

/webapp/backup.tgz

Details

Risk description:
These files can contain confidential information such as: application source code, configuration files, SSL certificates, etc. Manual review is required for the contents of these files.

Recommendation:
We recommend removing these files from the website directory if they are not needed for business purposes.

Remote Command Execution

URL	Remark
/webapp/cgi-bin/guestbook.cgi	May allow attackers to execute commands as the web daemon.

Details

Risk description:
The risk exists that an attacker will use this vulnerability to execute arbitrary commands on the server. As a result, the attacker could steal confidential information (user personal data, passwords, etc) or he could try to further penetrate the internal network and other servers from the same network.

Recommendation:
We recommend removing the affected script if it is not needed for business purposes or upgrading it to a current version which fixes this vulnerability.

More information about this issue:
https://www.owasp.org/index.php/Testing_for_Command_Injection_(OTG-INPVAL-013)

SQL Injection

URL	Remark
/webapp/CHANGELOG.txt	Version number implies that there is a SQL Injection in Drupal 7, can be used for authentication bypass (Drupageddon: see https://www.sektioneins.de/advisories/advisory-012014-drupal-pre-auth-sql-injection-vulnerability.html).

Details

Details

Risk description:
An atacker could exploit this vulnerability to execute arbitrary SQL commands on the database. As a result, he could extract sensitive data or further pivot to the operating system level.

Recommendation:
We recommend upgrading the web application and the vulnerable scripts to a recent version which fixes this vulnerability. Otherwise, the affected scripts should be removed from the server.

More information about this issue:
https://www.owasp.org/index.php/Testing_for_SQL_Injection_(OTG-INPVAL-005)

Arbitrary File Read

URL	Remark
/webapp/cgi-bin/banner.cgi	This CGI may allow attackers to read any file on the system.

Details

Risk description:
A malicious user could use this vulnerability to read arbitrary files from the web server including: source code files, configuration files, system files, etc. The information from these files could help the attacker to gain full access to the server.

Recommendation:
We recommend upgrading the web application and the vulnerable scripts to a recent version which fixes this vulnerability. Otherwise, the affected scripts should be removed from the server.

More information about this issue:
https://www.owasp.org/index.php/Testing_for_Local_File_Inclusion

Script Injection

URL	Remark
/webapp/phpinfo.php3?VARIABLE=<script>alert('Vulnerable')</script>	Contains PHP configuration information and is vulnerable to Cross Site Scripting (XSS).

Details

Risk description:
An attacker could inject arbitrary JavaScript code into the web browser of a victim user. As a result, the attacker could steal the victim's session cookies or steal confidential information from the victim's web application.

Recommendation:
We recommend upgrading the web application and the vulnerable scripts to a recent version which fixes this vulnerability. Otherwise, the affected scripts should be removed from the server.

More information about this issue:
https://www.owasp.org/index.php/Cross-site_Scripting_(XSS)

Directory listing is enabled

/webapp/download_folder/
/webapp/logs/

Details

Risk description:
An attacker can see the entire structure of files and subdirectories from the affected URL. It is often the case that sensitive files are 'hidden' among public files in that location and attackers can use this vulnerability to access them.

Recommendation:
We recommend reconfiguring the web server in order to deny directory listing. Furthermore, you should verify that there are no sensitive files at the mentioned URLs.

More information about this issue:
http://projects.webappsec.org/w/page/13246922/Directory%20Indexing.

Other security issues found

Server leaks inodes via ETags, header found with file /webapp/, fields: 0x2ba5 0x54d77c10df823
Allowed HTTP Methods: GET, HEAD, POST, OPTIONS
/webapp/test.php?%3CSCRIPT%3Ealert('Vulnerable')%3C%2FSCRIPT%3E=x: Output from the phpinfo() function was found.
/webapp/test.php: Output from the phpinfo() function was found.

Details

Risk description:
These findings should be manually analyzed and it must be decided if they present a security risk or not.

Recommendation:
We recommend taking appropriate actions according to the results of the risk analysis performed.

Server software is outdated

Apache/2.4.10 is outdated

Details

Risk description:
Outdated server software usually contains bugs and security vulnerabilities which could be exploited by malicious users to affect the confidentiality, integrity or availability of the application data.

Recommendation:
Upgrade to at least Apache/2.4.25

Server misconfiguration

URL	Remark
/webapp/.git/index	Git Index file may contain directory listing information.
/webapp/.git/HEAD	Git HEAD file found. Full repo details may be present.
/webapp/.git/config	Git config file found. Infos about repo details may be present.
/webapp/setup.sql	Setup SQL file found.

Details

Risk description:
These scripts are accessible because the server was badly configured and deployed. These scripts usually contain sensitive information which can be used by attackers to further compromise the system and steal confidential data.

Recommendation:
We recommend removing the above mentioned scripts if they are not needed for business purposes or to verify that they do not leak sensitive information.

Interesting files found

URL	Remark
/webapp/admin/	This might be interesting...
/webapp/logs/	This might be interesting...
/webapp/README.TXT	This might be interesting...
/webapp/INSTALL.txt	Default file found.
/webapp/CHANGELOG.txt	A changelog was found.
/webapp/admin/home.php	Admin login page/section found.
/webapp/admin/index.html	Admin login page/section found.
/webapp/test.php	This might be interesting...

/webapp/debug.php	Possible debug directory/program found.

Details

Risk description:
These files/folders usually contain sensitive information which may help attackers to mount further attacks against the server. Manual validation is required.

Recommendation:
We recommend you to analyze if the mentioned files/folders contain any sensitive information and restrict their access according to the business purposes of the application.

Server information disclosure

URL	Remark
/webapp/test.php	PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.

Details

Risk description:
An attacker could use these files to find information about the backend application, server software and their specific versions. This information could be further used to mount targeted attacks against the server.

Recommendation:
We recommend you to remove these scripts if they are not needed for business purposes.

More information about this issue:
http://projects.webappsec.org/w/page/13246936/Information%20Leakage

Server software and technology found

Technology	unknown
Server	Apache 2.4.10
Operating system	unknown

Details

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

Recommendation:
We recommend you to eliminate the information which permit the identification of software platform, technology, server and operating system: HTTP server headers, meta information, etc.

More information about this issue:
https://www.owasp.org/index.php/Fingerprint_Web_Server_(OTG-INFO-002).

Missing HTTP security headers

HTTP Security Header	Header Role	Status
X-Frame-Options	Protects against Clickjacking attacks	Not set
X-XSS-Protection	Mitigates Cross-Site Scripting (XSS) attacks	Not set
X-Content-Type-Options	Prevents possible phishing or XSS attacks	Not set

Details

Risk description:
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's consent (ex: delete user, subscribe to newsletter, etc). This is called a Clickjacking attack and it is described in detail here:
https://www.owasp.org/index.php/Clickjacking

The **X-XSS-Protection** HTTP header instructs the browser to stop loading web pages when they detect reflected Cross-Site Scripting (XSS) attacks. Lack of this header exposes application users to XSS attacks in case the web application contains such vulnerability.

vulnerability.

The HTTP **X-Content-Type-Options** header is addressed to 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.

Recommendation:
We recommend you to add the X-Frame-Options HTTP response header to every page that you want to be protected against Clickjacking attacks.
More information about this issue:
https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet

We recommend setting the X-XSS-Protection header to "X-XSS-Protection: 1; mode=block".
More information about this issue:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection

We recommend setting the X-Content-Type-Options header to "X-Content-Type-Options: nosniff".
More information about this issue:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options

Robots.txt file found

URL	Remark
/webapp/robots.txt	Contains 1 entries
/webapp/download_folder/	This URL from robots.txt is accessible

Details

Risk description:
There is no particular security risk in having a robots.txt file. However, this file is often misused to try to hide some web pages from the users. This should not be done as a security measure because these URLs can easily be read from the robots.txt file.

Recommendation:
We recommend you to remove the entries from robots.txt which lead to sensitive locations in the website (ex. administration panels, configuration files, etc).

More information about this issue:
https://www.theregister.co.uk/2015/05/19/robotstxt/

No interesting HTTP headers found

No security issue found regarding HTTP cookies

No WAF/Load Balancer/Proxy was detected

No security issue found regarding client access policies

HTTP PUT/DELETE methods are not enabled

No scripts vulnerable to Remote File Inclusion were found

No scripts vulnerable to Unauthorized File Upload were found

No scripts vulnerable to Authentication Bypass were found

No scripts vulnerable to Denial of Service were found

No administration consoles were found

No server software was identified

No web services were found

Scan coverage information

List of tests performed (26/26)

- ✓ Fingerprinting the server software and technology...
- ✓ Analyzing HTTP security headers...
- ✓ Checking for interesting HTTP headers...
- ✓ Analyzing HTTP cookies...
- ✓ Checking for directory listing...
- ✓ Detecting the presence of WAF/Load Balancer/Proxy...
- ✓ Testing for other security issues...
- ✓ Checking client access policies...
- ✓ Checking robots.txt file...
- ✓ Searching for sensitive files...
- ✓ Testing for HTTP PUT/DELETE methods...
- ✓ Checking for outdated server software...
- ✓ Checking for Remote Command Execution (known scripts) ...
- ✓ Checking for SQL injection (known scripts)...
- ✓ Checking for Arbitrary File Read (known scripts)...
- ✓ Checking for Remote File Inclusion (known scripts)...
- ✓ Checking for Unauthorized File Upload (known scripts)...
- ✓ Checking for Script Injection (known scripts)...
- ✓ Checking for Authentication Bypass (known scripts)...
- ✓ Checking for Denial of Service (known scripts)...
- ✓ Checking for administration consoles...
- ✓ Checking for server misconfiguration...
- ✓ Checking for interesting files...
- ✓ Checking for information disclosure (known scripts)...
- ✓ Checking for software identification (known scripts)...
- ✓ Checking for known web services...

Scan parameters

Website URL:	http://demo.pentest-tools.com/webapp/
Scan type:	Full