Website Vulnerability Scanner Report (Light)

✔ https://distrex.co.id

Summary

Overall risk level:

High

Risk ratings:

High:	1
Medium:	2
Low:	7
Info:	7

Scan information:

Start time:	2021-04-20 08:59:07 UTC+03
Finish time:	2021-04-20 08:59:33 UTC+03
Scan duration:	26 sec
Tests performed:	17/17
Scan status:	Finished

Findings

Vulnerabilities found for server-side software

Risk Level	CVSS	CVE	Summary	Exploit	Affected software
●	7.5	CVE-2020-11984	Apache HTTP server 2.4.32 to 2.4.44 mod_proxy_uwsgi info disclosure and possible RCE	N/A	http_server 2.4.38
●	7.2	CVE-2019-0211	In Apache HTTP Server 2.4 releases 2.4.17 to 2.4.38, with MPM event, worker or prefork, code executing in less-privileged child processes or threads (including scripts executed by an in-process scripting interpreter) could execute arbitrary code with the privileges of the parent process (usually root) by manipulating the scoreboard. Non-Unix systems are not affected.	N/A	http_server 2.4.38
●	6.4	CVE-2019-10082	In Apache HTTP Server 2.4.18-2.4.39, using fuzzed network input, the http/2 session handling could be made to read memory after being freed, during connection shutdown.	N/A	http_server 2.4.38

●	6	CVE-2019-0215	In Apache HTTP Server 2.4 releases 2.4.37 and 2.4.38, a bug in mod_ssl when using per-location client certificate verification with TLSv1.3 allowed a client to bypass configured access control restrictions.	N/A	http_server 2.4.38
●	6	CVE-2019-0217	In Apache HTTP Server 2.4 release 2.4.38 and prior, a race condition in mod_auth_digest when running in a threaded server could allow a user with valid credentials to authenticate using another username, bypassing configured access control restrictions.	N/A	http_server 2.4.38

Details

Risk description:
These vulnerabilities expose the affected applications to the risk of unauthorized access to confidential data and possibly to 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.

Recommendation:
We recommend you to upgrade the affected software to the latest version in order to eliminate the risk of these vulnerabilities.

Insecure cookie setting: missing HttpOnly flag

Cookie Name	URL	Evidence
PHPSESSID	https://distrex.co.id	Set-Cookie: PHPSESSID=7kkbn52cevb9gqqn0vbeblk7ed; path=/

Details

Risk description:
A cookie has been set without the HttpOnly flag, which means that it can be accessed by the JavaScript code running inside the web page. If an attacker manages to inject malicious JavaScript code on the page (e.g. by using an XSS attack) then the cookie will be accessible and it can be transmitted to another site. In case of a session cookie, this could lead to session hijacking.

Recommendation:
Ensure that the HttpOnly flag is set for all cookies.
https://owasp.org/www-community/HttpOnly

Insecure cookie setting: missing Secure flag

Cookie Name	URL	Evidence
PHPSESSID	https://distrex.co.id	Set-Cookie: PHPSESSID=7kkbn52cevb9gqqn0vbeblk7ed; path=/

Details

Risk description:
Since the Secure flag is not set on the cookie, the browser will send it over an unencrypted channel (plain HTTP) if such a request is made. Thus, the risk exists that an attacker will intercept the clear-text communication between the browser and the server and he will steal the cookie of the user. If this is a session cookie, the attacker could gain unauthorized access to the victim's web session.

Recommendation:
Whenever a cookie contains sensitive information or is a session token, then it should always be passed using an encrypted channel. Ensure that the secure flag is set for cookies containing such sensitive information.
https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/06-Session_Management_Testing/02-Testing_for_Cookies_Attributes.html

Missing security header: Strict-Transport-Security

URL	Evidence
https://distrex.co.id	Response headers do not include the HTTP Strict-Transport-Security header

Details

Risk description:
The HTTP Strict-Transport-Security header instructs the browser to initiate only secure (HTTPS) connections to the web server and deny any unencrypted HTTP connection attempts. Lack of this header permits an attacker to force a victim user to initiate a clear-text HTTP connection to

the server, thus opening the possibility to eavesdrop on the network traffic and extract sensitive information (e.g. session cookies).

Recommendation:
The Strict-Transport-Security HTTP header should be sent with each HTTPS response. The syntax is as follows:

```
Strict-Transport-Security: max-age=<seconds>[; includeSubDomains]
```

The parameter `max-age` gives the time frame for requirement of HTTPS in seconds and should be chosen quite high, e.g. several months. A value below 7776000 is considered as too low by this scanner check.
The flag `includeSubDomains` defines that the policy applies also for sub domains of the sender of the response.

Missing security header: Content-Security-Policy

URL	Evidence
https://distrex.co.id	Response headers do not include the HTTP Content-Security-Policy security header

Details

Risk description:
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.

Recommendation:
Configure the Content-Security-Header to be sent with each HTTP response in order to apply the specific policies needed by the application.

Read more about CSP:
https://cheatsheetseries.owasp.org/cheatsheets/Content_Security_Policy_Cheat_Sheet.html
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy

Missing security header: X-Frame-Options

URL	Evidence
https://distrex.co.id	Response headers do not include the HTTP X-Frame-Options security header

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://owasp.org/www-community/attacks/Clickjacking

Recommendation:
We recommend you 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.

More information about this issue:
https://cheatsheetseries.owasp.org/cheatsheets/Clickjacking_Defense_Cheat_Sheet.html

Missing security header: X-XSS-Protection

URL	Evidence
https://distrex.co.id	Response headers do not include the HTTP X-XSS-Protection security header

Details

Risk description:
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.

Recommendation:
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

Missing security header: X-Content-Type-Options

URL	Evidence
https://distrex.co.id	Response headers do not include the X-Content-Type-Options HTTP security header

Details

Risk description:
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.

Recommendation:
We recommend setting the X-Content-Type-Options header such as `X-Content-Type-Options: nosniff` .

More information about this issue:
https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-Content-Type-Options.

Missing security header: Referrer-Policy

URL	Evidence
https://distrex.co.id	Response headers do not include the Referrer-Policy HTTP security header

Details

Risk description:
The Referrer-Policy HTTP header controls how much referrer information the browser will send with each request originated 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 going to 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.

Recommendation:
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.

Read more:
https://developer.mozilla.org/en-US/docs/Web/Security/Referer_header:_privacy_and_security_concerns

Server software and technology found

Software / Version	Category
Debian	Operating Systems
Apache 2.4.38	Web Servers
PHP	Programming Languages
Twitter Bootstrap	Web Frameworks
jQuery 3.3.1	JavaScript Frameworks
jQuery UI	JavaScript Frameworks

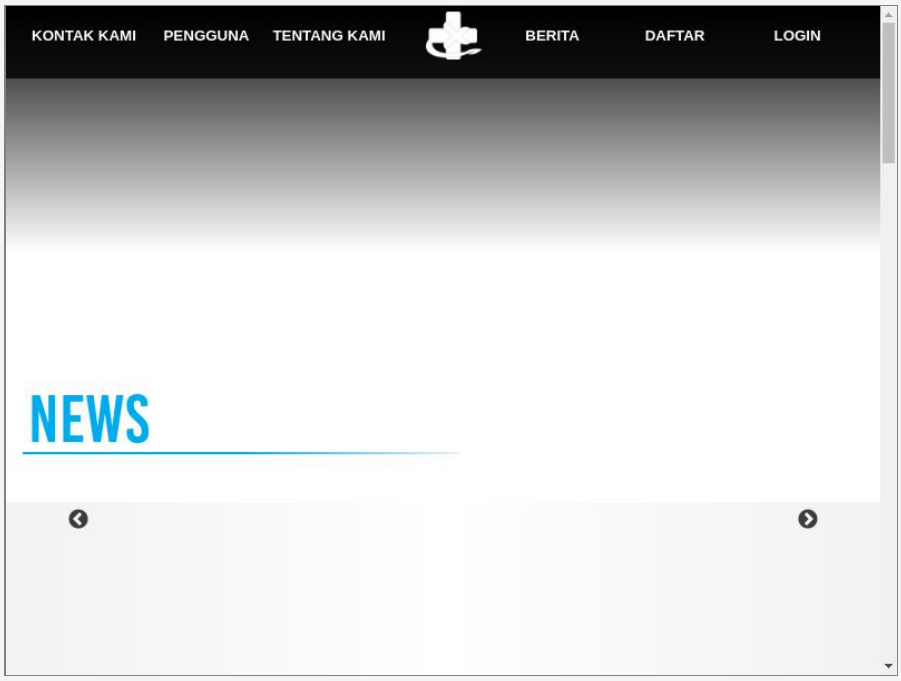
Details

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 permits the identification of software platform, technology, server and operating system: HTTP server headers, HTML meta information, etc.

More information about this issue:
https://owasp.org/www-project-web-security-testing-guide/stable/4-Web_Application_Security_Testing/01-Information_Gathering/02-Fingerprint_Web_Server.html.

Screenshot:



Website is accessible.

Nothing was found for client access policies.

Nothing was found for robots.txt file.

Nothing was found for use of untrusted certificates.

Nothing was found for domain too loose set for cookies.

Nothing was found for directory listing.

Nothing was found for secure communication.

Scan coverage information

List of tests performed (17/17)

- Checking for website accessibility...
- Checking for HttpOnly flag of cookie...
- Checking for Secure flag of cookie...
- Checking for missing HTTP header - Strict-Transport-Security...
- Checking for missing HTTP header - Content Security Policy...
- Checking for missing HTTP header - X-Frame-Options...
- Checking for missing HTTP header - X-XSS-Protection...
- Checking for missing HTTP header - X-Content-Type-Options...
- Checking for missing HTTP header - Referrer...
- Checking for website technologies...
- Checking for vulnerabilities of server-side software...
- Checking for client access policies...

- Checking for robots.txt file...
- Checking for use of untrusted certificates...
- Checking for domain too loose set for cookies...
- Checking for directory listing...
- Checking for secure communication...

Scan parameters

Website URL:	https://distrex.co.id
Scan type:	Light
Authentication:	False