Probely

SECURITY REPORT

Payroll Website - https://payrollmauritius.com

Report generated on 2020-05-17 at 11:45

Summary

This section contains the scan summary

TARGET https://payrollmauritius.com

Report generated on 2020-05-17 at 11:45

STARTED	ENDED	DURATION	SCAN PROFILE
May. 17, 2020, 07:34	May. 17, 2020, 07:50	15 minutes	Normal

NUMBER OF FINDINGS

	CURRENT SCAN	FROM LAST SCAN	PENDING FIX
HIGH	0	▼ 2	0
MEDIUM	0	▼ 1	0
LOW	0	= 0	0

TOP 5

Technical Summary

The following table summarizes the findings, ordered by their severity

Exhaustive Test List

The following pages contains the list of vulnerabilities we tested in this scan, taking into consideration the chosen profile

---

- Reflected cross-site scripting
- Cookie without HttpOnly flag
- Open redirection
- SQL Injection
- Missing cross-site request forgery protection
- Missing clickjacking protection
- Stored cross-site scripting
- Insecure crossdomain.xml policy
- SSL cookie without Secure flag
- HTTP TRACE method enabled
- Directory Listing
- ASP.NET tracing enabled
- Path traversal
- ASP.NET ViewState without MAC
- Session Token in URL
- Application error message
- Private IP addresses disclosed
- OS command injection
- XML external entity injection
- ASP.NET debugging enabled
- Insecure Silverlight clientaccesspolicy.xml policy
- PHP code injection
- Server-side JavaScript injection
- SQL injection (second order)
- Server-side template injection
- Unencrypted communications
- HSTS header not enforced
- Mixed content
- Cross Origin Resource Sharing: Arbitrary Origin Trusted
- Expired TLS certificate
- Insecure SSL protocol version 3 supported
- Outdated TLS protocol version 1.0 supported
- Secure TLS protocol version 1.2 not supported
- Weak cipher suites enabled
- Server Cipher Order not configured
- Untrusted TLS certificate
- Heartbleed
- Secure Renegotiation is not supported
- TLS Downgrade attack prevention not supported
- WordPress version with known vulnerabilities
- Joomla! version with known vulnerabilities
- Stored Open redirection
- Certificate without revocation information
- Full path disclosure
- HSTS header set in HTTP
- HSTS header with low duration and no subdomain protection

- HSTS header with low duration
- HSTS header does not protect subdomains
- Inclusion of cryptocurrency mining script
- Insecure SSL protocol version 2 supported
- Browser XSS protection disabled
- Browser content sniffing allowed
- Referrer policy not defined
- Insecure referrer policy
- Potential DoS on TLS Client Renegotiation
- JQuery library with known vulnerabilities
- AngularJS library with known vulnerabilities
- Bootstrap library with known vulnerabilities
- JQuery Mobile library with known vulnerabilities
- JQuery Migrate library with known vulnerabilities
- TLS certificate about to expire
- Moment.js library with known vulnerabilities
- Prototype library with known vulnerabilities
- React library with known vulnerabilities
- SWFObject library with known vulnerabilities
- TinyMCE library with known vulnerabilities
- Backbone library with known vulnerabilities
- Mustache library with known vulnerabilities
- Handlebars library with known vulnerabilities
- Dojo library with known vulnerabilities
- jPlayer library with known vulnerabilities
- CKEditor library with known vulnerabilities
- DWR library with known vulnerabilities
- Flowplayer library with known vulnerabilities
- DOMPurify library with known vulnerabilities
- Plupload library with known vulnerabilities
- easyXDM library with known vulnerabilities
- Ember library with known vulnerabilities
- YUI library with known vulnerabilities
- Sessvars library with known vulnerabilities
- jQuery UI library with known vulnerabilities
- WordPress plugin with known vulnerabilities
- Invalid referrer policy
- Insecure PHP Object deserialization

Detailed Finding Descriptions

This section contains the findings in more detail, ordered by severity

Glossary

Term	Definition
Vulnerability	A type of security weakness that might occur in applications (e.g. Broken Authentication, Information Disclosure). Some vulnerabilities take their name not from the weakness itself, but from the attack that exploits it (e.g. SQL Injection, XSS, etc.).
Findings	An instance of a Vulnerability that was found in an application.

Severity Legend

To each finding is attributed a severity which sums up its overall risk

The severity is a compound metric that encompasses the likelihood of the finding being found and exploited by an attacker, the skill required to exploit it, and the impact of such exploitation. A finding that is easy to find, easy to exploit and the exploitation has high impact, will have a greater severity.

Different findings of the same type could have a different severity: we consider multiple factors to increase or decrease it, such as if the application has an authenticated area or not.

The following table describes the different severities:

Severity	Description	Examples
HIGH	These findings may have a direct impact in the application security, either clients or service owners, for instance by granting the attacker access to sensitive information.	SQL Injection OS Command Injection
MEDIUM	Medium findings usually don't have immediate impact alone, but combined with other findings may lead to a successful compromise of the application.	Cross-site Request Forgery Unencrypted Communications
LOW	Findings where either the exploit is not trivial, the impact is low, or the finding cannot be exploited by itself.	Directory Listing Clickjacking

Category Descriptions

The following pages contain descriptions of each vulnerability. For each vulnerability you will find a section explaining its impact, causes and prevention methods.
These descriptions are very generic, and whenever they are not enough to understand or fix a given finding, more information is provided for that finding in the Detailed Finding Descriptions section.