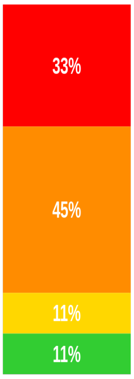
Security Alert: Dependency Health Report

Generated by ProSecureLabs Package Scanner

Security Assessment Complete

Repository Security Profile

Repository : `Arul-Prakash-R/Outdated_software` **Assessment Time :** `2025-10-12 06:08:51 UTC`
❗**Risk Classification :** `Medium Priority` **Branch :** `main`



Risk Assessment Matrix

Risk Level	Count	Percentage	Action	Response
🔴Critical	9	33%	Immediate	Address Immediately
🟠High Risk	10	45%	Priority	Resolve as High Priority
🟡Medium	3	11%	Schedule	Plan Timely Remediation
🟢Protected	3	11%	Maintain	Monitor and Maintain Compliance

System Security Status

Security Score: 68% - 24/27 packages need urgent updates!

Vulnerability Details

Node.js (npm)

Package	Current	Latest	Threat Level	Registry	Security Impact
mongoose	8.16.1	**8.19.1**	🟠**High**	npm	🟠**High Risk**

Python (pip)

Package	Current	Latest	Threat Level	Registry	Security Impact
PyGithub	2.6.1	**2.8.1**	🟠**High**	pip	🟠**High Risk**
flask	3.1.1	**3.1.2**	🟡**Medium**	pip	🟡**Medium Risk**

Rust

Package	Current	Latest	Threat Level	Registry	Security Impact
tokio	0.2.22	**1.47.1**	🔴**Critical**	rust	🔴**Critical Risk**
reqwest	0.10.10	**0.12.23**	🟠**High**	rust	🟠**High Risk**
serde	1.0.104	**1.0.228**	🟡**Medium**	rust	🟡**Medium Risk**

PHP (Composer)

Package	Current	Latest	Threat Level	Registry	Security Impact
laravel/framework	8.0.0	**v12.33.0**	🔴**Critical**	composer	🔴**Critical Risk**
symfony/http-foundation	5.2.0	**v7.3.4**	🔴**Critical**	composer	🔴**Critical Risk**
monolog/monolog	2.2.0	**3.9.0**	🔴**Critical**	composer	🔴**Critical Risk**
phpunit/phpunit	9.5.0	**12.4.1**	🔴**Critical**	composer	🔴**Critical Risk**
guzzlehttp/guzzle	7.0.0	**7.10.0**	🟠**High**	composer	🟠**High Risk**

mockery/mockery	1.4.0	**1.6.12**	🟠**High**	composer	🟠**High Risk**

Ruby (Gem)

Package	Current	Latest	Threat Level	Registry	Security Impact
rails	5.2.3	**8.0.3**	🔴**Critical**	ruby	🔴**Critical Risk**
pg	0.21.0	**1.6.2**	🔴**Critical**	ruby	🔴**Critical Risk**
httparty	0.14.0	**0.23.2**	🟠**High**	ruby	🟠**High Risk**

Go

Package	Current	Latest	Threat Level	Registry	Security Impact
github.com/gin-gonic/gin	v1.7.7	**v1.11.0**	🟠**High**	go	🟠**High Risk**
github.com/dgrijalva/jwt-go	v3.2.0	**v3.2.0+incompatible**	🟡**Medium**	go	🟡**Medium Risk**
golang.org/x/crypto	v0.0.0-20220315160706-3147a52a75dd	**v0.43.0**	⚫**Unknown**	go	⚫**Unknown Risk**

.NET

Package	Current	Latest	Threat Level	Registry	Security Impact
Newtonsoft.Json	12.0.1	**13.0.4**	🔴**Critical**	dotnet	🔴**Critical Risk**
Serilog	2.10.0	**4.3.1-dev-02387**	🔴**Critical**	dotnet	🔴**Critical Risk**
Microsoft.EntityFrameworkCore	5.0.0	**10.0.0-rc.1.25451.107**	⚫**Unknown**	dotnet	⚫**Unknown Risk**

Elixir

Package	Current	Latest	Threat Level	Registry	Security Impact

phoenix	~> 1.5.9	**1.8.1**	🟠High	elixir	🟠**High Risk**
ecto	~> 3.6	**3.13.3**	🟠High	elixir	🟠**High Risk**
plug_cowboy	~> 2.4.1	**2.7.4**	🟠High	elixir	🟠**High Risk**

🔴**Critical Priority** (9 packages require update)
* Immediate attention required to mitigate security and operational risks.
* Upgrade all dependencies with major version changes to maintain compatibility and stability.
* Prioritize updates with known security vulnerabilities or active risk exposure.
* Conduct comprehensive validation testing to ensure application integrity post-upgrade.
* Confirm seamless integration with production workloads before deployment.
* Identify and communicate breaking changes to all impacted stakeholders.

🟠**High Priority** (10 packages require update)
* Timely action recommended to maintain system health and reduce long-term risk
* Organize minor and patch updates by technology stack or ecosystem for efficiency.
* Schedule updates to development dependencies during off-peak hours to minimize disruption.
* Perform thorough regression testing following updates to detect functional issues.
* Utilize automated compatibility checks to identify safe versions.
* Maintain rollback strategies in case of failure or instability.
* Integrate updates into the CI/CD pipeline for structured and reliable deployment.
* Monitor update cadence to prevent accumulation of technical debt.

🟡**Medium Priority** (3 packages require update)
* Recommended for near-term action during planned maintenance cycles
* Schedule implementation during the next approved maintenance window.
* Communicate planned updates to all relevant stakeholders.
* Review release notes to ensure compatibility with existing systems.
* Deploy updates in a staging environment for pre-production testing.
* Confirm current system backups are available before applying updates.
* Monitor system performance and logs for anomalies post-deployment.

🟢**Low Priority** (3 packages are up to date)
* To be addressed as part of standard patch management processes
* Include in the organization's regular patching cycle.
* Maintain a log of outdated packages for audit and compliance purposes.
* Track vendor support timelines to avoid unexpected end-of-life scenarios.
* Evaluate the operational impact prior to rollout.
* Where feasible, bundle with other low-risk updates to optimize resource use.
* Reassess the priority level during the next vulnerability management review.

🛡Security Measures
* Conduct security audit of all dependencies
* Implement monitoring for vulnerability alerts
* Restrict access to affected systems
* Document all changes for compliance

📅Next Steps
* Review the changes required for each update
* Create update branches for testing
* Run comprehensive tests
* Update documentation
* Schedule updates during maintenance window

⚠Security Disclaimer

Your system has been identified as having **Critical Security Vulnerabilities**.
Immediate action is required to prevent potential:

- Data Breaches
- System Compromise
- Production Failures
- Compliance Violations

This report was generated by automated security scanning. Manual verification and immediate remediation are Required.

Generated by ProSecureLabs Package Scanner | 2025-10-12 06:08:51

Confidential Security Report - For authorized personnel only