

How to Reduce Your Alert Count Early in Development

Mend remediation and prioritization technologies help turn 104 alerts into 7

As organizations struggle to shift security left by integrating security testing tools early in the development process, many focus on vulnerability detection. While detection is an important step toward achieving DevSecOps maturity, security and development teams are now faced with a seemingly never-ending stream of security alerts and little to no means of prioritizing and remediating them.

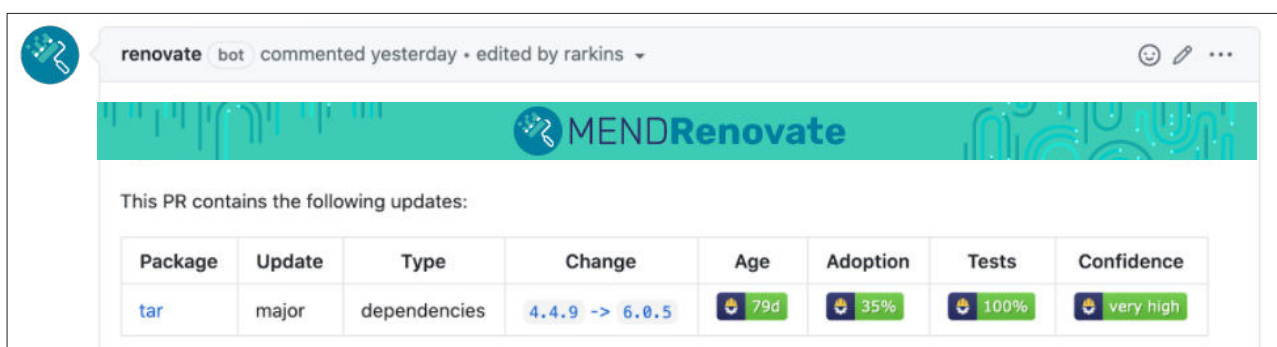
Mend's remediation-centric approach goes beyond detection to cut through the noise and zero in on security vulnerabilities that actually impact code. Using Mend, organizations significantly reduce the number of security alerts that need to be addressed.

This remediation-centric approach helps teams shift security left and fix security issues early in development without interrupting developers' workflows or delaying the release cycle. Mend's prioritization and remediation capabilities integrate into developers' native environments, and provide insights and auto-updates when they need them.

Mend Renovate

Mend Renovate works with package managers to help developers keep their open source dependencies up to date and remediate vulnerabilities within these libraries.

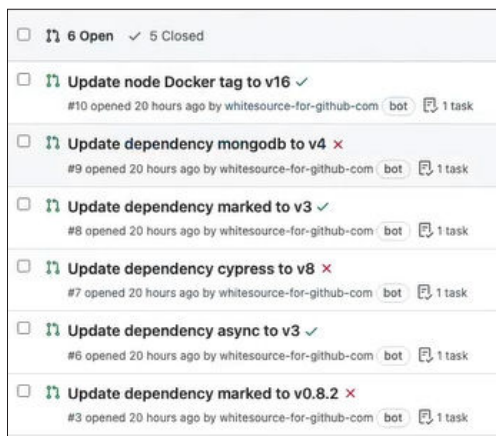
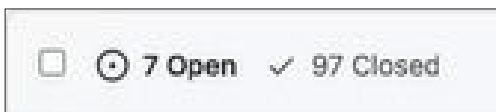
When an outdated version of an open source dependency is detected in a package, Renovate automatically generates a fix pull request with the latest version update. It is added to the repository along with release notes so that users can compare the versions and merge the fix with one click.



Package	Update	Type	Change	Age	Adoption	Tests	Confidence
tar	major	dependencies	4.4.9 -> 6.0.5	79d	35%	100%	very high

Using the fix pull requests helps developers significantly reduce the number of open source vulnerability alerts quickly, early in the development process.

In this JavaScript project on GitHub, Mend scanned for vulnerable versions on every commit. The issues tab – available for developers when they need it – shows that the first time running the scan, the project had a total **104 open issues**.



Upon running Renovate, fix pull requests were suggested to remediate the issues. A checkmark indicates which dependencies can be updated with ease:

Mend Merge Confidence

Mend Merge Confidence addresses one of the top barriers to dependency hygiene: keeping dependencies up to date. Merge Confidence shows whether a new version release has a breaking change, so that developers can update versions based on crowd-sourced data confirming that others have upgraded without problems, before making the jump themselves.

Every Mend Remediate pull request comes with a Merge Confidence rating, so developers know whether they can update the new release with ease:

Version	Age	Tests	Confidence
8.1.3	age 18 days	passing 100%	confidence high
8.1.4	age 17 days	passing 98%	confidence high
8.1.5	age 4 days	passing 74%	confidence low
8.1.6	age 4 days	passing 95%	confidence neutral

In this example, in addition to providing fix pull requests for detected vulnerabilities, Merge Confidence provides extra remediation support by letting developers know which of the vulnerable versions can be updated without breaking the build.

Once updated, the project has **reduced the number of alerts by 89**, leaving the developer with **only 15 remaining alerts** out of the original 104.

Mend Prioritize

Mend Prioritize enables development and security teams to further reduce the number of security alerts they need to address so that they can focus on fixing the most urgent issues first.

Mend Prioritize

Mend's patent-pending technology scans open source components with known vulnerabilities to determine which vulnerabilities directly impact your code. The advanced algorithms yield accurate results and minimize false positives.

Mend Prioritize also enables quicker remediation by providing a detailed trace analysis to help developers understand how they are using the vulnerable functionalities, enabling quicker remediation.



Red - An effective vulnerability has been found in your open-source code, demanding urgent remediation steps.



A vulnerability of unconfirmed effectiveness has been found in your open-source code; it should be treated as an effective (red) vulnerability.



Not an effective vulnerability and therefore does not require urgent remediation.



Vulnerability-related information changed for the scanned library after it was analyzed by WhiteSource Prioritize. Run the scan again for an updated report.

In our example, Mend Prioritize was used for the remaining fifteen alerts to determine which ones required immediate attention. The red shield alert indicates a security vulnerability that affects the software project and requires remediation.

A screenshot of a list of security alerts from Mend. Each alert entry includes a checkbox, a severity level in parentheses, the vulnerability name, the affected package name, and a red shield icon with the text 'security vulnerability'. The alerts are:

- WS-2019-0027 (Medium) detected in marked-0.3.9.tgz security vulnerability
- CVE-2021-23358 (High) detected in underscore-1.9.1.tgz security vulnerability
- WS-2019-0311 (Medium) detected in mongodb-2.2.36.tgz security vulnerability
- WS-2020-0163 (Medium) detected in marked-0.3.9.tgz security vulnerability
- WS-2018-0628 (Medium) detected in marked-0.3.9.tgz security vulnerability
- CVE-2020-7610 (High) detected in bson-1.0.9.tgz security vulnerability
- Dependency Dashboard

Prioritize also presents the trace call, which provides developers with the exact location of the vulnerable function, showing them where it starts in the application code and where it goes into the vulnerable library, so that they can easily locate the vulnerability.

A screenshot of the 'Trace View' interface in Mend. It shows a call path for CVE-2017-1000487. The path starts at the 'ORIGIN' (customerCode.UseLib7\$SampleInvocationHandler), goes through a 'PATH (2 points)' (customerCode.UseLib2[customerCode.UseLib2:Indirect reference]>> and customerCode.UseLib5[customerCode.UseLib5:Indirect reference]), and ends at the 'TARGET' (org.codehaus.plexus.util.cli.shell.Shell). Dashed blue arrows indicate the flow from origin to path and then to target.

Quick and Easy Remediation

Using Mend's remediation-centric tools helped minimize the time spent on researching and fixing 97 security alerts, by providing automated version updates, insights on their stability, and trace analysis to determine that only seven of the issues actually require immediate action.

Using Mend to Reduce 104 Alerts to 7

Mend helps you remediate vulnerabilities early in development. Here's how we reduce the alert count from 104 to 7.

