# Auditing security alerts

**In this article**



GitHub provides a variety of tools you can use to audit and monitor actions taken in response to security alerts.

## About security tools for auditors

GitHub provides tools for security auditors and developers to review and analyze responses to security alerts within an enterprise or organization. This guide describes the tools, which include historical timelines, security overview, audit logs, the API, and webhooks.

Security auditors can use these tools to ensure the appropriate actions are being taken to resolve security alerts and to identify areas for additional training. Developers can use these tools to monitor and debug their own security alerts. You will only see data for repositories and organizations to which you already have access.

## Security alert timelines

Each security alert has a historical timeline that shows when the alert was created or when a problem was detected. When the status of an alert changes this is recorded on the timeline, regardless of what caused the change, for example, Dependabot closing a fixed alert and a developer reopening an alert. You can see the historical timeline for an alert on the alert page under the description of the problem.

Many of the events in the timeline also create an event in the audit log, which you can query using the audit log UI or the API. For more information, see "Audit log."

## Security overview page

Security overview consolidates information about security alerts and provides high-level summaries of the security status of your enterprise or organization.

In security overview you can see repositories with open security alerts. You can also use security overview to filter and sort security alerts using interactive views.

For more information, see "About security overview."

## Audit log

You can access and search audit logs using the API or the audit log UI. The audit log lists events that are triggered by activities affecting your enterprise or organization, including events that are created when there are certain interactions with a security alert. Interactions that create an event can be triggered manually or by automation, for example, when Dependabot creates an alert.

- Secret scanning events track when an alert is created, resolved, or reopened, also when push protection is bypassed.
- Dependabot events track when an alert is created, dismissed, or resolved.
- Code scanning does not create timeline events in an audit log.

For a list of audit log events, see "Audit log events for your enterprise" and "Audit log events for your organization."

## Webhooks

You can set up `code_scanning_alert`, `dependabot_alert`, and `secret_scanning_alert` webhooks to receive payloads whenever there is a response to a security alert in an organization or repository. You can also define which responses to act on, for example, you might want to define a webhook that tracks secret scanning alerts created when someone bypasses push protection using the alert property `"push_protection_bypassed": true`.

You can also integrate webhook payloads into other tools you use to monitor and inform security behaviors. For example, a webhook fires when a secret alert is either created, resolved, or reopened. You can then parse the webhook payload and integrate it into tools your team uses like Slack, Microsoft Teams, Splunk, or email. For more information, see "About webhooks" and "Webhook events and payloads."

## API

You can use the API to list and interact with security alerts, for example, getting the most recent information about updates or dismissals of an alert. You can also use the API to make additional updates to the alert or to automate follow-up actions, such as creating a new issue for each alert that needs further action. Only the current status of an alert is reported by the API.

### Dependabot alerts API

You can list all Dependabot alerts for a repository, organization, or enterprise, or use path parameters to list only alerts that meet a specific set of criteria. For example, you might only want to list Dependabot alerts for Maven that were dismissed. Alternatively, you can get full details for an alert or update the alert.

For more information, see "Dependabot alerts."

### Secret scanning alerts API

You can list all secret scanning alerts for a repository, organization, or enterprise, or use path parameters to list only alerts that meet a specific set of criteria. Alternatively, you can get full details for an alert or update the alert.

To see which secret scanning alerts were the result of a push protection bypass, filter the results for `"push_protection_bypassed": true`.

For more information, see "Secret scanning."

### Code scanning alerts API

You can list all code scanning alerts for a repository, organization, or enterprise, or use path parameters to list only alerts that meet a specific set of criteria. Alternatively, you can get full details for an alert or update the alert.

For more information, see "Code scanning."

## Further reading

- Managing code scanning alerts for your repository
- Viewing and updating Dependabot alerts
- Managing alerts from secret scanning