GitHub Adds 48-Hour Expiry to Unvalidated npm Trusted Publishing Setups



GitHub now expires unvalidated npm trusted publishing configurations 48 hours after creation, following research showing an attacker could publish new versions of another user's package without stealing credentials.

- GitHub expires unvalidated npm trusted publishing configurations 48 hours after they are created.
- Researchers found an attacker could push new versions of someone else's package without stealing credentials.
- A configuration becomes validated and exempt from expiry after its first successful publish.
- The headline reports more than 1,500 packages were left vulnerable to hijacking.

GitHub has changed how npm trusted publishing works after researchers showed that a flaw allowed an

attacker to publish new versions of a package belonging to a different user, without needing to steal that user's credentials.

What changed

Under the new rule, a trusted publishing configuration that has not been validated expires 48 hours after it is created. GitHub states that a configuration becomes validated, and exempt from expiry, once its first publish succeeds. The source's headline says more than 1,500 packages were left exposed to hijacking by the flaw.

Why it matters

Trusted publishing is meant to remove long-lived tokens from release pipelines. A weakness in that mechanism is therefore significant, because a malicious version could reach downstream users through a normal update, with no credential theft to detect. Organisations that depend on npm packages, directly or through their vendors, inherit that exposure.

The source excerpt does not describe how the flaw was exploited, whether it was abused in the wild, or which packages were affected. Teams should avoid assuming either way and instead verify their own configurations and dependencies.

Practical view

The 48-hour expiry narrows the window in which a pending, unvalidated configuration can be misused. It does not replace review of existing setups. Maintainers should confirm that every trusted publishing entry is intentional and has completed a legitimate first publish.

What to do now

1. Review all npm trusted publishing configurations on packages your organisation publishes and remove any you do not recognise.
2. Confirm each configuration has completed a legitimate first publish and is still needed.
3. Check recent releases of your own published packages for versions your team did not author or approve.
4. Ask software vendors and internal teams whether they use npm trusted publishing and how they are responding.
5. Pin dependency versions and review changes in lockfiles before promoting updates to production.