CSRF allows attacker to finalize/unfinalize order adjustments in solidus_backend

Low **waiting-for-dev** published **GHSA-8639-qx56-r428** on Jun 1

Package

solidus_backend (RubyGems)

Affected versions	Patched versions
< 3.1.6, <3.0.6, <2.11.16	3.1.6, 3.0.6, 2.11.16

Description

Impact

CSRF vulnerability allowing attackers to change the state of an order's adjustments if they hold its number, and the execution happens on a store administrator's computer.

Reproduction steps:

- Take an order's number.
- Log in as an administrator.
- Visit that order's adjustments section (*Orders -> {Click on number} -> Adjustments*) and check that its adjustments are finalized (closed padlock under the **State** column).
- On another tab, visit `{your_site_url}/admin/orders/{order_number}/adjustments/unfinalize`.
- Notice how the adjustments are unfinalized (open padlock), even if the previous was a `GET` request which could have been linked from any other site.
- Visit `{your_site_url}/admin/orders/{order_number}/adjustments/finalize`.
- Notice how the adjustments are again finalized.

That happened because both routes were handled as `GET` requests, which are skipped by Rails anti-forgery protection.

Patches

Users should upgrade to solidus_backend v3.1.6, v3.0.6, or v2.11.16, depending on the major and minor versions in use.

References

- Rails CSRF protection.

For more information

If you have any questions or comments about this advisory:

- Open an issue or a discussion in Solidus.
- Email us at security@solidus.io
- Contact the core team on Slack

Severity

Low

CVE ID

CVE-2022-31000

Weaknesses

CWE-352