The jsoup cleaner may incorrectly sanitize crafted XSS attempts if SafeList.preserveRelativeLinks is enabled

Moderate jhy published GHSA-gp7f-rwcx-9369 on Aug 23

Package

org.jsoup:jsoup (Maven)

Affected versions	Patched versions
< 1.15.3	1.15.3

Description

jsoup may incorrectly sanitize HTML including `javascript:` URL expressions, which could allow cross-site scripting (XSS) attacks when a reader subsequently clicks that link. If the non-default `SafeList.preserveRelativeLinks` option is enabled, HTML including `javascript:` URLs that have been crafted with control characters will not be sanitized. If the site that this HTML is published on does not set a Content Security Policy, an XSS attack is then possible.

Impact

Sites that accept input HTML from users and use jsoup to sanitize that HTML, may be vulnerable to cross-site scripting (XSS) attacks, if they have enabled `SafeList.preserveRelativeLinks` and do not set an appropriate Content Security Policy.

Patches

This issue is patched in jsoup 1.15.3.

Users should upgrade to this version. Additionally, as the unsanitized input may have been persisted, old content should be cleaned again using the updated version.

Workarounds

To remediate this issue without immediately upgrading:

- disable `SafeList.preserveRelativeLinks`, which will rewrite input URLs as absolute URLs
- ensure an appropriate Content Security Policy is defined. (This should be used regardless of upgrading, as a defence-in-depth best practice.)

Background and root cause

jsoup includes a Cleaner component, which is designed to sanitize input HTML against configurable safe-lists of acceptable tags, attributes, and attribute values.

This includes removing potentially malicious attributes such as `<a href="javascript:...">`, which may enable XSS attacks. It does this by validating URL attributes against allowed URL protocols (e.g. `http`, `https`).

However, an attacker may be able to bypass this check by embedding control characters into the href attribute value. This causes the Java URL class, which is used to resolve relative URLs to absolute URLs before checking the URL's protocol, to treat the URL as a relative URL. It is then resolved into an absolute URL with the configured base URI.

For example, `java\tscript:...` would resolve to `https://example.com/java\tscript:...`.

By default, when using a safe-list that allows `a` tags, jsoup will rewrite any relative URLs (e.g. `/foo/`) to an absolute URL (e.g. `https://example.com/foo/`). Therefore, this attack attempt would be successfully mitigated. However, if the option SafeList.preserveRelativeLinks is enabled (which does not rewrite relative links to absolute), the input is left as-is.

While Java will treat a path like `java\tscript:` as a relative path, as it does not match the allowed characters of a URL spec, browsers may normalize out the control characters, and subsequently evaluate it as a `javascript:` spec inline expression. That disparity then leads to an XSS opportunity.

Sites defining a Content Security Policy that does not allow javascript expressions in link URLs will not be impacted, as the policy will prevent the script's execution.

For more information

If you have any questions or comments about this advisory:

- Open an issue in jsoup
- Email the author of jsoup at jonathan@hedley.net

Credits

Thanks to Jens Häderer, who reported this issue, and contributed to its resolution.

Severity

Moderate 6.1 / 10

CVSS base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	None
User interaction	Required
Scope	Changed
Confidentiality	Low
Integrity	Low
Availability	None

CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:L/I:L/A:N

CVE ID

CVE-2022-36033

Weaknesses

CWE-87