HSEC-2024-0004

See a problem?

Import Source	https://github.com/haskell/security-advisories/blob/generated/osv-export/2024/HSEC-2024-0004.json
JSON Data	https://api.osv.dev/v1/vulns/HSEC-2024-0004
Aliases	CVE-2026-40470
Published	2026-01-16T11:18:20Z
Modified	2026-04-14T18:00:10.355627Z
Summary	Hackage package and doc upload stored XSS vulnerability

Details

Hackage package and doc upload stored XSS vulnerability

Author: Fraser Tweedale (Haskell SRT)

Executive summary

A **critical XSS vulnerability** affected *hackage-server* and `hackage.haskell.org`. HTML and JavaScript files provided in source packages or via the documentation upload facility were served *as-is* on the main `hackage.haskell.org` domain. As a consequence, when a user with latent HTTP credentials browses to the package pages or documentation uploaded by a malicious package maintainer, their **session can be hijacked** to upload packages or documentation, amend maintainers or other package metadata, or perform any other action the user is authorised to do.

This issue was discovered and reported to the Haskell Security Response Team (SRT) in June 2024. Remediations were progressively developed and deployed throughout 2025. **The known issues have been fully mitigated on `hackage.haskell.org` as of January 2026.**

The mitigations involve serving user content from a sister domain. In the case of `hackage.haskell.org`, this domain is `hackage-content.haskell.org`.

Operators of other *hackage-server* instances should update to the `master` branch from the upstream repository (https://github.com/haskell/hackage-server; commit `9a1887607d9b8d1a3b8b02c990ee144c0d402b79` or later), and configure the user content domain (new flag `--user-content-uri`).

Scope of the issue

Package documentation uploads

Any Hackage user can upload documentation bundles for the packages they own. Typically those are prepared by Haddock, but they may be customised or modified. Malicious code can be delivered in several ways:

- In JavaScript files loaded by an HTML page;
- In `<script>` tags on an HTML page;
- In action attributes (e.g. `onclick`) on an HTML page.

`quick-jump` JavaScript

In addition to the supplied HTML pages, parts of the uploaded documentation bundle are referenced from the main package page (and candidate pages)—in particular, `quick-jump.min.js`, which implements the "quick jump" feature, and the corresponding CSS file (CSS can also present an XSS risk). These files are included by the following templates:

- `datafiles/templates/Html/package-page.html.st`
- `datafiles/templates/Html/candidate-page.html.st`

It is loaded via a `<script>` tag:

```
<script src="$doc.baseUrl$/quick-jump.min.js" type="text/
```

Source tarball browsing

HTML content in the source package is served as `text/html` or `application/xhtml+xml`, depending on file extension. The browser renders the page and executes scripts without restriction. The same scope of exploitation arises as for documentation uploads.

Impact

Malicious scripts can use latent HTTP credentials in the browser to perform any action the user is authorised to do, **without user interaction**. This could include uploading new package versions, editing packaging metadata, adding new maintainers, and more.

Additionally, malicious scripts and HTML could present a counterfeit login form to the user, tricking the user into divulging their credentials. The script can then abuse the credentials in the same way as the latent credential scenario.

- CVSS vector string: **CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L**
- CVSS score: **9.9 (Critical)**

Mitigations

This section describes the code and configuration changes to mitigate the issues.

User content domain

hackage-server was updated to serve user-uploaded content from a separate DNS domain. This includes package documentation (excluding the main package description) and source tarball browsing, for both published and candidate packages.

For the relevant resources, requests on the main domain are redirected to the user content domain via an HTTP 301 ("Moved Permanently") response status.

Doc and source tarballs uploaded prior to the deployment of these mitigations are exempted from the redirection, and remain accessible via the main domain (this may be changed in the future).

Alternative approaches considered:

- Serving HTML from the source tarball with `Content-Type: text/plain` would prevent the browser from interpreting them as HTML and executing inline or referenced scripts. But several packages have custom doc bundles uploaded, offering a rich documentation experience to their users. Therefore, changing this behaviour would surprise and probably disappoint many users.

quick-jump: reject unrecognised JavaScript and CSS

A small number of versions of `quick-jump.min.js` and `quick-jump.css` have appeared in official releases of of the Haddock documentation builder program. *hackage-server* matches the SHA-256 digests of the files in the

doc bundle against the list of "known good" files. If there is no match, *hackage-server* refuses to serve the file (403 Forbidden).

The digest lists are currently hard-coded; the initial values are mentioned at the end of this document. The list needs to be updated whenever the official Haddock quick-jump implementation changes.

Alternative approaches considered:

- Using *Subresource Integrity (SRI)* to limit to known-good hashes. This would require the same amount of development and maintenance overhead whilst depending on browser implementation for enforcement.
- Ignoring the quick-jump implementation from the doc bundle and serving a single "golden" implementation for all packages. Because quick-jump reads the generated `doc-index.json` this approach ran the risk of breaking the feature on older (or newer) uploads if the format of the file changes. Formally specifying the `doc-index.json` would allow the quick-jump feature to be evolved and improved to bring benefits to all packages and reduce security risks.

There was considerable debate between the digest allow-list and "golden" approaches. The *hackage-server* maintainers had the final say, and the SRT is satisfied that the chosen approach resolves the security issues.

Acknowledgements

- **Zubin Duggal** reported the issue in documentation uploads.
- **Duncan Coutts** noticed that the issue also affects source tarball browsing.
- **Gershom Bazerman** and **Janus Troelsen** implemented the mitigations, and Gershom handled redeploys of the `hackage.haskell.org` service.
- **Fraser Tweedale** verified and analysed the issue, and coordinated the security response.

Appendix A.1. Known-good variants of `quick-jump.min.js`

This appendix records known variants of `quick-jump.min.js` from the GHC and Haddock repositories. The commit ID, Git object (short) ID, and SHA-256 digests are noted. The SHA-256 digests can be used with Subresource Integrity (SRI) or server-side checks to prevent delivery or execution of unrecognised script content.

From `https://gitlab.haskell.org/ghc/haddock.git` – `master` branch

```
commit: e99aefb50ca63e2dbcc95841efbb53cea90151d8 (Sep 23 2
object: c9f2b445b9
sha256: e1da96b0d7ab3d72cfe3786def923c5af91ba331858852f1f4

commit: 8e88615a23a9f1980a55bd1b3ef9dcc938d95237 (Oct 10 2
object: cb24f8bdea
sha256: a273a3ef19c21032afc5f65d1e09933146f183da906ca9d0b4

commit: b4982d87f41d9a4d3f6237bacfd819145723e35b (Oct 30 2
object: f22f8f2881
sha256: 8aed621ac2b746751585cbe271631394cacc0e01cca4ef589

commit: 93c1e6eb9e829a66ff213ec076d529ab008880b3 (Dec 16 2
object: bfdf04a372
sha256: 4b10c18a7ad35f032e8cdc0d263716a93878bf06d998b1b66

commit: 59812a09eb69cbf12407206381f4c214987b1efd (Apr 3 20
object: c03e083607
sha256: ce86bba43edb0534c0faa2d6d0f504877576c5271321e3fbd9

commit: a69311708493efe8524aed0e9d19365f79f2fab3 (Oct 24 2
object: 06c35c7454
sha256: 548d676b3e5a52cbfef06d7424ec065c1f34c230407f9f5dc0
```

From `https://gitlab.haskell.org/ghc/ghc.git` – `master` branch

Note: all of the *objects* listed above also occur in the GHC repo, but the commit history is different (as expected).

```
commit: 7776566531e72c415f66dd3b13da9041c52076aa (Nov 13 2
object: 0b0eeb27d1
sha256: 7ca43fc2058574846e032bc5493a0ad4568e4fa14fb58558f

commit: 14e554cf682ae975ba356b07672c0f9d465e2a78 (May 24 2
object: 06c35c7454
sha256: (seen already)
```

Appendix A.2. Known-good variants of `quick-jump.css`

This appendix records known variants of `quick-jump.css` from the GHC and Haddock repositories. The commit ID, Git object (short) ID, and SHA-256 digests are noted. The SHA-256 digests can be used with Subresource Integrity

(SRI) or server-side checks to prevent delivery or use of unrecognised CSS content.

From `https://gitlab.haskell.org/ghc/haddock.git` — `master` branch

No changes since integration of haddock into the ghc repo.

From `https://gitlab.haskell.org/ghc/ghc.git` — `master` branch

```
commit: 9511e587701349093cbe3ac7c00f13583820774f (Feb 7 2
object: cf10eee4
sha256: 6bd159f6d7b1cfef1bd190f1f5eadcd15d35c6c567330d746

commit: 05ccce6e07731f9788a434d6e06f4cadeff3d6ba (Dec 8 2
object: d656f51c
sha256: 6997c223e09b340f5f1bb970c930b458f768a0bbbe787cb87

commit: fa5ec121e2a700137bab8bd48cc30b1e80f58fd4 (Feb 27
object: 8772809c
sha256: 29fe483bd37ad3feba12f646e966173112752 6f246c246b00

commit: fc069bf200f930c21f96ddbbec1d7c5c69f8ba72 (Jan 15
object: 468d8036
sha256: 1d51573b72bc8a7b9b0dda3beffb7882db78d22a37840203f

commit: 0997eb61803a37803ddb6cf7116eb9db1046b2ce (Oct 10
object: ede05042
sha256: 59693ef3f0d793031b3af58b214af7884c0f63ce6db659ffd

commit: d41abb0f606bf5fdbdc0a7bd3758e0c30601b121 (Sep 23
object: b69903c3
sha256: f95b8b12a8a13dd31add93527e1239fdff6997c7f2396e975
```

Database specific


```
{
    "home": "https://github.com/haskell/security-advisories",
    "osvs": "https://raw.githubusercontent.com/haskell/security-advisories/refs/heads/generated/osv-export",
    "repository": "https://github.com/haskell/security-advisories"
}
```


References

https://github.com/haskell/hackage-server/commit/5a0dc3357b042502fcf684ed7e2464fde1407809

https://github.com/haskell/hackage-server/commit/33d9ea3de9a298b21b53ba23386af2742384e627

https://github.com/haskell/hackage-server/commit/354a7de3b7f21ad1346677df515d8e8ec75016e6

Affected packages

Hackage
hackage-server

Package

Name hackage-server

Purl pkg:hackage/hackage-server

Severity

9.9 (Critical) CVSS_V3 - CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:C/C:H/I:H/A:L CVSS Calculator

Affected ranges

Type ECOSYSTEM

Events
Introduced
0.1
Fixed
0.6

Affected versions

▸ 0.*

Database specific

▸ source

▸ osv

▸ human_link
