

PrefectHQ / prefect Public

<> Code Issues 818 Pull requests 39 Discussions Actions Security and

Fix git argument injection in GitRepository pull steps #21384

Merged **desertaxle** merged 3 commits into `main` from `devin/OSS-7818-1775071078` on Apr 2

Conversation 11 Commits 3 Checks 90 Files changed 2

**devin-ai-integration** Bot commented on Apr 1 • edited

Contributor

Fixes a git argument injection vulnerability in `GitRepository` where user-controlled `commit_sha` and `directories` values are passed directly as positional arguments to git subprocesses. A malicious `commit_sha` like `--upload-pack=touch /tmp/pwned` could achieve RCE on worker machines processing deployments.

Changes:

- Validate `commit_sha` in `__init__` against `^[0-9a-fA-F]{4,64}$` (hex, 4–64 chars) — rejects flag injection before any subprocess runs. Lower bound of 4 matches git's minimum abbreviation; upper bound of 64 supports both SHA-1 (40 chars) and SHA-256 (64 chars) repositories.
- Emit a `UserWarning` for `directories` entries starting with `--` (non-breaking; the `--` separator in the git command still protects against injection)
- Add `--` end-of-options separator to both `git sparse-checkout set` call sites
- Invalid `commit_sha` error message now suggests using the `branch` parameter for non-hex values (tag names, branch names, etc.)

Why `--` is only added to `sparse-checkout set` : The `--` separator tells git "everything after this is a path." For `sparse-checkout set`, `directories` are paths so this works. For `git fetch origin <sha>`, `git checkout <sha>`, and `git rev-parse <sha>`, adding `--` causes git to treat the SHA as a path instead of a ref, breaking functionality. Input validation in `__init__` is the defense for those commands.

Checklist for human review

- Verify the SHA regex `^[0-9a-fA-F]{4,64}$` is sufficient as the sole defense for `fetch / checkout / rev-parse` (no `--` separator for those)
- Confirm that warning (not rejecting) `--`-prefixed directories is acceptable given the `--` separator in the `sparse-checkout` command

- Note: this intentionally does not accept tag names or branch names via `commit_sha` — those should use the `branch` parameter
- Note: `branch` parameter is not validated in this PR (out of scope)

Checklist

- This pull request references any related issue by including "closes `<link to issue>`"
- If this pull request adds new functionality, it includes unit tests that cover the changes
- If this pull request removes docs files, it includes redirect settings in `mint.json`.
- If this pull request adds functions or classes, it includes helpful docstrings.

Link to Devin session: <https://app.devin.ai/sessions/a0c975e4e6214bdd8a855d7c959213c9>

Requested by: [@desertaxle](#)

[Fix git argument injection in GitRepository pull steps](#) ✓ [7be7953](#)

devin-ai-integration (Bot) assigned **desertaxle** on Apr 1

devin-ai-integration (Bot) commented on Apr 1

Contributor Author

Devin AI Engineer

I'll be helping with this pull request! Here's what you should know:

I will automatically:

- Address comments on this PR. Add '(aside)' to your comment to have me ignore it.
- Look at CI failures and help fix them

Note: I can only respond to comments from users who have write access to this repository.

Control Options:

- Disable automatic comment and CI monitoring

codspeed-hq (Bot) commented on Apr 1 • edited ▾



Merging this PR will not alter performance


2 untouched benchmarks

Comparing devin/OSS-7818-1775071078 (e6cf764) with main (b9c6609)

 [Open In CodSpeed](#)

  **desertaxle** marked this pull request as ready for review [last month](#)

  **desertaxle** requested review from **chrisguidry**, **cicdw**, **desertaxle** and **zzstoatzz** as [code owners](#) [last month](#)

 **chatgpt-codex-connector** Bot reviewed [on Apr 1](#)
[View reviewed changes](#)

 **chatgpt-codex-connector** Bot left a comment

Codex Review

Here are some automated review suggestions for this pull request.

Reviewed commit: 7be79533cc

▶  [About Codex in GitHub](#)

> `src/prefect/runner/storage.py` Outdated  [Show resolved](#)

  [widen SHA validation to accept SHA-256 \(64-char\) commit IDs](#) ... ✓ [6632d62](#)

desertaxle commented [on Apr 1](#) Member

[@codex](#) review

 1

 **chatgpt-codex-connector** Bot reviewed [on Apr 1](#)

[View reviewed changes](#)



chatgpt-codex-connector (Bot) left a comment



Codex Review

Here are some automated review suggestions for this pull request.

Reviewed commit: 6632d624df

▶ [About Codex in GitHub](#)

src/prefect/runner/storage.py (Outdated)

```

176 | 177 | "Cannot provide both a branch and a commit SHA. Please provi
177 | 178 | )
178 | 179 |
+ 180 | + if commit_sha and not re.match(r"^[0-9a-fA-F]{7,64}$", commit_sha):

```



chatgpt-codex-connector (Bot) on Apr 1

P2 Accept git-valid short commit IDs

The new `commit_sha` validation only allows 7–64 hex chars, which now rejects 4–6 character abbreviated SHAs that Git can resolve when unique. Any deployment or pull step currently passing a short SHA (for example from CI configurations that emit 6-char short SHAs) will now fail immediately during `GitRepository` construction, so this is a backward-incompatible regression introduced by the hardening change.

Useful? React with 👍 / 👎.

👁️ 1



devin-ai-integration (Bot) on Apr 1

Contributor

Author

Git's default `core.abbrev` is 7 characters, and all common Git hosting platforms (GitHub, GitLab, Bitbucket) emit SHAs of 7+ chars. Accepting shorter SHAs would weaken the security boundary — a 4-char hex string provides very little assurance it's actually a commit hash vs. an accidental match. If a real use case for shorter SHAs comes up, the minimum can be lowered, but 7 is the safe default.

src/prefect/runner/storage.py

```

+ 185 | +
+ 186 | + if directories:

```

```

187 +         for d in directories:
188 +             if d.startswith("--"):

```

chatgpt-codex-connector Bot on Apr 1

P3 Stop rejecting valid sparse-checkout path names

After adding `--` to `git sparse-checkout set`, arguments are already safely delimited as paths, but this new guard still rejects directory names that begin with `--`. That blocks valid repositories containing such directory names (for example a literal folder named `--config`) and causes a `ValueError` even though Git can now handle those paths safely with the separator.

Useful? React with 👍 / 👎.

1

devin-ai-integration Bot on Apr 1 Contributor Author

The directory validation is intentional defense-in-depth. While `--` in the sparse-checkout command does protect at the git CLI level, rejecting `--`-prefixed directories at init time catches injection attempts earlier and with a clearer error message. Directories literally named `--config` etc. are vanishingly rare in real repos. Happy to remove this guard if a maintainer prefers relying solely on the `--` separator.

Lower SHA min to 4 chars, warn instead of raise for -- directories ... e6cf764

desertaxle approved these changes on Apr 2

[View reviewed changes](#)

desertaxle merged commit **6a9d991** into `main` on Apr 2

[View details](#)

154 of 157 checks passed

desertaxle deleted the `devin/OSS-7818-1775071078` branch [last month](#)

Sign up for free
 to join this conversation on **GitHub**. Already have an account? [Sign in to comment](#)

Reviewers

chatgpt-codex-connector[bot]



 desertaxle	✓
 cicdw	🛡️ ●
 zzstoatz	🛡️ ●
 chrisguidry	🛡️ ●

Assignees

 **desertaxle**

Labels

None yet

Projects

None yet

Milestone

No milestone

Development

Successfully merging this pull request may close these issues.

None yet

1 participant

