0-DAY NO CVE HIGH PUBLIC

Forgejo, restore-repo file:// os.Open

Forgejo 16.0.5 restore-repo rewrites release asset URLs with raw filepath.Join and prefixes file://. uri.Open still os.Open's that path. Relative .. walks out of the dump. Not live-migrate LFI. No CVE yet.

NAME	TYPE
Forgejo, restore-repo file:// os.Open	0-day analysis
CVE	**CVE RISK**
n/a	high
DISCLOSURE STATUS	**VENDOR**
public	Forgejo
AFFECTED	**PUBLISHED**
Forgejo through v16.0.5 (8d24b7d); unpublished. Local operator forgejo restore-repo on an untrusted dump. Absolute DownloadURL dump-root escape does not land. Gitea localFileURL is not in this tree.	12 Oct 2026
UPDATED	**TAGS**
12 Oct 2026	0-day, forgejo, lfi, restore, cwe-22, cwe-73, authenticated

TL;DR

Lab files and proof-of-concept **here**.

01 file:// still calls os.Open

I am @abraxas_null. The proof of concept is on GitHub: **abraxas/forgejo-restore-file-open** (loopback client). The lab stack is `lab/`: Dockerfile, docker-compose.yml, run.sh, poc.py. Authorized lab only. It talks to loopback.

This is **Forgejo v16.0.5** (`8d24b7d52df327118cd776cf461d34e52e0ec1ce`). No CVE yet. CWE-22 / CWE-73. 7.3 High (`CVSS:3.1/AV:L/AC:L/PR:N/UI:R/S:C/C:H/I:N/A:N`). Local file read. Operator restore.

`forgejo restore-repo` of an untrusted dump rewrites every release asset `download_url` with `filepath.Join(baseDir, value)` and prefixes `file://`. `uri.Open` still has a file branch that `os.Open`s that path. There is no containment under the dump. Gitea's `localFileURL` helper is not in this tree.

02 Why Forgejo

I had two restore ideas on Gitea 1.27.3. Absolute join failed. Dump-side symlink succeeded. Forgejo never grew `OpenWithClient` or `localFileURL`. Raw `filepath.Join` plus `file://` is still here. Independently re-checked.

I only wanted Critical or High. Local is in-scope.

03 Pinning v16.0.5

Latest GA: **v16.0.5**, commit `8d24b7d`. Image `codeberg.org/forgejo/forgejo:16.0.5`.

04 The skip list is half the hunt

Closed on this pin: CVE-2026-89094 `.git` recreate, ApplyDiffPatch RCE, git HTTP redirect SSRF, CGNAT, draft attachments, OpenID association CSRF, org-mode host file read, Actions pwn-request. Instance `dump` always packs `app.ini`. That is secrets by design, archive mode 0600. Do not treat "backup contains secrets" as a vuln.

Live `POST /repos/migrate` sets `DownloadFunc`. That path does not `os.Open` an attacker `file://`. PR `PatchURL` is prefix-checked against the clone HTTP base. `file://` clone is only `doer.CanImportLocal()`. This leftover is restore of a dump, not remote LFI.

Gitea 1.27.3 absolute `DownloadURL` join-escape **FAIL**ed. Go `filepath.Join` does not drop an absolute second argument the way Python `os.path.join` does. `Join("/dump", "/etc/passwd")` stays `/dump/etc/passwd`. `FilePathJoinAbs` is in-tree and is **not** used by restore. Same FAIL here. The working input is relative `..`.

05 Twenty surfaces, one restore join

Templates, git-apply, Actions, packages, migrate, hooks, tokens, federation, OpenID, attachments, restricted users, hostmatcher, LFS, keys, restore, OAuth, markup, internal API, secrets, visibility.

Restore of a dump has no `DownloadFunc`. `GiteaLocalUploader.CreateReleases` then opens the rewritten `file://` URL. Empty `--units` restores releases.

06 The join that walks out

GO Copy

```
*asset.DownloadURL = "file://" + filepath.Join(r.baseDir, *asset.DownloadURL)
```

GO Copy

```go
case "file":
    return os.Open(u.Path)
```

`NewRepositoryRestorer` takes `filepath.Abs(baseDir)` only. No containment. No `EvalSymlinks`. `Clean` on a relative `..` walks out of the dump. `os.Open` follows a dump-side symlink if you plant one instead.

A YAML `download_url` that is already `file:// ...` is not the working path: join treats it as a relative name under the dump. Pre-rewrite `https:// ...` also becomes `file://` + join, so restore is not HTTP SSRF.

CLI posts `RepoDir` to internal `POST /api/internal/restore_repo`. That is the operator path, not unauth HTTP restore.

07 Wrong turns, in the order I made them

Absolute `DownloadURL` dump-root escape. Go `filepath.Join` keeps the dump prefix. FAIL on Gitea 1.27.3. FAIL here.

Live migrate `file://` LFI. `DownloadFunc` is set. Not this sink.

`X-Forgejo-Internal-Auth`. Restore is CLI as the forgejo uid, not a stolen internal token.

Treating restore HTTP as remote unauth. CVSS is local, user interaction. The operator has to run restore-repo.

A reverse shell. Theatre. The witness is a planted file next to the dump.

08 What an attacker can do

Hand the operator a dump directory (or already have `INTERNAL_TOKEN` and a path the server can read). Units include `releases` / `release_assets`.

PRIVILEGE	METHOD / SINK
Read a host file the forgejo uid can open, via `..` out of the dump	`release.yml` `c` `WITNESS.txt`; re bytes stored as
Same result with a symlink planted inside the dump	Joined path stay follows the link
Same sink on PR patches	`pull_request.y` same way, then `repo/pulls/<n>`
Readers of the restored repo download the host file	`GET /<owner>/<repo>`

PRIVILEGE	METHOD / SINK
Typical host files: `app.ini`, `INTERNAL_TOKEN`, `SECRET_KEY`, **DB password**	forgejo uid can r

Closed here: live `POST /repos/migrate` `file://` LFI. Absolute `DownloadURL` dump-root join escape. Unauth HTTP restore. Backup `dump` packing `app.ini`. Not OS LPE.

09 The lab

Source of truth is `lab/`. Image `codeberg.org/forgejo/forgejo:16.0.5`. Host bind **127.0.0.1:19340**. Dump sat at `/data/restore-dump`. `download_url` was `../FORGEJO-FILE-WITNESS.txt`. A decoy inside the dump did not match. Restore returned 0. `GET /labadmin/restored/releases/download/v1.0/witness.txt` was the host witness.

PLAIN TEXT Copy

```
git clone https://github.com/abraxas/forgejo-restore-file-open
cd forgejo-restore-file-open
python3 forgejo-restore-file-open-Abraxas-Labs.py
```

Last lab run, trimmed:

PLAIN TEXT Copy

```
IOC asset name='witness.txt' size=34
IOC download status=200 bytes=34 snippet=b'FORGEJO-RESTORE-FILE-OPEN-WITNESS\n'
```

```
3  SUCCESS forgejo-restore-file-open how=live ..-join file:// os.Open FORGEJO-RESTOR
```

I am not printing a dump YAML you can hand an operator. The `file://` → `os.Open` branch is the useful part.

10 The fix

Contain joined paths under `baseDir` after `EvalSymlinks` (Gitea PR 38215 `localFileURL`). Refuse `..`. Do not follow dump-side symlinks out of the dump. Drop `file://` from `uri.Open` except a dump-rooted helper.

Same tag leftovers: keys IDOR, webhook `0.0.0.1` SSRF, OpenID discovery SSRF.

11 References

- Proof of concept: abraxas/forgejo-restore-file-open · forgejo-restore-file-open-Abraxas-Labs.py
- Lab: `lab/` · Dockerfile · docker-compose.yml · run.sh · poc.py
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-22 · CWE-73
- Tree: forgejo/forgejo v16.0.5 (`8d24b7d52df327118cd776cf461d34e52e0ec1ce`) · `restore.go` · `uri.go`
- Same class on Gitea: gitea-restore-file-open

Abraxas Labs

Independent security research by abraxas.

Analysis of 0-days, n-days, malware, and the

leftovers that still matter.

GitHub @abraxas_null abraxas.null@proton.me abraxaslabs.tech

Independent research. Private analysis by arrangement. No ads. abraxas.null@proton.me · For defensive awareness and authorized testing only.

© 2026 Abraxas Labs