SERIES 0-DAY NO CVE CRITICAL PUBLIC

Stirling-PDF 3.2.0, H2 CREATE OR REPLACE ALIAS leftover of GHSA-mrr8

Stirling-PDF 3.2.0 import-database denylist is CREATE(FORCE)?ALIAS. CREATE OR REPLACE ALIAS still RUNSCRIPTs a Java alias. Backup download contains attacker SQL. GHSA-mrr8 closed the original strings. No CVE yet.

03 SERIES Stirling-PDF v3.2.0 leftovers

00 Stirling-PDF 3.2.0, H2 CREATE OR REPLACE ALIAS leftover of GHSA-mrr8

00 The pin

00 Stirling-PDF 3.2.0, processing-folder implied storage root IDOR

NEXT →
Phase 0: The pin

NAME	SERIES
Stirling-PDF 3.2.0, H2 CREATE OR REPLACE ALIAS leftover of GHSA-mrr8	Stirling-PDF v3.2.0 leftovers
PHASE	TYPE
0	0-day analysis
CVE	CVE RISK
n/a	critical
DISCLOSURE STATUS	VENDOR
public	Stirling-PDF

AFFECTED	PUBLISHED
Stirling-PDF 3.2.0 (a7e593728dd9aa81e75318a949789d3c2e9ee754), image stirlingtools/stirling-pdf:3.2.0-fat. ROLE_ADMIN, default embedded H2.	**17 Oct 2026**
UPDATED	**TAGS**
17 Oct 2026	**0-day, stirling-pdf, h2, cwe-94, leftover, remote**

TL;DR

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

01 They banned CREATE ALIAS. They left OR REPLACE in the middle.

I am @abraxas_null. The proof of concept is on GitHub: **abraxas/stirling-h2-import-leftover** (loopback client). The lab stack is `lab/`: docker-compose.yml, run.sh, poc.py. Authorized lab only. It talks to loopback.

This is leftover #1 of the Stirling-PDF v3.2.0 leftovers series. H2 first because it restores the original GHSA-mrr8 class through a regex gap. Sibling: stirling-storage-folder-idor (processing-folder implied storage root). A third candidate, markdown WeasyPrint SSRF leftover of GHSA-ff33, failed in lab. It is not this post.

This is **Stirling-PDF v3.2.0** (`a7e593728dd9aa81e75318a949789d3c2e9ee754`), image `stirlingtools/stirling-pdf:3.2.0-fat`. No CVE yet. CWE-94 / CWE-184. 9.1 (`CVSS:3.1/AV:N/AC:L/PR:H/UI:N/S:C/C:H/I:H/A:H`). **Reach:** Remote ROLE_ADMIN. **Class:** H2 RUNSCRIPT leftover of GHSA-mrr8-934j-4g8m / CVE-2026-85714.

`POST /api/v1/database/import-database` is `@PreAuthorize("hasRole('ADMIN')")`. `validateSqlContent` is supposed to stop `CREATE ALIAS`. The denylist is `CREATE\s+(FORCE\s+)?ALIAS`. H2 `parseCreate` allows `OR REPLACE` before `ALIAS`. The

allowlist still sees `\bCREATE\b`. The validator sanitizes a **copy**. `verifyBackup` and production `RUNSCRIPT FROM ?` the **original file**.

A ROLE_ADMIN import creates a Java alias the denylist is supposed to stop. A later backup download contains attacker SQL. Host `FILE_READ` / FileUtils RCE from the original advisory is inferred, not lab-proven here.

02 Why Stirling-PDF

I wanted a self-hosted PDF toolbox people actually docker-run, with an import-database button next to a default embedded H2. Stirling is that product. GHSA-mrr8 already taught the world that `FILE_READ` / `CREATE ALIAS` / `CSVWRITE` through RUNSCRIPT is the whole game. v2.13.2 shipped a denylist. v3.2.0 still ships that denylist. I sat on the current GA anyway.

Template `security.enableLogin` is true. Official compose and the README `docker run` set `SECURITY_ENABLELOGIN=false`, then `anyRequest().permitAll()`. `@EnableMethodSecurity` stays on. Class `@PreAuthorize("hasRole('ADMIN')")` still 403s anonymous. Default first admin is `admin` / `stirling` when `initialLogin` is empty. Lite / `DISABLE_ADDITIONAL_FEATURES` omits `DatabaseController`. Custom PostgreSQL disables it (`H2SQLCondition`). The HIT deploy is fat plus login-on plus default H2.

Vendor intake is the GitHub Security Advisory tab or security@stirlingpdf.com. Unpaid. I am not opening a public issue on Stirling-Tools/Stirling-PDF.

03 Pinning v3.2.0

Latest stable on the day I cloned. Tag object **v3.2.0**, commit `a7e593728dd9aa81e75318a949789d3c2e9ee754`. Image `stirlingtools/stirling-pdf:3.2.0-fat`. Stay on **3.2.0+** for the disclosed GHSA-mrr8 strings. Do not treat "I banned `CREATE ALIAS`" as "I banned H2 Java aliases."

The lab binds **only** `127.0.0.1:18152`. Fat. Login-on. Default H2. Witness in a backup `.sql`.

04 The skip list is half the hunt

Before you look for leftovers, you write down what is already public so you do not "discover" a CVE with a press release. On v3.2.0 the interesting rows were already closed:

- GHSA-mrr8-934j-4g8m / CVE-2026-85714. Bare `CREATE ALIAS`, `FILE_READ(`, `CSVWRITE(` are denied. **This is the sanitizer `OR REPLACE` missed.**
- GHSA-ff33-grr6-rmvp markdown SSRF. Nested leftover **FAILED** in lab (OWASP 20260924.2). Not this post.
- GHSA-xw8v-9mfm-g2pm HTML-to-PDF SSRF. Same sanitizer.
- GHSA-wccq-mg6x-2w22 markdown ZIP traversal. `startsWith` holds.
- GHSA-3xxh-mm3g-c9w5 pipeline INTERNAL_API_USER. Allowlist holds.
- GHSA-998c-x8hx-737r URL-to-PDF. Default-off. Sibling CVE-2026-71270 is public.
- Public n-day issue #7628 `.fodt` / `.fods`. Do not "discover" it.

GHSA-mrr8 is the interesting skip. It named three strings. It did not name `OR REPLACE`.

05 Surfaces that returned nothing

Unauth import 401s. Method security ADMIN stays on when HTTP is `permitAll`. Lite image has no controller. PostgreSQL datasource disables it. Bare `CREATE ALIAS` is 500 disallowed. `EXECUTE IMMEDIATE` plus `FILE_READ('/etc/os-release')` 500s `Backup verification failed`. Comment-mismatch and unicode-space bypasses stayed source-only; I did not need them once `OR REPLACE` worked.

The temperature changed on the denylist line sitting next to H2 `parseCreate`. Tests cover the obvious strings. They do not cover `OR REPLACE`.

06 The denylist is the whole argument

`DatabaseService.DENIED_PATTERNS`:

PLAIN TEXT Copy

```
Pattern.compile("(?i)\\bCREATE\\s+(FORCE\\s+)?ALIAS\\b"),
```

H2 accepts `CREATE OR REPLACE ALIAS name FOR "java.lang.System.nanoTime"`. `OR REPLACE` sits where `FORCE` was optional. The regex never sees `ALIAS` after `CREATE` plus optional `FORCE`. Allowlist `\bCREATE\b` matches. No string hiding required.

`validateSqlContent` sanitizes a copy for regex. `verifyBackup` RUNSCRIPTs the original into `jdbc:h2:mem:backupVerify_*`. Production import RUNSCRIPTs the original again. The copy can be clean. The file that runs is not.

Lab SQL, trimmed:

PLAIN TEXT Copy

```
CREATE TABLE IF NOT EXISTS lab_w(n VARCHAR);
CREATE OR REPLACE ALIAS lab_nano FOR "java.lang.System.nanoTime";
INSERT INTO lab_w VALUES('STIRLING-H2-IMPORT-WITNESS');
```

200 `importIntoDatabaseSuccessed`. `GET /api/v1/database/createDatabaseBackup` then `GET /api/v1/database/download/backup_202610172328.sql` is 50432 bytes and contains the witness.

I used `System.nanoTime` as the alias target. I did not replay FileUtils / CSVWRITE / a Python `.pth`. That chain is GHSA-mrr8's original write-up. This leftover restores the primitive. It does not need me to re-prove the rest of the advisory.

07 The lab

Source of truth is `lab/`. Image `stirlingtools/stirling-pdf:3.2.0-fat`. Port **18152**. Bind it to loopback.

PLAIN TEXT Copy

```
git clone https://github.com/abraxas/stirling-h2-import-leftover
cd stirling-h2-import-leftover
python3 stirling-h2-import-leftover-Abraxas-Labs.py
```

or `cd lab && ./run.sh`. The loopback client is `lab/run.sh` with a banner on it. Target **only** `http://127.0.0.1:18152`. Do not publish the port off loopback.

Last lab run, trimmed:

PLAIN TEXT Copy

```
IOC unauth-import http=401
IOC negative-import http=500 ... disallowed operations ...
IOC leftover-import bypass=create-or-replace-alias http=200 ... importIntoDatabas
IOC download file=backup_202610172328.sql http=200 bytes=50432 witness=1
SUCCESS stirling-h2-import-leftover negative-denied=yes bypass=create-or-replace-
```

Witness string: `STIRLING-H2-IMPORT-WITNESS`. It is an INSERT into `lab_w`. The backup download is the oracle.

Default admin `admin` / `stirling` has `firstLogin=true`. The lab uses `change-password-on-login`, then re-logs.

08 Wrong turns, in the order I made them

Treating unauth import as in-scope. Method security ADMIN. 401. Last line requires `unauth-import=401`.

Treating the original `CREATE ALIAS` string as still live. 500 disallowed. That is GHSA-mrr8 working. The leftover is `OR REPLACE`.

Treating `EXECUTE IMMEDIATE` plus `FILE_READ('/etc/os-release')` as the leftover. 500 verifyBackup. Inner `FILE_READ(` is still denied on the unstripped allow/deny mix, or verifyBackup refuses the result. I kept it as extra FAIL. Do not claim host file read as lab-proven.

Replaying a reverse shell / FileUtils `.pth`. Theatre. The oracle is a witness string in a backup. GHSA-mrr8 already described that chain. This pack proves the denylist miss.

Login-off official compose as the HIT deploy. Import is still ADMIN. Fat is required because lite omits the controller.

09 What an attacker can do

ROLE_ADMIN on login-on proprietary/fat with the default embedded H2. Default first admin is `admin` / `stirling` until changed. They do not need unauth.

Privilege table (exhaustive). Same bar as Signal-Server Dropwizard admin.

Integrity (lab-proven)

METHOD + PATH
ROLE_ADMIN `POST /api/v1/database/import-database` with `CREATE OR REPLACE ALIAS lab_n`

Recon (lab-proven)

METHOD + PATH
ROLE_ADMIN `GET /api/v1/database/createDatabaseBackup` then `GET /api/v1/database/dowr`

Recon / host file (lab extra FAIL)

METHOD + PATH	GRANTS
same import wrapping `FILE_READ('/etc/os-release')` in `EXECUTE IMMEDIATE`	500 `Backup verificatio failed`. Host bytes not in a backup

Integrity / RCE (inferred)

METHOD + PATH
alias `org.h2.store.fs.FileUtils` plus `CSVWRITE` of a Python user-site `.pth`, then office cor

Adjacent surface that stays closed

METHOD + PATH	WHY IT IS CLOSED
unauth `POST /api/v1/database/import-database`	401 `Authentication required`
ROLE_ADMIN import of bare `CREATE ALIAS`	500 disallowed
lite / `DISABLE_ADDITIONAL_FEATURES`	`DatabaseController` absent
custom PostgreSQL datasource	`@Conditional(H2SQLCondition.class)`

I am not claiming a reverse shell, and I am not claiming host `FILE_READ` as lab-proven. The leftover is the denylist miss that lets `CREATE OR REPLACE ALIAS … FOR "java.lang.…"` through RUNSCRIPT.

10 What this is not

It is not "GHSA-mrr8 never shipped." Bare `CREATE ALIAS` is denied. This `OR REPLACE` form is not. It is not unauthenticated. It is not a reverse shell. It is not lab-proven host FILE_READ.

If the instance was ever **<= 2.11.0**, rotate and patch for the disclosed GHSA-mrr8 wave anyway. That is the original strings, not this leftover.

11 The fix

Extend the denylist so `OR REPLACE` cannot sit between `CREATE` and `ALIAS`. Run `validateSqlContent` on the same bytes `RUNSCRIPT` will execute. Deny `EXECUTE IMMEDIATE` unless the inner SQL is also scanned.

PLAIN TEXT Copy

```
Pattern.compile("(?i)\\bCREATE\\s+(OR\\s+REPLACE\\s+)?(FORCE\\s+)?ALIAS\\b")
```

Re-run the loopback client against a patched build: leftover import must 400/500 the way bare `CREATE ALIAS` already does.

Until then, do not treat "I banned `CREATE ALIAS`" as "I banned H2 Java aliases."

I am not going to print a walk of `import-database` aimed at a live host. The regex gap is the useful part. If you own the box, run the proof of concept against loopback.

Same product, other unpublished lab on this tag: processing-folder storage IDOR. Series hub: Stirling-PDF v3.2.0 leftovers.

12 References

- Proof of concept: abraxas/stirling-h2-import-leftover · stirling-h2-import-leftover-Abraxas-Labs.py
- Lab: `lab/` · docker-compose.yml · run.sh · poc.py
- Series: Stirling-PDF v3.2.0 leftovers · stirling-storage-folder-idor
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me

- CWE-94 · CWE-184
- Tree: Stirling-PDF v3.2.0 (`a7e593728dd9aa81e75318a949789d3c2e9ee754`) · `DatabaseService.java` · `DatabaseController.java`
- Nearby patched (not this leftover): GHSA-mrr8-934j-4g8m
- Product: Stirling-PDF

03 SERIES Stirling-PDF v3.2.0 leftovers

00 Stirling-PDF 3.2.0, H2 CREATE OR REPLACE ALIAS leftover of GHSA-mrr8

00 The pin

00 Stirling-PDF 3.2.0, processing-folder implied storage root IDOR

NEXT →
Phase 0: The pin

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