SERIES 0-DAY NO CVE HIGH PUBLIC

Plausible Analytics, editor People HTML leaks transfer nanoid

Plausible Analytics master d44e104 GET /:domain/settings/people puts a pending ownership-transfer nanoid in HTML id= that a guest editor can load. Default CE invite-only plus email verification off: register as the invitee and accept. Site team_id moves. No CVE yet.

03 SERIES Plausible Analytics leftovers

00 The pin

00 Plausible Analytics, editor People HTML leaks transfer nanoid

00 Plausible Analytics, traffic-change notification site_id IDOR

← PREVIOUS
Phase 0: The pin

NEXT →
Phase 0: Plausible Analytics, traffic-change notification site_id IDOR

NAME	SERIES
Plausible Analytics, editor People HTML leaks transfer nanoid	Plausible Analytics leftovers
PHASE	TYPE
0	0-day analysis
CVE	CVE RISK
n/a	high
DISCLOSURE STATUS	VENDOR

public	**Plausible**
AFFECTED **Plausible Analytics GitHub master d44e104ae3943a9240d5cb6b62e67dff0f069d3a (2026-10-16). CE MIX_ENV=ce. Default DISABLE_REGISTRATION=invite_only, ENABLE_EMAIL_VERIFICATION=false. Authenticated guest/team editor.**	PUBLISHED **17 Oct 2026**
UPDATED **17 Oct 2026**	TAGS **0-day, plausible, privilege-leftover, cwe-200, cwe-269, authenticated**

TL;DR

Lab files and proof-of-concept here.

01 The CSS id is the bearer

I am @abraxas_null. The proof of concept is on GitHub: **abraxas/plausible-editor-invite** (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 Plausible Analytics leftovers series. Same pin, other pack: plausible-traffic-notify (`PUT` traffic-change-notification `site_id` leftover of Shield/Goal). This post is the site steal.

This is **Plausible Analytics** GitHub master `d44e104ae3943a9240d5cb6b62e67dff0f069d3a` (2026-10-16). Community edition, `MIX_ENV=ce`. No CVE yet. CWE-200 / CWE-269. 8.1 High (`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:L`). **Reach:** Remote. **Class:** Privilege leftover (invitation-token leak). Authenticated guest editor or team editor of that site. Not unauth. Not guest viewer.

`Sites.list_people/1` selects raw `guest_invitations.invitation_id` and `team_site_transfers.transfer_id`. `settings_people.html.heex` puts that 21-char nanoid in `id="invitation-#{invitation.invitation_id}"` for every pending row. People settings is not in `SiteController` `@special_cased_actions`, so `AuthorizeSiteAccess` allows `:editor`. Guest editor maps to site role `:editor`.

`Nanoid.generate/0` is the register-from-invite bearer. Unauth `GET /register/invitation/:invitation_id` looks up by id only. LiveView `register` overwrites posted email with `invitation.email`. Default CE is `DISABLE_REGISTRATION=invite_only` and `ENABLE_EMAIL_VERIFICATION=false`. Accept is `POST /sites/invitations/:id/accept` via citext email match.

A guest editor who can already see People HTML reads a pending ownership-transfer nanoid, registers as the invitee, and accepts. The site moves onto the attacker personal team.

Vendor intake is security@plausible.io. Unpaid. I am not opening a public issue on plausible/analytics.

02 Why Plausible

I wanted a self-hosted analytics product people actually run, with a team model that pretends an editor cannot become owner. Plausible is that product. Guest editor is a real role. Ownership transfer is a real flow. The invite token is a nanoid because it is supposed to be a secret in a mail. Putting it in a CSS id is how the delete button finds the row. Delete is owner/admin only. The `id=` is not.

whoareme Bug 02 closed register without a valid invite. It did not stop the People page from printing a valid one.

I only wanted Critical or High. Remote preferred. Medium stays on the floor. EE `extra/lib` is another tree. This pin is CE `lib/`.

03 Pinning master

SECURITY.md patches only latest MAJOR.MINOR and tells operators to run `:latest`. Last CE tag v3.2.1 is 2026-05-15. I cloned GitHub HEAD on 2026-10-16. Commit `d44e104ae3943a9240d5cb6b62e67dff0f069d3a`. A leftover on v3.2.1 would be five months of "we already shipped that." A leftover on HEAD is current.

The lab binds **only** `127.0.0.1:18180`. Image `plausible-lab:d44e104ae3943a`. `DISABLE_REGISTRATION=invite_only`. `ENABLE_EMAIL_VERIFICATION=false`. First launch allows the first user. Password min 12. Captcha off unless `FRIENDLY_CAPTCHA_*` is set.

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 this pin the interesting rows were already closed:

- CVE-2026-8467 / GHSA-55hg-8qxv-qj4p storybook RCE. `/storybook` is gone. Not this leftover.
- whoareme Bug 02 invite-only LiveView register without a valid invite. `check_registration_enabled(:default)` vs `:invitation`. Patched. The leftover is a **valid** invitation id that an editor was never supposed to hold.
- whoareme Bug 02 Shield / Goal `site_id` mass-assignment. Patched. Sibling leftover is traffic-change notification.
- whoareme Bug 03 EE SSO domain-verification SSRF. EE. Out of this lab.
- LiveView CSRF: socket `check_origin` true. Password reset uses `BASE_URL`, not Host. Tracker XSS is `Jason.encode!`. ClickHouse queries always `AND`

`site_id`.

The temperature changed on `list_people` selecting the same nanoid that `GET /register/invitation/:id` treats as a bearer.

05 Surfaces that returned nothing

I split the tree the way you split a Phoenix analytics app: register, invite, transfer, people, LiveView CSRF, Stats API, ingest, shared links, plugins MCP, S3, 2FA, password reset, tracker, ClickHouse, super-admin, CSV import, funnels, TeamSetup, installation tracker script.

Most of that is Plausible doing what the whoareme writeups said it would do.

Guest viewer 404s on people. Editor accept of a leaked transfer id does not move `team_id` (email match). Owner/admin can already invite and transfer. 21-char nanoid brute is out. No OS shell. No ClickHouse admin. No super-admin LiveView. Funnels nested `goal_id` is dangling FK recon of conversion names, not High. TeamSetup guest rename of an unsetup personal team is Medium config.

The leftover is the HTML `id=` on a page an editor is allowed to load.

06 The HTML id is the whole argument

`Sites.list_people/1` (`sites.ex:176-201`):

PLAIN TEXT Copy

```
invitations =
  from(
    gi in Teams.GuestInvitation,
    inner_join: ti in assoc(gi, :team_invitation),
    where: gi.site_id == ^site.id,
    select: %{
```

```
      invitation_id: gi.invitation_id,
      email: ti.email,
      role: gi.role
    }
  )
  |> Repo.all()

site_transfers =
  from(
    st in Teams.SiteTransfer,
    where: st.site_id == ^site.id,
    select: %{
      invitation_id: st.transfer_id,
      email: st.email,
      role: :owner
    }
  )
  |> Repo.all()

%{memberships: memberships, invitations: site_transfers ++ invitations}
```

Transfers are folded into the same invitation list. The template does not distinguish. `settings_people.html.heex:108-132`:

PLAIN TEXT Copy

```
row_attrs={fn invitation -> %{id: "invitation-#{invitation.invitation_id}"} end}
```

The delete button is owner/admin only. The row `id=` is not.

`RegisterForm` `find_by_id_unified/1` looks up team invitation, guest invitation, then site transfer by that same string. `handle_event("register", ...)`:

PLAIN TEXT Copy

```
user =
  params["user"]
  |> Map.put("email", invitation.email)
  |> Auth.User.new()
```

Posted email is theatre. The account is the invitee. Default CE `check_registration_enabled(:invitation)` allows it. Email verification is off. `InvitationController.accept_invitation` matches citext email and moves the site.

I am not going to print a walk of a live host's People page. The missing authorization on the nanoid is the useful part.

07 What an attacker can do

Authenticated guest editor (`:guest_member, :editor`) or team editor of that site. Needs a normal session cookie. No owner, no admin, no super-admin LiveView. They do not brute the 21-char nanoid. They read it from HTML they are allowed to load, then register as the invitee.

Privilege table (exhaustive). Same bar as Signal-Server Dropwizard admin: method + path, what that call grants, recon vs integrity vs availability, lab-proven vs inferred, adjacent surface that stays closed.

AXIS	METHOD + PATH	GRANTS	PROVENANCE
Recon	`GET /:domain/settings/people` as guest/team editor	Pending rows render `id="invitation-<nanoid>"`. Nanoid is `invitation_id` or `transfer_id`. Same site only. Invitee email is also in the table.	**Lab-proven.** Editor `Hc8DiBC2q08mv5LF`

AXIS	METHOD + PATH	GRANTS	PROVENANCE
Recon	unauth `GET /register/invitation/:id`	Invitee email on the register form. Lookup is by id only.	**Lab-proven.** Regist `newowner@lab.inva`
Integrity	LiveView `register` on `/register/invitation/:id`	Posted `user[email]` is overwritten with `invitation.email`. New CE account at the invitee address. Default CE email verification off. Invite-only still allows `:invitation`.	**Lab-proven.** Poste `overwrite@lab.inv` `newowner@lab.inva`
Integrity	`POST /sites/invitations/:id/accept` as that new account	Pending ownership transfer moves `sites.team_id` onto the newowner personal team. Transfer row is gone.	**Lab-proven.** `sites`
Integrity	same accept on a pending **guest** invite for that email	New membership on that site (editor-for-editor is not a gain). Other pending team/guest/transfer rows for that email also become accept-able.	Source (`InvitationContro`

AXIS	METHOD + PATH	GRANTS	PROVENANCE
Availability	after transfer accept	Original team loses the site. Stats keep serving on the new team.	**Lab-proven.** team_

Stays closed on this pin

AXIS	METHOD + PATH	WHY IT IS CLOSED
Recon	guest viewer `GET /:domain/settings/people`	404. Viewer is not `:editor`. **Lab-proven.**
Integrity	editor `POST /sites/invitations/:id/accept` of the leaked transfer id	Email match. Editor is not `newowner@lab.invalid`. Site stays on team 1. **Lab-proven.**
Integrity	unauth register without a valid invitation id	`check_registration_enabled(:default)` vs `:invitation`. whoareme Bug 02, patched.
Recon	21-char nanoid brute	URL-safe alphabet, 21 chars. Out of this leftover.
Recon	ClickHouse events, SMTP, other sites' people pages	Same-site only. No ClickHouse admin. No mailbox read.
Integrity	owner/admin invite and transfer	They already can. This leftover is the editor who cannot.
Integrity	super-admin LiveView, OS shell, `extra/lib` EE SSO	Closed. CE lab. No shell.

I am not claiming unauthenticated account takeover of an already-registered user, a reverse shell, or ClickHouse dump. The High instance is a pending **owner** transfer (or invites to **other** sites/teams for that email) on default CE: invite-only plus email

verification off. EE always sets `email_verified: false` and `/activate` needs the invitee mailbox.

Preconditions: a pending guest invite and/or ownership transfer on that site; invitee email is not already a `users.email` row (citext unique).

08 The lab

Source of truth is `lab/`. Image `plausible-lab:d44e104ae3943a`. Port **18180**. Bind it to loopback. First run clones https://github.com/plausible/analytics at `d44e104ae3943a9240d5cb6b62e67dff0f069d3a` and builds with `MIX_ENV=ce` if the image is missing.

PLAIN TEXT Copy

```
git clone https://github.com/abraxas/plausible-editor-invite
cd plausible-editor-invite
python3 plausible-editor-invite-Abraxas-Labs.py
```

or `cd lab && ./run.sh`. The loopback client is `lab/run.sh` with a banner on it. Target **only** `http://127.0.0.1:18180`. Do not publish the port off loopback.

Last lab run, trimmed:

PLAIN TEXT Copy

```
IOC editor-people-leak invitation-id=Hc8DiBC2q08mv5LFmBysu
IOC viewer-people http=404
IOC register-email-overwrite posted=attacker-overwrite@lab.invalid stored=newowne
IOC editor-accept-no-move team_id=1
IOC newowner-accept team_id 1 → 2
SUCCESS plausible-editor-invite PLAUSIBLE-H02-EDITOR-INVITE-TOKEN-WITNESS
```

Witness string: `PLAUSIBLE-H02-EDITOR-INVITE-TOKEN-WITNESS`. Oracle is the people HTML nanoid plus `sites.team_id` after accept.

`run.sh` seeds owner, guest editor, guest viewer, a pending ownership transfer to `newowner@lab.invalid`. Editor loads people. Viewer 404s. LiveView register overwrites email. Editor accept is a negative. Newowner accept moves the site.

09 Wrong turns, in the order I made them

Treating guest register as auto-accept. First lab FAIL was `editor-people-404` because I assumed invite-only register also accepted the membership. It does not. You still `POST /sites/invitations/:id/accept` after login. The leftover is the token plus accept, not register alone.

Treating editor accept of the leaked transfer as the steal. Email match. Editor is `editor@lab.invalid`. Transfer is `newowner@lab.invalid`. Site stays on team 1. Lab last line requires the editor-accept negative and the newowner-accept move.

Login 429. CE `Auth.rate_limit` `login_ip=5` per minute. Keep one session per user. Do not log in a sixth time in the same minute and call it a product bug.

Claiming viewer can read people. Viewer 404s. The leftover is editor. If your fixture leaks to viewer, the run is FAIL.

Bruting the nanoid. 21 chars, URL-safe alphabet. Out. The HTML already has it.

A reverse shell. Theatre. The oracle is a nanoid in HTML and a `team_id` column. I am not claiming ClickHouse dump or super-admin LiveView.

Client bugs that look like product bugs. Phoenix LiveView register is a websocket `lv:` join on `/live/websocket?vsn=2.0.0`. A helper that POSTs `/register` as a dead form never fires `handle_event("register", ...)`. I mention that once because it cost time and looked, for a minute, like invite-only had finished the job.

10 The fix

Do not put the bearer nanoid in HTML an editor can load. People settings can key the row by a server-side id that is not `invitation_id` / `transfer_id`, or hash it for the DOM. Keep the real token on the owner/admin delete form only.

Stop selecting raw `invitation_id` / `transfer_id` in `Sites.list_people/1` for the editor view. Authorize People as owner/admin, the way danger-zone already is, if editors do not need the pending table.

On register-from-invite, bind the new account to a one-time hashed token, not a nanoid that also lives in a CSS id.

Re-run the loopback client against a patched build: editor people HTML must not contain `invitation-<21-char>`. Newowner accept of a token that was only on People must fail.

Until then, do not treat "editors cannot transfer a site" as "editors cannot read the transfer bearer." Default CE invite-only plus email verification off makes that bearer a register-and-accept path.

I am not going to print a walk of `/settings/people` aimed at a live host. The HTML `id=` is the useful part. If you own the box, run the proof of concept against loopback.

Same product, other unpublished lab on this pin: traffic-change notification `site_id`. Series hub: Plausible Analytics leftovers.

11 References

- Proof of concept: abraxas/plausible-editor-invite · plausible-editor-invite-Abraxas-Labs.py
- Lab: `lab/` · docker-compose.yml · run.sh · poc.py
- Series: Plausible Analytics leftovers · plausible-traffic-notify

- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-200 · CWE-269
- Tree: plausible/analytics master `d44e104ae3943a9240d5cb6b62e67dff0f069d3a` · `sites.ex` · `settings_people.html.heex` · `register_form.ex` · `invitation_controller.ex`
- Nearby patched (not this lookup): whoareme 2026-08-25 Bug 02 invite-only LiveView register without a valid invite
- Product: Plausible Analytics

03 SERIES Plausible Analytics leftovers

00 The pin

00 Plausible Analytics, editor People HTML leaks transfer nanoid

00 Plausible Analytics, traffic-change notification site_id IDOR

← PREVIOUS
Phase 0: The pin

NEXT →
Phase 0: Plausible Analytics, traffic-change notification site_id IDOR

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