SERIES 0-DAY NO CVE HIGH PUBLIC

Plausible Analytics, traffic-change notification site_id IDOR

Plausible Analytics master d44e104 PUT /sites/:domain/traffic-change-notification/:type still casts site_id. whoareme Bug 02 patched Shield and Goal. This controller was not in that pass. An editor of their own site re-parents spike notify onto a victim. Oban mails attacker about victim traffic.

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: Plausible Analytics, editor People HTML leaks transfer nanoid

NAME	SERIES
Plausible Analytics, traffic-change notification site_id IDOR	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 + EE lib/. Authenticated owner/admin/editor of a site they already control. CSRF required.	PUBLISHED 17 Oct 2026
UPDATED 17 Oct 2026	TAGS 0-day, plausible, idor, cwe-915, cwe-639, authenticated

TL;DR

Lab files and proof-of-concept here.

01 Shield learned. This changeset did not.

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

This is leftover #2 of the Plausible Analytics leftovers series. Same pin, other pack: plausible-editor-invite (People HTML leaks transfer nanoid to editors). This post is the Shield/Goal sibling that still casts `:site_id`.

This is **Plausible Analytics** GitHub master `d44e104ae3943a9240d5cb6b62e67dff0f069d3a` (2026-10-16). Community edition plus the same `lib/` on EE. No CVE yet. CWE-915 / CWE-639. 7.1 High (`CVSS:3.1/AV:N/AC:L/PR:L/UI:N/S:U/C:H/I:L/A:L`). **Reach:** Remote. **Class:** IDOR (mass-assignment). Authenticated owner / admin / editor of a site they already control. Viewer cannot. CSRF required (same-session IDOR).

`Plausible.Site.TrafficChangeNotification.changeset/2` still casts `:site_id`. `SiteController.update_traffic_change_notification/2` loads the row by the authorized path domain, then `Repo.update!` of the client map. Route `PUT /sites/:domain/traffic-change-notification/:type` sits under `:browser` + CSRF + `AuthorizeSiteAccess` `[:owner, :admin, :editor, :super_admin]`. The HTML form only posts `threshold`. Phoenix still accepts extra keys.

whoareme Bug 02 taught Shield and Goal to bind `socket.assigns.site.id` and stop casting `:site_id`. This controller was not in that pass. Enable (`POST .../enable`) stamps `site_id: site.id` from assigns. Attacker enables on their own site, then PUTs `traffic_change_notification[site_id]=<victim_pk>`. Unique index `(site_id, type)`. Default install has no spike/drop row, so the update succeeds and the row now belongs to the victim. Table name is legacy `spike_notifications`.

An editor of their own site re-parents the spike/drop notify row onto a victim site. Recipients stay the attacker mailbox. Oban mails attacker about victim traffic.

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

02 Why Plausible

I wanted the leftover of a public mass-assignment class on a product people self-host. whoareme named Shield and Goal. Grep for the next `cast(.*:site_id` is the whole hunt. Traffic-change notification is the one that was still there.

Spike mail is supposed to be a convenience for the site you already own. Subject names the domain. Body has live visitors, top sources, top pages. That is recon of a site you are not a member of, delivered to an inbox you already read.

I only wanted Critical or High. Remote preferred. Medium stays on the floor. Installation tracker `cast :site_id` is ingest, and ingest is already unauth `POST /api/event`. Dropped.

03 Pinning master

Same pin as the editor invite leftover. GitHub HEAD `d44e104ae3943a9240d5cb6b62e67dff0f069d3a`. `SECURITY.md` says `:latest`. Last CE tag v3.2.1 is five months behind.

The lab binds **only** `127.0.0.1:18183`. Image `plausible-lab:d44e104ae3943a`. `DISABLE_REGISTRATION=false` so two independent teams can register. `SECRET_KEY_BASE` must be at least 64 bytes or `Plug.Session.COOKIE` refuses the session.

04 The skip list is half the hunt

- whoareme Bug 02 Shield / Goal `site_id`. Patched. **This is the sanitizer traffic-change notification missed.**
- whoareme Bug 02 stored XSS in goal `display_name`. Patched. Not this leftover.
- whoareme Bug 02 invite-only register without a valid invite. Patched. Sibling leftover is People HTML.
- CVE-2026-8467 storybook RCE. `/storybook` gone.
- whoareme Bug 03 EE SSO SSRF. EE. Out of this lab.
- Tracker ingest is already unauth. Casting `site_id` on installation config does not raise this leftover.

The temperature changed on a changeset that still listed `:site_id` next to handlers that had stopped.

05 Surfaces that returned nothing

PUT with the victim **path domain** is 404. `AuthorizeSiteAccess` is the path domain, not the body `site_id`. Attacker is not a member of victim. That negative is required. The leftover is PUT on **your** domain with a foreign integer PK.

Viewer cannot hit the route. Shield / Goal / funnel / segment / shared-link write is patched. No ClickHouse dump. No dashboard HTML. `Repo.update!` 500s if the victim already has that type. Default install does not, so first plant works. No OS shell.

`site_id` is the integer PK. Enumerable. whoareme used `site_id=1`. Dashboard `data-*` does not emit it. You guess, or you have another oracle. Sequential sites on a fresh CE box start at 1.

06 The cast is the whole argument

`traffic_change_notification.ex:19-25`:

PLAIN TEXT Copy

```
def changeset(schema, attrs) do
  schema
  |> cast(attrs, [:site_id, :recipients, :threshold, :type])
  |> validate_required([:site_id, :recipients, :threshold, :type])
  |> validate_number(:threshold, greater_than_or_equal_to: 1)
  |> unique_constraint([:site_id, :type])
end
```

`site_controller.ex:489-504`:

PLAIN TEXT Copy

```
def update_traffic_change_notification(conn, %{
      "traffic_change_notification" => params,
```

```
      "type" ⇒ type
    }) do
  site = conn.assigns[:site]

  notification =
    Repo.get_by(Plausible.Site.TrafficChangeNotification, site_id: site.id, type

  Plausible.Site.TrafficChangeNotification.changeset(notification, params)
  ▷ Repo.update!()
```

Load by authorized `site.id`. Update from `params`. Phoenix nested params accept `traffic_change_notification[site_id]`. Enable is clean: it stamps assigns. The leftover is the PUT.

`Plausible.Workers.TrafficChangeNotifier` emails `notification.recipients` for the **row's** site. Recipients never moved. The site did. First send is immediate (`last_sent` nil). Repeat at most every 12 hours. Subject `Traffic spike on <victim.domain>`. Body: current visitor count, top 3 sources, top 3 pages. Dashboard HTML / installation links in the mail are omitted unless the recipient is a team/guest member. Attacker is not. The recon is the subject and the counts.

I am not going to print a walk of a live host's email-reports form. The extra `cast :site_id` is the useful part.

07 What an attacker can do

Owner, admin, or editor of any site they already control. Needs a session cookie and the CSRF token from that session. They do not need membership on the victim. They do not PUT the victim path domain (that 404s). They PUT their own domain with a foreign `site_id` integer PK.

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	PROVENA
Integrity	`POST /sites/:attacker_domain/traffic-change-notification/:type/enable` then `PUT /sites/:attacker_domain/traffic-change-notification/:type` with `traffic_change_notification[site_id]=<victim_pk>`	Re-parent attacker `spike_notifications` row onto `victim.sites.id`. Recipients stay the attacker address. Owns the unique `(site_id, type)` slot. Threshold is attacker-chosen (lab used 1).	**Lab-pr** 1, recipi
Recon	Oban `TrafficChangeNotifier` email to `notification.recipients`	Subject `Traffic spike on <victim.domain>`. Body: current visitor count, top 3 sources, top 3 pages. First send is immediate (`last_sent` nil). Repeat at most every 12 hours. Threshold 1 yields a mail whenever the victim has any current visitor. Dashboard HTML / installation links in the mail are omitted unless the recipient is a team/guest member (attacker is not).	**Inferre** capture (`MAILE`
Availability	victim `POST .../enable` of the same type	Unique `(site_id, type)` rejects. Victim cannot turn the feature	Source

AXIS	METHOD + PATH	GRANTS	PROVENA
		on until the planted row is gone.	
Availability	victim `disable` (`delete_all` by `site_id`)	Victim can clear the planted row if they notice the extra recipient.	Source (`disab`

Stays closed on this pin

AXIS	METHOD + PATH	WHY IT IS CLOSED
Integrity	`PUT /sites/:victim_domain/traffic-change-notification/:type`	Path domain is `AuthorizeSiteAccess`. Attacker is not a member. **Lab-proven 404.** `attacker-not-member-of-victim=1`
Integrity	Shield / Goal / funnel / segment / shared-link write	whoareme Bug 02 patched those `cast :site_id` handlers. They bind assigns.
Recon	ClickHouse dump, Stats API, dashboard HTML	No membership, no shared link. Notifier mail omits dashboard links for a non-member.
Integrity	tracker ingest	Already unauth `POST /api/event` with a client-chosen domain. Not this leftover.
Integrity	`Repo.update!` when victim already has that type	500 on unique `(site_id, type)`. Default install has no row, so first plant works.
Integrity	viewer of attacker site	`AuthorizeSiteAccess` is owner/admin/editor. Viewer cannot.

AXIS	METHOD + PATH	WHY IT IS CLOSED
Integrity	OS shell, super-admin LiveView	Closed. No shell.

I am not claiming ClickHouse dump, a reverse shell, or dashboard login as the victim. The dump is traffic recon in mail the attacker already receives: domain, live visitors, top sources, top pages.

08 The lab

Source of truth is `lab/`. Image `plausible-lab:d44e104ae3943a`. Port **18183**. Bind it to loopback. First run clones the pin if the image is missing.

PLAIN TEXT Copy

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

or `cd lab && ./run.sh`. Target **only** `http://127.0.0.1:18183`.

Last lab run, trimmed:

PLAIN TEXT Copy

```
IOC victim.lab id=1 team=1 attacker.lab id=2 team=2
IOC attacker-not-member-of-victim=1
IOC put-victim-path http=404
IOC spike_notifications row 1 site_id 2 → 1 recipients={attacker@lab.invalid}
SUCCESS plausible-traffic-notify PLAUSIBLE-H03-TRAFFIC-NOTIFY-SITEID-WITNESS
```

Witness string: `PLAUSIBLE-H03-TRAFFIC-NOTIFY-SITEID-WITNESS`. Oracle is the `spike_notifications.site_id` move. Notifier mail recon stays inferred.

Two teams. Enable spike notify on attacker. PUT own domain with `site_id=1`. Row moves. Victim path PUT 404s. Attacker is not a member of victim.

09 Wrong turns, in the order I made them

`SECRET_KEY_BASE` **shorter than 64 bytes.** First FAIL. `Plug.Session.COOKIE` wants 64. The leftover is not a session bug. Pad the secret.

PUT on the victim path domain. 404. That is the closed adjacent. If your fixture "succeeds" there, you accidentally made the attacker a member. The leftover is PUT on **attacker** domain.

Treating enable as the bug. Enable stamps `site.id` from assigns. Clean. Update takes the client map. That is the leftover.

Capturing mail as the oracle. Bamboo local adapter. I did not parse the mailbox. The row move is the witness. Mail recon is inferred from `TrafficChangeNotifier`. Privilege table says so.

Claiming Shield is still open. It is not. whoareme patched it. Grep the remainder.

A reverse shell. Theatre. The oracle is a `site_id` column. I am not claiming ClickHouse dump.

10 The fix

Stop casting `:site_id`. Bind `site.id` from `conn.assigns` the way Shield and Goal already do after whoareme Bug 02.

PLAIN TEXT Copy

```
def changeset(schema, attrs) do
  schema
  ▷ cast(attrs, [:recipients, :threshold, :type])
```

```
    ▷ validate_required([:recipients, :threshold, :type])
    ▷ validate_number(:threshold, greater_than_or_equal_to: 1)
    ▷ unique_constraint([:site_id, :type])
  end
```

Keep `site_id` on insert from assigns in `enable`. Do not take it from the PUT body.

Re-run the loopback client against a patched build: after PUT with a foreign `site_id`, the row must stay on attacker `sites.id`. Victim path PUT must stay 404.

Until then, do not treat "I patched Shield and Goal" as "every changeset that belongs_to Site stopped casting `:site_id`."

I am not going to print a walk of the email-reports PUT aimed at a live host. The extra cast is the useful part. If you own the box, run the proof of concept against loopback.

Same product, other unpublished lab on this pin: editor People HTML. Series hub: Plausible Analytics leftovers.

11 References

- Proof of concept: abraxas/plausible-traffic-notify · plausible-traffic-notify-Abraxas-Labs.py
- Lab: `lab/` · docker-compose.yml · run.sh · poc.py
- Series: Plausible Analytics leftovers · plausible-editor-invite
- @abraxas_null · github.com/abraxas · abraxaslabs.tech · abraxas.null@proton.me
- CWE-915 · CWE-639

- Tree: plausible/analytics master `d44e104ae3943a9240d5cb6b62e67dff0f069d3a` · `traffic_change_notification.ex` · `site_controller.ex`
- Nearby patched (not this lookup): whoareme 2026-08-25 Bug 02 Shield / Goal `site_id` mass-assignment
- 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: Plausible Analytics, editor People HTML leaks transfer nanoid

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

Home Research Open Source Reading Room About GitHub @abraxas_null Console

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

© 2026 Abraxas Labs