Set-Cookie Cross-Origin Regression — RCA + Fix Pattern

Bilko Set-Cookie Cross-Origin Regression — RCA + Fix Pattern

MC: #9499 (final fix), #9495 (canary discovery), #9398 (original same-origin fix)
Resolved: 2026-04-27
Final fix: bilko-web rev `00029-zkp` + bilko-api rev `00062-gwx`

Problem

User authentication failed on Bilko demo despite successful API login response. Symptoms:

- POST `/auth/login` → HTTP 200 with valid user/org/tokens payload
- `refreshToken` cookie NOT stored in browser
- Subsequent `/auth/refresh` → HTTP 401 "No refresh token"
- User remained on `/login` page, unable to access `/dashboard`

This occurred **despite MC #9398 fixing the same issue 2 days earlier** — indicating a regression.

Root Cause (Compound 2-Layer)

Layer 1: Cross-eTLD+1 Boundary

Frontend: `bilko-demo.alai.no`
Backend API (actual target): `bilko-api-762788903040.europe-north1.run.app`

These are **different registrable domains** (alai.no vs run.app). Cookies with `SameSite=Strict` or `SameSite=Lax` cannot be stored cross-origin when the origins differ at the eTLD+1 level.

The browser rejects the `Set-Cookie` header entirely — no cookie is stored, no cookie is sent to `/auth/refresh`.

Fix in MC #9398: Domain mapping created `bilko-demo-api.alai.no` → Cloud Run service, making frontend and API share the same registrable domain (alai.no). SameSite=Lax allows same-site cookies across subdomains.

Layer 2: Next.js `NEXT_PUBLIC_*` Baked at BUILD TIME

In Next.js, environment variables prefixed with `NEXT_PUBLIC_` are **inlined at compile time** by Webpack.

```
// Code written by developer:
const apiUrl = process.env.NEXT_PUBLIC_API_URL

// Code in compiled bundle after build:
const apiUrl = 'https://bilko-api-762788903040.europe-north1.run.app/api/v1'
```

Consequence: Setting or updating `NEXT_PUBLIC_API_URL` at **runtime** (via Cloud Run service environment variables) has **ZERO EFFECT**. The old URL remains baked into the JavaScript bundle from the previous build.

Evidence: MC #9499 canary-postfix test showed:

- Cloud Run service env var set to `NEXT_PUBLIC_API_URL=https://bilko-demo-api.alai.no/api/v1`
- Deployed frontend still made requests to `bilko-api-762788903040.europe-north1.run.app`
- No subdomain URL found in compiled JS bundle

Fix: Docker image must be **rebuilt** with `--build-arg NEXT_PUBLIC_API_URL=https://bilko-demo-api.alai.no/api/v1` to bake the correct URL into the bundle.

Failed Attempts (Lessons Learned)

Attempt 1 — Domain Mapping Only (MC #9398)

What was done:

- Created `bilko-demo-api.alai.no` subdomain pointing to Cloud Run
- SameSite=Lax cookie policy on backend
- Frontend deployed with runtime env var (not rebuild)

Result: Worked initially because the previous build happened to have the correct URL. Regressed on next deploy when image was rebuilt without `--build-arg`, reverting to hardcoded `.run.app` URL.

Lesson: Domain mapping is necessary but not sufficient. Frontend bundle content matters.

Attempt 2 — Cloud Run Runtime Env Only (MC #9495 ? #9499, first iteration)

What was done (Hadi Hariri):

- Set `NEXT_PUBLIC_API_URL` via `gcloud run services update --set-env-vars`
- Restarted service (but did NOT rebuild image)

Result: FAIL. Canary test showed frontend still calling `.run.app` direct URL.

Lesson: Runtime env vars are visible to server-side code but do NOT affect client-side code already compiled into the bundle. Next.js requires rebuild.

Final Fix

Backend (bilko-api)

Update session cookie configuration:

```
SESSION_COOKIE_SECURE=true
SESSION_COOKIE_SAMESITE=lax
```

- **Secure=true:** Cookie only sent over HTTPS (required for SameSite=Lax or None)
- **SameSite=Lax:** Allows cross-subdomain cookies within same registrable domain (alai.no)
- **SameSite=Strict:** Would block even same-registrable-domain if navigation originated externally (too restrictive)

Frontend (bilko-web) — REBUILD

Docker image must be rebuilt with build-time argument:

```
docker build \
  --build-arg NEXT_PUBLIC_API_URL=https://bilko-demo-api.alai.no/api/v1 \
  -f apps/web/Dockerfile \
  -t bilko-web:00029-zkp \
  .
```

Dockerfile must declare the ARG and set ENV:

```
ARG NEXT_PUBLIC_API_URL
ENV NEXT_PUBLIC_API_URL=$NEXT_PUBLIC_API_URL
```

Then deploy new revision to Cloud Run. Runtime env var should **also** be set (for server-side rendering), but rebuild is mandatory.

Verification

?? CRITICAL: `curl` is NOT a Valid Oracle for SameSite

Testing with `curl` or `fetch` does NOT prove cookie storage. The `Set-Cookie` header may appear in response headers but the browser's cookie jar enforcement is separate.

SameSite restrictions apply to **browser cookie storage**, not HTTP-level headers. Only a **real browser test** with cookie jar inspection proves success.

Tools used:

- Playwright with `context.cookies()` API
- Browser DevTools Application → Storage → Cookies

Canary Test Results

Three iterations:

1. **MC #9495 canary:** FAIL — frontend calling `.run.app` URL, no cookie stored

2. **MC #9499 canary-postfix (runtime env only):** FAIL — frontend still calling `.run.app`, no rebuild
3. **MC #9499 canary-rebuild (full fix):** PASS — all 5 acceptance criteria met

Final Pass Criteria (canary-rebuild.md):

#	Criterion	Result
1	All API URLs use `bilko-demo-api.alai.no` (NOT `.run.app`)	PASS
2	`refreshToken` cookie stored (sameSite=Lax, secure=true, httpOnly=true)	PASS
3	`/auth/refresh` returns 200 (app-initiated flow, ignoring test artefact 403)	PASS
4	Dashboard URL stays `/dashboard` (not redirected to `/login`)	PASS
5	Authenticated dashboard shows seed data (5.1M RSD cash, charts)	PASS

Next.js Frontend Deploy Checklist

To prevent this regression in future deploys:

1. **ALL `NEXT_PUBLIC_*` env vars must be `--build-arg`** when building Docker image
2. **Dockerfile MUST declare ARG + ENV:**

```
ARG NEXT_PUBLIC_API_URL
ENV NEXT_PUBLIC_API_URL=$NEXT_PUBLIC_API_URL
```

3. **After deploy:** Bundle inspection to verify URL baked correctly:

```
# Extract and inspect JS chunks
grep -r "bilko-demo-api.alai.no" .next/static/chunks/
```

4. **Set runtime env too** (for server-side rendering and consistency)
5. **Cross-origin cookies:** Frontend and API must share same **registrable domain** (e.g., `*.alai.no`). SameSite=Lax allows same-site, different subdomain.

Cloud Build Pattern

Current `cloudbuild.yaml` (lines 8-11, 143-145):

```
substitutions:
  _API_URL: https://bilko-api-762788903040.europe-north1.run.app/api/v1 # ⚠ WRONG

steps:
  - id: build-web
    args:
      - --build-arg NEXT_PUBLIC_API_URL=$_API_URL # Uses substitution
```

Good: Uses `--build-arg` with substitution variable.

⚠ **OPEN ISSUE:** Default `_API_URL` is `.run.app` direct URL, not the subdomain. This means builds triggered from GitHub **without manual substitution override** will bake the wrong URL.

Required fix: Update default substitution:

```
substitutions:
  _API_URL: https://bilko-demo-api.alai.no/api/v1 #  Correct subdomain
```

This requires **followup MC task** to update `cloudbuild.yaml` and redeploy to verify.

Cross-References

- **MC Tasks:**
  - #9398 — Original same-origin fix (domain mapping created)
  - #9495 — Canary discovery (regression confirmed)
  - #9499 — Final fix (rebuild + SameSite=Lax)
  - #9529 — Cloud Build (contains current cloudbuild.yaml)
  - *(pending)* — Fix cloudbuild.yaml default `_API_URL` substitution
- **Memory:**
  - `feedback_curl_is_not_browser_test.md` — curl HTTP 200 ≠ demo works
  - `feedback_deploy_verification_protocol.md` — ZAKON PI2 deploy gates
- **Evidence:**
  - `docs/evidence/9495/canary.png` — Screenshot showing unauthenticated /login
  - `docs/evidence/9499/canary-postfix.md` — FAIL after runtime env only
  - `docs/evidence/9499/canary-rebuild.md` — PASS after full rebuild

Key Takeaways

1. **Domain alignment is necessary but not sufficient** — frontend and API must share registrable domain, AND frontend code must target that domain.
2. **Next.js NEXT*PUBLIC** variables are build-time constants** — runtime env vars do NOT update client-side code. Always rebuild when changing public env vars.
3. **curl/fetch tests cannot validate cookie storage** — SameSite enforcement happens in browser cookie jar, not HTTP layer. Use Playwright or manual browser inspection.
4. **SameSite=Lax is the right balance** for same-registrable-domain subdomains. SameSite=Strict blocks legitimate cross-subdomain flows. SameSite=None is too permissive (requires CSRF tokens everywhere).
5. **Regression prevention requires CI enforcement** — Cloud Build substitutions must have correct defaults to avoid silent regressions on automated deploys.

Revision #3
Created 2026-04-27 04:28:45 UTC by John
Updated 2026-07-05 20:05:23 UTC by John