WAF Rules

WAF Rules — Drop Payment App

MC #1229 — Web Application Firewall configuration for Drop fintech.

Overview

Drop runs on Fly.io which does not provide a built-in WAF. Protection is layered:

1. **Middleware-level** (Next.js Edge Middleware) — first line of defense
2. **Fly.io Proxy** — TLS termination, DDoS mitigation at network edge
3. **Application-level** — input validation, parameterized SQL, CSRF checks

Middleware WAF Rules (Implemented in `src/drop-app/src/middleware.ts`)

1. CSRF Origin Validation

- **Rule:** All mutation requests (POST/PUT/PATCH/DELETE) to `/api/*` must have valid `Origin` or `Referer` header
- **Action:** Block with 403
- **Bypass:** None

2. Rate Limiting

- **Rule:** Per-IP rate limits on auth endpoints (10 req/window)
- **Action:** Block with 429
- **Scope:** `/api/auth/*`

3. Content-Security-Policy

- **Rule:** Strict CSP with nonce-based script/style loading in production
- **Action:** Browser enforcement (block inline scripts/styles without nonce)
- **Dev mode:** `unsafe-inline` permitted for HMR

Recommended Reverse Proxy Rules (Fly.io / Cloudflare)

If a CDN or reverse proxy is added in front of Fly.io, configure these rules:

SQL Injection (SQLi)

- **Pattern:** Block requests containing SQL keywords in query params and body:
  - `UNION SELECT`, `OR 1=1`, `DROP TABLE`, `; --`, `' OR '`
- **Action:** Block with 403
- **Note:** Drop uses parameterized queries exclusively — this is defense-in-depth

Cross-Site Scripting (XSS)

- **Pattern:** Block requests containing:
  - `<script>`, `javascript:`, `on\w+=`, `<img.*onerror`
- **Action:** Block with 403
- **Note:** React auto-escapes output; CSP blocks inline scripts

Path Traversal

- **Pattern:** Block requests containing:
  - `../`, `..\\`, `%2e%2e`, `/etc/passwd`, `/proc/self`
- **Action:** Block with 403

Request Size Limits

- **Rule:** Max request body 1MB (API), 10KB (auth endpoints)
- **Action:** Block with 413

Geo-blocking (Optional)

- **Rule:** Drop targets Norway/Scandinavia. Consider restricting to EU/EEA IPs for reduced attack surface.
- **Action:** Block with 403 for non-allowed regions
- **Note:** Requires Cloudflare or similar CDN with geo-IP support

Bot Protection

- **Rule:** Rate limit on `/api/auth/*` endpoints (already in middleware)
- **Supplemental:** Add CAPTCHA challenge after 3 failed BankID attempts
- **Action:** Challenge or block

Implementation Priority

Priority	Rule	Status
P0	CSRF Origin check	Implemented (middleware.ts)
P0	CSP headers	Implemented (middleware.ts + next.config.ts)
P0	Rate limiting	Implemented (per-endpoint)
P1	Trivy container scan	Implemented (CI/CD)
P1	npm audit	Implemented (CI/CD)
P2	SQLi WAF rules	Pending — requires CDN/proxy
P2	XSS WAF rules	Pending — requires CDN/proxy
P2	Path traversal rules	Pending — requires CDN/proxy
P3	Geo-blocking	Pending — requires CDN/proxy
P3	Bot protection (CAPTCHA)	Pending — requires frontend integration

Testing WAF Rules

When WAF rules are deployed via CDN:

```
# Test SQLi blocking
curl -X POST "https://getdrop.no/api/test" -d "id=1 OR 1=1"
# Expected: 403 Forbidden

# Test XSS blocking
```

```
curl -X POST "https://getdrop.no/api/test" -d "name=<script>alert(1)</script>"
# Expected: 403 Forbidden

# Test path traversal blocking
curl "https://getdrop.no/../../etc/passwd"
# Expected: 403 Forbidden
```

Monitoring

- All WAF blocks should be logged with: timestamp, rule ID, client IP, request path, matched pattern
- Alert on >100 blocks/hour from single IP (potential attack)
- Weekly WAF report for security review

Revision #6
Created 2026-02-18 08:44:29 UTC by John
Updated 2026-05-23 10:58:52 UTC by John