Security Architecture

Overview

This document describes the security measures implemented in Spotify Genre Sorter.

Authentication

OAuth 2.0 Flows

Two authentication modes are supported:

1. **Spotify-Only Mode** (Default)
    - Direct Spotify OAuth authentication
    - No GitHub dependency required
2. **GitHub + Spotify Mode**
    - GitHub OAuth for access control (whitelist)
    - Spotify OAuth for music data access

Token Security

Measure	Implementation
Server-side storage	OAuth tokens stored in Cloudflare KV, never sent to browser
Session ID only	Browser receives only a UUID session identifier
HttpOnly cookies	Session cookie cannot be accessed by JavaScript
Secure flag	Cookies only transmitted over HTTPS
SameSite=Lax	Prevents CSRF attacks
Single-use state tokens	OAuth state tokens deleted after verification
Session expiry	Sessions expire after 7 days

Session ID Entropy

- Generated using `crypto.randomUUID()` (UUIDv4)
- 122 bits of entropy
- Cryptographically secure random generation

API Security

Rate Limiting

```
30 requests per minute per IP address
```

- In-memory rate limiter tracks requests per IP
- Returns HTTP 429 with `Retry-After` header when exceeded

- Periodic cleanup of expired entries

Input Validation

Input	Validation
Track IDs	Must match `/^[a-zA-Z0-9]{22}$/` (Spotify format)
Genre names	Max 100 characters, dangerous chars stripped (`<>"'&`)
Bulk requests	Max 50 genres, max 10,000 tracks

CORS Policy

```
cors({
  origin: (origin) => origin,  // Same-origin only
  allowMethods: ['GET', 'POST'],
  maxAge: 3600,
})
```

- No cross-origin API access allowed
- Preflight caching for 1 hour

HTTP Security Headers

All responses include:

```
Content-Security-Policy:
  default-src 'self';
  script-src 'self' 'unsafe-inline';
  style-src 'self' 'unsafe-inline';
  img-src 'self' data: https:;
  font-src 'self';
  connect-src 'self' https://api.spotify.com https://api.github.com;
  frame-ancestors 'none';

X-Frame-Options: DENY
X-Content-Type-Options: nosniff
Referrer-Policy: strict-origin-when-cross-origin
Strict-Transport-Security: max-age=31536000; includeSubDomains
Permissions-Policy: camera=(), microphone=(), geolocation=()
```

Header Purposes

Header	Protection
CSP	Prevents XSS, limits external resources
X-Frame-Options	Prevents clickjacking
X-Content-Type-Options	Prevents MIME sniffing
HSTS	Enforces HTTPS

Referrer-Policy	Limits referrer leakage
Permissions-Policy	Disables unnecessary APIs

Data Storage

Cloudflare KV

Data Type	Key Pattern	TTL	Encryption
Sessions	`session:{uuid}`	7 days	At-rest (CF managed)
OAuth state	`state:{uuid}`	10 minutes	At-rest (CF managed)
Genre cache	`genre_cache_{userId}`	1 hour	At-rest (CF managed)
User registry	`user:{spotifyId}`	No expiry	At-rest (CF managed)

Data Minimisation

- Only essential user data stored (Spotify ID, display name)
- No music library data persisted (cached temporarily)
- OAuth tokens stored server-side only

Spotify API Interaction

Scopes Requested

```
user-library-read      - Read liked songs
playlist-modify-public - Create public playlists
playlist-modify-private - Create private playlists
user-read-private      - Read user profile
```

Rate Limit Handling

- Exponential backoff retry (3 attempts)
- Respects `Retry-After` header from Spotify
- Network error recovery

CI/CD Security

Automated Scanning

Tool	Purpose
CodeQL	Static code analysis on every push
Snyk	Dependency vulnerability scanning
npm audit	Package security audit

Secret Management

- Secrets stored in Cloudflare Workers secrets (not in code)

- GitHub Actions secrets for deployment credentials
- No secrets in repository

Threat Model

Considered Threats

Threat	Mitigation
Session hijacking	HttpOnly + Secure cookies, SameSite
CSRF	SameSite cookies, OAuth state tokens
XSS	CSP headers, input sanitisation
Clickjacking	X-Frame-Options: DENY
Token theft	Server-side storage only
Brute force	Rate limiting
Spotify API abuse	Rate limiting, token refresh

Out of Scope

- DDoS protection (handled by Cloudflare)
- Network-level attacks (handled by Cloudflare)

Incident Response

Suspected Token Compromise

1. User logs out (session deleted)
2. Re-authentication required
3. Spotify tokens automatically refreshed

Suspected Session Compromise

1. Delete session from KV: `wrangler kv:key delete session:{id}`
2. User must re-authenticate

Security Contacts

Report security issues to: Create GitHub Issue

Last updated: December 2025