

STRIDE threat model for an authentication flow

Captured 2026-05-14T07:05:00Z · team tier · claude-opus-4-7

Scenario

You are designing a new B2B authentication flow using OAuth 2.1 + OIDC + Keycloak. Your security architect needs a STRIDE threat model per component with controls.

What was asked

Build a STRIDE threat model for the following authentication architecture: user → Keycloak → resource server → PostgreSQL. List threats per component, mitigations, and cite OWASP / NIST 800-53 / STRIDE patterns where relevant.

Answer

STRIDE threat model — OAuth 2.1 + OIDC + Keycloak

Four components in scope: **user agent** → **Keycloak (auth server)** → **resource server** → **PostgreSQL**. STRIDE per component with mitigations grounded in the gateway's OWASP WSTG corpus and the STRIDE MCP threat-pattern library.

1. User agent ↔ Keycloak (login + token issuance)

Category	Threat	Cited mitigation
Spoofing	Stolen credentials replayed at login	OAuth 2.1 PKCE; MFA. See OWASP — Testing Session Fixation
Spoofing	OIDC redirect-URI tampering steers tokens to an attacker	Strict redirect-URI allowlist; exact-match only. See STRIDE MCP — OAuth 2.0 Access Token Theft and Replay
Tampering	Auth-code interception in the redirect	PKCE S256 challenge binds the code to the originating client. Same OWASP/STRIDE references.
Repudiation	User denies they consented to the scopes they were granted	Keycloak's consent screen + audit log of grants. (No on-point citation in this gateway run — flagged for review.)
Information disclosure	Refresh token leak grants long-lived access	STRIDE MCP — Predictable Token Generation via Weak Random Number Generators . Use rotating refresh tokens; bind tokens to the client via DPoP.
DoS	Brute-force on login endpoint	Rate-limit by IP + by username; account lockout escalation.
Elevation		OWASP — Testing for Session Hijacking . Regenerate session ID on auth-state change.

Category	Threat	Cited mitigation
	Session fixation: attacker plants a session ID before login	

2. Resource server (validates Bearer tokens)

Category	Threat	Cited mitigation
Spoofting	Forged JWT signed by an attacker key	Validate <code>iss</code> + JWKS pinned to Keycloak; reject <code>alg: none</code> . See OWASP — Testing for Weak Encryption .
Spoofting	Token replayed after logout	OWASP — Testing for Logout Functionality . Keep token TTL short (≤ 15 min); use a revocation list for the gap.
Tampering	WebSocket upgraded from an attacker origin then attached to the user's session	STRIDE MCP — WebSocket Origin Validation Bypass . Validate <code>Origin</code> on the upgrade; require an Origin-bound token.
Repudiation	API write is later disputed	Structured audit log on every state-changing call: actor, resource, request hash, JWT <code>jti</code> .
Information disclosure	Token leaked in URL query string in referer headers	Carry tokens in <code>Authorization</code> header only; never in query.
DoS	Expensive endpoint hit unauthenticated	Rate-limit by JWT <code>sub</code> ; cap unauthenticated paths separately.
Elevation	Scope-creep: a token with <code>read</code> accepted on a write endpoint	Per-endpoint scope check; default-deny if scope absent.

3. Resource server ↔ PostgreSQL

Category	Threat	Cited mitigation
Tampering	SQL injection via crafted input	OWASP — Testing for SQL Injection , plus its Objectives 1–3. Parameterised queries; ORM with bound parameters; principle-of-least-privilege DB role.
Information disclosure	Verbose DB errors leak schema in API responses	Generic 5xx; structured logs (server-side only).
Elevation	DB role shared across services has SUPERUSER	Per-service DB role; row-level security where tenants share tables.

4. Cross-cutting (audit, logging, key management)

- **Repudiation, fleet-wide:** forensic-grade structured logs per Ansvr ADR-001 (actor, action, target, request hash, signing-key id, timestamp from a monotonic clock).
- **Information disclosure, fleet-wide:** TLS 1.2+ on every hop; secrets in HashiCorp Vault; no static API keys checked into code.
- **Tampering, fleet-wide:** signed audit chain — each event references the prior event's hash.

Gateway run notes

The **Security Controls MCP** (the NIST 800-53 corpus) returned only 1 weakly-relevant row when searched for IA-2 IA-5 AC-3 . The gateway did not produce on-point NIST 800-53 control IDs for this query in this run, so the model declined to cite specific NIST control numbers from training data. Re-run after a corpus refresh for richer 800-53 grounding.

For your three-component architecture, the **load-bearing controls today** are: PKCE, exact-match redirect URI allowlist, session-ID regeneration on state change, parameterised queries, per-service DB roles, and short token TTL + revocation list. Everything else is reinforcement.

Tool calls

#	Tool	Result
1	search	6 results from OWASP MCP — Session Timeout (×4), Weak Encryption (×2)
2	search	1 weakly-relevant result from Security Controls MCP; insufficient for NIST 800-53 citations
3	search	4 results from OWASP MCP — Testing for SQL Injection + Objectives 1/2/3
4	search	10 results: OWASP Session Fixation/Logout/Management/Hijacking + STRIDE OAuth replay, Predictable Tokens, WebSocket Origin Bypass

Citations

Type	Jurisdiction	Reference	URL
guidance	intl	OWASP WSTG — Testing for Session Hijacking	https://github.com/OWASP/wstg/blob/v4.2/document/4-Web_Application_Security_Testing/06-Session_Management_Testing/09-Testing_for_Session_Hijacking.md
guidance	intl	OWASP WSTG — Testing for Session Fixation	https://github.com/OWASP/wstg/blob/v4.2/document/4-Web_Application_Security_Testing/06-Session_Management_Testing/03-Testing_for_Session_Fixation.md
guidance	intl	OWASP WSTG — Testing for Logout Functionality	https://github.com/OWASP/wstg/blob/v4.2/document/4-Web_Application_Security_Testing/06-Session_Management_Testing/06-Testing_for_Logout_Functionality.md
guidance	intl	OWASP WSTG — Testing for Weak Encryption	https://github.com/OWASP/wstg/blob/v4.2/document/4-Web_Application_Security_Testing/09-Testing_for_Weak_Cryptography/04-Testing_for_Weak_Encryption.md
guidance	intl	OWASP WSTG — Testing for SQL Injection	https://github.com/OWASP/wstg/blob/v4.2/document/4-Web_Application_Security_Testing/07-Input_Validation_Testing/05-Testing_for_SQL_Injection.md
	intl		https://github.com/Ansvar-Systems/stride-mcp

Type	Jurisdiction	Reference	URL
threat-pattern		STRIDE MCP — OAuth 2.0 Access Token Theft and Replay (Spring Boot and Express.js)	
threat-pattern	intl	STRIDE MCP — Predictable Token Generation via Weak Random Number Generators	https://github.com/Ansvar-Systems/stride-mcp
threat-pattern	intl	STRIDE MCP — WebSocket Origin Validation Bypass	https://github.com/Ansvar-Systems/stride-mcp

Generated by Ansvar AI Gateway. This is the same artifact a customer would receive.

<https://gateway.ansvar.eu> · 2026-05-14T07:05:00Z