**Document: 17 – Threat Model**

**Project:** MCPX-KendoBridge Admin Portal  
**Document ID:** TJ-MCPX-DOC-17  
**Version:** 1.0.0  
**Status:** Draft  
**Date:** 2025-09-27  
**Owner:** Security & Compliance (Technijian)  
**Confidentiality:** Technijian Internal

**Document Control**

| **Version** | **Date** | **Author** | **Change Summary** | **Status** |
| --- | --- | --- | --- | --- |
| 1.0.0 | 2025-09-27 | Security & Compliance | Initial STRIDE threat model | Draft |

**Approvals**

| **Role/Team** | **Name** | **Signature/Date** | **Comment** |
| --- | --- | --- | --- |
| Director of Software Eng. |  |  |  |
| Systems Architect |  |  |  |
| DevSecOps / SRE |  |  |  |
| DBA Lead |  |  |  |
| QA Lead |  |  |  |

**1. Purpose & Method**

Establish a **STRIDE**-based threat model for the MCPX-KendoBridge Admin Portal and map **mitigations → controls → evidence** so risks are consistently treated through design, implementation, and release.

* **Method:** STRIDE (Spoofing, Tampering, Repudiation, Information disclosure, Denial of service, Elevation of privilege) across **processes, data flows, data stores, and trust boundaries**.
* **Inputs:** 04 Context & Architecture, 05 NFRs, 07/07a DB contracts & grants, 08/08a UI, 09 Test Strategy, 10 CI/CD, 11 Monitoring, 12 Evidence, 13 Compliance.

**2. System Overview & Assets**

**2.1 Primary Assets (what we protect)**

* **Administrative access** to MCP servers and operations.
* **Tokens/claims** (AAD ID/Access tokens with roles).
* **Configuration/flags/lookups** (non-secret runtime state).
* **AuditEvent log** (who/what/when/before→after).
* **SSE channels** (job progress, notifications).
* **OpenAPI contract & service endpoints**.
* **Evidence Pack & parity reports** (release integrity).

**2.2 Data Classification (quick view)**

* **Regulated**: PHI/PII (if present under tenant scope) — **not stored/returned** by API.
* **Confidential**: config values (non-secret), audit events, parity reports.
* **Internal**: logs/metrics, release artifacts (SBOM/SARIF).
* **Public**: none.

**3. Trust Boundaries & DFD**

**3.1 Trust Boundaries**

1. **Browser ↔ Admin API** (TLS; JWT; CORS allow-list).
2. **Admin API ↔ SQL Server** (SP-only; least-privilege EXECUTE).
3. **Admin API ↔ Azure AD / Microsoft Graph** (OIDC/MSAL; Graph optional).
4. **Admin API ↔ Child MCP** (local STDIO bridge).
5. **RTM ↔ Prod parity** (promotion gate).

**3.2 Level-0 Data Flow Diagram (PlantUML)**

@startuml

left to right direction

actor "Portal Admin" as Admin

actor "Portal Viewer" as Viewer

rectangle "Azure AD (Entra ID)" as AAD

rectangle "Microsoft Graph" as Graph

node "Admin Web\n(React + KendoReact Fluent 2)" as Web

node "Admin API\n(.NET 8)" as API

database "SQL Server 2022\nSP-only, add-only" as DB

cloud "Child MCP\n(STDIO)" as MCP

Admin --> Web

Viewer --> Web

Web --> AAD : MSAL PKCE\n(OIDC)

Web --> API : HTTPS (JWT Bearer)\nCORS allow-list

API --> DB : Stored Procedures

API --> MCP : JSON-RPC via STDIO

API --> Graph : Directory ops (optional)

@enduml

**4. STRIDE by Element (Top Risks & Mitigations)**

Severity: **H**igh / **M**edium / **L**ow. *Controls reference existing docs.*

**4.1 Browser ↔ API (boundary)**

* **S – Token spoofing / session fixation** (**H**)  
  **Mitigations:** MSAL **PKCE**; validate iss/aud/exp/sig; short-lived access tokens; no tokens in localStorage (session/secure cookie); logout clears session.  
  **Evidence:** OpenAPI security; login E2E; Code review; pen-test notes.  
  **Monitoring:** auth failures spike alert; invalid token codes.
* **T – Tampering with requests / CSRF** (**M**)  
  **Mitigations:** Bearer-only API (no cookie-implicit auth); idempotency keys on mutations; server-side RBAC; request validation against OpenAPI schema.  
  **Evidence:** Contract tests; negative tests (invalid schema); logs with requestId.
* **R – Repudiation of admin actions** (**H**)  
  **Mitigations:** **AuditEvent** (who/what/when/before→after + requestId); immutable store; clock sync.  
  **Evidence:** Audit exports; evidence pack.
* **I – Info disclosure via errors/CORS** (**H**)  
  **Mitigations:** **Error envelope** (no stack traces); **CORS allow-list** deny-by-default; /config/effective exposes **non-secrets only**.  
  **Evidence:** Error catalog; CORS tests; contract tests for config.
* **D – DoS / resource abuse (JSON & SSE)** (**H**)  
  **Mitigations:** **Rate limits** (per principal/IP/session); max SSE connections; server timeouts & heartbeats; circuit breakers to child/DB.  
  **Evidence:** NFR policy; 429 tests; k6 SSE thresholds; alerts.
* **E – Elevation via UI-only checks** (**H**)  
  **Mitigations:** **Server-enforced RBAC** (app roles/groups); UI hides actions but server authoritative; Graph write behind admin consent.  
  **Evidence:** Negative tests (viewer → 403); code review.

**4.2 Admin API ↔ SQL Server**

* **S – Impersonation at DB** (**M**)  
  **Mitigations:** Dedicated **app\_mcpx** user; no table DML; **EXECUTE-only** on whitelisted SPs.  
  **Evidence:** Grants script; runtime connection user check.
* **T – SQL injection / tampering** (**H**)  
  **Mitigations:** **SP-only**, parameterized inputs; add-only schema; constraints; SARGable predicates.  
  **Evidence:** SAST; integration tests; Query Store baselines.
* **R – Repudiation** (**M**)  
  **Mitigations:** Every mutation SP writes **AuditEvent**; API includes requestId param.  
  **Evidence:** SP bodies; audit rows for test mutations.
* **I – Sensitive data exposure** (**M**)  
  **Mitigations:** Secrets not stored/returned; non-secret policy enforced in SPs; separate secret store.  
  **Evidence:** Contract & unit tests.
* **D – DB saturation** (**M**)  
  **Mitigations:** Indexing; perf SLOs (reads p95 ≤ 75 ms, writes ≤ 150 ms); rate-limits; query timeouts.  
  **Evidence:** k6 + Query Store; alerts on p95 breach.
* **E – Privilege escalation** (**M**)  
  **Mitigations:** No table rights; grants PR review; signature policy; \_v2 SP for breaking change.  
  **Evidence:** Signature snapshot in Evidence Pack.

**4.3 Admin API ↔ Child MCP (STDIO)**

* **S/T – Untrusted child process** (**M**)  
  **Mitigations:** Strict spawn args/WorkingDir; input/output validation; timeouts; kill on idle; map errors to envelope.  
  **Evidence:** Config keys; integration tests.
* **D – Backpressure / event flood** (**M**)  
  **Mitigations:** SSE heartbeats; output throttling; max message size; per-job quotas.  
  **Evidence:** SSE metrics; k6 thresholds; alerts.

**4.4 Azure AD / Microsoft Graph**

* **S – App spoofing / consent phishing** (**M**)  
  **Mitigations:** Verified app publisher; admin consent reviewed; least scopes; conditional access (if policy).  
  **Evidence:** Tenant app reg; consent logs.
* **E – Unauthorized role assignment** (**H**)  
  **Mitigations:** App roles bound to approved groups; Graph write requires explicit consent; or **Change Request** path only.  
  **Evidence:** Access flows in UI; audit entries; approvals.

**4.5 Parity Gate (RTM ↔ Prod)**

* **T/I – Misconfig leading to prod impact** (**H**)  
  **Mitigations:** **Config Parity diff**; **0 critical** required; promotion blocked; evidence retained ≥ 1 year.  
  **Evidence:** Parity report in release; gate logs.

**5. Attack Scenarios (Top 10) → Controls**

| **#** | **Scenario (abbrev)** | **Risk** | **Key Controls** |
| --- | --- | --- | --- |
| 1 | Stolen token replay | H | MSAL PKCE; short-lived tokens; JWT validation; session/secure cookie; logout |
| 2 | CORS origin spoof | H | DB-backed allow-list; preflight enforced; deny-by-default; audited edits |
| 3 | JSON injection | H | OpenAPI schema validation; SP-only; parameterized SPs |
| 4 | SSE exhaust | H | Per-principal/IP stream caps; timeouts; heartbeats; k6 perf gate |
| 5 | RBAC bypass (UI only) | H | Server-side RBAC; 403 on forbidden; negative tests |
| 6 | Flag/config drift | M | AuditEvent; parity gate; approval on risky changes |
| 7 | Legacy endpoints exposure | M | Feature flag **OFF** by default; 403 feature\_disabled; audits |
| 8 | DB privilege creep | M | EXECUTE-only grants; signature snapshot; grants PR approval |
| 9 | Secret leak via config | M | /config/effective returns **non-secrets** only; tests |
| 10 | CSP/XSS enabling token theft | M | CSP (report-only → enforce); no tokens in localStorage; a11y scan also catches some issues |

**6. Mitigation Plan & Residual Risk**

| **Risk** | **Owner** | **Current Control** | **Residual** | **Action / Due** |
| --- | --- | --- | --- | --- |
| SSE DoS | SRE | Rate limits; timeouts; k6 gate | Low-M | Add reconnect jitter test (09) / Sprint 3 |
| CORS misconfig | DL/SEC | DB allow-list; audit | Low | Two-person approval toggle (09/11) / Sprint 2 |
| Token storage | UX/FE | Session/secure cookie | Low | Verify no localStorage writes in CI / Sprint 1 |
| Graph write abuse | SEC | Consent gates; CR path | Low-M | Default to CR until policy approved / Sprint 4 |
| SP drift | DBA/SA | Signature snapshot | Low | Nightly drift check & ticketing / Sprint 4 |

**7. Security Testing Plan (extract)**

* **SAST:** CodeQL (C# + JS/TS) — fail on high/critical.
* **DAST (auth’d):** ZAP or equivalent against staging (login flow scripted).
* **Contract:** OpenAPI conformance; negative tests for envelope and 403/429.
* **E2E:** Playwright flows incl. **Viewer** negative paths.
* **Perf:** k6 SSE TTFB ≤ 200 ms; heartbeat ≤ 10 s; cap reconnects.
* **A11y & CSP:** axe CI **0 critical**; CSP report-only logs reviewed → enforce.

**8. Security Requirements (normative)**

* TLS 1.2+, HSTS, secure cookies; **no** mixed content.
* **MSAL PKCE**; JWT validation (iss/aud/exp/sig).
* **RBAC** server-side; app roles/groups; least privilege.
* **CORS allow-list** per env; deny by default; audit edits.
* **Rate limits**: JSON 60 req/min/principal; 300 req/min/IP; SSE 5 streams/principal, 20/IP (configurable).
* **SP-only DAL**; add-only schema; grants EXECUTE-only.
* **Error envelope** always; never leak stack traces or secrets.
* **Parity gate** requires 0 critical diffs for Prod promotion.
* **Evidence Pack** retained ≥ 1 year (includes signature snapshot, parity, perf smoke).

**9. Secrets & Keys**

* **Never** in repo or /config/effective.
* Store in a **secret store** (Key Vault/ENV).
* Rotate **Kendo license** via CI secret; follow runbook.

**10. Supply Chain & CI/CD Hardening**

* OpenAPI lint/diff; CodeQL; Dependency Review; Secret Scan; SBOM.
* Pin critical GitHub Actions where feasible; review advisories.
* DocX→MD mirror & TREE.md ensure docs are indexable for review/audit.

**11. Incident Response Summary**

* P1 engage ≤ 15 min; MTTR ≤ 60 min.
* Steps: Detect (alerts) → Contain (rate-limit/flag/rollback) → Eradicate → Recover → **Postmortem**.
* Evidence: timeline, requestId samples, diffs, remediation tasks.

**12. Acceptance Criteria (Threat Model)**

1. STRIDE risks enumerated for each boundary and mapped to specific controls.
2. Controls are **testable** and tied to CI/CD and Monitoring.
3. **Top-10 scenarios** have owners and actions.
4. **Evidence Pack** includes parity, perf smoke, and SP signature snapshot.
5. Threat model reviewed quarterly or on major architectural change (ADR).

**13. Open Issues**

* Confirm CSP enforcement timeline (after report-only burn-in).
* Decide Graph write policy (enable vs CR-only).
* Calibrate final rate-limit numbers based on Beta telemetry.
* Define “critical parity keys” (blocking) vs “major/minor” (advisory).

**End of Document — TJ-MCPX-DOC-17 v1.0.0**