DOWNLOADABLE ENGINEERING REFERENCE / PDF EDITION

Token Lifecycle Review Guide

This guide is a structured way to review access and refresh token behavior across mobile clients and backend services. It is written to surface the mistakes that show up in production: unsafe storage, inconsistent enforcement, and refresh flows that fail under real-world conditions.

It is not a compliance checklist. It is an engineering review guide.

> [!SECURITY] Defensive intent This guide is published to support mitigation and engineering decisions. It is intended for client-owned or explicitly authorized systems and materials.

1) System Map

- Identify all token types used (access, refresh, device tokens, service tokens).
- List where tokens are issued, validated, rotated, and revoked.
- Document which components are allowed to mint or extend sessions.

2) Storage and Exposure

For each token type, answer:

- Where does it live on the client?
  - memory
  - keychain/keystore
  - local database
  - logs/crash reports
- What is the exposure model on compromised devices (root/jailbreak/instrumentation)?
- What is the exposure model on shared devices or backups?

3) Transport and Boundaries

- Verify all token transport uses modern TLS and does not allow downgrade paths.
- Confirm tokens are never placed in URLs, redirects, or referrers.
- Validate CORS and origin boundaries for web-adjacent flows (if any).
- Ensure the backend is skeptical of the client:
  - authorization decisions should not rely on client-provided claims without server enforcement

4) Rotation and Replay Risk

- Define rotation policy for refresh tokens:
  - rotation on use
  - rotation on risk events
  - rotation on privilege changes
- Identify replay windows introduced by:
  - concurrent refresh requests
  - retry storms
  - offline mode
- Confirm the system has a clear answer for “what happens if a refresh token leaks?”

5) Session Scope and Least Privilege

- Avoid “god sessions” where a token silently expands privileges over time.
- Keep session scope tight:
  - user vs admin scopes
  - device-specific session constraints where meaningful
  - risk-based escalation where justified
- Make server-side enforcement explicit and testable.

6) Failure Modes (Real-World)

Test refresh and session behavior under:

- clock skew and time jumps
- background/foreground transitions
- network flaps and captive portals
- partial outages (auth service down, API up, or vice versa)
- token cache desynchronization across devices

7) Observability Without Leakage

- Ensure authentication telemetry exists, but does not log secrets:
  - correlation IDs
  - token “kind” and reason codes
  - rotation events
  - revocation reasons
- Ensure incident responders can reconstruct session history without raw token exposure.

8) Clear Ownership and Review Outputs

Outputs that make the work actionable:

- trust-boundary map for auth and session extension
- prioritized risk list tied to exploitability and impact
- recommended token lifecycle model
- validation plan and retest criteria

9) Practical deliverables (what “done” looks like)

Artifact	Why it matters
Trust-boundary map	Makes authority and enforcement points explicit across app, API, and internal teams.
Invariants list	Converts “security intent” into testable, server-side statements.
Failure-mode matrix	Ensures refresh/session logic survives retries, offline mode, and partial outages.
Retest plan	Lets teams validate remediation without ambiguous “it seems fixed” outcomes.

Public engineering reference, not a completed assessment, legal advice, audit, certification, or customer deliverable.

Editable Markdown source: demonicbinary.com/downloads/token-lifecycle-review-guide.md