Security Pillars

License Integrity & User Authentication (KA-MOD-018)

Ensuring that both the software used and the individuals using it are verified, authorized, and compliant.

1. User Authentication: Verifying Identity

Identity Proofing

The initial step of establishing trust. Verify that the asserted identity (e.g., username, role) maps accurately to a real, known entity.

- Requires multi-factor checks.
- Prevents impersonation.

Access Control (Authorization)

Authentication only confirms identity; authorization defines **what they can do.** Ensure users only access resources relevant to their role (Principle of Least Privilege).

- Role-Based Access Control (RBAC).
- Strict permissions auditing.

Session Integrity

Maintain continuous validation throughout the session. Implement short session timeouts and continuous authentication checks (e.g., token verification).

- Regular token renewal.
- Monitoring for anomalous activity.

2. License Integrity: Ensuring Compliance & Validity

Software Validation

Verify the software's identity through **digital signatures** and **certificate checks** to ensure it hasn't been tampered with or replaced by malicious code.

- Code signing verification.
- Trusted source repository checks.

Compliance Auditing

Ensure all software usage adheres to the purchased license terms (e.g., user count, environment type, duration). Prevents legal risk and unauthorized expansion.

- Automated license server checks.
- Usage metering and reporting.

Anti-Tampering Measures

Implement countermeasures to prevent or detect unauthorized attempts to bypass license controls (e.g., cracking, license key generation).

- Obfuscation techniques.
- Regular environment integrity checks.

Consequences of Failure

Authentication Failure

Leads to **Unauthorized Access**, data breaches, and compromise of system trust boundaries.

License Integrity Failure

Leads to **Legal Penalties**, non-compliance, and operational risk from unpatched, pirated software.

KA-MOD-018 | The foundation of a secure platform is the **dual verification** of both the user and the tools they use.