CHECKLIST

Mobile App Deployment Security Checklist for Smart Home Ecosystems

The Smart Home - with multiple IoT devices, mobile apps, APIs, and a rapidly evolving ecosystem - is shaping up to look quite similar to the world of connected cars. The bad news is that if we are not careful, the proliferation of apps and APIs could open up opportunities for hackers. See this blog for an overview of the issues and best practices.

This checklist is designed to complement the blog, offering developers a practical guide for securing Smart Home apps and APIs. It outlines a comprehensive set of potential issues, along with recommended mitigation strategies for each one.

Each section represents a category of mobile app security challenges and includes a clear description of **why it matters** and **how to address it**.

1. Weak App Integrity Protections

Security Issues	Why It Matters	How to Address It
Lack of App Attestation	APIs can’t distinguish real apps from emulators, bots, or fake clients.	Integrate **mobile attestation** (e.g., Approov) to verify app authenticity for every API request.
Repackaged or Tampered Apps	Malicious code can be injected into legitimate apps without detection.	Use attestation and **tamper detection** to block altered apps; sign builds securely.
No Detection of Rooted/Jailbroken Devices	Compromised OS environments enable privilege escalation and bypass of controls.	Implement **runtime checks** for rooted/jailbroken status and block API access accordingly.
Bypass of Obfuscation	Attackers reverse-engineer code using Frida, JADX, or dynamic instrumentation.	Combine obfuscation with **RASP, debugger detection**, and API-side attestation.

©2025 Approov Limited. All rights reserved. Version 1.0

🔐 2. Hardcoded Secrets and Credentials

Security Issues	Why It Matters	How to Address It
API Keys Hardcoded in the APK/IPA	Exposed secrets can be reused in attacks or scripts.	Remove all hardcoded secrets. Use **just-in-time secret delivery** tied to verified app integrity.
Embedded OAuth Tokens	Tokens can be replayed or hijacked for unauthorized access.	Never store long-lived tokens. Use **short-lived, bound tokens** with encrypted storage.
Static TLS Certificate Pins	Pins can be outdated or bypassed by tampered clients.	Use **dynamic TLS pinning** with cloud-side control (e.g., Approov Pinning).

3. Insecure Communication Channels

Security Issues	Why It Matters	How to Address It
No TLS Enforcement	Data and tokens can be intercepted over insecure networks.	Enforce **HTTPS-only connections** and reject any plaintext fallback.
TLS MitM Attacks	Without pinning, TLS can be spoofed to intercept app traffic.	Apply **certificate pinning** and validate TLS using secure APIs.
Lack of DPoP / Token Binding	Tokens can be captured and reused from another environment.	Implement **token binding** to app/device ID or request signature.

🤖 4. Bot and Automation Abuse

Security Issues	Why It Matters	How to Address It
API Access Without Real App Context	Attackers bypass the UI and use scripts or fake apps.	Require attestation and **enforce device context** checks (e.g., platform, build).

©2025 Approov Limited. All rights reserved. Version 1.0

No Device Binding or Token Replay Protection	Tokens are reused across virtual or emulated devices.	Bind tokens to a **verified device** identity and monitor for duplication.
Overuse of Webview/Hybrid Bridges	Insecure JS bridges allow local privilege escalation or injection.	Harden bridges with **whitelisting**, sandboxing, and disable unused interfaces.

5. Privacy and Data Leakage Risks

Security Issues	Why It Matters	How to Address It
Geolocation or Device Data Leak	Sensitive user data may be unintentionally exposed.	Enforce **data minimization** and mask device identifiers in API responses.
Insecure Local Storage	Credentials or sensitive config can be extracted from disk.	Use **encrypted storage** (Keychain, Keystore) and avoid storing secrets client-side.
Telemetry Overexposure	APIs leak more data than necessary (e.g., camera URLs).	Sanitize API responses to expose only what's required for the feature.

6. Improper Runtime Protections

Security Issues	Why It Matters	How to Address It
No RASP (Runtime Application Self-Protection)	Apps can't detect or react to live attacks.	Integrate a **RASP framework** to detect hooking, debugging, or tampering in real time.
Lack of Emulator Detection	Attackers simulate real devices to test or exploit APIs.	Add **emulator detection logic** and block access when detected.
Failure to Lock/Protect API Invocation Paths	Business logic can be exploited by bypassing the UI.	Enforce **server-side validation** and API request signing or attestation.

©2025 Approov Limited. All rights reserved. Version 1.0

7. Inadequate Authorization & Contextual Control

Security Issues	Why It Matters	How to Address It
No Granular Access Control	Any user can perform critical actions beyond their scope.	Implement **fine-grained, role-based API permissions** (RBAC/ABAC).
No Rate Limiting or Lockout Logic	Enables brute-force attacks on PINs or pairing codes.	Add **per-user/device rate limits** and lockouts for sensitive operations.
Over-privileged API Tokens	Tokens grant more access than necessary, increasing risk.	Use **scoped access tokens** with minimal necessary privileges.

8. Slow Security Updates and Patchability

Security Issues	Why It Matters	How to Address It
User-Delayed Updates	Users may run old, vulnerable versions indefinitely.	Encourage updates and **gracefully deprecate old app versions** via API policy.
Hardcoded Logic	Vulnerabilities persist until the next version is deployed.	Move logic and controls to the **backend or dynamic configuration**.
Monolithic Builds	Security updates require full app re-release.	Modularize apps to support **feature flagging and remote config updates**.

✓ Bonus: Top 5 Quick Wins for Mobile App/API Security

1. Add **Approov Mobile Security** for runtime app attestation and secret protection.
2. Remove all **hardcoded secrets** from apps.
3. Block requests from **rooted, emulated, or tampered devices.**

©2025 Approov Limited. All rights reserved. Version 1.0

4. Use **cloud-managed TLS pinning.**
5. **Bind API tokens** to a verified app and device session.

About Approov

Approov's app attestation technology has been adopted by major organisations in high-stakes industries, demonstrating its real-world effectiveness. By reducing API attacks by over 95% and preventing bot attacks, man-in-the-middle exploits, and app tampering, Approov is creating a safer digital ecosystem.

For more information about Approov's mobile security solutions, please visit www.approov.io.

©2025 Approov Limited. All rights reserved. Version 1.0