Security Audit

dunno

Version 1.2.0 · Build 30

Privacy-Focused Android Messaging

Document: DUNNO-SEC-2026-004

Date: 06 August 2026

Classification: Confidential

Scope: Full source code audit (Kotlin / ~130+ files)

Perspective: Published release build (.aab / .apk)

Independent security audit of dunno v1.2.0 (build 30). Audited from the perspective of the published release artifact.

Table of Contents

1. Executive Summary

dunno v1.2.0 (build 30) is the latest release of this privacy-focused Android messaging application. This report presents a security audit of the source code from the perspective of a published release build (.aab on Google Play, .apk on the project website).

The core application — text chat, PTT walkie-talkie, and file attachments — maintains its strong, multi-layered security posture from v1.1.0. All messaging is end-to-end encrypted (NaCl/libsodium) over the Tor network. Local storage is protected by SQLCipher with hardware-backed Keystore passphrases. The UserDataKey remains device-bound via DeviceKeyWrapper. Input validation, rate limiting, and signature verification remain consistent across all core protocols.

This version introduces shared PTT/Fast-Call audio routing (PttAudioRoute), a new speaker toggle UI (PttSpeakerToggle), active-call indicator banner (AppNav), and robust FGS SecurityException fallback handling in both CallForegroundService and FastCallForegroundService. A VISIBILITY_PUBLIC issue identified during the initial v1.2.0 review was promptly resolved — both FGS notifications now correctly use VISIBILITY_PRIVATE, consistent with PttNotifier and CallNotifier.

The core messaging functionality contains no critical or high-severity findings. Two findings (1 MEDIUM, 1 LOW) of limited impact are documented — both carry-overs from v1.1.0 in unchanged source files.

The two optional WebRTC-based features — Fast Call (audio) and Video Call — remain unchanged from v1.1.0. Their privacy trade-offs are protected by mandatory user consent and documented separately in Section 6.

2. Scope & Methodology

In Scope

- Full source code review of ~130+ Kotlin/Java production files
- All files modified for v1.2.0 (10+ files: PTT audio, FGS, UI)
- AndroidManifest.xml, build.gradle.kts, proguard-rules.pro
- Verification that all previous fixes remain intact

Out of Scope

- Server-side infrastructure
- Debug/log code (ProGuard-stripped in release)
- Build configuration not in the published artifact

Methodology

Manual code review with architectural analysis. OWASP Mobile Top 10 (2024) reference. Severity: CRITICAL, HIGH, MEDIUM, LOW.

3. Changes in v1.2.0

Version 1.2.0 focuses on PTT/Call audio infrastructure, FGS robustness, and UI polish. No changes were made to cryptographic, networking, storage, authentication, or WebRTC-related source files.

3.1 PTT & Call Infrastructure

- PttAudioRoute: Shared audio routing class with reference-counted acquire/release. Manages MODE_IN_COMMUNICATION + setCommunicationDevice for speaker/earpiece toggle. Thread-safe via synchronized instanceLock. No security issues.
- CallForegroundService: Refactored with SecurityException fallback. mediaPlayback-only when

microphone FGS type is blocked. Uses MicAwareFgsTypes. State via AtomicReference. FGS notification now uses VISIBILITY_PRIVATE (resolved during audit).
- FastCallForegroundService: Same SecurityException fallback. Uses FastCallFgsLifecycle. FGS notification now uses VISIBILITY_PRIVATE (resolved during audit).

3.2 User Interface

- PttSpeakerToggle: New Compose speaker/earpiece toggle with sound-wave animation. Uses MaterialTheme colors. No security implications.
- AppNav: Active-call indicator banner in navigation shell. Reads CallController.session. No security implications.
- ContactsScreen, IdentityScreen, ChatsScreen, PttScreen, SettingsScreen: Minor updates. No security-relevant changes.

3.3 Verification of Previous Fixes

i All fixes from v1.1.0 remain intact: (a) FailedAttemptManager timestamp (MIGRATION_16_17), (b) ProGuard OkHttp keep-rules (lines 78-83), (c) FLAG_SECURE + recents blackout on incoming call/PTT activities, (d) PttNotifier/CallNotifier + both FGS services now all use VISIBILITY_PRIVATE on lock screen, (e) PinKeyDeriver Arrays.fill() zeroing.

4. Findings — Core Application

The core application has no CRITICAL or HIGH findings. Two findings of limited impact. Both are carry-overs from v1.1.0; their source files were not modified in v1.2.0.

Ref	Severity	Component	Title
M-01	MEDIUM	Networking	Custom HTTP/1.1 parser in OnionHttpClient
L-01	LOW	Networking	XOR obfuscation of relay .onion (self-documented)

i No new findings in the core application for v1.2.0. All FGS notifications (PttNotifier, CallNotifier, CallForegroundService, FastCallForegroundService) now consistently use VISIBILITY_PRIVATE.

MEDIUM M-01: Custom HTTP/1.1 parser in OnionHttpClient

Finding: OnionHttpClient implements a hand-written HTTP/1.1 parser instead of a proven library. Input validation is present (64 header cap, 128-char status line, 8192-char header line, 64 KiB response body). Parsing logic itself is hand-rolled. Unchanged from v1.1.0.

Impact: Limited to interactions with malicious Tor hidden services. All peers are trusted contacts verified via Ed25519 fingerprints.

Recommendation: Consider fuzz-testing the parser as defense-in-depth.

LOW L-01: XOR obfuscation of relay .onion in TurnCredentialClient

Finding: Rolling XOR used to hide TURN relay .onion in the APK. Code explicitly states this is not a security measure. Self-documented. Unchanged from v1.1.0.

Impact: Negligible. Address trivially extractable from the APK.

Recommendation: Acceptable as-is.

5. Security Strengths

All security strengths from v1.1.0 remain applicable. Key highlights:

STRONG S-01: Transport v2 + Inbox v4: E2EE with minimal RAM exposure

ECDH-based per-message keys via cryptoSecretBoxEasy. UserDataKey zeroed after unlock. SessionKey-only in RAM. TEE compromise alone cannot decrypt stored messages.

STRONG S-02: Hardware-backed key protection

DeviceKeyWrapper binds UserDataKey to physical device (non-exportable Keystore key). IdentityKeyWrapper protects identity secrets with user authentication (4h timeout).

STRONG S-03: Comprehensive screen & notification privacy

FLAG_SECURE + recents blackout on incoming call/PTT activities. All notifications — PttNotifier, CallNotifier, CallForegroundService, FastCallForegroundService — now consistently use VISIBILITY_PRIVATE on lock screen.

STRONG S-04: DeviceIntegrity + ProGuard anti-tamper

Runtime Frida/root detection triggers session key wipe. Crypto classes deliberately obfuscated in release builds.

STRONG S-05: BridgeConfig + EmergencyWipeManager

12 embedded obfs4 bridges with auto-rotation. Secure delete with zero-overwrite. Tor HS key and Keystore alias cleanup. Crash recovery via persistent markers.

STRONG S-06: Robust FGS fallback (new in v1.2.0)

CallForegroundService and FastCallForegroundService handle SecurityException when microphone FGS type is blocked. Falls back gracefully without crashing.

6. Optional Features — WebRTC Audio & Video Calling

NOTE: These features are OPTIONAL and operate in a reduced-security environment. An explicit security warning is shown before use. User consent is mandatory. No changes were made to WebRTC source files in v1.2.0 — all findings from v1.1.0 remain applicable.

Ref	Severity	Component	Title
C-01	CRITICAL	WebRTC Audio	ICE ALL exposes IP via STUN
C-02	CRITICAL	WebRTC Video	Clearnet signaling bypasses Tor
M-02	MEDIUM	WebRTC Video	Server auth without nonce

CRITICAL C-01: ICE policy ALL may expose real IP via STUN (WebRTC Audio)

Finding: WebRtcCallSession.kt line 201: iceTransportsType = ALL. STUN candidates reveal user's public IP to TURN/STUN server operator.

Impact: Real IP visible to TURN/STUN infrastructure. Mitigated by mandatory user consent.

Recommendation: Consider RELAY-only ICE as optional privacy mode.

CRITICAL C-02: Clearnet signaling bypasses Tor (WebRTC Video)

Finding: VideoCallServer.kt connects to wss://assistant.eelke-sp.cc via OkHttp — clearnet TCP outside Tor. STUN uses clearnet endpoint.

Impact: IP, call timing, metadata visible to VPS operator. Mitigated by mandatory user consent.

Recommendation: Consider certificate pinning or Tor-based signaling.

MEDIUM M-02: Server auth without nonce (WebRTC Video)

Finding: VideoCallServer.sendAuth() signs (contactId + timestamp) without random nonce. Replay possible within timestamp window.

Impact: Intercepted auth message can be replayed.

Recommendation: Add random nonce to auth message.

Privacy Comparison

Property	Chat + PTT	Fast Call (Audio)	Video Call
Transport	Tor (.onion)	Tor + STUN/TURN	Clearnet (WSS+STUN)
IP visible	No	Yes (STUN)	Yes (VPS+STUN)
Signaling	Tor (E2EE)	Tor (E2EE)	Clearnet WSS
Media	N/A	DTLS-SRTP	DTLS-SRTP
Metadata protected	Yes	Partial	No
User warning	No	Yes	Yes
Privacy level	Maximum	Reduced	Minimal

7. Conclusion & Recommendations

7.1 Overall Assessment

dunno v1.2.0 (build 30) maintains the strong security posture of v1.1.0. The changes — audio routing, FGS robustness, and UI polish — introduced no security regressions. The one LOW finding identified during the audit (VISIBILITY_PUBLIC in FGS notifications) was promptly resolved.

The core messaging functionality has no critical or high-severity findings. Two low-impact findings carry over from v1.1.0 in unchanged files. All previously resolved issues remain fixed. All notification channels now consistently protect lock screen privacy.

The application's security architecture — proven cryptography, Tor-exclusive transport with bridge support, hardware-backed key storage, runtime integrity monitoring, and comprehensive screen/notification privacy — places it among the strongest privacy-focused communication tools available.

7.2 Recommendations by Priority

- [HIGH] Consider RELAY-only ICE as optional privacy mode for Fast Call.
- [HIGH] Implement certificate pinning for Video Call WSS connection.
- [HIGH] Add random nonce to Video Call auth message.
- [MEDIUM] Consider fuzz-testing the OnionHttpClient HTTP/1.1 parser.

7.3 Complete Findings Table

Ref	Severity	Scope	Component	Title
C-01	CRITICAL	WebRTC	Audio	ICE ALL exposes IP via STUN
C-02	CRITICAL	WebRTC	Video	Clearnet signaling bypasses Tor
M-01	MEDIUM	Core	Networking	Custom HTTP/1.1 parser
M-02	MEDIUM	WebRTC	Video	Server auth without nonce
L-01	LOW	Core	Networking	XOR obfuscation (self-documented)

Total: 5 findings (2 CRITICAL, 0 HIGH, 2 MEDIUM, 1 LOW). Core application: 2 findings (1 MEDIUM, 1 LOW). No new findings in v1.2.0. Optional WebRTC features: 3 findings (2 CRITICAL, 1 MEDIUM) — unchanged. Both CRITICAL findings protected by mandatory user consent. All previously resolved issues remain fixed. No regressions.