

Cure53 Security Assessment of Gem Wallet Rust Wallet Core, Management Summary, 09.2026

Cure53, Dr.-Ing. M. Heiderich, Dr. A. Pirker, Dr. N. Kobeissi, MSc. C. Mayr

Cure53, a Berlin-based IT security consulting firm, was engaged to conduct a penetration test and source code audit conducted against the Gem Wallet (Rust core).

To give some context regarding the assignment's origination and composition, Gem Wallet LLC contacted Cure53 in July 2026. The test execution was scheduled for September 2026, namely in CW39. A total of thirteen days were invested to reach the coverage expected for this project, and a team of four senior testers was assigned to its preparation, execution, and finalization.

The work was split into three separate work packages (WPs), defined as:

- **WP1:** White-box audits & reviews against the v4 keystore & UniFFI secret boundary
- **WP2:** White-box audits & reviews against the derivation, import & signing primitives
- **WP3:** White-box audits & reviews against the iOS/Android password custody

The review conformed to a comprehensive white-box testing model designed to give assessors complete operational and architectural visibility. By providing full source code access and comprehensive technical resources upfront, Gem Wallet enabled Cure53 to perform an exhaustive evaluation of system internals and underlying design logic.

Operational setup concluded in CW38 (September 2026) to establish complete testing readiness. Project communication was centered within a private, shared Telegram channel that unified Gem Wallet engineering teams and Cure53 security assessors, enabling direct cross-functional collaboration and rapid information sharing throughout the engagement lifecycle.

Operational engagement proceeded efficiently under clear, well-defined scoping parameters, resulting in minimal clarification requests and zero testing roadblocks. Cure53 provided real-time visibility throughout the audit lifecycle, delivering frequent status updates, immediate findings disclosures, and live reporting via shared Markdown documentation in the dedicated Telegram channel.

Cure53 achieved thorough test coverage across all scoped components, uncovering twenty-one total findings. Risk classification divided these results into eleven exploitable security vulnerabilities and ten general security weaknesses characterized by lower exploitation potential.

Identified Vulnerabilities

- GEM-01-001 WP3: No authorization required on wallet deletion (Medium) **FIXED**
- GEM-01-006 WP2: Swap permits signed with unrelated wallet (Low) **FIXED**
- GEM-01-007 WP1: Wallet deletion removes secrets before DB commit (Low)
- GEM-01-008 WP3: User authentication not configured by default (Medium) **MITIGATED**
- GEM-01-010 WP3: Unreadable Android storage can overwrite password (Low) **FIXED**
- GEM-01-012 WP3: Biometric authorization bypass via hooks (Medium) **MITIGATED**
- GEM-01-013 WP3: Potential phishing via deep link hijacking (Medium) **FIXED**
- GEM-01-014 WP3: DoS of Android app via malformed intents (Low) **FIXED**
- GEM-01-016 WP3: Biometric authorization bypass via file manipulation (Medium) **MITIGATED**
- GEM-01-017 WP1: Missing zeroization in mnemonic generation (Low) **FIXED**
- GEM-01-020 WP2: Missing zeroization when deriving private keys (Low) **FIXED**

Miscellaneous Issues

- GEM-01-002 WP2: No checks on entropy size for seed on deriving priv. keys (Info) **FIXED**
- GEM-01-003 WP1: No minimum checks on Argon2id parameters (Info) **FIXED**
- GEM-01-004 WP1: Missing expected type check on decrypting secrets (Info)
- GEM-01-005 WP2: Signing keys lack binding to approved accounts (Info)
- GEM-01-009 WP3: Concurrent iOS password creation can lock-out wallet (Low) **FIXED**
- GEM-01-011 WP3: Android Keystore failure exposes encryption keys (Medium) **MITIGATED**
- GEM-01-015 WP3: Android application supports outdated SDK versions (Low)
- GEM-01-018 WP3: Missing screenshot protection for seedphrase export (Low) **FIXED**
- GEM-01-019 WP2: Existing wallet check blocks secrets recovery (Low) **FIXED**
- GEM-01-021 WP2: Secp256k1 malleability in recover_ethereum_address (Info) **FIXED**

Overall evaluation reflected a balanced operational security posture. The total absence of High and Critical severity findings confirms that core architecture is well-protected against severe exploitation attempts, validating baseline risk management practices while highlighting opportunity areas for secondary control refinement.

Cryptographic resilience was strongly confirmed across the architecture. Auditing of low-level Rust libraries and mobile endpoints verified robust cryptographic primitive choices, sound parameter selection, and secure entropy generation, ensuring underlying mathematical protections remain uncompromised.

However, the testing team observed several areas where further attention was required. Most notably, systemic authorization flaws were discovered, including missing access checks, authorization bypasses, and insecure default configurations. Positively, the Gem Wallet team demonstrated their commitment to security by quickly addressing the majority of findings shortly following the testing phase. The rolled-out fixes were provided to Cure53 and subsequently verified through follow-up fix verification.

Cure53 would like to thank Herman and Tim from the Gem Wallet LLC team for their excellent project coordination, support and assistance, both before and during this assignment.