



Clear Balance

v2.1 Android Security Test Report

Owner-led penetration testing and privacy/security verification

PUBLIC EDITION

Assessment date	1 October 2026
Target	Clear Balance v2.1 production Android app
Package	com.ryan.clearbalanceuk
Distribution	Google Play production installation
Primary device	Physical Android device, non-rooted
APK SHA-256	Retained in internal evidence record
Known-good backup SHA-256	Retained in internal evidence record
Overall outcome	No security vulnerabilities identified within the completed test scope; one hostile/rooted-device test area deferred.

Public edition: This report is derived from the full internal test record. Exact artefact hashes and seeded test values have been withheld from publication. This was an owner-led, source-aware assessment of the production Android application; it is not an independent third-party certification, formal audit, or claim of absolute security.

1. Executive summary

Clear Balance v2.1 was subjected to a structured Android security test focused on the controls introduced or strengthened for the release: local privacy protections, App Lock, encrypted backups, Android platform exposure, network privacy, notification privacy, screenshot protection, and production dependency exposure.

Across the completed scope, no exploitable security issue was identified. The production build was confirmed to be non-debuggable, Android backup was disabled, cleartext network traffic was disabled, exported components were limited and appropriately protected, seeded financial data was not found in logs or readable shared storage, App Lock resisted the tested external-launch and lifecycle bypass attempts, and sensitive screens were protected from recent-apps and ADB screenshot capture.

The encrypted backup workflow received the deepest adversarial testing. A known-good backup restored correctly with the correct password, while incorrect passwords, modified ciphertext, truncated files, unsupported envelope versions, unsupported cipher declarations, and unsupported KDF declarations were all rejected without damaging the existing plan.

One planned category - rooted/hostile-device extraction of the private application sandbox and runtime instrumentation - was deliberately deferred. iOS was also outside the scope of this Android test run.

2. Scope and objectives

The purpose of the assessment was to test security claims against the shipped Android application rather than relying only on source-code configuration. The assessment focused on whether a normal attacker, another installed application, a person with ADB access to a non-rooted device, or someone in possession of an exported backup could obtain or alter sensitive Clear Balance plan data.

In scope:

- Production Google Play artefact and manifest hardening.
- Exported Android activities, receivers, providers, permissions and deep-link surface.
- App Lock behaviour across direct external launch, background/resume and force-stop paths.
- Sensitive-data leakage to Android logs and readable shared storage.
- Encrypted backup confidentiality, integrity, version handling and failure behaviour.
- Uninstall/reinstall data removal and Android backup behaviour.
- Observed network/privacy traffic during representative app workflows.
- Static SDK/dependency review for unexpected analytics, crash-reporting and attribution SDKs.
- Recent-apps privacy, screenshot/screen-capture protection and notification privacy.

Out of scope / deferred:

- Rooted-device private-sandbox extraction and runtime memory inspection.
- Frida/Objection method hooking and hostile runtime instrumentation.
- Independent external penetration testing.
- Formal source-code cryptographic review.
- iOS penetration testing.
- Exhaustive verification of every possible application workflow or third-party service path.

3. Test environment and tooling

Device	Physical Android device; production device remained non-rooted.
Build source	Installed Google Play production app; base.apk and split APKs pulled from the installed package.
ADB	Package inspection, direct activity launching, log capture, storage checks, screenshots and package metadata.
apkanalyzer	Production manifest and DEX package namespaces.
PowerShell	Hashing, backup manipulation, content searches and evidence capture.
HTTP Toolkit	Observed representative network traffic and searched for seeded plan-data canaries.

4. Result convention

Each test was recorded as PASS, DEFERRED/NOT TESTED, or a security finding if unsafe behaviour was observed. No completed test produced a security finding requiring remediation.

Result	Meaning	Use in this report
PASS	Expected security behaviour observed in the tested scenario.	Completed tests
DEFERRED	Planned area not executed; neither pass nor failure.	CB-PT-006B
FINDING	Unsafe or unexpected behaviour requiring assessment/remediation.	None identified

5. Test summary

ID	Area	Result	Summary
CB-PT-001	Production Android exposure	PASS	Non-debuggable release; backup and cleartext disabled; exported surface reviewed.

ID	Area	Result	Summary
CB-PT-002	App Lock bypass/lifecycle	PASS	No bypass via direct activity launch, back/cancel, background/resume or force-stop.
CB-PT-003	Sensitive-data logging	PASS	Seeded debt data not recovered from Android logcat.
CB-PT-004	Shared/external storage leakage	PASS	No seeded plan data found in readable shared storage.
CB-PT-005	Encrypted backup security suite	PASS	Confidentiality, wrong-password, tamper, truncation, version/cipher/KDF and good-restore tests passed.
CB-PT-006A	Uninstall/reinstall data removal	PASS	Local plan and App Lock state removed; no Android backup resurrection.
CB-PT-006B	Rooted-device vault extraction	DEFERRED	Requires hostile/root-capable environment; not assessed in this run.
CB-PT-007	Network/privacy traffic	PASS	No seeded plan values found in observed traffic for tested workflows.
CB-PT-008	Static SDK/dependency review	PASS	No AppsFlyer/Adjust SDK namespaces; no tested analytics/crash SDKs detected.
CB-PT-009	Recent-apps snapshot privacy	PASS	Sensitive plan information not exposed in Android Recents.
CB-PT-010	Screenshot/screen capture	PASS	ADB screencap control worked; Clear Balance rendered black.

ID	Area	Result	Summary
CB-PT-011	Notification privacy	PASS	No debt/balance/budget canary data exposed in notification surfaces.

6. Detailed test results

CB-PT-001 - Production Android exposure

Objective	Verify that the shipped Google Play build does not expose avoidable Android attack surface.
Evidence and actions	Pulled the installed production APK; captured a SHA-256 fingerprint; attempted ADB run-as access; inspected package flags and the shipped manifest; reviewed exported activities/receivers/providers and requested permissions.
Observed	ADB run-as returned package not debuggable. The manifest contained android:allowBackup="false" and android:usesCleartextTraffic="false". MainActivity was exported only as the launcher entry point. The AndroidX ProfileInstallReceiver was exported but protected by android.permission.DUMP. Content providers were non-exported. No inbound BROWSABLE/VIEW activity filter was identified.

CB-PT-002 - App Lock external launch and lifecycle bypass

Objective	Determine whether protected plan data can be exposed by launching the exported activity directly or abusing lifecycle transitions.
Evidence and actions	Force-stopped the app, launched MainActivity directly via ADB, cancelled biometric prompts, used Back, switched apps, resumed from background and repeated direct relaunch/force-stop paths.
Observed	No dashboard flash, debt name, balance or other protected content was exposed. Cancelling authentication, pressing Back, app switching and force-stop/relaunch did not bypass App Lock.

CB-PT-003 - Sensitive data in Android logs

Objective	Test whether financial-plan values are written to Android system logs.
Evidence and actions	Seeded distinctive fake financial values; cleared logcat; exercised create/edit/view workflows; searched the resulting log capture.
Observed	No seeded financial values were found. Normal Android lifecycle messages exposed only package/activity identifiers.

CB-PT-004 - Shared/external storage leakage

Objective	Check for unintended copies of plan data outside the private application sandbox.
Evidence and actions	Inspected app-related shared storage paths and searched readable Download/Documents areas and filenames/content for Clear Balance and seeded canary values.
Observed	No unintended sensitive plan data was found in readable shared storage.

CB-PT-005 - Encrypted backup security

Objective	Assess confidentiality, integrity, format handling and failure safety of exported encrypted backups.
Evidence and actions	Exported a production backup to Google Drive, downloaded the file, hashed it, inspected the versioned envelope, searched for plaintext canaries and the test password, tested wrong passwords, modified ciphertext, truncation, unsupported envelope version, unsupported cipher declaration, unsupported KDF declaration, and a known-good restore.

Observed	The envelope declared format=clear_balance_encrypted_backup, envelopeVersion=1, cipher=AES-256-GCM, and PBKDF2-HMAC-SHA256 with 600000 iterations. Seeded debt values and the test password were not present in plaintext. Valid backup + correct password restored successfully. Wrong password, tampered ciphertext, truncated file and unsupported metadata variants were rejected without partial restore or damage to the existing plan.
-----------------	---

CB-PT-006A - Uninstall/reinstall data removal

Objective	Verify that deleting the Android app removes local Clear Balance data and that Android backup does not silently restore it.
Evidence and actions	Created unmistakable test data, uninstalled the app, reinstalled from Google Play and checked for plan/App Lock resurrection; then verified intentional restore from the known-good encrypted backup.
Observed	The reinstall began with fresh app state; the local test plan and App Lock state did not return. The user-controlled encrypted backup remained independently restorable.

CB-PT-006B - Rooted/hostile-device extraction

Objective	Test vault confidentiality and key handling after the Android sandbox itself is compromised.
Evidence and actions	Planned but not executed. The physical production device correctly refused adb root; a separate hostile emulator/rooted environment was deliberately not pursued.
Observed	No result. This area remains outside the completed assurance scope.

CB-PT-007 - Network and privacy traffic

Objective	Look for outbound transmission of seeded debt/budget data during representative workflows.
------------------	--

Evidence and actions	Routed a physical device through HTTP Toolkit, exercised Dashboard, Debts, Budget, Settings, Premium/paywall and Restore Purchases, and searched observed traffic for distinctive seeded plan-data canaries.
Observed	No seeded plan values were identified in the traffic observed during the tested workflows.

CB-PT-008 - Static SDK/dependency review

Objective	Check the shipped APK for unexpected analytics, crash-reporting or attribution SDKs.
Evidence and actions	Reviewed DEX package namespaces. Broad string scans were abandoned as too noisy. Exact SDK namespace checks were then used.
Observed	Adapty was present as expected. No exact com.appsflyer or com.adjust.sdk namespaces were present; initial AppsFlyer/Adjust keyword hits were references inside Adapty's optional attribution helper. Sentry, Amplitude, Mixpanel, Crashlytics, Facebook SDK and Firebase Analytics were not detected by the targeted package checks.

CB-PT-009 - Recent-apps snapshot privacy

Objective	Check whether sensitive debt/balance content remains visible in Android Recents after backgrounding/locking.
Evidence and actions	Viewed sensitive plan screens, backgrounded the app and inspected the Recent Apps preview.
Observed	Sensitive financial-plan content was not exposed in the task preview.

CB-PT-010 - Screenshot/screen-capture protection

Objective	Test whether Android OS-level screenshot capture can recover Clear Balance UI content.
Evidence and actions	Used ADB screencap with a home-screen control, then repeated while Clear Balance was unlocked on a sensitive screen and while locked.

Observed	The home-screen control was captured normally. Clear Balance captures were black, and no debt/budget data was recoverable.
-----------------	--

CB-PT-011 - Notification privacy

Objective	Verify that reminders do not expose plan details outside the app.
Evidence and actions	Triggered the scheduled budget-review reminder, inspected lock-screen/collapsed/expanded notification surfaces and searched Android notification data for seeded canary values.
Observed	Only generic reminder content was exposed; no debt names, balances, APRs, minimum payments, budgets or seeded test values were identified.

7. Backup format observations

The exported backup was a readable JSON envelope containing non-secret metadata plus an encrypted payload. This is an appropriate pattern: algorithm identifiers, version information, salt and nonce do not need to be secret; the protected financial plan is contained in the ciphertext.

Format identifier	clear_balance_encrypted_backup
Envelope version	1
Declared cipher	AES-256-GCM
Declared KDF	PBKDF2-HMAC-SHA256
KDF iterations	600000
Visible metadata	Format/version, KDF parameters, salt, nonce, ciphertext field and mac field.
Plaintext plan data	Not observed in the exported backup during the tested searches.
Backup password	Not observed in plaintext.

The field name 'mac' was observed in the envelope. This report does not infer its exact implementation semantics from the field name alone; cryptographic implementation details would require source-level review.

8. Limitations

- No rooted-device or jailbroken-device testing was completed.
- No runtime method hooking, memory scraping or instrumentation was completed.
- The assessment was performed by the developer/owner and is not independent.
- Network conclusions are limited to the workflows and traffic observed during the test session.
- Static analysis was targeted rather than exhaustive; a broad DEX string scan was discarded because it produced excessive false-positive noise.
- The assessment does not prove that an attacker controlling an unlocked rooted device could never recover runtime data.
- iOS was not assessed in this report.

9. Recommendations

- Repeat the relevant privacy, App Lock, backup and notification tests on iOS.
- If stronger assurance is desired, complete the deferred rooted/hostile-device phase using a disposable test environment and runtime instrumentation.
- Consider an independent external review before making any public claim of third-party penetration testing or certification.
- Re-run the backup tamper/version tests after any future change to the backup format, KDF, cipher or restore pipeline.
- Re-run targeted dependency and manifest checks after major SDK upgrades, especially Adapty, Flutter, billing and notification dependencies.
- Keep public wording precise: describe what was tested and what was observed rather than claiming absolute security.

10. Overall conclusion

Within the completed Android scope, Clear Balance v2.1 behaved as intended under the tested security and privacy attacks. No security vulnerability requiring remediation was identified. The strongest assurance came from the encrypted backup tests, App Lock bypass attempts, platform exposure checks, data-leakage searches and screen/notification privacy tests.

The assessment supports the conclusion that the v2.1 production Android build has a small exposed attack surface and fails safely across the tested backup corruption and authentication scenarios. The principal remaining assurance gap is hostile/rooted-device testing, which was explicitly deferred rather than treated as a pass.