COMPUTER FORENSICS LAB

London digital forensics & e-discovery · Established 2007

Android App Data and Third-Party Application Forensics

Sandboxes, Sideloading and the Open Ecosystem: Every App Its Own World, and Some Worlds Stranger Than Others

On Android, as on iOS, every app is its own world of evidence, storing its data in its own databases and files, and each must be understood on its own terms before its data is relied on. But Android's openness adds dimensions the iPhone lacks. Apps can be installed from outside the official store, sideloaded from a file or an alternative store, so a device may run apps that are obscure, modified, or built to deceive, including cloned and "dual" apps that run a second copy of a messaging app under a different account. App data may sit on the SD card as well as in the app's protected sandbox. And the Google account backs up app data selectively, according to each app's own settings. For the lawyer, this makes Android app forensics both richer and more demanding: the apps must be inventoried, including the unexpected ones, each understood precisely, and the evidence inside them recovered, corroborated and attributed with care. This guide sets out for UK lawyers how Android apps store their data, what the open ecosystem adds, how app evidence is recovered and read, and how to avoid the confident nonsense that a misread unfamiliar app produces.

⏱ Estimated reading time: 24 min read

§ ABOUT THE AUTHOR

PREPARED BY COMPUTER FORENSICS LAB E-DISCOVERY TEAM

Computer Forensics Lab is a London-based digital forensics and electronic disclosure practice established in 2007. Android app analysis is a demanding, high-value strand of its mobile work: inventorying every app on a device, including sideloaded, cloned and obscure ones, recovering each app's data from its sandbox, the SD card and the Google account, understanding each app on its own terms rather than guessing, and reading the evidence inside them correctly. The analysis corroborates across the estate and providers and is reported under CPR Part 35 and CrimPR Part 19.

ESTABLISHED 2007 · LONDON | ISO 17025-ALIGNED PROCEDURES | ANDROID APP & SANDBOX FORENSICS | SIDELOADED & CLONED APP ANALYSIS | CPR PART 35 EXPERT REPORTS | FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

In this guide

§ 0 1 · ORIENTATION

Executive summary

The headline point: on Android every app is its own world, storing data in its own sandboxed databases and files that must be understood on their own terms, and the open ecosystem adds dimensions iOS lacks, sideloaded and obscure apps, cloned and "dual" apps running second accounts, app data on the SD card, and selective Google backup, so the apps must be inventoried including the unexpected ones, each understood precisely, and the evidence inside recovered and attributed with care.

How apps store data (§3): each app in its own sandbox, with private databases (usually SQLite) and files, plus shared and external storage it may use. **What openness adds (§4):** sideloading from outside the official store, cloned and dual apps, modified apps, and app data on removable storage. **Inventory and recovery (§5):** every installed app enumerated, including hidden and sideloaded ones, and each app's data recovered at depth from sandbox, external storage and backup. **Interpretation (§6):** fields read for what they mean, interpretations validated, and activity attributed to a person with care. **Deployment (§7):** the evidence inside the apps, matches, journeys, transactions, listings, the user's own words, deployed honestly.

- **Every app its own world:** sandboxed data, understood on its own terms.
- **Openness adds strangeness:** sideloaded, obscure, modified and cloned apps.
- **Inventory everything:** including apps that are hidden or unexpected.
- **Data in more places:** sandbox, SD card and selective Google backup.
- **Read, validate, attribute:** a misread unfamiliar app is confident nonsense.

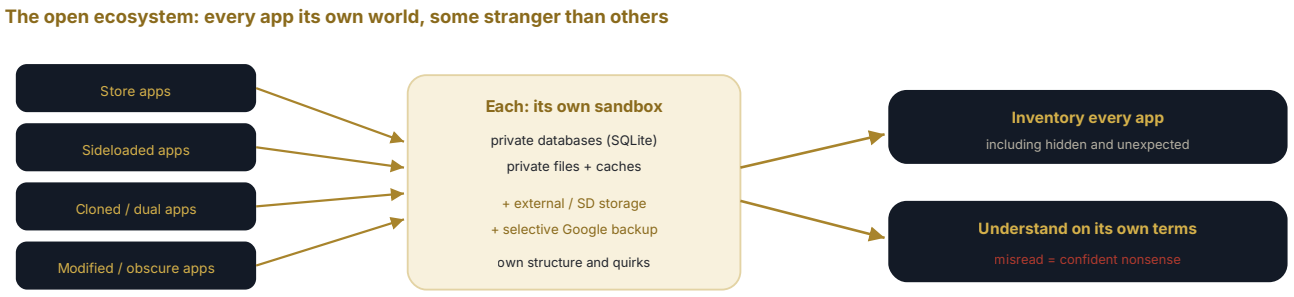

Figure 1 - store, sideloaded, cloned and modified apps each keep their own sandboxed data, plus external storage and selective backup, so every app is inventoried and understood on its own terms before its evidence is relied on.

§ 0 2 · FIRST PRINCIPLES

The problem in plain English: an open ecosystem of worlds

The iPhone app guide (guide 151) made a point that holds for every platform: a phone is a collection of apps, each its own world of evidence, storing its data in its own way, and each must be understood on its own terms before its data is relied on, because a field misread in an unfamiliar app produces confident nonsense. Everything in that guide applies to Android. A dating app still records matches and meetings, a marketplace still records listings and buyers, a rideshare app still records journeys, a payment app still records transactions, a notes app still records the user's own words, and each still stores it in its own databases, on Android as on iOS usually SQLite (guide 155), inside its own protected space, the sandbox, that other apps cannot read.

What Android adds is openness, and openness brings strangeness. On an iPhone, apps come from one store, vetted by one company. On Android, an app can be installed from outside the official store, sideloaded from a file or from an alternative store, so a device may run apps that are obscure, that have been modified, or that are built to deceive: cloned or "dual" apps that run a second copy of a messaging app under a different account, apps disguised as something innocuous, apps that hide themselves from the launcher. App data may live not only in the sandbox but on the SD card and in shared external storage, where it is less protected and may survive the phone's encryption (guide 166). And the Google account (guide 164) backs up app data selectively, according to each app's own settings, so some apps' data is in the cloud and some is not. For the lawyer this makes Android app forensics richer, because more may be there, and more demanding, because the apps must first be inventoried, including the hidden and unexpected ones, then each understood precisely, then the evidence inside them recovered from wherever it lives, and finally attributed and corroborated with the same care as anywhere else. This guide sets out how Android apps store their data, what the open ecosystem adds, how app data is inventoried and recovered, and how it is read without producing confident nonsense.

§ 0 3 · HOW APPS STORE DATA

How Android apps store their data

- **The sandbox:** each app given its own private storage area that other apps cannot read, holding its databases, files and caches, so each app's data is a self-contained world reached only at sufficient extraction depth (guide 162): the per-app sandbox.
- **SQLite databases:** most apps keeping their data in SQLite databases within the sandbox, so the deleted-record recovery disciplines apply (guide 155): the common database.
- **Shared and external storage:** apps also writing to shared storage and the SD card, for media, downloads and sometimes data, where it is less protected and may outlive the sandbox (guide 166): the external footprint.
- **Caches, preferences and logs:** apps keeping caches, preference files and internal logs that record settings, sessions and activity beyond the main database (guide 138): the wider app trace.
- **Selective Google backup:** the Google account backing up app data only for apps that opt in, so the cloud holds some apps' data and not others' (guide 164): the partial cloud copy.

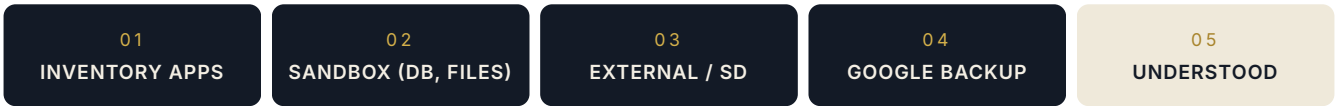

Figure 2 - inventory the apps, recover each one's sandbox, external storage and any Google backup, and understand each on its own terms before relying on it.

§ 0 4 · WHAT OPENNESS ADDS

What the open ecosystem adds

- **Sideloading:** apps installed from files or alternative stores rather than the official store, so a device may run apps unknown to mainstream analysis, modified versions, or apps built for concealment (guide 139): the unofficial app.
- **Cloned and dual apps:** manufacturer and third-party features running a second copy of an app under a different account, so a person may have two WhatsApps or two social accounts on one phone, each with its own data (guide 165): the second account.
- **Hidden and disguised apps:** apps hidden from the launcher or disguised as innocuous tools (a calculator that is a vault), so the app inventory must look beyond the visible icons (§5): the concealed app.
- **Modified apps:** altered versions of popular apps with changed behaviour, including messaging apps modified to defeat deletion or add features, so an app's behaviour is not assumed from its name (guide 156): the altered app.
- **Work profiles and multiple users:** Android's work profiles and multi-user support keeping separate app instances and data per profile or user, so all profiles are examined (guide 144): the parallel profiles.

§ 0 5 · INVENTORY AND RECOVERY

Inventorying and recovering app data

- **Enumerate every app:** all installed apps enumerated from the system's package records, not just the visible icons, so hidden, sideloaded, cloned and profile-specific apps are found (§4): the complete inventory.
- **Identify the relevant ones:** the apps bearing on the matter identified from the inventory, with attention to the unexpected, so effort is directed and nothing telling is missed (guide 151): the targeted enquiry.
- **Recover the sandbox at depth:** each relevant app's sandbox, databases, files, caches and preferences, recovered by a file-system or deeper extraction, since a logical extraction may not reach it (guide 162): the sandbox recovered.
- **Recover external and cloud copies:** the app's data on shared storage and the SD card, and its Google backup where the app opted in, recovered and reconciled with the sandbox (guides 166, 164): the other copies.
- **Recover deleted app data:** deleted records within the app's databases recovered from remnants and journals, and from backups and earlier states (guide 155): the deleted app data.

§ 0 6 · INTERPRETATION AND HONEST LIMITS

Interpretation, attribution and honest limits

- **Read fields for what they mean:** each app's fields, statuses and timestamps interpreted for their true meaning in that app, since an assumed meaning produces confident error (guide 151): the discipline against misreading.
- **Validate the interpretation:** the understanding of an app's data validated by testing, documentation or corroboration, especially for obscure, sideloaded or modified apps whose behaviour is undocumented (guide 100): the checked interpretation.
- **Establish the app's identity and version:** the app confirmed as what it claims to be, its version and any modification identified, so conclusions rest on how this app actually behaves (§4): the verified app.
- **Attribute to a person:** app activity coming from an account on the device, and cloned apps meaning one phone may hold several accounts, so activity is attributed to a person with corroboration (guide 105): the who-did-it question.
- **State confidence and limits:** what an app's data shows, and the limits of interpreting an undocumented or modified app, expressed at honest confidence (guide 100): the honest app account.

§ 0 7 · DEPLOYMENT

Deployment: the evidence inside the apps

- **Proving contact, dealing and movement:** dating, marketplace, rideshare and payment app data establishing contact, transactions and journeys, as on iOS (guide 151): the substantive app evidence.
- **Exposing the hidden app:** a hidden, disguised or sideloaded app found and its contents recovered, revealing what a party concealed behind an innocuous icon (§4): the concealment exposed.
- **Revealing the second account:** a cloned or dual app recovered, revealing a second messaging or social account and its conversations (§4): the parallel life revealed.
- **Recovering the user's own words:** notes, journal and productivity app data recovering a person's candid records (guide 135): the self-account recovered.
- **Corroborated, honest presentation:** the app findings understood on their own terms, corroborated with the app's cloud, the provider and counterparts, and presented at honest confidence with attribution stated, under CPR Part 35 or CrimPR Part 19 (guides 100, 130): the app evidence deployed defensibly.

§ 0 8 · THE WIDER MAP

Source architecture: where else the evidence lives

Per this series' source-architecture discipline: an app's data on an Android phone lives in several places, and the open ecosystem multiplies both the apps and the places. The phone holds each app's sandbox, its external and SD card data, and any cloned or profile-specific instances. But the Google account (guide 164) holds the data of apps that opted into backup, reachable regardless of the phone's lock; the app's own cloud and account hold what it syncs, often more and older than the device; the manufacturer's cloud may hold app backups; the SD card may hold app data readable when the phone is sealed (guide 166); the counterparts hold their side of a match, trade or journey; and the app provider holds authoritative records by disclosure. The discipline is to inventory every app, including the hidden and cloned, recover each from the sandbox, external storage and cloud, and corroborate with the provider and counterpart, so that an app's local data is one source among several and a hidden or second-account app is caught by its traces across the estate. When the sandbox is inaccessible, the SD card or Google backup may hold the data; when a field's meaning is doubtful, the provider's records confirm it; and a cloned app's second account is corroborated by the provider and the counterparts who dealt with it.

EVIDENCE	APP SANDBOX	EXTERNAL / SD CARD	GOOGLE BACKUP	APP CLOUD / ACCOUNT	COUNTERPART	PROVIDER RECORDS
App activity	Y: at depth	Some (media, data)	If opted in	Y: often more	Their side	Y: by disclosure
Deleted app data	Remnants	Remnants (card)	Earlier backup	Cloud copy	Their copy	Provider retention
Cloned / dual account	Y: separate instance	Some	Varies	Y: second account	Their side	Y: account records
Hidden / sideloaded app	Y: package records	Installer file (APK)	Usually no	Varies	n/a	Alt-store records
Field interpretation	Structure on device	n/a	n/a	Confirms meaning	Corroborates	Authoritative

Fallback strategy in one line: inventory every app including hidden and cloned ones, recover each from sandbox, external storage and cloud, and corroborate with the provider and counterpart; when the sandbox is sealed or a field doubtful, the card, backup, cloud or provider resolves it.

§ 0 9 · IN THE WILD

Worked examples

EXAMPLE 1 · THE SECOND WHATSAPP THE INVENTORY REVEALED

A party disclosed one WhatsApp account and denied any other. The §5 complete inventory, enumerated from the system's package records rather than the visible icons, found a §4 cloned "dual" instance of WhatsApp running under a second number, with its own §3 sandbox and conversations. Its data was recovered at §5 depth (guide 162), deleted messages from its database remnants (guide 155), and the §8 second account corroborated by the provider's records and the counterparts who had messaged it. The §6 attribution tied both accounts to the party with care. The undisclosed second life was on the same phone. The lesson is §4's and §5's: Android's cloned and dual apps let one phone run a second copy of an app under a different account, so the app inventory is built from the system's package records, not the launcher, and a second instance of a familiar app is recovered and understood as its own world.

EXAMPLE 2 · THE CALCULATOR THAT WAS A VAULT

A device showed nothing unusual among its icons, but the §5 enumeration found a §4 sideloaded app disguised as a calculator that was in fact a "vault" app hiding photos and files behind a PIN. Its §3 sandbox and external storage were recovered, and its contents, including deleted items from its database and the §8 SD card's remnants (guide 166), were restored. The app's §6 identity and behaviour were verified rather than assumed from its name, and its contents §6 attributed with care. What the party had hidden behind an innocuous icon was found. The lesson is §4's: the open ecosystem lets apps be sideloaded, hidden and disguised, so the inventory looks beyond the visible icons and verifies what each app actually is, because the innocuous calculator may be the vault holding the evidence.

EXAMPLE 3 · THE MODIFIED APP THAT BEHAVED DIFFERENTLY

A messaging app on a device appeared to be a familiar one, and its data was initially read on that assumption. The §6 verification discipline found it was a §4 modified version, sideloaded, with altered behaviour: it retained messages the genuine app would have deleted and stored them differently. Reading its data as though it were the genuine app would have produced §6 confident nonsense about what had and had not been deleted and when. Once its real version and behaviour were established and §6 validated, the data was read correctly, and the §8 provider's records corroborated the interpretation. The name was familiar; the behaviour was not. The lesson is §6's: on Android an app's behaviour cannot be assumed from its name, since sideloaded and modified versions behave differently, so each app's identity, version and modification are established and its interpretation validated before any conclusion is drawn from it.

§ 10 · WHERE IT GOES WRONG

Common mistakes and technical limitations

Common mistakes

- **Inventorying from the launcher:** the cardinal Android error, missing hidden, cloned and sideloaded apps that only the package records reveal (Examples 1 and 2, §5).
- **Assuming an app from its name:** reading a modified or sideloaded app as the genuine one (Example 3, §6).
- **Misreading an unfamiliar app:** assuming a field's meaning and producing confident nonsense (§6, guide 151).
- **Ignoring external and SD storage:** recovering the sandbox but not the app's data on the card or shared storage (§3, §5).
- **Assuming Google backed it up:** expecting cloud app data when the app did not opt in (§3).
- **Missing profiles and users:** examining one profile while a work profile or second user holds the app (§4).
- **Shallow extraction:** a logical extraction that never reaches the sandbox (§5, guide 162).
- **Attributing carelessly:** app activity, especially from a cloned app, attributed to a person without corroboration (§6).

Technical limitations

- **No single method:** every app stores data differently, and the open ecosystem adds unknown apps: the core challenge (§2).
- **Sandboxes need depth:** app data is reached only by a file-system or deeper extraction (§3).
- **Obscure apps are undocumented:** interpretation carries uncertainty that is stated (§6).
- **Modified apps defy assumptions:** behaviour must be established, not presumed (§4).
- **The ecosystem evolves constantly:** apps, clones and stores change continually (§4).

§ 11 · INTERROGATORIES & DRAFTING AIDS

Questions to ask · Suggested wording

Ask your client

- Which apps might hold relevant evidence, and might there be sideloaded, cloned, hidden or dual-account apps on the device?
- Is there an SD card or a work profile, and are the app accounts and the Google account available?
- Is the deeper extraction available to reach the app sandboxes?

Ask your opponent

- Please preserve and disclose all apps installed on the device at Schedule 1, including sideloaded, cloned, dual-account and hidden apps and all user profiles, with their data, including deleted app records, external storage and the associated app and Google accounts.
- Please confirm all accounts used in each messaging and social app on the device, including any second or cloned instance.
- Please confirm whether any app has been installed from outside the official store, or modified.

Ask your eDiscovery / forensic provider

- Has every app been inventoried from the package records, including hidden, cloned and sideloaded ones?
- Has each app's identity, version and behaviour been verified, and its data interpretation validated?
- What does each app hold, across sandbox, card and cloud, and how is it attributed and corroborated?

SUGGESTED WORDING · INSTRUCTION FOR AN ANDROID APP ANALYSIS

> "We instruct you to examine the application data on the Android device at Schedule 1 relevant to Schedule 2. Please, on the acquired device at sufficient depth and preserving integrity: (1) enumerate every installed app from the system's package records across all user and work profiles, including hidden, disguised, sideloaded, cloned and dual-account apps, and identify those bearing on the matter; (2) verify each relevant app's identity, version and any modification, and establish how it actually stores and deletes data rather than assuming from its name; (3) recover each app's sandbox (databases, files, caches and preferences), its data on shared storage and any SD card, and its Google or own-cloud backup where it exists, including deleted records from remnants, journals and earlier states; (4) interpret each app's fields, statuses and timestamps for their true meaning and validate that interpretation by testing, documentation or corroboration with the provider and counterparts; and (5) attribute app activity, including from cloned instances, to a person with corroboration, and state what each app's data shows, and the limits of interpreting obscure or modified apps, at honest confidence."

§ 12 · QUICK CONTROL

Checklist and red flags · When to involve a digital forensic expert

The Android app checklist

- ☐ Every app enumerated from package records, all profiles
- ☐ Hidden, disguised, sideloaded, cloned and dual apps found
- ☐ Each relevant app's identity, version and modification verified
- ☐ Sandbox recovered at sufficient extraction depth
- ☐ External storage and SD card app data recovered
- ☐ Google and own-cloud backups recovered where they exist
- ☐ Deleted app records recovered from remnants and backups
- ☐ Field meanings validated; interpretation corroborated
- ☐ Activity attributed to a person with corroboration
- ☐ Confidence and interpretive limits stated honestly

Red flags

- Apps inventoried from the launcher, not the package records: Examples 1 and 2.
- An app's behaviour assumed from its name: Example 3.
- A cloned or dual-account instance missed.
- External or SD card app data ignored.
- A work profile or second user not examined.
- A shallow extraction relied on for sandbox data.
- An unfamiliar app's field read on an assumed meaning.

When to involve a digital forensic expert

Whenever the evidence may lie inside an app on an Android device, and especially where a party may have concealed an app, an account or a second life on the phone: the expert enumerates every app from the system's package records across all profiles, finding the hidden, disguised, sideloaded, cloned and dual-account apps that the launcher does not show (Examples 1 and 2, §4, §5), and verifies each relevant app's identity, version and modification so that a familiar name is never taken for genuine behaviour (Example 3, §6). Each app's data is recovered at depth from its sandbox, from external storage and the SD card, and from Google and own-cloud backups, including deleted records (§5, guides 162, 166, 164); its fields are read for their true meaning and the interpretation validated; and the activity is attributed to a person with corroboration from the provider and counterparts and reported at honest confidence under CPR Part 35 or CrimPR Part 19. This guide continues the Android block. Through **cflab.uk** and **e-discovery.uk**, Android app work reaches the evidence inside every app on the device, including the ones a party hoped no one would find, and reads each on its own terms so that what the apps hold becomes reliable evidence rather than confident nonsense.

§ 13 · COMMON QUESTIONS

Frequently asked questions

How does Android app forensics differ from iPhone app forensics?

The core is the same; the openness adds more (§2). On both, every app is its own world, storing data in its own databases that must be understood on their own terms. Android adds sideloading, so devices run obscure, modified or deceptive apps; cloned and dual apps running second accounts; hidden and disguised apps; app data on the SD card and shared storage; work profiles and multiple users; and selective Google backup. So Android app work starts with a complete inventory from the package records and verifies each app's identity, then applies the same recovery and interpretation disciplines.

What is a cloned or dual app?

A second copy of an app running under a different account on the same phone (§4, Example 1). Manufacturer features and third-party tools let a user run, say, two WhatsApps or two social-media accounts, each with its own sandbox and data. A party may disclose one account and deny the other. The complete inventory, built from the system's package records rather than the visible icons, finds the second instance, whose data is recovered and understood as its own world, and whose account is corroborated with the provider and the counterparts who dealt with it.

Can apps really be hidden on a phone?

Yes, and the open ecosystem makes it easy (§4, Example 2). Apps can be hidden from the launcher, disguised as innocuous tools (a calculator that is a vault for hidden photos), or sideloaded from outside the official store. So the app inventory is never built from what is visible on the screen; it is enumerated from the system's package records across all user and work profiles, and each app's real identity and behaviour is verified rather than assumed from its icon or name, because the innocuous app may be the one holding the evidence.

Why verify an app's version and whether it's modified?

Because a modified app behaves differently, and reading it as genuine produces confident nonsense (§6, Example 3). Sideloaded and modified versions of popular apps, including messaging apps altered to defeat deletion or change storage, are common on Android, and their data structures and deletion behaviour differ from the genuine app's. So each app's identity, version and any modification are established, and its interpretation validated by testing, documentation or corroboration with the provider, before any conclusion is drawn about what its data shows, particularly about what was and was not deleted.

Is app data backed up to the Google account?

Only selectively (§3, guide 164). The Google account backs up app data only for apps that opt in, so some apps' data is in the cloud and some is not, and it cannot be assumed either way. Where an app opted in, its backup is a source reachable regardless of the phone's lock; where it did not, the sandbox, the SD card, the app's own cloud and the counterparts are the routes. Each app's backup status is established, and the device, backup and cloud copies are reconciled rather than assumed to be complete or identical.

Does app activity prove the phone's owner did it?

It shows account activity; attributing it to a person takes care, and cloned apps make this more important (§6, guide 105). One phone may hold several accounts in the same app, and devices and accounts may be shared, so activity is attributed to the account first and to a person only with corroboration from other evidence, the provider and counterparts. Attribution is established, not presumed, and stated at honest confidence, so a match, a transaction or a conversation is tied to a particular person only where the evidence supports it.

§ 14 · REFERENCE

Glossary

Sandbox
An app's private, protected storage area.

Sideloading
Installing an app from outside the official store.

APK
An Android app installer file.

Cloned / dual app
A second instance of an app under another account.

Hidden / disguised app
An app concealed from the launcher or in disguise.

Modified app
An altered version of an app with changed behaviour.

Package records
The system's list of every installed app.

Work profile
A separate Android profile with its own apps.

External storage
Shared storage and the SD card apps may use.

Selective backup
Google backing up only opted-in apps' data.

Sources and authoritative references

The Android app disciplines in this guide draw on materials published by our laboratory and e-discovery practice, referenced here as sources:

- cflab.uk, digital forensics services (Android app and sandbox forensics, sideloaded and cloned app analysis, CPR Part 35 and CrimPR Part 19 reporting): cflab.uk/digital-forensics-services · guides library: cflab.uk/guides
- e-discovery.uk, EDRM Phase 02 · Identification and Phase 06 · Analysis (app identification and data interpretation as practised): e-discovery.uk/edrm/identification · e-discovery.uk/edrm/analysis
- e-discovery.uk, practice method and Knowledge Centre: e-discovery.uk · e-discovery.uk/knowledge
- Android developer documentation, app data storage, sandboxing and backup: developer.android.com, data storage · developer.android.com, Auto Backup · Android security (app sandbox): source.android.com, app sandbox
- Forensic Science Regulator, Code of Practice v2 (validation and honest reporting) and ISO/IEC 27037: gov.uk, FSR Code · iso.org, 27037
- PD 57AD and CPR Part 31 · CPR Part 35 · UK GDPR: justice.gov.uk, PD 57AD · justice.gov.uk, Part 35 · ico.org.uk

§ HOW A SPECIALIST LABORATORY CAN ASSIST

Working with Computer Forensics Lab

Android app work at the laboratory reaches the evidence inside every app on the device, including the ones a party hoped no one would find. Every installed app is enumerated from the system's package records across all user and work profiles, so that hidden, disguised, sideloaded, cloned and dual-account apps are found rather than missed because they do not appear on the screen (Examples 1 and 2). Each relevant app's identity, version and any modification are verified, because on Android a familiar name does not guarantee genuine behaviour and a modified app read as the original produces confident nonsense (Example 3, guide 156). Each app's data is recovered at sufficient depth from its sandbox, its databases, files, caches and preferences, and from shared storage, any SD card, and Google or own-cloud backups where they exist, with deleted records restored from remnants, journals and earlier states (guides 162, 166, 164, 155). Fields, statuses and timestamps are read for their true meaning and the interpretation validated by testing, documentation and corroboration with the provider and counterparts, and activity, including from cloned instances, is attributed to a person with care. Findings are reported at honest confidence, with the limits of interpreting obscure or modified apps stated, under CPR Part 35 or CrimPR Part 19, and handled proportionately under the UK GDPR. This guide continues the Android block. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and the guiding discipline is the open ecosystem's own demand: inventory everything, verify what each app is, and understand each on its own terms.

INSTRUCT THE LAB

Speak to a forensic examiner, not a salesperson.

Describe your matter in confidence and an examiner will respond the same working day. Conflicts checks and scoping calls with US counsel are routine.

NEW ENQUIRIES
+44 (0)20 7164 6971

EMAIL
info@cflab.uk

E-DISCOVERY
e-discovery.uk

Computer Forensics Lab Ltd · Euro House, 133 Ballards Lane, Finchley, London N3 1LJ, United Kingdom
cflab.uk · Customer service +44 (0)20 7164 6915 · ICO ZB395023 · NPCC · ISO 17025 aligned practice · UK Gov Digital Marketplace