JONE / CLIENT SECURITY RESEARCH

Defensive Reverse Engineering of WeCom Clients

Client storage, protocol structures, text, groups, and media lifecycle

SAMPLE EDITION

One complete experiment + selected chapters

JONE Security Research

Independent research | English edition | September 2026

Publication Note

Defensive Reverse Engineering of WeCom Clients

Client Storage, Protocol Structures, Messaging, and Controlled Experiments

Author: JONE Security Research. English public redacted edition V2.0. Research manuscript dated September 2, 2026; publication package prepared September 3, 2026.

This is a SAMPLE EDITION. It contains selected opening chapters, the laboratory baseline, and one complete experiment (WC-L02). Excerpts retain their original chapter and experiment numbering. The full English edition includes 14 experiment cards.

Independent research

This is an independent publication. It is not affiliated with, sponsored by, endorsed by, or an official manual of the companies or products discussed. Product names identify the subjects of analysis; their trademarks remain with their respective owners.

Evidence and scope

The findings belong to the fixed samples described in the text. Observation, inference, negative evidence, and unresolved questions retain their stated grades. Reproducing an experiment requires an authorized sample and a separate reader record; this edition makes no claim to have retested later versions.

WeCom is also known as WeChat Work. Results are tied to the documented PC and Android samples; unresolved fields and version-dependent names retain their evidence limits.

AI assistance disclosure

AI tools assisted with organizing research records, drafting, translation, editing, and publication layout. AI-generated prose is not experimental evidence. Evidence claims must be read together with the fixed-build records, controls, and limitations stated in the manuscript.

Intended use

For authorized defensive research using owned samples, read-only copies, and synthetic data. The publication does not distribute real accounts, credentials, private messages, working decryption or key-recovery tools, authentication bypasses, active-send callers, or bulk extraction tooling.

Navigation Contents

Part I - Evidence Narrative and Architectural Findings

This part preserves the complete fixed-build evidence narrative: sample boundaries, key objects, fields, call chains, state machines, counterevidence, and defensive implications. It answers what the findings are and where their evidence resides. Part II rewrites those findings as experiment cards so a reader can determine how to validate them on an owned sample, what constitutes a pass, and when a claim must be downgraded.

> Reproducibility principle: reproduce methods, structures, criteria, and evidence grades-not third-party accounts, secret material, or active-call capability. Every public lab uses owned samples, read-only copies, and synthetic data.

Scope and Security Boundary

This book is based on fixed samples of WeCom Android 4.1.41, Android 5.0.8, and Windows 5.0.8.6009. It documents database key material, a login-response protobuf, local business models, group-member relationships, media persistence, text/image message objects, and cross-version validation.

The public edition retains responsibilities of concrete functions and classes, parameter types, field numbers, protocol nesting, data relationships, state machines, validation coverage, and refuted claims. It removes live tenants, accounts, customers, conversations, devices, paths, servers, raw keys, seeds, input and outer keys, login packets, RVAs, and working database-recovery, active-send, reflection, or Hook calls.

The root material differs by platform. Android closes a deterministic local pure function from rawkey to database seed. Windows shows a server-provided account key that is stable across devices and not persistently recoverable locally. The two must not be merged into a fictional universal algorithm.

> The book explains how a claim was proven without delivering a way to execute that claim against a live account. Every example value is synthetic or empty.

Chapter 1 - Samples and Evidence Management

Platform	Sample	Principal task
Android	4.1.41.55184 arm64	Main database and message-object evidence
Android	5.0.8 / versionCode 70675	Item-by-item isomorphism check for text and image paths
Windows	5.0.8.6009	Login response, input key, database derivation, and cross-device validation

The study preserves APK/PE hashes, ABI, decompiler versions, runtime timeline, and an evidence index. Final coverage includes 7/7 offline validation samples and 10/10 critical call-location checks; exact addresses remain internal. At least 16 rounds of candidate conclusions were corrected, and retired claims remain in the audit trail with reasons.

Evidence grades are E8 for runtime/database/final-result closure; E7 for static and dynamic corroboration; E6 for strong field-flow, offline, or controlled-variable evidence; E5 for cross-version/cross-device confirmation; E3 for sample statistics; H for hypothesis; and R for refuted.

Chapter 2 - Cross-Platform Local-State Map

Android divides into account-scoped root material, deterministic derivation, SQLCipher business databases, and media files. The Windows client obtains a root input key from the login response, manages it in DbKeyManager, and derives outer and database-opening material. When the process ends without a new login response, disk does not provide equivalent root context.

Dimension	Android	Windows
Root source	Controlled local sample state	Server login response
Account isolation	Rawkey/derived result varies by account	Input key stable per account

Dimension	Android	Windows
Device relation	Conclusion bounded to local samples	Same account agrees across devices
Persistence	Verifiable remnants after logout	No persistent equivalent input key found
Business DB	SQLCipher/private wrapper	Proprietary page container and derivation

Part II - Full Experimental Validation Manual

Laboratory Baseline

Create a separate directory for every experiment and begin with a `sample-card`, `toolchain-card`, and `experiment-log`. The sample card records app version, versionCode, ABI, SHA-256, acquisition date, and authorization scope. The toolchain card records versions of decompilers, signature parsers, database tools, and scripts. The experiment log records start/end time, unique lab id, actions, observations, anomalies, and hashes of raw evidence. No live account, device, contact, conversation, message, media, credential, key, endpoint, or exact address may enter the public manuscript.

Common Isolation Conditions

1. Use only test accounts, devices, and application artifacts that you own or are explicitly authorized to study.

2. Preserve originals read-only; analyze copies and recompute hashes before and after each lab.

3. Dynamic observation is read-only and pass-through: never alter returns, bypass checks, or act on behalf of third parties.

4. Network-facing tests use minimal synthetic messages; structural work that can be completed offline remains offline.

5. Every causal claim requires at least one negative control; correlation without a control is not promoted to causation.

6. For a new build, remap tags, fields, states, and outcome anchors before using any old obfuscated name or address.

Record Format

Each card has fourteen elements: research question; sample/build; environment and prerequisites; static anchors; inputs; structure; outputs and side effects; analysis path; positive procedure; expected and observed evidence; negative controls; failures and corrections; version/evidence/publication boundary; and defensive meaning with pass/fail/inconclusive criteria. “Observed result” in this book belongs to the author's fixed-build evidence; readers must record their own result separately rather than overwrite it.

Experiment 02 | WC-L02 | Map Android Local State with Minimal-Action Diffs

1. Research Question

How are account material, database derivation, business tables, and media files separated, and why does readable not equal usable?

2. Sample, Build, and Evidence Scope

Lab id: `WC-L02`; evidence mapping: `WC-02`; fixed sample: `WeCom Android 4.1.41-5.0.8`. This lab makes claims only for the listed sample.

3. Environment, Authorization, and Prerequisites

Follow the laboratory baseline in this part. Keep originals read-only, pin tool versions, restrict dynamic observation to owned accounts with pass-through behavior, and publish synthetic values only.

4. Static Locators and Evidence Anchors

Preferences/account material, database header/open call, schema, media magic, write time, and callers.

5. Inputs, Provenance, and Constraints

Before/after read-only snapshots for login, text, group/member change, image, and logout actions.

6. Data, Protocol, or Object Structure

State map: layer, object, account scope, create/update/delete event, reader/writer, business meaning, and evidence grade.

7. Outputs, State Changes, and Side Effects

Four layers separate and primary write targets remain stable. A recognizable media file does not prove its index, conversation relation, or lifecycle is understood.

8. Analysis Path and Call Chain

Minimal action → file/page diff → container identification → schema/caller → UI/business outcome.

9. Positive Validation Procedure

1) No-action baseline; 2) one action per run; 3) stop and snapshot; 4) diff hash/mtime/pages; 5) identify magic/schema; 6) trace readers/writers; 7) repeat three times; 8) separate byte readability, structural parsing, and business-semantic recovery.

10. Author Observation and Reader Record

Expected/observed pattern in the author's fixed build: Four layers separate and primary write targets remain stable. A recognizable media file does not prove its index, conversation relation, or lifecycle is understood. Readers create a separate result column containing actual values, timing window, raw-evidence hash, and any divergence.

11. Negative and Control Experiments

Extension-only classification fails. A no-action baseline removes background noise. An orphan file without database relation supports container facts only.

12. Failed Hypotheses, Anomalies, and Corrections

Paths and table names invite overinterpretation. Business roles need diff, caller, and UI agreement.

13. Version, Evidence Grade, and Publication Boundary

Evidence E1+E3; publication B. Live directories, account material, and business data are C.

14. Defensive Meaning and Pass/Fail Criteria

Pass: action-object-caller relation repeats. Fail: filename equals semantics. Inconclusive: background noise not isolated.

About the Full Edition

The full edition of **Defensive Reverse Engineering of WeCom Clients**, by JONE Security Research, contains the complete evidence narrative and 14 experiment cards. Each experiment includes scope, inputs, structures, positive validation, controls, failed hypotheses, version boundaries, and pass/fail criteria.

Find the full edition on Leanpub by its exact title and author. This sample preserves original chapter and experiment identifiers; the navigation contents above list only the included excerpts.

WeCom is also known as WeChat Work. Results are tied to the documented PC and Android samples; unresolved fields and version-dependent names retain their evidence limits.