Agents as Auditors: Detecting Malicious UI Flows via Trajectory-Level Analysis

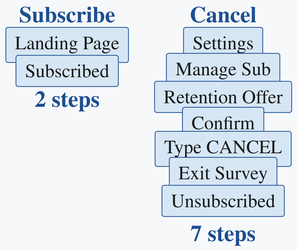
Aryan Kaul[1,†] Yuhang Chen[2,†] Zhuo Zhang[1] Tianlong Chen[2]

[1]Columbia University [2]UNC Chapel Hill † equal contribution

The Problem

Mobile apps trap users with sequences of individually benign screens. The deception lives in the flow, not the frame.

Subscribe in 2 taps. Cancel in 7 screens.



Sept 2025: FTC settled with Amazon for **$2.5B** over a cancellation flow designed to prevent users from leaving Prime.

Why Existing Detectors Fail

Every prior automated detector operates on **single screenshots**. None accept interaction sequences as input.

A per-screen classifier cannot count consecutive retention offers, or compare the length of a subscribe path to a cancel path.

The signal is temporal. They cannot see it.

Our Insight

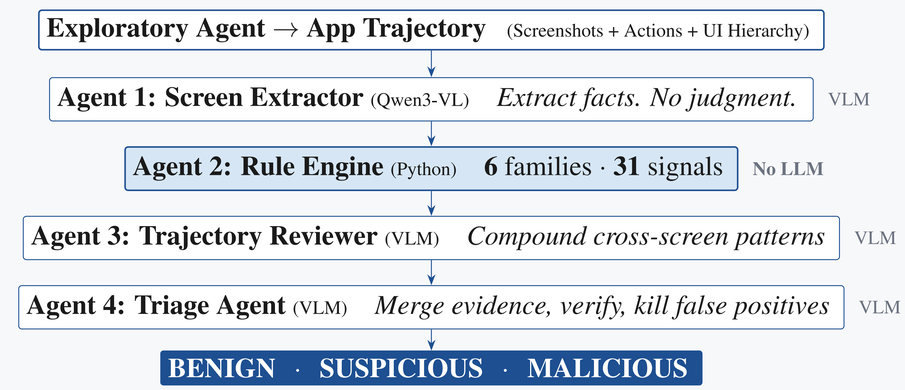
VLM agents navigating apps already produce ordered interaction traces of screenshots, actions, and UI state.

These traces contain the temporal signal that per-screen detectors miss. Prior work shows agents traverse manipulative flows end to end (DECEPTICON: >70% susceptibility).

We model each interaction as an intent-response pair and detect where the app systematically works against user intent.

DPSentry Pipeline

The agent is not just the user simulator. It is the evidence collector.



Rule families: Forced Action (4 signals), Subscription Traps (7), Nagging (3), Permission Abuse (5), System Impersonation (5), Disruptive Ads (7).

Why Split Extraction from Judgment?

When a VLM is asked to both observe and judge a screen, it hallucinates dark patterns on benign apps. Chrome's sync dialog was labeled obstruction. A Contacts app was labeled malicious.

Separating extraction from judgment fixes this. Agent 1 extracts facts. Agent 2 applies deterministic rules. No single VLM call makes the final decision.

Method	Accuracy	False Positives
Zero-shot VLM judge	62%	3/8
Single-screenshot baseline	75%	0/8
DPSentry (ours)	**100%**	**0/8**

Evaluated on 8 labeled real trajectories (5 benign + 2 malicious + 1 neutral). Reproduced on Qwen3-VL-8B and Qwen2.5-VL-7B.

Preliminary Results

Score separation across 8 real trajectories.

Stable across 22 threshold configurations.

Real-world findings.

10 top-downloaded Google Play apps (>1B installs combined) showed policy-relevant behavior within minutes of normal use. All 10 reported to Google.

Malicious app sample.

Of 8 apps flagged by anti-malware: 7/8 were permission harvesters. 1/8 was a fake antivirus.

Worked Example: Fake Antivirus

Per-screen analysis: each screen looks like a system tool. The trajectory reveals the pattern:

1. Security scanner branding
2. Scans system packages
3. Always reports problems found
4. Offers no real remediation
5. APK only 3.4MB (too small for real AV)

Verdict: System Impersonation → MALICIOUS

Next Step: Scaling the Audit Agent

- **Exploratory agent.** Automate trace collection via VLM-driven Android exploration.
- **Sub-trace selection.** Segment traces into onboarding, subscription, cancellation flows.
- **Scale. 200+** apps; measure agreement with Google enforcement.