INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-05-19 18:49 UTC

# Trapdoor Android Ad Fraud Network: 455-App Malvertising Pipeline Generating 659M Fraudulent Bid Requests Daily

THREAT CAMPAIGN | HIGH | CVSS 7.5

| | |
|---|---|
| SCC Item ID | SCC-CAM-2026-0340 |
| Type | Threat Campaign |
| Severity | HIGH |
| CVSS Base Score | 7.5 |
| Affected Products | Android mobile devices; Google Play Store-hosted applications (455 apps, now removed); Google attribution and install tracking infrastructure |
| Published | 2026-05-19T12:38:12 |
| Discovery Source | Rss |

## Executive Summary

A large-scale Android advertising fraud operation named 'Trapdoor' used 455 malicious apps distributed through Google Play Store to generate 659 million fraudulent ad bid requests daily, siphoning advertising revenue from brands and media buyers. The campaign affected any organization running programmatic advertising on mobile platforms, with fraudulent installs and attribution data corrupting campaign performance metrics. Google has removed the identified apps; the primary residual risk is financial loss from ad spend already consumed and ongoing exposure if related clusters remain active.

## Technical Analysis

Trapdoor is a campaign-based Android ad fraud operation documented by HUMAN Security's Satori Threat Intelligence team. No CVE is assigned; the threat is behavioral malware rather than a discrete software vulnerability. Relevant CWEs: CWE-940 (improper verification of source of a communication channel), CWE-1021 (improper restriction of rendered UI layers, clickjacking-adjacent), CWE-693 (protection mechanism failure). The operation distributed 455 apps via Google Play Store backed by 183 C2 domains. Architecture was multi-stage: initial installs via threat actor-controlled ad campaigns triggered malvertising payloads that drove secondary installs; secondary installs performed the core RTB fraud. Evasion relied on selective payload activation tied to install attribution data, the fraud module only activated when the install originated through Trapdoor-controlled ad channels, preventing detection by researchers installing apps outside those channels. MITRE ATT&CK techniques include T1418 (software discovery), T1624.001 (event-triggered execution:

broadcast receivers), T1583.006 (web services for C2 infrastructure), T1036.001 and T1036.005 (masquerading), T1406 (obfuscated files), T1628.002 (hide artifacts), T1437 (standard application layer protocol), T1444 (masquerade as legitimate application), T1583.001 (acquire infrastructure: domains), T1664 (exploitation for installation), T1627 (exploitation for defense evasion), and T1517 (access notifications). Related clusters include SlopAds, Low5, and BADBOX 2.0. All 455 identified apps have been removed from Google Play; C2 domain status should be independently verified.

## Action Checklist

1. Containment: Audit Mobile Device Management (MDM) or enterprise Android deployments for the 455 Trapdoor apps. Compare installed apps against HUMAN Security's published IOC list. Block the 183 known C2 domains at DNS and network egress immediately.
2. Detection: Review mobile ad attribution logs for anomalous install spikes, mismatched attribution sources, or install events that do not correlate to legitimate campaign spend. Query network logs for DNS resolution or HTTP/S connections to the 183 known Trapdoor C2 domains. Look for broadcast receiver abuse patterns (T1624.001) in mobile threat detection tooling if deployed.
3. Eradication: Remove any identified Trapdoor-affiliated apps from managed devices via MDM. Revoke any mobile advertising SDK tokens or attribution keys associated with compromised campaigns. Rotate ad platform API credentials if those credentials were accessible from affected devices.
4. Recovery: Validate that C2 domain blocks are in place and confirm no further beaconing from previously infected devices. Audit programmatic ad campaign performance data for the period of exposure; flag invalid traffic reports to ad platforms and demand-side platforms (DSPs) for credits or refunds where applicable. Re-baseline attribution data.
5. Post-Incident: Review mobile app vetting procedures for enterprise-approved app lists; establish a periodic review cadence against threat intelligence feeds. Evaluate whether your ad fraud detection vendor or ad measurement partner has controls for install attribution spoofing. Map control gaps against CWE-940 and CWE-693 to identify where attribution verification and channel validation can be hardened.

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | URGENT |
| **Escalation Criteria** | Escalate immediately to legal and finance leadership if Trapdoor-affiliated app installs or fraudulent attribution events are identified on devices used by employees with access to ad platform billing credentials or DSP media budgets exceeding organizational materiality thresholds, or if IVT exposure triggers contractual clawback clauses with advertising partners that could constitute a reportable financial event. |
| **Recovery Notes** | Post-eradication, monitor DNS egress logs and MMP attribution pipelines daily for a minimum of 30 days for any recurrence of beaconing to the 183 C2 domains or anomalous install attribution patterns, as Trapdoor-affiliated SDKs may persist in apps not yet identified in the 455-app IOC list. Re-baseline all programmatic campaign KPIs (CPI, CTR, ROAS) against the cleaned attribution dataset and establish a new fraud-adjusted performance baseline before resuming full media spend. Retain all forensic exports — DNS logs, MMP CSVs, device GAID lists, and network captures — for a minimum of 12 months to support any DSP refund disputes or audit requirements. |

| | |
|---|---|
| **Forensic Artifacts** | MMP (AppsFlyer, Adjust, or Branch) raw attribution event logs for the full exposure window: specifically install events with click-to-install times under 10 seconds, organic installs from devices with no paid touchpoint in the attribution window, and installs attributed to sources inconsistent with active campaign spend — all forensic signatures of Trapdoor's broadcast receiver-based click injection (MITRE T1624.001). \| DNS resolver query logs (Pi-hole, BIND query log, or Windows DNS debug log) filtered for the 183 known Trapdoor C2 domains: preserve the full query record including source device IP, query timestamp, response IP, and query frequency — periodic high-frequency queries at regular intervals are the behavioral fingerprint of Trapdoor SDK beaconing. \| ADB-extracted APK files and app data directories (/data/data/[trapdoor_package_name]/shared_prefs/ and /cache/) from at least one representative infected device prior to eradication: these directories may contain cached fraudulent bid request payloads, SDK configuration files referencing C2 endpoints, and locally stored GAID values used in fraudulent attribution events. \| Network capture (PCAP) of HTTP/S traffic from affected devices to Trapdoor C2 infrastructure: specifically the POST request bodies containing fabricated OpenRTB bid request JSON, which will include spoofed device metadata (make, model, OS version, GAID) used to generate the 659 million daily fraudulent bid requests — this payload structure is the primary technical evidence of the fraud mechanism. \| Google Play Store install history and MDM enrollment records correlating device IDs to the install timestamps of the 455 Trapdoor-affiliated apps: this artifact establishes the exposure window per device and is required to scope the financial impact of fraudulent attribution to specific campaign flight dates and ad spend periods. |

### Per-Action IR Details

**Containment — Audit any Mobile Device Management (MDM) or enterprise Android deployment for the 455 apps identified in the Trapdoor campaign. Cross-reference installed app lists against HUMAN Security's published IOC list. Block the 183 known C2 domains at DNS and network egress layers immediately.**

**NIST Phase:** Containment

**Reference:** NIST 800-61r3 §3.3 — Containment Strategy

**Controls:** NIST IR-4 (Incident Handling), NIST SI-3 (Malicious Code Protection), NIST SC-7 (Boundary Protection), CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.3 (Address Unauthorized Software)

**Compensating:** For teams without MDM: use ADB (Android Debug Bridge) across managed devices to dump installed package lists via 'adb shell pm list packages -f > installed_pkgs.txt', then diff against the HUMAN Security IOC app package name list. For DNS blocking without enterprise DNS filtering: push a blocklist of the 183 Trapdoor C2 domains to Pi-hole (free, deployable in under 1 hour) or add them as null-route entries in your local DNS resolver (e.g., BIND or Windows DNS via PowerShell: 'Add-DnsServerPrimaryZone' for each domain). For network egress, create deny rules on the perimeter firewall or pfSense for the 183 C2 FQDNs and their resolved IPs before blocking, capturing one final DNS resolution to document the IP-to-domain mapping.

**Evidence:** Before executing blocks, capture: (1) Full DNS query logs from your resolver for the 183 Trapdoor C2 domains — note timestamps, source device IPs, and query frequency, as high-frequency periodic queries indicate active SDK beaconing from still-installed apps. (2) NetFlow or firewall session logs showing HTTP/S connections to Trapdoor C2 infrastructure, specifically POST requests used by ad fraud SDKs to exfiltrate fabricated bid request telemetry. (3) MDM enrollment records showing which devices had the flagged apps installed, correlated with the install timestamps relative to HUMAN Security's identified campaign window. (4) Screenshot or export of the MDM app inventory report before removal, preserving app version numbers and install dates for each of the 455 package names.

**Detection — Review mobile ad attribution logs for anomalous install spikes, mismatched attribution sources, or install events that do not correlate to legitimate campaign spend. Query network logs for DNS resolution or HTTP/S connections to the 183 known Trapdoor C2 domains. Look for broadcast receiver abuse patterns (T1624.001) in mobile threat detection tooling if deployed.**

**NIST Phase:** Detection Analysis

**Reference:** NIST 800-61r3 §3.2 — Detection and Analysis

**Controls:** NIST SI-4 (System Monitoring), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-2 (Event Logging), NIST IR-5 (Incident Monitoring), CIS 8.2 (Collect Audit Logs), CIS 7.1 (Establish and Maintain a Vulnerability Management Process)

**Compensating:** Without a mobile SIEM or MTD platform: (1) Export attribution data from your MMP (AppsFlyer, Adjust, or Branch) as CSV and run a Python script or Excel pivot to flag installs where click-to-install time is under 10 seconds (a strong indicator of Trapdoor-style click injection via T1624.001 broadcast receiver abuse). (2) Use Zeek or Wireshark on a network tap or egress mirror to capture DNS queries and HTTP POST bodies to the 183 C2 domains — filter in Wireshark with 'dns.qry.name contains [domain]' for each known C2. (3) For host-level detection on Android, if devices are rooted or in a test environment, use 'adb shell dumpsys package [package_name]' to enumerate broadcast receiver registrations for ACTION_PACKAGE_ADDED or INSTALL_REFERRER intents, which Trapdoor abused to hijack attribution. (4) Deploy the free HUMAN Security GIVT/IVT detection API if your ad platform supports it for post-hoc traffic validation.

**Evidence:** Capture before concluding detection scope: (1) MMP (mobile measurement partner) raw attribution logs for the exposure period — specifically look for installs attributed to organic or direct channels from devices with no corresponding paid media touchpoint, which is the fingerprint of Trapdoor's fraudulent attribution injection. (2) Network proxy or firewall logs showing the specific URI paths used by Trapdoor SDK callbacks (typically formatted as programmatic OpenRTB bid request JSON POSTed to C2 subdomains). (3) Android logcat output from affected devices (via 'adb logcat -d > device_log.txt') filtered for package names matching the 455 Trapdoor apps, capturing SDK initialization events and outbound connection attempts. (4) Ad platform impression and click logs cross-referenced against the device IDs (GAID — Google Advertising ID) of enrolled MDM devices to identify which device IDs were submitting fraudulent bid requests.

**Eradication — Remove any identified Trapdoor-affiliated apps from managed devices via MDM. Revoke any mobile advertising SDK tokens or attribution keys associated with compromised campaigns. Rotate ad platform API credentials if those credentials were accessible from affected devices.**

**NIST Phase:** Eradication

**Reference:** NIST 800-61r3 §3.4 — Eradication

**Controls:** NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST SI-7 (Software, Firmware, and Information Integrity), NIST IA-3 (Device Identification and Authentication), CIS 2.3 (Address Unauthorized Software), CIS 5.1 (Establish and Maintain an Inventory of Accounts)

**Compensating:** Without MDM for bulk removal: use ADB in a scripted loop — 'for device in $(adb devices | grep -v List | awk "{print $1}"); do adb -s $device uninstall [package_name]; done' — iterating across all 455 package names. For credential rotation without a secrets manager: export current MMP SDK keys and DSP API keys to a temporary audit log before rotation (preserving the compromised key values as forensic evidence), then generate new keys via each platform's API or console and redeploy via your CI/CD pipeline or manual config update. For apps where removal is not immediately possible due to user-owned BYOD devices, push a network policy via MDM to block those devices' GAIDs from submitting attribution events until the app is removed.

**Evidence:** Before eradication actions: (1) Preserve a full forensic image or ADB backup ('adb backup -apk -shared -all -f trapdoor_backup.ab') of at least one representative affected device as an evidentiary sample before app removal, capturing the Trapdoor app APK, its shared preferences, and any locally cached fraudulent bid request payloads stored in the app's data directory (/data/data/[package_name]/). (2) Document the specific MMP SDK token values and DSP API key IDs that were in-scope — these are the credentials that Trapdoor's C2 infrastructure could have harvested to generate authenticated fraudulent attribution events. (3) Export a final list of all device GAIDs (Google Advertising IDs) associated with affected installs — these identifiers were the primary vehicle for fraudulent bid requests and will be needed for invalid traffic claims with DSPs.

**Recovery — Validate that C2 domain blocks are in place and confirm no further beaconing from previously infected devices. Audit programmatic ad campaign performance data for the period of exposure; flag invalid traffic reports to ad platforms and demand-side platforms (DSPs) for credits or refunds where applicable. Re-baseline attribution data.**

**NIST Phase:** Recovery

**Reference:** NIST 800-61r3 §3.5 — Recovery

**Controls:** NIST IR-4 (Incident Handling), NIST AU-6 (Audit Record Review, Analysis, and Reporting), NIST AU-11 (Audit Record Retention), NIST SI-4 (System Monitoring), CIS 8.2 (Collect Audit Logs), CIS 7.2 (Establish and Maintain a Remediation Process)

**Compensating:** Without a commercial IVT (Invalid Traffic) detection platform: query your DNS resolver logs or Pi-hole query log for any continued resolution attempts against the 183 C2 domains post-block — any hits after block deployment indicate a device that still has a Trapdoor app installed and active. For campaign data re-baselining without a BI platform: export raw impression, click, and install CSVs from your DSP and MMP for the exposure window, remove all rows where the device GAID appears on your affected-device list, and recalculate CPI (cost per install) and ROAS (return on ad spend) on the cleaned dataset. Submit the delta as an IVT dispute to the DSP using the IAB Tech Lab's IVT classification taxonomy (GIVT/SIVT) to support refund claims.

**Evidence:** During recovery validation: (1) DNS resolver negative-query logs confirming zero successful resolutions of the 183 Trapdoor C2 domains post-block, with timestamps, as proof of effective containment for any DSP dispute documentation. (2) MMP attribution reports for the re-baselined period, exported before and after removal of fraudulent install events, documenting the volume and dollar value of invalid attribution — this is required for financial recovery claims against ad platforms. (3) Firewall or proxy egress logs from the 72 hours post-eradication showing absence of HTTP POST traffic to previously identified Trapdoor C2 IP addresses, confirming no residual SDK activity from re-installed or missed apps.

**Post-Incident — Review mobile app vetting procedures for enterprise-approved app lists; establish a periodic review cadence against threat intelligence feeds. Evaluate whether your ad fraud detection vendor or ad measurement partner has controls for install attribution spoofing. Map control gaps against CWE-940 and CWE-693 to identify where attribution verification and channel validation can be hardened.**

**NIST Phase:** Post Incident

**Reference:** NIST 800-61r3 §4 — Post-Incident Activity

**Controls:** NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST RA-3 (Risk Assessment), NIST SI-10 (Information Input Validation), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 2.1 (Establish and Maintain a Software Inventory)

**Compensating:** Without a commercial mobile threat intelligence feed: subscribe to HUMAN Security's free threat advisories and the Google Play Protect threat bulletin (both public) and schedule a monthly review against your MDM-enforced app allowlist. For CWE-940 (Improper Verification of Source of a Communication Channel) gap assessment: audit your MMP configuration for whether S2S (server-to-server) postback verification with a shared secret or signed callbacks is enabled — this is a free MMP feature that prevents Trapdoor-style unauthenticated attribution injection. For CWE-693 (Protection Mechanism Failure) mapping: review whether your ad SDK configurations enforce certificate pinning on attribution postbacks, using the free MobSF (Mobile Security Framework) static analysis tool to scan your own app APKs for missing pinning implementations.

**Evidence:** For the lessons-learned record: (1) Final incident timeline documenting the earliest evidence of Trapdoor C2 beaconing in DNS/network logs versus the HUMAN Security disclosure date — this gap quantifies your detection latency and informs future TI feed subscription decisions. (2) Complete financial impact summary: total ad spend attributed to fraudulent installs across the exposure window, calculated from the re-baselined MMP data, segmented by campaign and DSP — required for executive reporting and any insurance or refund claims. (3) App vetting procedure documentation (or evidence of its absence) as a control gap artifact for the post-incident review, specifically noting whether the 455 Trapdoor apps bypassed any existing enterprise app approval process and why.

## Detection Guidance

Primary detection surface is network and DNS telemetry. Query DNS logs for resolution requests to the 183 Trapdoor C2 domains published by HUMAN Security. Query proxy or firewall logs for HTTP/S connections to

those domains from Android devices on your network or VPN. For mobile ad operations teams: pull invalid traffic (IVT) reports from your DSP or ad measurement platform and look for install events where attribution source does not match any active paid campaign, installs clustering from narrow device cohorts at abnormal rates, or install-to-engagement ratios near zero (indicating non-human activity). Behavioral indicator: broadcast receiver triggering on install events (T1624.001) that initiates network activity to external domains without user interaction. If you have mobile EDR or MTD (mobile threat defense) deployed, query for app behaviors matching obfuscated payload execution post-install (T1406) and hidden artifact creation (T1628.002). Note: Apps are confirmed removed from Google Play per HUMAN Security threat intelligence; detection priority is residual infections on devices that installed apps prior to removal and network-level C2 communication.

## Indicators of Compromise

| Type | Value | Context | Confidence |
|---|---|---|---|
| DOMAIN | `183 C2 domains — full list published by HUMAN Security Satori team` | Command-and-control infrastructure used by Trapdoor campaign to coordinate fraudulent RTB activity and payload activation | HIGH |

## Framework Mappings

### MITRE-ATTACK

- **T1418** — Software Discovery
- **T1624.001** — Broadcast Receivers
- **T1583.006** — Web Services
- **T1036.001** — Invalid Code Signature
- **T1406** — Obfuscated Files or Information
- **T1628.002** — User Evasion
- **T1036.005** — Match Legitimate Resource Name or Location
- **T1598** — Phishing for Information
- **T1437** — Application Layer Protocol
- **T1628** — Hide Artifacts
- **T1444**
- **T1583.001** — Domains
- **T1664** — Exploitation for Initial Access
- **T1627** — Execution Guardrails
- **T1517** — Access Notifications

### ISO-27001-2022

- **A.8.8** — Management of technical vulnerabilities
- **A.5.34** — Privacy and protection of personal information

## MITRE ATT&CK Mapping

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1418 | Software Discovery | Discovery |
| T1624.001 | Broadcast Receivers | Persistence |
| T1583.006 | Web Services | Resource-Development |
| T1036.001 | Invalid Code Signature | Defense-Evasion |
| T1406 | Obfuscated Files or Information | Defense-Evasion |
| T1628.002 | User Evasion | Defense-Evasion |
| T1036.005 | Match Legitimate Resource Name or Location | Defense-Evasion |
| T1598 | Phishing for Information | Reconnaissance |
| T1437 | Application Layer Protocol | Command-And-Control |
| T1628 | Hide Artifacts | Defense-Evasion |
| T1444 | | |
| T1583.001 | Domains | Resource-Development |
| T1664 | Exploitation for Initial Access | Initial-Access |
| T1627 | Execution Guardrails | Defense-Evasion |
| T1517 | Access Notifications | Collection |

## Sources

| Source | URL | Tier |
|---|---|---|
| **Security News** | https://thehackernews.com/2026/05/trapdoor-android-ad-fraud-scheme-... | T3 |
| **Use Google Play Protect to help keep your apps safe & your data ...** | https://support.google.com/googleplay/answer/2812853?hl=en | T3 |
| **Google allegedly stealing app projects without consent - Facebook** | https://www.facebook.com/groups/chatgpt/posts/1256721253102654/ | T3 |
| **How we kept the Google Play & Android app ecosystems safe in 2024** | https://blog.google/security/how-we-kept-google-play-android-app-ec... | T1 |

| Source | URL | Tier |
|---|---|---|
| **[PDF] 50 Ways to Leak Your Data: An Exploration of Apps' Circumvention ...** | https://www.ftc.gov/system/files/documents/public_events/1415032/pr... | T1 |



Generated 2026-05-19 18:49 UTC by TJS Security Command Center