INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-04-21 06:39 UTC

# FakeWallet Campaign Exploits App Store Trust to Harvest Crypto Seed Phrases via iOS Provisioning Abuse

**THREAT CAMPAIGN** | **HIGH** | CVSS 7.5

| | |
|---|---|
| **SCC Item ID** | SCC-CAM-2026-0191 |
| **Type** | Threat Campaign |
| **Severity** | HIGH |
| **CVSS Base Score** | 7.5 |
| **Affected Products** | Apple App Store (iOS), MetaMask, Coinbase Wallet, Trust Wallet, OneKey, Ledger, users in China |
| **Published** | 2026-04-20T17:52:54 |
| **Discovery Source** | Rss |

## Executive Summary

A coordinated campaign attributed to SparkKitty placed 26 trojanized cryptocurrency wallet apps on Apple's App Store, targeting users primarily in China. The apps impersonated MetaMask, Coinbase Wallet, Trust Wallet, and other legitimate wallets, capturing BIP-39 seed phrases through phishing overlays and abusing iOS enterprise provisioning profiles to deliver secondary payloads. Any user who entered a seed phrase into these apps faces irreversible loss of all funds in the original wallet unless they migrate funds to a new wallet immediately. Once a seed phrase is exfiltrated, recovery of that original wallet is impossible.

## Technical Analysis

FakeWallet is a sub-operation of the SparkKitty campaign. Threat actor submitted 26 trojanized wallet apps to Apple's App Store, successfully bypassing App Store review. Two exploitation mechanisms were used: (1) phishing overlay injection (CWE-1021) intercepting BIP-39 mnemonic seed phrases at entry or display, and (2) iOS enterprise provisioning profile abuse (CWE-345, CWE-940) to sideload secondary malware payloads outside App Store review scope. Mapped MITRE ATT&CK techniques include T1476 (Deliver Malicious App via Authorized App Store), T1444 (Masquerade as Legitimate Application), T1036.005 (Match Legitimate Name or Location), T1417 (Input Capture), T1598.003 (Spearphishing Link), T1517 (Access Notifications), T1567 (Exfiltration Over Web Service), and T1532 (Archive Collected Data). No CVE has been assigned; exploited weaknesses are process and trust-model failures rather than discrete software flaws. Apple removed all 26 apps

following Kaspersky disclosure. No patch exists, the attack surface is the App Store review process and user trust in store provenance. Affected wallet software (MetaMask, Coinbase Wallet, Trust Wallet, OneKey, Ledger) was impersonated, not directly compromised.

## Action Checklist

1. Containment: Audit any corporate-issued or BYOD iOS devices enrolled in MDM for installation of the 26 removed apps. Cross-reference installed app bundle IDs against the Kaspersky disclosure app list. Revoke or quarantine devices with confirmed installs immediately.
2. Detection: Query MDM telemetry and mobile threat defense (MTD) logs for apps impersonating MetaMask, Coinbase Wallet, Trust Wallet, OneKey, or Ledger with non-official bundle IDs. Flag any iOS devices with enterprise provisioning profiles not issued by your organization. Review network logs for outbound connections to domains associated with SparkKitty C2 infrastructure as listed in Kaspersky's disclosure.
3. Eradication: Remove identified trojanized apps via MDM remote wipe or app removal command. Revoke unauthorized enterprise provisioning profiles. Require device re-enrollment with attestation for any confirmed-infected device before returning to production use.
4. Recovery: For any user confirmed to have entered a seed phrase into a flagged app, treat the associated wallet as fully compromised. Advise immediate fund transfer to a new wallet generated on a verified, uncompromised device using a trusted source. Verify the replacement wallet app against official publisher developer IDs and App Store listing histories before use.
5. Post-Incident: This campaign exposed a gap in App Store review controls for impersonation-class attacks. Formalize a mobile app vetting policy for any org-approved crypto wallet usage. For high-value crypto custody, mandate hardware wallet use (physical device, not software app). Add SparkKitty and FakeWallet campaign IOCs to threat intelligence feeds. Review BYOD policy to require MTD enrollment for devices accessing corporate resources.

## IR / Forensic Enrichment

| | |
|---|---|
| **Triage Priority** | IMMEDIATE |
| **Escalation Criteria** | Escalate to CISO and legal counsel immediately if any confirmed-infected device belongs to an employee who used the trojanized wallet for business-related crypto transactions, or if organizational funds were custodied in a wallet whose seed phrase was entered into a SparkKitty app — both conditions create potential financial loss reporting obligations and, depending on jurisdiction, regulatory notification requirements. |
| **Recovery Notes** | Recovery for seed phrase compromise is not reversible — the only valid recovery action is fund migration to a new wallet with a seed phrase never entered into any software application on the affected device. Monitor the blockchain addresses associated with all confirmed-compromised wallets for 90 days post-incident using free explorer tools (Etherscan, blockchain.com) to detect delayed or staged fund sweeps, as SparkKitty operators may not drain wallets immediately. Before returning any previously infected device to production use, perform a full factory reset and MDM re-enrollment rather than relying solely on app removal, given the enterprise provisioning profile abuse mechanism which may have delivered persistent secondary payloads beyond the trojanized wallet app. |

| | |
|---|---|
| **Forensic Artifacts** | MDM app inventory records: installed app bundle IDs with timestamps — SparkKitty trojanized apps used bundle IDs mimicking legitimate wallets (e.g., variants of com.metamask.ios, com.coinbase.Coinbase) but with differing developer certificate signatures; the delta between official and observed bundle IDs is the primary artifact. \| iOS provisioning profile configuration: Settings > General > VPN & Device Management — unauthorized enterprise profiles installed by SparkKitty secondary payloads will show issuer certificate chains not matching Apple's App Store distribution or your organization's MDM authority; export profile UUID and certificate hash before removal. \| Network egress logs (DNS + TLS SNI): outbound connections from affected devices to SparkKitty C2 domains as listed in Kaspersky's FakeWallet campaign disclosure — seed phrase exfiltration would appear as a single HTTPS POST (small payload, ~100-500 bytes) to a non-Apple, non-wallet-provider domain shortly after the user entered their seed phrase into the phishing overlay. \| Blockchain transaction history: pull complete transaction records from the wallet addresses derived from any seed phrases entered into flagged apps using Etherscan (EVM), blockchain.com (BTC), or Solscan (SOL) — unauthorized outbound transfers to unknown addresses following the infection window confirm active exploitation and establish financial impact. \| Device backup app plist (iMazing or iTunes backup): the file at HomeDomain/Library/Preferences/com.apple.mobile.installation.plist within an unencrypted iOS backup contains installation history including apps subsequently deleted; this captures evidence of the trojanized app even after MDM-directed removal has occurred. |

### Per-Action IR Details

**Containment — Audit any corporate-liable or BYOD iOS devices enrolled in MDM for installation of the 26 removed apps. Cross-reference installed app bundle IDs against the Kaspersky disclosure app list. Revoke or quarantine devices with confirmed installs immediately.**

**NIST Phase:** Containment

**Reference:** NIST 800-61r3 §3.3 — Containment Strategy: isolate affected assets to prevent further seed phrase exfiltration to SparkKitty C2 infrastructure while preserving forensic state prior to remediation.

**Controls:** NIST IR-4 (Incident Handling), NIST SI-3 (Malicious Code Protection), CIS 1.1 (Establish and Maintain Detailed Enterprise Asset Inventory), CIS 2.3 (Address Unauthorized Software)

**Compensating:** Without MDM, use Apple Configurator 2 (free) to query installed app bundle IDs manually on each device. Export the device app list via `cfgutil get installedApps` and diff against the Kaspersky-disclosed bundle IDs (e.g., com.fake.metamask variants). For BYOD devices not enrolled in any management tool, distribute a written self-attestation form and require users to screenshot their App Library and submit to the IR team within 2 hours.

**Evidence:** Before quarantine, capture a full MDM device inventory report showing installed app bundle IDs, installation timestamps, and provisioning profile UUIDs. Screenshot or export the MDM console record showing app installation date — this timestamps when the trojanized wallet app entered the device, which is critical for establishing the seed phrase exposure window. Do not remove the app before capturing this record.

**Detection — Query MDM telemetry and mobile threat defense (MTD) logs for apps impersonating MetaMask, Coinbase Wallet, Trust Wallet, OneKey, or Ledger with non-official bundle IDs. Flag any iOS devices with enterprise provisioning profiles not issued by your organization. Review network logs for outbound connections to domains associated with SparkKitty C2 infrastructure as listed in Kaspersky's disclosure.**

**NIST Phase:** Detection Analysis

**Reference:** NIST 800-61r3 §3.2 — Detection and Analysis: correlate MDM telemetry, provisioning profile inventory, and network egress logs to scope all devices that may have exfiltrated seed phrases to SparkKitty infrastructure.

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

**Compensating:** Without MTD, use Apple's free Screen Time or Configurator 2 to enumerate provisioning profiles on each device: navigate to Settings > General > VPN & Device Management and photograph any profile not issued by your org. For network detection without a SIEM, run a 48-hour Wireshark or pfSense capture filtered on DNS queries and TLS SNI fields matching SparkKitty domains from the Kaspersky IOC list. Use the free iMazing tool to pull a device backup and inspect the installed apps plist at `HomeDomain/Library/Preferences/com.apple.mobile.installation.plist` for non-App-Store bundle IDs.

**Evidence:** Capture MDM enrollment records showing provisioning profile UUID, issuer, and installation timestamp for every enrolled iOS device. Pull firewall or proxy logs for DNS queries and HTTPS connections to SparkKitty C2 domains as listed in Kaspersky's FakeWallet disclosure — these connections would occur at or shortly after first app launch when the trojanized app transmitted the entered seed phrase. Preserve raw PCAP or NetFlow records for the 30-day window prior to detection, as seed phrase exfiltration is a one-time high-value event that may appear as a single small HTTPS POST to a suspicious domain.

**Eradication — Remove identified trojanized apps via MDM remote wipe or app removal command. Revoke unauthorized enterprise provisioning profiles. Require device re-enrollment with attestation for any confirmed-infected device before returning to production use.**

**NIST Phase:** Eradication

**Reference:** NIST 800-61r3 §3.4 — Eradication: remove the SparkKitty trojanized wallet apps and any associated enterprise provisioning profiles that enabled secondary payload delivery, then verify clean state before re-admission to the environment.

**Controls:** NIST IR-4 (Incident Handling), NIST SI-2 (Flaw Remediation), NIST CM-2 (Baseline Configuration) — implied by re-enrollment attestation requirement, CIS 4.6 (Securely Manage Enterprise Assets and Software), CIS 2.3 (Address Unauthorized Software)

**Compensating:** Without MDM remote-wipe capability, provide users with a step-by-step written procedure: (1) long-press the trojanized app icon and select Remove App; (2) navigate to Settings > General > VPN & Device Management, tap each unauthorized provisioning profile, and select Remove Management; (3) reboot the device. For attestation without a formal MDM enrollment flow, require the user to submit a screenshot of Settings > General > VPN & Device Management showing only org-issued profiles, and a screenshot of App Library showing the offending app absent, before device is reconnected to corporate Wi-Fi or email.

**Evidence:** Before executing app removal, capture the full provisioning profile configuration from the device — specifically the profile UUID, certificate chain, and any embedded payload configurations that may reveal the secondary payload delivery mechanism used by SparkKitty. Export the MDM removal command log with timestamp as proof of eradication for post-incident documentation. If the device can be briefly examined before wipe, check Safari or in-app browser history for phishing overlay domains that mimicked MetaMask or Coinbase Wallet login pages, as these URLs are distinct from legitimate wallet infrastructure.

**Recovery — For any user confirmed to have entered a seed phrase into a flagged app, treat the associated wallet as fully compromised. Advise immediate fund transfer to a new wallet generated on a verified, uncompromised device using a trusted source. Verify the replacement wallet app against official publisher developer IDs and App Store listing histories before use.**

**NIST Phase:** Recovery

**Reference:** NIST 800-61r3 §3.5 — Recovery: restore user ability to conduct crypto transactions only through verified, uncompromised wallet infrastructure — seed phrase compromise is irreversible, so recovery focuses entirely on new wallet provisioning and fund migration, not restoration of the affected wallet.

**Controls:** NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-7 (Software, Firmware, and Information Integrity), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 4.6 (Securely Manage Enterprise Assets and Software)

**Compensating:** Without a formal app vetting tool, verify replacement wallet apps manually using two cross-checks: (1) confirm the App Store developer name matches the exact string used by the legitimate publisher (e.g., 'MetaMask' by ConsenSys Software Inc., 'Coinbase Wallet' by Coinbase, Inc.) and check that the first release date in App Store history predates 2023 — the trojanized SparkKitty apps would have recent first-publish dates; (2) use a second

unrelated device or desktop browser to independently look up the official publisher's developer ID from their website and compare to the App Store listing. Document both verification steps in writing before the user installs.

**Evidence:** Before advising fund migration, capture a record of the wallet address(es) associated with the compromised seed phrase so blockchain transaction history can be monitored for unauthorized transfers — use a free blockchain explorer (e.g., Etherscan for EVM wallets, blockchain.com for BTC) to pull and preserve the full transaction history as of the incident date. This serves as the financial impact baseline for post-incident reporting and any potential law enforcement referral. Document whether funds were already drained, as SparkKitty operators may have automated near-instant sweeping upon seed phrase receipt.

**Post-Incident — This campaign exposed a gap in App Store review controls for impersonation-class attacks. Formalize a mobile app vetting policy for any org-approved crypto wallet usage. For high-value crypto custody, mandate hardware wallet use (physical device, not software app). Add SparkKitty and FakeWallet campaign IOCs to threat intelligence feeds. Review BYOD policy to require MTD enrollment for devices accessing corporate resources.**

**NIST Phase:** Post Incident

**Reference:** NIST 800-61r3 §4 — Post-Incident Activity: document lessons learned specific to App Store impersonation-class supply chain attacks, update BYOD and mobile app policies, and operationalize SparkKitty IOCs to improve detection posture against this threat actor's continued activity.

**Controls:** NIST IR-4 (Incident Handling), NIST IR-8 (Incident Response Plan), NIST SI-5 (Security Alerts, Advisories, and Directives), NIST AU-6 (Audit Record Review, Analysis, and Reporting), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process), CIS 6.3 (Require MFA for Externally-Exposed Applications)

**Compensating:** Without a commercial threat intelligence platform, ingest SparkKitty IOCs (C2 domains, bundle IDs, provisioning profile certificate hashes from the Kaspersky disclosure) into a free MISP instance or a simple Pi-hole blocklist for DNS-layer blocking. Use free YARA rules to scan any future app IPA files for obfuscation patterns associated with the FakeWallet campaign — the Kaspersky disclosure should include behavioral indicators suitable for YARA authoring. Publish the mobile app vetting policy as a one-page checklist distributed via email and require written acknowledgment from all employees who use crypto wallets for any business purpose.

**Evidence:** Compile the full incident record including: MDM audit logs showing affected device count and app installation timestamps, network logs showing any confirmed C2 connections to SparkKitty infrastructure, blockchain explorer snapshots of any compromised wallet addresses showing pre- and post-incident balances, and user attestation records. This documentation package supports both the internal lessons-learned review and any external reporting obligations (e.g., to Apple's App Store fraud team or law enforcement). Preserve all logs per your retention policy — minimum 12 months recommended for a financial-impact incident of this class.

## Detection Guidance

Detection requires mobile-layer visibility. In MDM/UEM platforms, query for installed applications with display names matching MetaMask, Coinbase Wallet, Trust Wallet, OneKey, or Ledger but with bundle IDs that do not match the verified publisher entries. In mobile threat defense (MTD) tools, look for enterprise provisioning profile installations not tied to your organization's certificate authority. Network-layer: inspect proxy or DNS logs for domains and IPs listed in Kaspersky's FakeWallet/SparkKitty disclosure as C2 endpoints (see sources section for Kaspersky press release and validation guidance). Behavioral indicator: any iOS app requesting accessibility permissions or overlay permissions that identifies as a cryptocurrency wallet should be treated as suspicious pending verification. IOCs published by Kaspersky in their disclosure are the primary reference for this campaign. Validate all IOCs against current Kaspersky advisory before deploying to detection systems.

## Indicators of Compromise

| Type | Value | Context | Confidence |
|---|---|---|---|
| DOMAIN | `See Kaspersky FakeWallet/SparkKitty disclosure` | C2 domains and exfiltration endpoints published by Kaspersky; not independently verified in this session — retrieve directly from official Kaspersky press release | MEDIUM |

## Framework Mappings

### MITRE-ATTACK

- **T1476**
- **T1517** — Access Notifications
- **T1581**
- **T1444**
- **T1036.005** — Match Legitimate Resource Name or Location
- **T1627** — Execution Guardrails
- **T1598.003** — Spearphishing Link
- **T1417** — Input Capture
- **T1567** — Exfiltration Over Web Service
- **T1532** — Archive Collected Data

### OWASP-TOP10-2021

- **A08:2021** — Software and Data Integrity Failures

### NIST-800-53R5

- **SI-7** — Software, Firmware, and Information Integrity
- **AT-2** — Literacy Training and Awareness

### CIS-V8

- **2.5** — Allowlist Authorized Software
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks

### HIPAA-SECURITY

- **164.308(a)(5)(i)** — Security Awareness and Training

### ISO-27001-2022

- **A.5.34** — Privacy and protection of personal information
- **A.5.23** — Information security for use of cloud services

## MITRE ATT&CK Mapping

| Technique ID | Technique Name | Tactic |
|---|---|---|
| T1476 | | |
| T1517 | Access Notifications | Collection |
| T1581 | | |
| T1444 | | |
| T1036.005 | Match Legitimate Resource Name or Location | Defense-Evasion |
| T1627 | Execution Guardrails | Defense-Evasion |
| T1598.003 | Spearphishing Link | Reconnaissance |
| T1417 | Input Capture | Collection |
| T1567 | Exfiltration Over Web Service | Exfiltration |
| T1532 | Archive Collected Data | Collection |

## Sources

| Source | URL | Tier |
|---|---|---|
| **Security News** | https://www.bleepingcomputer.com/news/security/chinas-apple-app-sto... | T3 |
| **Kaspersky finds 26 fake crypto wallet apps on Apple's App Store that ...** | https://www.kaspersky.com/about/press-releases/kaspersky-finds-26-f... | T3 |
| **China's Apple App Store infiltrated by crypto-stealing wallet apps** | https://www.instagram.com/p/DXXou9HEvcQ/ | T3 |
| **Apple Removes Fake Crypto Wallet App That Stole $9.5 Million From** | https://www.facebook.com/MacRumors/posts/apple-removes-fake-crypto-... | T3 |
| **Apple Removes Fake Crypto Wallet App That Stole $9.5 ... - Reddit** | https://www.reddit.com/r/apple/comments/1sliil7/apple_removes_fake_... | T3 |

DISCLAIMER

This intelligence report is produced by Tech Jacks Solutions Security Command Center (SCC) for informational purposes only. It does not constitute professional security advice, legal counsel, or an incident response engagement. The information herein is derived from publicly available sources and AI-assisted analysis; while every effort is made to ensure accuracy, Tech Jacks Solutions makes no warranties regarding completeness or timeliness. Organizations should conduct their own validation and consult qualified security professionals before taking action based on this report. Tech Jacks Solutions is not liable for any damages resulting from the use of this information.

Generated 2026-04-21 06:39 UTC by TJS Security Command Center