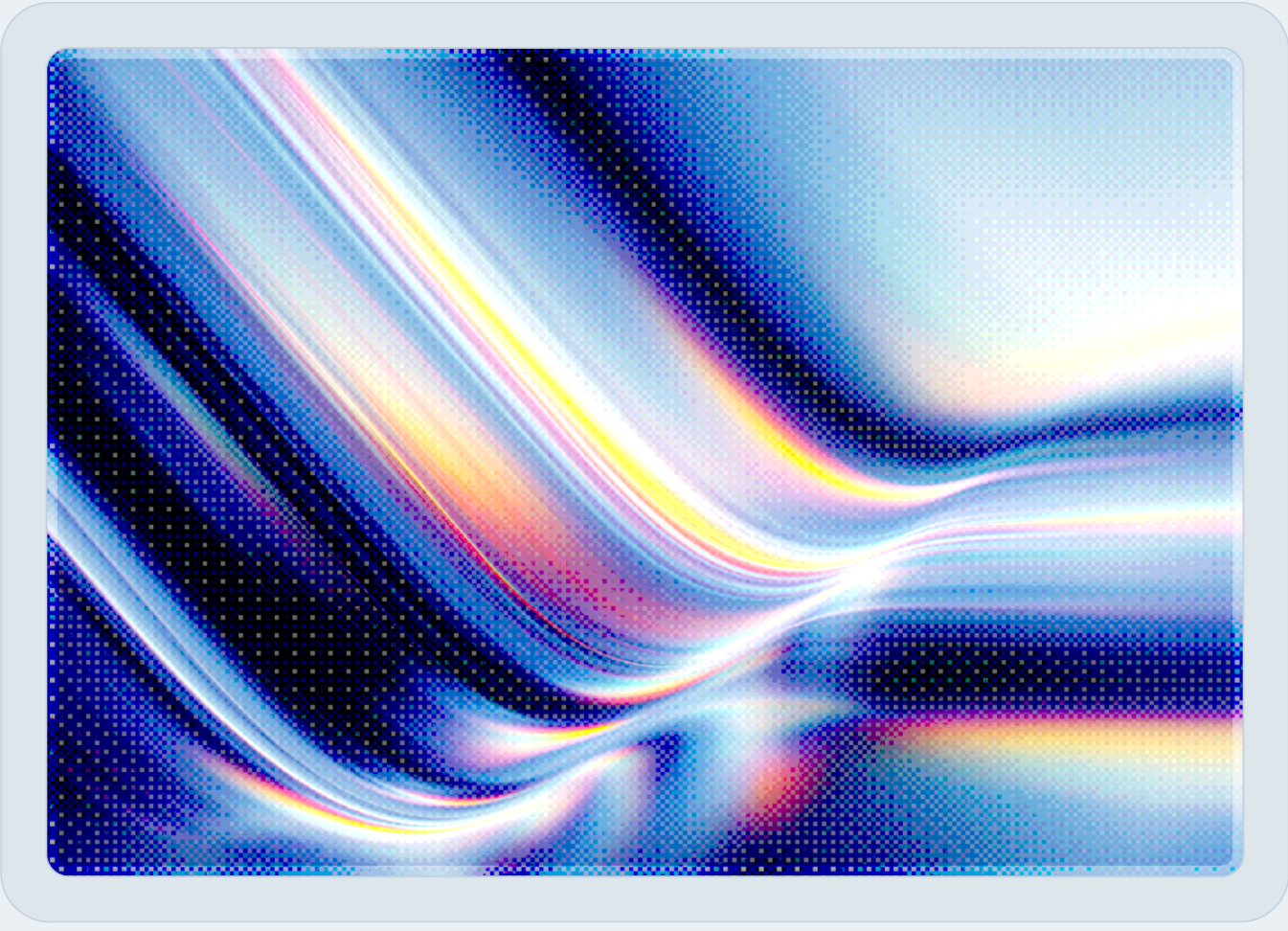
CSAI Foundation | Cloud Security Alliance

# TrickMo Goes Dark: TON Blockchain C2 Evades Enterprise Defense

## TrickMo.C Migrates Command-and-Control to The Open Network, Expanding Android Banking Trojan Capabilities

2026-05-12

AI-assisted Rapid Research



CSAI

CSA cloud security alliance®

**© 2026 Cloud Security Alliance. Some rights reserved.**

You may download, store, display, view, print, redistribute, and link to this document in its original, unmodified form, provided that attribution to the Cloud Security Alliance is maintained and all trademark and copyright notices remain intact.

This document may not be modified or altered. You may quote portions of the document as permitted by the Fair Use provisions of the United States Copyright Act, provided that attribution is given to the Cloud Security Alliance.

This document may be shared on professional and social media platforms in its original form with attribution.

*This document was generated with AI assistance and has not undergone official CSA review and approval processes.*

# Key Takeaways

- **A new TrickMo variant tracked as TrickMo.C** was identified between January and February 2026, migrating its entire command-and-control (C2) transport from conventional internet infrastructure onto The Open Network (TON), a decentralized peer-to-peer overlay originally built for the Telegram messaging platform [1][2].
- **Traditional domain takedowns and IP-based blocking are largely ineffective** against TrickMo.C because the operator's endpoints exist as opaque `.adnl` identities resolved entirely within the TON overlay, never touching the public DNS hierarchy or the public IP routing table [1][3].
- **TrickMo.C retains full device takeover capabilities** against banking, fintech, wallet, and authenticator applications, including WebView credential overlays, keylogging, screen streaming, bidirectional remote control, and silent suppression of one-time passwords (OTPs); active campaigns have been observed targeting users in France, Italy, and Austria [1][4].
- **New network-oriented features transform infected devices into attacker-controlled network relay nodes with active reconnaissance capability**: the updated runtime-loaded APK adds SSH tunneling, SOCKS5 proxying, and five network reconnaissance commands (curl, DNS lookup, ping, telnet, traceroute), giving operators shell-equivalent access to any network the handset is connected to [1][2].
- **Reserved capabilities indicate planned expansion**: the variant carries full NFC permissions and bundles the Pine runtime hooking framework but activates neither; ThreatFabric assesses both as capabilities provisioned for future delivery without requiring an APK update [1].
- **The blockchain C2 pattern is not TrickMo-specific.** Multiple threat actors across ransomware, botnet, and banking-trojan ecosystems are adopting decentralized infrastructure, creating a structural challenge for AI-based behavioral detection and network-layer security controls alike [5][6].

# Background

TrickMo is an Android banking trojan with a documented history stretching back to September 2019, when it was first identified targeting German financial institutions in concert with the TrickBot cybercrime group [7][9]. Its early design centered on overlay attacks — displaying fake HTML login pages atop legitimate banking applications to harvest credentials — combined with SMS interception to defeat SMS-based one-time passwords. The malware became a standalone, independently distributed platform by 2021, at which point it adopted WebView-based overlays as its primary credential-theft vector [8].

Between 2023 and 2024, TrickMo received substantial feature additions that transformed it from a credential harvester into a full device-takeover tool. The 2024 generation of TrickMo exploited Android's Accessibility Services API to obtain a privileged, persistent foothold on infected devices, enabling real-time screen streaming, bidirectional remote control, and the ability to silently dismiss or capture authentication notifications without alerting the device owner [9][10]. Researchers at Zimperium identified exactly 40 distinct TrickMo samples in 2024, while a concurrent Cleafy Labs investigation exposed a misconfigured C2 server leaking 12 GB of exfiltrated victim data — including credentials and personal documents — belonging to more than 13,000 compromised devices [13][8]. By 2024, TrickMo had established itself among the most feature-complete and actively maintained Android banking trojans under researcher observation, based on the breadth of documented capabilities and active campaign activity [9][11].

The 2026 TrickMo.C variant, identified and analyzed by ThreatFabric's Mobile Threat Intelligence team, represents a deliberate architectural re-engineering of the malware's infrastructure layer rather than a wholesale redesign of its on-device capabilities [1]. The targeting profile — banking customers, fintech platform users, and wallet holders in France, Italy, and Austria — is consistent with prior TrickMo campaigns, and the on-device feature set is a direct evolution of the 2024 tooling. What distinguishes TrickMo.C is the decision to abandon conventional internet C2 infrastructure entirely in favor of The Open Network.

The Open Network (TON) was originally developed by the Telegram messaging platform and subsequently transferred to an independent open-source foundation. TON operates as a decentralized, peer-to-peer overlay network with its own routing protocol (ADNL — Abstract Datagram Network Layer), its own naming layer, and its own distributed hash table. Nodes and services within TON are addressed not by IP addresses or DNS hostnames but by opaque base32-encoded strings with a `.adnl` pseudo-TLD, and these identifiers are resolved entirely within the overlay by its participating nodes [1][2]. TON has seen growing adoption as a legitimate payment and messaging layer within

Telegram's ecosystem. In organizations where Telegram is permitted on managed mobile devices, outbound TON traffic may already be present in network logs, complicating efforts to block it wholesale without disrupting legitimate communications [2][3].

# Security Analysis

## Architectural Shift: From DNS to ADNL

The most consequential change in TrickMo.C is not any individual capability but the structural decision to move C2 communications off the public internet. In prior TrickMo generations, operators relied on conventional C2 servers registered under standard domain names. This design made the infrastructure susceptible to takedown operations: threat intelligence teams, domain registrars, and hosting providers could collaborate to seize or sinkhole the C2 domains, severing the operator's communication channel with infected devices. Law enforcement agencies and private-sector threat intelligence groups have used this mechanism to disrupt numerous banking trojan campaigns over the past decade.

TrickMo.C eliminates this attack surface by embedding a native TON proxy directly within the host APK. At process launch, the APK starts this embedded proxy on a loopback port, and the bot's HTTP client is configured to route all C2 traffic through it. Every outbound command-and-control request is addressed to a `.adnl` hostname, which the embedded proxy resolves via the TON overlay network — entirely bypassing the public DNS infrastructure and the public IP routing table [1][2]. The operator's C2 endpoints have no registrar-revocable domain names, no IP addresses that can be blocked through threat-intelligence feeds, and no hosting provider that can be served a takedown notice. Sinkholing the C2 through conventional domain seizure mechanisms is not applicable to this architecture, and no equivalent single-point infrastructure intervention has been demonstrated against TON-routed C2 communications to date [1][2].

From a network visibility standpoint, traffic-pattern detection at the enterprise perimeter sees only encrypted TON overlay traffic. TON communications are encrypted at the protocol layer and are structurally indistinguishable from traffic generated by any legitimate TON-enabled application — including the Telegram client itself, which has broad installation across enterprise mobile fleets [3][4]. AI-based anomaly detection systems trained on behavioral baselines that treat TON traffic as benign are unlikely to flag this C2 communication channel on traffic characteristics alone, without additional behavioral context from the endpoint. Detection requires correlation of behavioral indicators on the device itself, not network traffic inspection.

## Expanded Operator Capabilities: Infected Devices as Network Pivots

Beyond the C2 transport change, TrickMo.C introduces a meaningful expansion of operator capabilities through a runtime-loaded APK module. Five new operator commands expose curl, DNS lookup, ping, telnet, and traceroute primitives that execute from the infected device's network context [1]. The practical implication of this is significant: when a TrickMo-infected employee handset connects to a corporate Wi-Fi network, internal VPN, or home network shared with enterprise-connected systems, the operator gains the ability to perform reconnaissance against those internal networks from within the perimeter — without requiring any additional compromise of network infrastructure.

More consequential still are the additions of SSH tunneling and SOCKS5 proxying capabilities. These features allow operators to route their own traffic through infected devices, using the handset as an exit node. This enables adversaries to make requests that appear to originate from a legitimate corporate endpoint or from a genuine residential IP address, bypassing IP-reputation filtering and geolocation controls. For financial institutions that rely on device fingerprinting and IP reputation as fraud signals, this capability directly undermines those controls [1][2]. The transformation of infected mobile devices into programmable network pivots — attacker-controlled relay nodes through which adversary traffic can be routed and reconnaissance executed — represents a qualitative escalation in the threat model beyond credential theft.

## Reserved Capabilities and the Runtime Delivery Model

ThreatFabric's analysis identified two capabilities present in TrickMo.C's code that are provisioned but not yet active: comprehensive NFC permissions and the Pine runtime hooking framework [1]. NFC-based attacks against banking applications have been observed in other Android banking trojans and could enable relay attacks against tap-to-pay transactions or NFC-enabled authentication tokens. The Pine framework, when activated, would allow the operators to intercept and modify function calls within other applications on the device — including calls to networking libraries and authentication SDKs — without the need to deploy a new version of the host APK.

The significance of bundling inactive capabilities lies in the delivery model. Because these features are included in the host APK but driven by runtime-loaded modules, operators can activate them across the entire infected device population simply by updating the payload delivered through the C2 channel. This design provides adversaries with a scalable update mechanism for capability expansion while minimizing the detection risk associated with publishing a new APK to distribution channels. Security teams assessing the current TrickMo.C risk profile must account not only for its documented active capabilities but for the capability envelope that can be unlocked without any change visible to app store monitoring or static analysis of the installed APK [1].

### The Broader Blockchain C2 Trend

TrickMo.C's adoption of TON infrastructure is not an isolated tactical choice; it reflects a documented and growing pattern among threat actors migrating C2 operations to decentralized networks [5][6]. The Aeternum botnet, disclosed in March 2026, abuses the Polygon blockchain to encode operator commands within on-chain data, making the C2 instructions permanent, globally replicated, and impossible to delete [6]. The DeadLock ransomware group has similarly adopted Polygon smart contract-based communication channels to increase infrastructure resilience [14]. Earlier precedents include the EtherHiding technique, which uses Binance Smart Chain contract storage to host C2 payloads, and a range of threat actors that have used public blockchain transaction memos or IPFS-hosted content as staging for second-stage payloads [5][12].

Each of these implementations exploits a common structural property of blockchain and decentralized networks: the absence of a single authority capable of removing or blocking content, combined with encryption that makes the traffic difficult to distinguish from legitimate use. For AI-based security detection systems, this pattern poses a fundamental challenge. Behavioral models trained primarily on historical data in which C2 traffic is characterized by connections to known-bad IP addresses or newly registered domains are likely to face significant generalization challenges when operator communications are routed through legitimate, widely adopted decentralized infrastructure [5]. Effective detection in this environment requires behavioral indicators from the endpoint — application behavior, permission abuse patterns, and anomalous inter-process activity — rather than reliance on network-layer signals alone.

## Recommendations

### Immediate Actions

Enterprise security teams with responsibility for mobile device fleets should treat TrickMo.C as an active threat requiring immediate defensive posture review. Mobile Device Management (MDM) and Unified Endpoint Management (UEM) platforms should be queried for Android devices that have granted Accessibility Service permissions to applications outside a known-approved list; Accessibility Services access is a mandatory precondition for TrickMo.C's core device-takeover functionality. Any device with unauthorized Accessibility Service grants warrants immediate investigation and isolation pending forensic review.

Network security teams should audit egress filtering rules for TON network traffic. Outbound connections to TON overlay peers on UDP port 51400 and related ports [1], or HTTPS connections to TON API endpoints from mobile network segments, should be flagged for review. While blocking TON traffic wholesale may have collateral impact on legitimate Telegram functionality, anomalous patterns — such as TON traffic originating from devices not running Telegram, or connections that persist outside of normal Telegram usage patterns — warrant investigation. SIEM correlation rules should be created to surface these anomalies.

## Short-Term Mitigations

Organizations should enforce application allowlisting on managed Android devices through MDM policy, restricting installation to applications distributed through managed channels. TrickMo.C is distributed through a dropper application that uses malformed ZIP files and JSONPacker obfuscation to evade static analysis; this delivery mechanism requires sideloading or installation from unmanaged sources [1] [8]. Enforcing Google Play Protect and restricting "Install Unknown Apps" permissions across the managed fleet reduces the attack surface materially.

Mobile threat defense (MTD) solutions should be evaluated for behavioral detection coverage of the specific indicators associated with TrickMo.C: Accessibility Service abuse, runtime APK loading, loopback proxy process creation, and anomalous network socket patterns. Static and signature-based detection is insufficient against a threat that uses obfuscated dropper apps and runtime-loaded payloads; behavioral detection from an on-device agent is the appropriate control layer. MTD vendors should be queried specifically about their detection efficacy against TON-routed C2 traffic.

For financial institutions and fintech platforms, fraud detection systems that rely on IP reputation and device fingerprinting as fraud signals should be reviewed in light of TrickMo.C's SOCKS5 proxying capability. Transactions originating through a SOCKS5 proxy running on an infected device will appear to come from a legitimate, known-clean device IP address. Layering behavioral biometrics, transaction anomaly detection, and out-of-band transaction confirmation — none of which depends on network-origin signals — provides more resilient fraud detection against this threat profile.

## Strategic Considerations

The migration of malware C2 infrastructure to decentralized networks appears to represent a structural shift in the threat landscape rather than a transient tactical adoption, given the breadth of threat actor categories — banking trojans, botnets, and ransomware groups — now employing this technique [5][6] [14]. Security programs that have built detection and response capabilities primarily around network-layer indicators — IP reputation feeds, domain blocklists, DNS monitoring, and traffic volume anomalies —

face a growing blind spot as more threat actors adopt blockchain and peer-to-peer overlay infrastructure. Investment in endpoint behavioral detection, on-device mobile threat defense, and application-layer telemetry should be prioritized over further refinement of network-layer detection against this threat class.

Enterprise mobility programs should treat Accessibility Services permissions on Android as a privileged capability requiring the same governance rigor applied to root access or administrative privileges on other platforms. Applications requesting Accessibility Services should be subject to security review, and any unanticipated Accessibility Services grants — whether from user error, social engineering, or malware-driven coercion — should trigger automatic incident response workflows. This policy stance is warranted independent of TrickMo specifically, given that Accessibility Services abuse is a common pattern across numerous documented Android banking trojan families [9][11].

The reserved NFC and Pine hooking capabilities in TrickMo.C should be treated as a forward-looking risk indicator. If operators choose to activate NFC relay functionality, compromised devices could be used to relay tap-to-pay transactions in real time against point-of-sale terminals, ATMs, and NFC-enabled building access systems. Financial institutions should monitor for activated NFC capabilities in future TrickMo samples and prepare incident response playbooks for NFC relay attack scenarios before such capabilities are deployed, not after.

## CSA Resource Alignment

The TrickMo.C threat intersects with several CSA AI Safety Initiative frameworks and research areas, each providing guidance relevant to the defensive challenges described above.

CSA's MAESTRO threat modeling framework for agentic AI systems is directly applicable to the AI detection evasion dimension of this threat [15]. MAESTRO's layer model addresses the risk of AI-based security systems being circumvented by adversaries who understand the detection models' assumptions and design their infrastructure to remain outside those assumptions. TrickMo.C's routing of C2 traffic through TON — traffic that AI behavioral models may classify as benign — is a concrete instance of this evasion pattern in production malware. Organizations deploying AI-based network anomaly detection should apply MAESTRO's adversarial stress-testing recommendations to their mobile threat detection models, specifically testing whether those models generalize to decentralized and blockchain-based C2 communication patterns.

The CSA AI Controls Matrix (AICM), as a superset of the Cloud Controls Matrix (CCM), provides relevant control guidance in the mobile security and endpoint protection domains [16]. AICM mobile security controls address application permission governance, MDM policy enforcement, and behavioral

monitoring requirements that map directly to the defensive measures recommended above. Organizations should ensure that Accessibility Services permission governance, application allowlisting, and mobile threat defense requirements are explicitly represented in their AICM-aligned control frameworks.

CSA's Zero Trust guidance is particularly relevant to the network pivot capability introduced in TrickMo.C [17]. A mature Zero Trust architecture, which requires continuous verification of device posture and user identity at every access decision rather than relying on network perimeter as a trust boundary, can substantially limit the value of an infected mobile device as a network pivot, provided device health attestation is enforced at the access layer. An infected handset connected to a corporate network that enforces Zero Trust policies — with device health attestation, per-session authentication, and least-privilege microsegmentation — provides the operator with minimal reconnaissance value and limited lateral movement opportunity even if the SOCKS5 proxy is active.

Finally, CSA's AI Organizational Responsibilities framework addresses the governance dimension of AI-assisted fraud detection and mobile threat defense. As TrickMo.C demonstrates, AI detection systems that are not continuously evaluated against adversarial infrastructure innovations will develop blind spots that threat actors can exploit at scale. Organizational AI governance programs should include requirements for adversarial red-teaming of AI-based security detection systems, specifically testing their resilience against decentralized and blockchain-based C2 patterns that were not present in training data.

# References


[1] ThreatFabric. "New TrickMo Variant: Device Take Over Malware Targeting Banking, Fintech, Wallet & Auth Apps." ThreatFabric Mobile Threat Intelligence, 2026.

[2] BleepingComputer. "TrickMo Android Banker Adopts TON Blockchain for Covert Comms." BleepingComputer, May 2026.

[3] Infosecurity Magazine. "TrickMo Variant Routes Android Trojan Traffic Through TON." Infosecurity Magazine, May 2026.

[4] Security Affairs. "Android Banking Trojan TrickMo Evolves Using TON Network for C2." Security Affairs, May 2026.

[5] Endor Labs. "The Unkillable C2: How Attackers Are Moving Command and Control to the Blockchain." Endor Labs, 2025.

[6] Cyberwarzone. "Aeternum C2 Botnet Abuses Polygon Blockchain to Hide Malware Commands and Evade Takedowns." Cyberwarzone, March 2026.

[7] Malpedia / Fraunhofer FKIE. "TrickMo (Malware Family)." Malpedia, updated 2026.

[8] Cleafy Labs. "A New TrickMo Saga: From Banking Trojan to Victim's Data Leak." Cleafy Labs, 2024.

[9] The Hacker News. "TrickMo Android Trojan Exploits Accessibility Services for On-Device Banking Fraud." The Hacker News, September 2024.

[10] The Hacker News. "TrickMo Banking Trojan Can Now Capture Android PINs and Unlock Patterns." The Hacker News, October 2024.

[11] Packetlabs. "Updated TrickMo Banking Trojan Has New Tricks Up Its Sleeves." Packetlabs, 2025.

[12] Elastic. "Potential EtherHiding C2 via Blockchain Connection." Elastic Security Documentation, 2025.

[13] Zimperium. "Expanding the Investigation: Deep Dive into Latest TrickMo Samples." Zimperium, 2024.

[14] The Register. "DeadLock Ransomware Uses Polygon Smart Contracts to Hide C2 Infrastructure." The Register, January 2026.


© 2026 Cloud Security Alliance. AI-assisted Rapid Research. 

[15] Cloud Security Alliance. "MAESTRO: Multi-Agent Environment, Security, Threat, and Risk Overview." CSA AI Safety Initiative, 2024.

[16] Cloud Security Alliance. "AI Controls Matrix (AICM)." Cloud Security Alliance, 2024.

[17] Cloud Security Alliance. "Zero Trust Advancement Center." Cloud Security Alliance, 2024.