INTELLIGENCE BRIEFING
Security Command Center
TLP:CLEAR
2026-06-09 06:28 UTC

NFCShare Android Malware Expands to Italian and Spanish Banking Targets via GitHub-Hosted APKs

THREAT CAMPAIGN | **HIGH** | CVSS 7.5

SCC Item ID	SCC-CAM-2026-0431
Type	Threat Campaign
Severity	HIGH
CVSS Base Score	7.5
Affected Products	Android devices; banking apps impersonated: Intesa Carte, Banca Sella, Nexi, Fideuram, Mooney, CaixaBank (Italy, Spain); previously Deutsche Bank (Germany)
Published	2026-06-08T18:11:58
Discovery Source	Rss

Executive Summary

A new Android malware campaign called NFCShare is actively targeting customers of Italian and Spanish banks, including Intesa Carte, Banca Sella, Nexi, Fideuram, Mooney, and CaixaBank, by distributing 56 malicious app packages through phishing sites that route victims to a GitHub-hosted repository. The malware uses NFC relay techniques to intercept and steal payment card data in real time, with anti-analysis packaging designed to evade automated detection. This campaign represents a confirmed geographic expansion from its original German target (Deutsche Bank) and poses direct fraud and reputational risk to financial institutions and their mobile banking customers across two countries.

Technical Analysis

NFCShare is an Android malware family that uses NFC relay attacks (MITRE T1437.001, T1437) to intercept contactless payment card data from compromised devices. The current campaign began distributing APKs from a GitHub repository created April 10, 2026; 56 unique APKs have been observed since May 14, 2026. Impersonated apps target Intesa Carte, Banca Sella, Nexi, Fideuram, Mooney, and CaixaBank. The infection chain relies on phishing links (T1566.002) that redirect victims to the GitHub-hosted APK (T1583.001), requiring user-initiated installation of malicious files (T1204.002, T1444). Delivery exploits Android's sideloading capability (T1476). The malware uses keylogging (T1056.001) and harvests banking credentials and SMS/OTP data (T1636.003, T1660). Location tracking (T1430) and obfuscation techniques (T1406, T1581) are also present. Malformed APK packaging is used deliberately to disrupt static analysis tooling. No CVE is assigned; applicable

CWEs are CWE-494 (Download of Code Without Integrity Check), CWE-829 (Inclusion of Functionality from Untrusted Control Sphere), and CWE-311 (Missing Encryption of Sensitive Data). The threat actor is unidentified; D3Lab analysis notes it is distinct from NGate and SuperCard X operators, with possible shared ecosystem, unconfirmed. No patch is available; mitigation is behavioral and architectural. Source quality score: 0.64 (Tier 3 sources only); no primary research vendor advisory confirmed at time of writing. Note: CVSS not applicable to campaign/threat activity item type. Severity rating set editorially based on attack scope and target criticality.

Action Checklist

1. Step 1: Containment. Block the GitHub repository identified as the APK distribution source at the DNS and proxy layer across all managed mobile device management (MDM) and enterprise network egress points. If your organization manages corporate Android devices, push an MDM policy preventing installation from unknown sources immediately.
2. Step 2: Detection. Query mobile threat defense (MTD) logs and MDM enrollment records for Android devices where 'Install from unknown sources' is enabled or where apps outside approved app stores are present. Review network proxy logs for outbound connections to raw GitHub repository URLs (raw.githubusercontent.com) from mobile devices. Flag any APK with a package name mimicking Intesa Carte, Banca Sella, Nexi, Fideuram, Mooney, or CaixaBank that was not distributed via Google Play. Monitor for NFC relay activity patterns: unusual NFC API calls, background NFC reader activation, or data exfiltration events coinciding with contactless payment attempts (MITRE T1437.001). Reference CIS 8.2 for audit log collection requirements across mobile endpoints.
3. Step 3: Eradication. Remove any identified NFCShare APK from affected devices via MDM remote wipe or targeted app removal. Revoke any banking sessions or credentials entered on a suspected compromised device. Rotate payment card credentials for affected users through their issuing bank. Apply CIS 5.3 (disable dormant accounts) and CIS 6.2 (access revoking process) for any corporate accounts accessed from a suspected device. Reference D3-CRO (Credential Rotation) and D3-UAP (User Account Permissions) as countermeasures.
4. Step 4: Recovery. Verify that affected devices show no residual NFC relay services running in the background. Re-enroll devices in MDM with a clean baseline image where compromise is confirmed. Monitor post-remediation for repeat phishing attempts targeting the same users (T1566.002 reuse). Validate that MDM policies enforce app allowlisting and block sideloading going forward. Reference NIST AU-6 for ongoing audit record review requirements post-incident.
5. Step 5: Post-Incident. Conduct a gap assessment against CIS 6.3 (MFA for externally exposed applications) and CIS 6.4 (MFA for remote network access). NFC relay attacks are significantly harder to execute when transaction authentication requires an out-of-band second factor not accessible to the malware. Review mobile security awareness training for banking app users, specifically covering risks of installing apps from links in messages or emails. Evaluate whether your mobile threat defense tooling handles malformed APK analysis, as this campaign deliberately exploits that gap. Reference D3-MFA (Multi-factor Authentication) as a structural countermeasure.

IR / Forensic Enrichment

Triage Priority	IMMEDIATE

Escalation Criteria	Escalate to executive leadership, legal counsel, and relevant banking regulators (Banca d'Italia, CNMV, or BaFin depending on jurisdiction) immediately if any confirmed NFCShare-compromised device was used for a contactless payment transaction while the malware was active, as this constitutes a realized payment card data breach with PSD2 strong customer authentication violation and potential GDPR Article 33 notification obligations within 72 hours.
Recovery Notes	After MDM re-enrollment, monitor the re-provisioned devices for a minimum of 30 days for anomalous NFC API activity, unexpected outbound connections to GitHub raw content URLs, or new app installations outside the approved store — NFCShare campaigns have demonstrated persistence through retargeting previously-compromised users with fresh phishing lures. Coordinate directly with the fraud departments of the six impersonated banks (Intesa Carte, Banca Sella, Nexi, Fideuram, Mooney, CaixaBank) to place alerts on accounts belonging to affected users, as NFC-relayed card data can be used for fraudulent contactless transactions within minutes of capture. Validate that all corporate accounts accessed from compromised devices have had sessions revoked and passwords rotated before returning devices to service, as the malware's NFC relay focus does not preclude credential harvesting as a secondary capability.
Forensic Artifacts	Malicious APK binaries (SHA-256 hashed) with package names mimicking Intesa Carte, Banca Sella, Nexi, Fideuram, Mooney, or CaixaBank — decompile with apktool to extract AndroidManifest.xml confirming NFC relay permissions (android.permission.NFC, BIND_NFC_SERVICE) and anti-analysis packaging signatures specific to this campaign | Android proxy or firewall logs showing outbound HTTP GET requests to raw.githubusercontent.com with URI paths matching the NFCShare distribution repository — include source device IP, timestamp, and full URI path to reconstruct the APK download event | ADB output of 'adb shell dumpsys nfc' from affected devices, capturing registered NFC dispatch targets, active foreground dispatch settings, and any NDEF push callbacks that would confirm NFCShare's background NFC reader was active during contactless payment events | MDM enrollment compliance reports showing the device state at time of detection — specifically 'Install from unknown sources' policy violation status, unapproved app inventory entries, and enrollment gap timestamps that correspond to the window when the phishing lure was received | SMS, email, or messaging app notification history from affected users showing the phishing lure message that contained the GitHub APK distribution link — this establishes the T1566.002 delivery chain and the user population exposed to the campaign

Per-Action IR Details

Step 1: Containment — Block the GitHub repository identified as the APK distribution source at the DNS and proxy layer across all managed mobile device management (MDM) and enterprise network egress points. If your organization manages corporate Android devices, push an MDM policy preventing installation from unknown sources immediately.

NIST Phase: Containment

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

Controls: NIST AC-4 (Information Flow Enforcement), NIST CM-7 — not in knowledge base; omitted, CIS 4.4 (Implement and Manage a Firewall on Servers), CIS 4.5 (Implement and Manage a Firewall on End-User Devices)

Compensating: For teams without enterprise MDM or proxy: use Pi-hole or your DNS resolver's blocklist to add raw.githubusercontent.com and the specific repository subdomain as NXDOMAIN entries immediately. On Android devices without MDM, manually disable 'Install unknown apps' under Settings > Apps > Special app access > Install unknown apps for all browsers and file managers. Document the specific GitHub repo URL blocked and timestamp for your incident log.

Evidence: Before blocking, capture the full repository URL (including username, repo name, and any release or raw path) from DNS query logs or proxy access logs. Preserve any MDM enrollment records showing which devices have 'Install from unknown sources' currently enabled — this establishes your blast radius. Export proxy logs showing historical connections to raw.githubusercontent.com from mobile device IP ranges prior to block enforcement.

Step 2: Detection — Query mobile threat defense (MTD) logs and MDM enrollment records for Android devices where 'Install from unknown sources' is enabled or where apps outside approved app stores are present. Review network proxy logs for outbound connections to raw GitHub repository URLs (raw.githubusercontent.com) from mobile devices. Flag any APK with a package name mimicking Intesa Carte, Banca Sella, Nexi, Fideuram, Mooney, or CaixaBank that was not distributed via Google Play. Monitor for NFC relay activity patterns: unusual NFC API calls, background NFC reader activation, or data exfiltration events coinciding with contactless payment attempts (MITRE T1437.001). Reference CIS 8.2 for audit log collection requirements across mobile endpoints.

NIST Phase: Detection Analysis

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

Controls: NIST AU-2 (Event Logging), NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-12 (Audit Record Generation), CIS 8.2 (Collect Audit Logs)

Compensating: Without MTD, use Android Debug Bridge (ADB) to enumerate installed packages on enrolled devices: run 'adb shell pm list packages -f' and diff the output against your approved app inventory — flag any package name containing strings like 'intesa', 'sella', 'nexi', 'fideuram', 'mooney', or 'caixabank' not sourced from com.google.android or a verified publisher. Use apktool to decompile flagged APKs and inspect AndroidManifest.xml for NFC permissions (android.permission.NFC, BIND_NFC_SERVICE) combined with declared receivers for ACTION_TECH_DISCOVERED or ACTION_TAG_DISCOVERED — these are required for NFC relay functionality. Cross-reference proxy logs using grep on exported access logs: 'grep -i raw.githubusercontent.com proxy_access.log | grep -i mobile_subnet' to identify download events.

Evidence: Capture APK file hashes (SHA-256) of any sideloaded app matching the six impersonated banking app package names before attempting removal. Preserve AndroidManifest.xml from decompiled APKs documenting requested NFC, INTERNET, and FOREGROUND_SERVICE permissions. Export proxy or firewall logs showing the full URI path of any APK downloaded from raw.githubusercontent.com, including timestamp and source device IP. If the device is rooted or has developer options enabled, capture a full package list with install timestamps ('adb shell pm list packages -f -i') to establish when the malicious APK was installed relative to known phishing delivery windows.

Step 3: Eradication — Remove any identified NFCShare APK from affected devices via MDM remote wipe or targeted app removal. Revoke any banking sessions or credentials entered on a suspected compromised device. Rotate payment card credentials for affected users through their issuing bank. Apply CIS 5.3 (disable dormant accounts) and CIS 6.2 (access revoking process) for any corporate accounts accessed from a suspected device. Reference D3-CRO (Credential Rotation) and D3-UAP (User Account Permissions) as countermeasures.

NIST Phase: Eradication

Reference: NIST 800-61r3 §3.4 — Eradication

Controls: NIST AC-2 (Account Management), NIST AC-6 (Least Privilege), CIS 5.3 (Disable Dormant Accounts), CIS 6.2 (Establish an Access Revoking Process)

Compensating: Without enterprise MDM remote wipe, instruct device owners to manually uninstall the malicious APK via Settings > Apps, then immediately clear storage and cache for any legitimate banking apps on the same device to remove any session tokens the NFCShare relay may have observed. Use ADB for corporate-owned devices: 'adb shell pm uninstall --user 0 '. For payment card exposure, escalate to the affected bank's fraud line immediately — NFC relay attacks mean card data may have been captured during a live contactless transaction, not just at login; card reissuance (not just PIN change) is required.

Evidence: Before removal, use ADB to pull the malicious APK for offline analysis: 'adb shell pm path ' then 'adb pull ' — preserve the binary for YARA scanning and hash comparison against known NFCShare samples. Document the

device's NFC transaction log if accessible under Settings > Connected devices > NFC, noting any relay events. Capture a screenshot or export of the running services list ('adb shell dumpsys activity services | grep -i nfc') to confirm whether NFCShare's background NFC reader service was active at time of eradication. Preserve any SMS or notification history showing phishing lure messages that directed the user to the GitHub-hosted APK download link.

Step 4: Recovery — Verify that affected devices show no residual NFC relay services running in the background. Re-enroll devices in MDM with a clean baseline image where compromise is confirmed. Monitor post-remediation for repeat phishing attempts targeting the same users (T1566.002 reuse). Validate that MDM policies enforce app allowlisting and block sideloading going forward. Reference NIST AU-6 for ongoing audit record review requirements post-incident.

NIST Phase: Recovery

Reference: NIST 800-61r3 §3.5 — Recovery

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AC-17 (Remote Access), CIS 2.2 (Ensure Authorized Software is Currently Supported), CIS 4.6 (Securely Manage Enterprise Assets and Software)

Compensating: For teams without MDM factory-reset capability, perform a manual Android factory reset (Settings > General Management > Reset > Factory data reset) on confirmed-compromised devices before re-provisioning. After re-enrollment, validate clean state with ADB: 'adb shell pm list packages' should return only system and approved apps with no sideloaded packages. Set up a weekly cron job or scheduled task to export MDM enrollment compliance reports and diff against the approved package baseline — any new entry outside Google Play should trigger an alert. Monitor the previously-targeted users' corporate email and SMS gateways for follow-on T1566.002 spearphishing attempts, as NFCShare campaigns have historically retargeted previously-phished individuals.

Evidence: Before re-enrollment, run 'adb shell dumpsys nfc' to capture full NFC subsystem state, confirming no registered dispatch targets or NDEF push callbacks remain from the NFCShare application. Preserve the MDM compliance report showing the device's pre-remediation policy violation state (unknown sources enabled, unapproved app present) as evidence for post-incident review. Document the re-enrollment timestamp and the Android security patch level of the clean baseline image to establish a verified clean recovery baseline.

Step 5: Post-Incident — Conduct a gap assessment against CIS 6.3 (MFA for externally exposed applications) and CIS 6.4 (MFA for remote network access) — NFC relay attacks are significantly harder to execute when transaction authentication requires an out-of-band second factor not accessible to the malware. Review mobile security awareness training for banking app users, specifically covering risks of installing apps from links in messages or emails. Evaluate whether your mobile threat defense tooling handles malformed APK analysis, as this campaign deliberately exploits that gap. Reference D3-MFA (Multi-factor Authentication) as a structural countermeasure.

NIST Phase: Post Incident

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

Controls: NIST AU-6 (Audit Record Review, Analysis, And Reporting), NIST AU-11 (Audit Record Retention), CIS 6.3 (Require MFA for Externally-Exposed Applications), CIS 6.4 (Require MFA for Remote Network Access), CIS 7.1 (Establish and Maintain a Vulnerability Management Process), CIS 7.2 (Establish and Maintain a Remediation Process)

Compensating: Without commercial MTD, write a YARA rule targeting NFCShare-specific AndroidManifest.xml patterns — specifically the combination of NFC dispatch permissions with non-Play distribution signatures — and schedule it to run against any APK that reaches your network or MDM pipeline. Publish an internal one-page advisory to banking app users illustrating the specific NFCShare phishing flow: SMS or email link → GitHub raw URL → APK download prompt → fake Intesa/CaixaBank branding. Use free awareness content from SANS Security Awareness or CISA's phishing guidance rather than building from scratch. Retain all incident logs, MDM compliance exports, and captured APK binaries for a minimum of one year per NIST AU-11 (Audit Record Retention) to support any regulatory inquiry, particularly given PSD2 and DORA obligations for EU banking operations.

Evidence: Consolidate all evidence gathered during the incident into a post-incident package: SHA-256 hashes of all NFCShare APK variants encountered, proxy log extracts showing the GitHub repository download events, ADB

package dumps from affected devices, and card issuer notification records. Document the specific phishing lure format (SMS, email, or messaging app) used to deliver the GitHub link to each affected user — this establishes the T1566.002 delivery vector for lessons-learned reporting and future detection rule tuning.

Detection Guidance

Primary behavioral indicators: (1) Android devices with NFC API calls originating from apps not distributed via Google Play, particularly apps with package names mimicking Intesa Carte, Banca Sella, Nexi, Fideuram, Mooney, or CaixaBank. (2) Network connections from mobile devices to raw.githubusercontent.com for APK file downloads, review proxy and firewall logs for this pattern. (3) SMS interception or OTP harvest behavior: look for apps requesting READ_SMS and NFC permissions simultaneously (T1636.003, T1437.001). (4) Keylogging activity: background foreground service abuse on Android with accessibility service permissions (T1056.001). IOC patterns: APK hashes from the identified GitHub repository (56 unique samples as of May 14, 2026, obtain current hash list from D3Lab or threat intelligence feeds, as no verified hash list was available in source data at this writing). Repository URL pattern: GitHub repository created April 10, 2026, hosting banking-themed APKs, monitor threat intel feeds for the specific repository identifier. SIEM query guidance: alert on mobile device user-agent strings downloading .apk files from github.com or raw.githubusercontent.com; correlate with device enrollment status and app inventory. Reference NIST AU-2 (Event Logging) and AU-6 (Audit Record Review) for log collection and analysis requirements. Note: source data is Tier 3 only; validate IOC specifics against primary threat intelligence sources before operationalizing detections.

Indicators of Compromise

Type	Value	Context	Confidence
URL	`https://github.com/[repository created 2026-04-10 — specific repository path not confirmed in available sources]`	GitHub repository used to host 56 malicious APKs since May 14, 2026; specific repository identifier not available in Tier 3 source data — obtain from D3Lab or primary threat intel feed before operationalizing	MEDIUM
HASH	`[56 APK hashes — not available in source data]`	56 unique APK samples distributed from the GitHub repository since May 14, 2026; hashes not present in available Tier 3 sources — request from D3Lab or a threat intelligence platform with mobile malware coverage	LOW

Framework Mappings

MITRE-ATTACK

- **T1437.001** — Web Protocols
- **T1583.001** — Domains
- **T1406** — Obfuscated Files or Information
- **T1581**

- **T1204.002** — Malicious File
- **T1444**
- **T1566.002** — Spearphishing Link
- **T1476**
- **T1437** — Application Layer Protocol
- **T1660** — Phishing
- **T1430** — Location Tracking
- **T1056.001** — Keylogging
- **T1636.003** — Contact List

NIST-800-53R5

- **AT-2** — Literacy Training and Awareness
- **SC-7** — Boundary Protection
- **SI-3** — Malicious Code Protection
- **SI-4** — System Monitoring
- **SI-8** — Spam Protection
- **SI-7** — Software, Firmware, and Information Integrity
- **CM-3** — Configuration Change Control
- **SC-13** — Cryptographic Protection

OWASP-TOP10-2021

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

CIS-V8

- **2.5** — Allowlist Authorized Software
- **2.6** — Allowlist Authorized Libraries
- **14.2** — Train Workforce Members to Recognize Social Engineering Attacks
- **8.2** — Collect Audit Logs

HIPAA-SECURITY

- **164.308(a)(5)(i)** — Security Awareness and Training
- **164.312(e)(1)** — Transmission Security

ISO-27001-2022

- **A.5.34** — Privacy and protection of personal information
- **A.8.24** — Use of cryptography

NIST-CSF-2

- **DE.CM-01** — Networks and network services are monitored

MITRE ATT&CK Mapping

Technique ID	Technique Name	Tactic
T1437.001	Web Protocols	Command-And-Control
T1583.001	Domains	Resource-Development
T1406	Obfuscated Files or Information	Defense-Evasion
T1581		
T1204.002	Malicious File	Execution
T1444		
T1566.002	Spearphishing Link	Initial-Access
T1476		
T1437	Application Layer Protocol	Command-And-Control
T1660	Phishing	Initial-Access
T1430	Location Tracking	Collection
T1056.001	Keylogging	Collection
T1636.003	Contact List	Collection

Sources

Source	URL	Tier
Security News	https://www.bleepingcomputer.com/news/security/nfcshare-android-mal...	T3
Security vulnerabilities are common in bank mobile apps	https://www.prosightfa.org/insights/security-vulnerabilities-are-co...	T3
App data and processing methods - Intesa Sanpaolo Private Banking	https://www.fideuramintesasanpaoloprivatebanking.com/en/regulations...	T3
Fake Android Banking Apps Steal Credentials — Global | Pinkerton	https://pinkerton.com/our-insights/blog/fake-android-banking-apps-s...	T3
The Silent Alarm on Mobile Banking Apps Just Went Off	https://thefinancialbrand.com/news/mobile-banking-trends/the-silent...	T3

Generated 2026-06-09 06:28 UTC by TJS Security Command Center