

The Rise of Arsink Rat

 zimperium.com/blog/the-rise-of-arsink-rat

Rajat Goyal

January 29, 2026

Executive Summary

Arsink is a cloud-native Android Remote Access Trojan (RAT) that aggressively harvests private data and gives remote operators intrusive control over infected devices. We observed multiple variants that use **Google Apps Script** to upload larger files and media to **Google Drive**, or **Firebase Realtime Database + Firebase Storage & Telegram** for C2 and exfiltration.

The operation's significant scale is evidenced by the **1,216 distinct APK hashes** identified across the observation period (Figure 1). Notably, **774 of these samples** incorporate Google Apps Script or "macro" upload mechanisms, pointing to the extensive use of Google services for media and file exfiltration. The operation leverages **317 distinct Firebase Realtime Database endpoints** as C2/data sinks, and our infrastructure enumeration extracted **45,000 unique victim IPs**, demonstrating both scale and breadth of exposure.

Samples Distribution Timeline

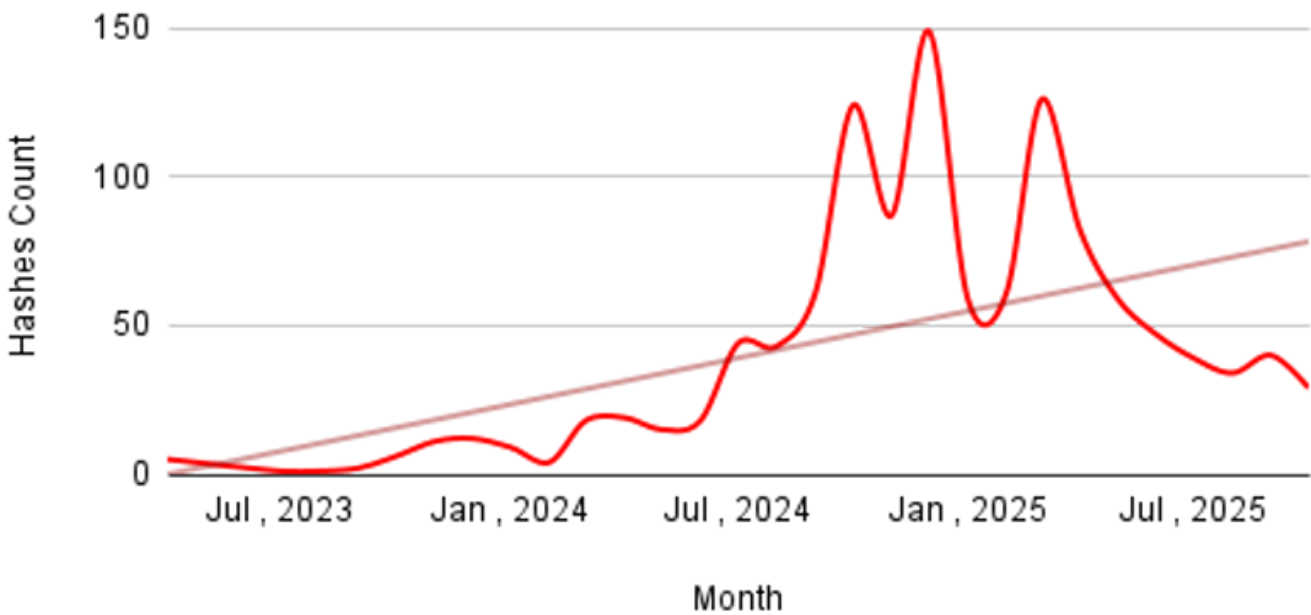


Figure 1. Showing samples found over the period of time

Distribution is broad and social-engineered. The malware is pushed via **Telegram, Discord, and MediaFire links**, as well as similar channels, while impersonating dozens of popular brands: Figure 2 shows some of the brands that were impersonated in this campaign.



Figure 2. Brands that were impersonated in this campaign

The **zLabs team has been tracking this campaign over the past few months**, observing a consistent increase in sample distribution and C2 infrastructure reuse across multiple clusters.

Distribution in the Wild: Arsink in Action

Arsink samples are distributed through a wide range of social-engineered channels rather than a single vector. In our collection, we consistently observed malicious APKs delivered via **Telegram channels, Discord posts and direct MediaFire-hosted APK links**.

Key traits of the campaign are:

- **Social engineering & impersonation:** Operators routinely impersonate well-known consumer platforms and apps. The malicious campaigns involved impersonating over 50 well-known brands, such as Google, YouTube, WhatsApp, Instagram, Facebook, and TikTok, to trick users into downloading the malicious software and granting it extensive permissions.
- **Deceptive filenames & lures:** Messages and links push users to download APKs; file names are intentionally deceptive (presented as “mod” or “pro” versions of legitimate apps) so victims believe they are getting enhanced or premium functionality.
- **Scale of artefacts:** From our dataset: **1,216 unique APK hashes** and **317 unique Firebase RTDB endpoints**, plus 45,000 unique victim IPs, evidence of many distinct builds and multiple C2 instances operated in parallel.

To better reflect the diversity in the wild, we identified four operational variants that were found in this campaign:

- **Firestore + Apps Script / Drive variant**
Structured telemetry and small data are written to the **Firestore Realtime Database (RTDB)**. Audio is written to **Firestore Storage**. **Google Apps Script** is used to collect big base64 encoded files and media.
- **Telegram exfiltration variant**
Rather than Firestore, this variant pushes stolen data directly to an attacker-controlled **Telegram bot** (via the Bot API). SMS, device info, and other harvested data are sent as Telegram messages. This is a simpler, low-latency pipeline suitable for quick exfiltration.

Embedded-payload dropper variant

The installed APK contains a secondary malicious payload inside its **assets**. The app extracts that embedded archive to external storage and repackages it as an **.apk** (renaming **Ai_App.zip** → **Ai_App.apk** → **App.apk**)

This method avoids downloading the second stage over the network, making detection and takedown harder and enabling the delivery of a RAT offline.

In most cases, the apps don’t deliver real features, they display a minimal UI, immediately request sensitive permissions, and then operate silently.

Infrastructure Disruption and Takedown Efforts

In parallel with tracking and detection, zLabs worked directly with Google to report and disrupt Arsink’s abuse of legitimate cloud services. As part of this coordinated effort, multiple malicious Firestore Realtime Database endpoints and attacker-controlled Google Apps Script instances used for command-and-control and data exfiltration were identified and dismantled. On top of this, Google disabled malicious Apps Scripts accounts and confirmed that known versions of Arsink Rat malware do not exist on Google Play and that Android users are automatically protected by

Google Play Protect, which can warn users or block apps known to exhibit malicious behavior, even when those apps come from sources outside of Play. These actions significantly degraded the campaign's operational capacity and forced threat actors to retool their infrastructure.

While takedowns help reduce active abuse, Arsink's rapid variant churn and shifting exfiltration paths underscore why detection and prevention at the device level remain critical — attackers can reconstitute infrastructure faster than centralized blocklists or takedown processes can fully contain.

They Are Watching You: What it Exfiltrates & Capabilities

Arsink is a sophisticated **malware** that actively exfiltrates information, transmitting it directly to its operators while affording them complete remote control. The analysis of the samples revealed this stealthy tool is highly versatile, possessing multiple methods to exfiltrate compromised data:

Device snapshot & identifiers

- Public IP: used by the server to index the user.
- Device model, OS, battery state, and coarse reverse-geocoded location text.
- Google account email addresses discovered on the device.

• Communications & identity

- **SMS** harvesting and continuous exfiltration of newly received messages.
- **Call logs** (numbers, types, timestamps, durations).
- **Contacts** list (names and phone numbers).

• Audio & media exfiltration

- **Microphone recordings** and metadata get exfiltrated to Firebase Storage
- **Photos and arbitrary files** are enumerated on external storage, and larger files in some samples are uploaded via **Google Apps Script**, then to **Google Drive**.

• Remote device control

- Flashlight toggle, vibration, play bundled audio, set wallpaper, display operator messages (toasts), and text-to-speech output.
- Ability for a **phone call initiation** (CALL_PHONE).
- Remote file operations: list directories, create directories, upload arbitrary file content, and delete files.
- **Destructive command**: an operator-triggered wipe of the external storage root.

• Stealth & persistence

- Hides its launcher icon to reduce detection by casual users.
- It launches a foreground service that keeps running despite task killers and displays a **persistent notification** to prevent the service from being terminated.
- Uses timers and frequent RTDB listeners/polling to maintain active connectivity with operator C2.

Region-Wide Spread: Nobody is Left

The Arsink operation has a truly global footprint, as it is not confined to any specific geographic area. From the analysis of victim telemetry and publicly accessible C2 dumps, we identified **≈45,000 unique infected IP addresses spanning some 143 countries** across the Middle East, Asia, Africa, Europe, and the Americas.

The largest concentrations were observed in **Egypt (≈13,000 devices)**, **Indonesia (≈7,000)**, **Iraq (≈3,000)**, **Yemen (≈3,000)** and **Türkiye (≈2,000)**. Notable clusters also appear in **Pakistan (≈2,500)**, **India (≈2,500)**, **Bangladesh (≈1,600)**, and North African countries such as **Algeria** and **Morocco** (each ≈1,000), regions where third-party APK distribution and Telegram sharing are common. Figure 3 shows a world map visualising infection density by country and highlights the campaign's multi-regional scale.

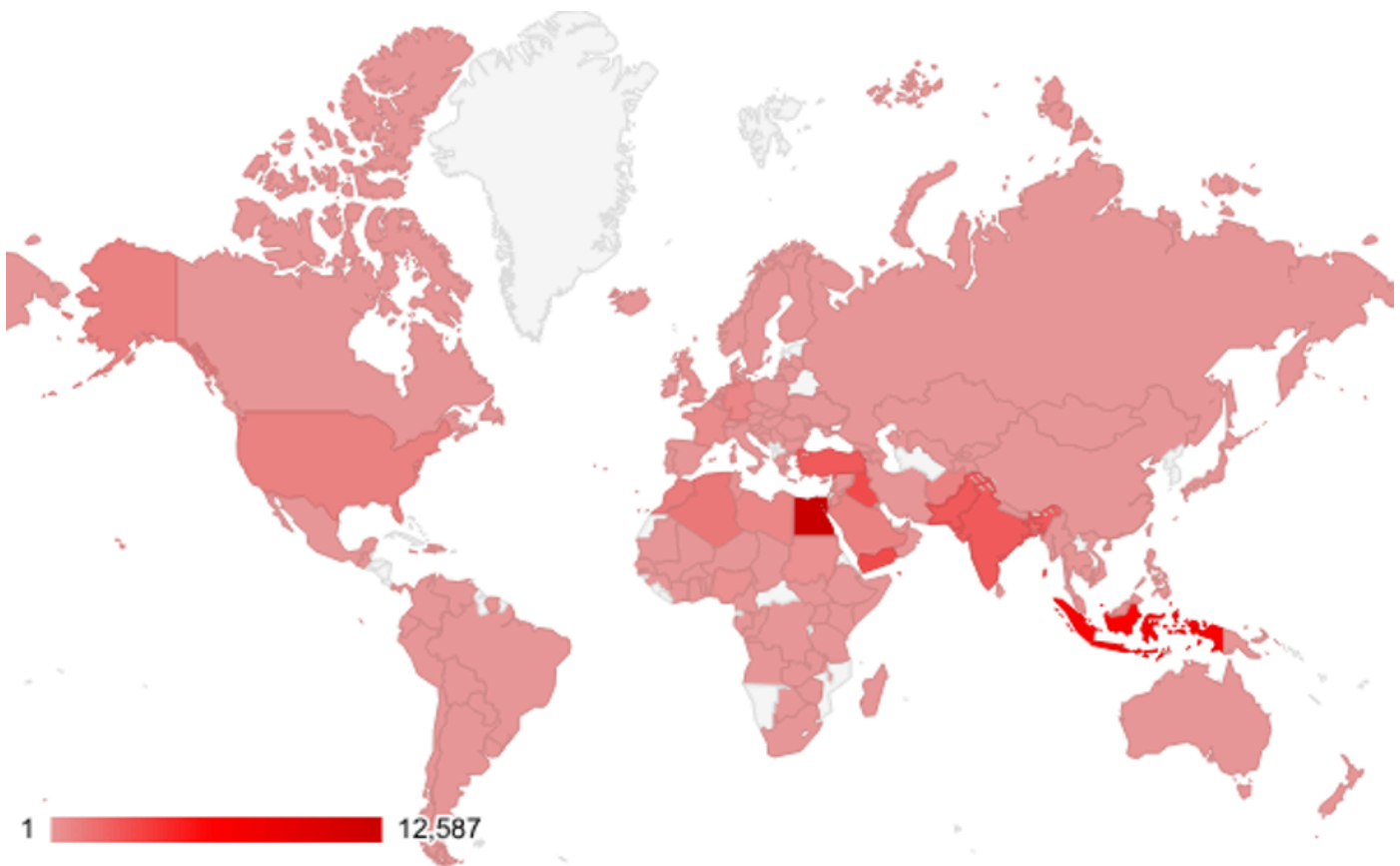


Figure 3. Geographic distribution of victim IPs extracted from misconfigured C2 databases.

This geographic diversity reinforces that **Arsink is an opportunistic, mass-distribution threat** rather than a regionally targeted campaign, leveraging brand impersonation and social platforms to achieve worldwide penetration.

Conclusion

The Arsink campaign represents one of the most persistent and evolving Android surveillance operations observed in recent months. What began as a single-family threat abusing Firebase as a command-and-control channel has expanded into a large, modular ecosystem of variants, each using different exfiltration paths such as Firebase Storage, Google Apps Script and Drive, Telegram bots, and embedded payload droppers.

From a user’s perspective, these apps appear harmless, most offer no real functionality beyond intrusive permission prompts, yet behind the scenes, they perform continuous exfiltration of messages, contacts, call logs, location data, and media content, while allowing operators to issue remote commands and even wipe files.

As the zLabs team continues to track this operation, we expect further retooling and diversification of its infrastructure. The ongoing campaign serves as a reminder that modern mobile malware no longer relies solely on dedicated servers or overt phishing sites, it thrives within legitimate ecosystems, exploiting user trust in familiar brands and widely used cloud services.

Zimperium vs. Arsink RAT

Despite Arsink’s heavy reliance on legitimate cloud platforms and its rapid variant churn, Zimperium’s [Mobile Threat Defense](#) (MTD) and [Mobile Runtime Protection](#) (zDefend) provide effective protection against this threat through on-device, behavior-based detection. Across the Arsink RAT variants analyzed, Zimperium detects malicious activity in a zero-day fashion, without relying on static signatures or pre-existing indicators of compromise.

For enterprises, Arsink represents more than a consumer spyware threat—it is a direct risk to corporate data, operations, and brand trust. Devices compromised by Arsink can silently exfiltrate SMS messages (including OTPs), contacts, call logs, audio, and files, enabling account takeover, fraud, and lateral abuse of enterprise services. The malware’s use of trusted cloud platforms further complicates traditional network-based defenses, allowing it to blend into normal traffic. In organizations where mobile devices access corporate email, authentication flows, messaging platforms, or customer data, an infection like Arsink can quickly escalate into credential theft, unauthorized access, regulatory exposure, and reputational damage—making on-device, behavior-based protection a critical control for modern enterprises.

MITRE ATT&CK Techniques

Tactic	ID	Name	Description
Initial Access	T1476	Deliver Malicious App via Other Means	Distribution of malicious APKs outside official stores (direct links, DMs, file hosts, sideloading).

Initial Access	T1660	Phishing	Use of messages or links on Telegram, WhatsApp, Discord, and MediaFire to lure users into installing or sideloading APKs.
Discovery	T1426	System Information Discovery	Collects OS, build, model, version, serial, and other device identifiers used to profile victims.
Discovery	T1422	System Network Configuration Discovery	Gathers network-related information such as IMSI/IMEI, interfaces, and public IP.
Collection	T1533	Data from Local System	Enumerates files and media on external storage and local data stores for exfiltration.
Collection	T1636.004	Protected User Data: SMS Messages	Reads and continuously exfiltrates SMS messages, including OTPs.
Collection	T1636.002	Protected User Data: Call Log	Harvests call history including numbers, timestamps, durations, and call types.
Collection	T1636.003	Protected User Data: Contact List	Reads the device contact list for bulk exfiltration.
Collection	T1429	Audio Capture	Records microphone audio and stages or uploads recordings to cloud storage.
Execution / Persistence	T1541	Foreground Persistence	Starts a foreground service with a sticky notification to remain active and retain sensor access.
Defense Evasion	T1628.001	Hides Artifacts: Suppress Application Icon	Hides the launcher icon to avoid casual detection or uninstallation.
Command and Control	T1437	Application Layer Protocol (Web / HTTPS)	Uses cloud APIs and services such as Firebase, Apps Script, Google Drive, and Telegram Bot API for C2.
Exfiltration	T1646	Exfiltration over C2 Channel	Sends stolen data over the same channels used for command and control.
Impact	T1630.002	Delete Device Data / Data Destruction	Operator-triggered destructive wipe of external storage and device data.

Indicators of Compromise

The IOCs for this campaign can be found in [this repository](#).

[zLabs](#)

[Threat Intelligence](#)