Hook Version 3: The Banking Trojan with The Most Advanced Capabilities

By Vishnu Pratapagiri

Published: 2025-08-25 · Archived: 2026-04-05 16:58:09 UTC

Executive Summary

Zimperium's zLabs research team has uncovered a new variant of the **Hook Android banking trojan**, now featuring some of the most advanced capabilities we've seen to date. This version introduces:

- **Ransomware-style overlays** that display extortion messages
- **Fake NFC overlays** to trick victims into sharing sensitive data
- **Lockscreen bypass** via deceptive PIN and pattern prompts
- **Transparent overlays** to silently capture user gestures
- **Stealthy screen-streaming sessions** for real-time monitoring

In total, the malware now supports **107 remote commands** — with **38 newly added** in this update.

There is growing evidence that the malware is being distributed on a large scale, not only through phishing websites but also via GitHub, where threat actors are actively leveraging the platform to host and spread malicious APK files.

Distribution Methods

We have been actively monitoring multiple GitHub repositories and have observed both old and new variants of malware such as Hook and Ermac being hosted (**Figure 1**). It is also evident that this method of distribution is not limited to these families alone, other malware strains like Brokewell and various SMS spyware trojans are also being disseminated through the same channels.

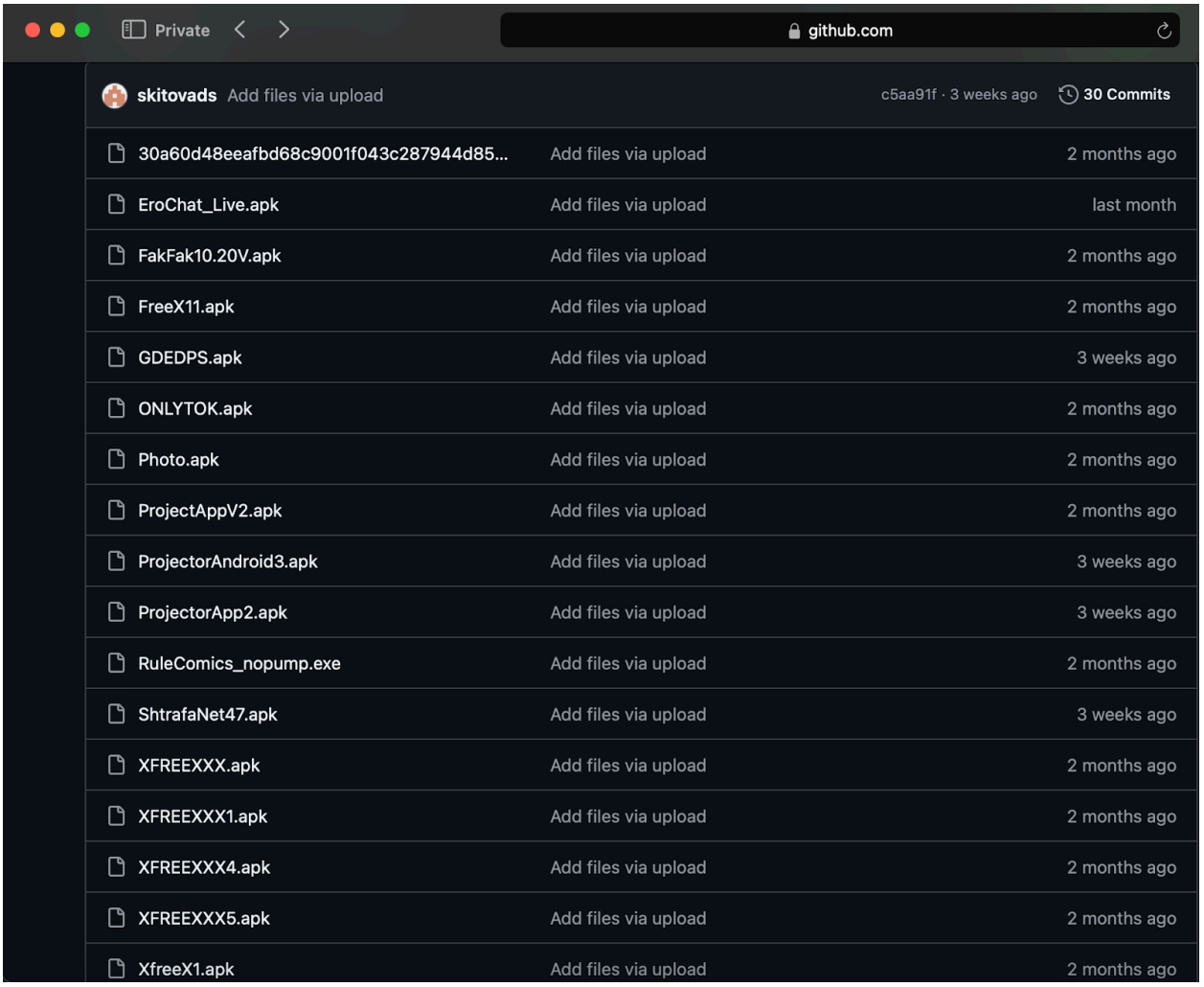

Fig.1: Threat actors hosting different malware on github repository

Technical analysis

As with prior versions, Hook abuses **Android Accessibility Services** to automate fraud and control devices remotely. The difference: its growing command set and overlay techniques give attackers even more flexibility in stealing data, hijacking sessions, and bypassing defenses.

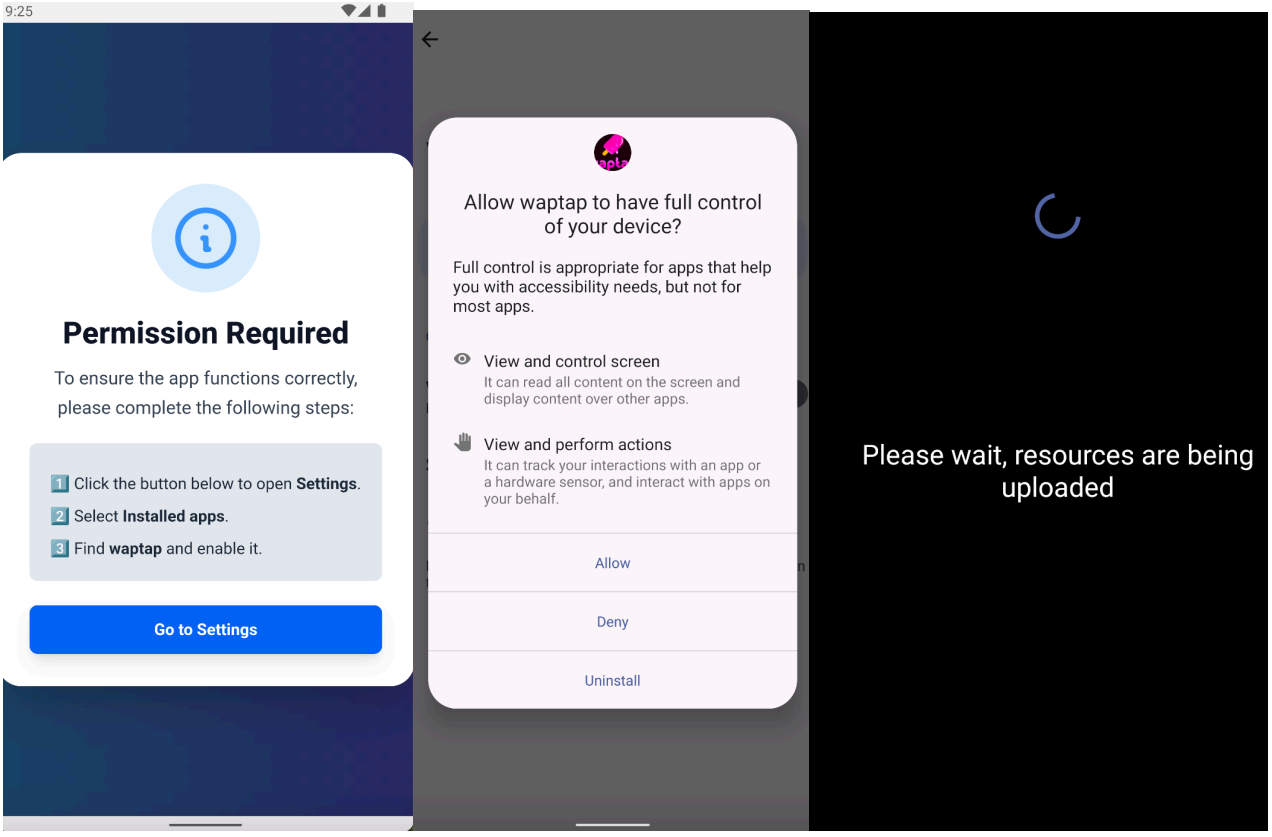

Fig.2: Malware requesting accessibility services to the victim

New Capabilities in Hook v3

In this section we analyse some of the most notorious new commands Hook implements. However, the complete list of commands utilized by Hook v3 is presented in the table after the conclusion of this document, owing to its extensive nature.

Ransomware-style overlay

A prominent characteristic of the latest variant is its capacity to deploy a full-screen **ransomware overlay**, which aims to coerce the victim into remitting a ransom payment. This overlay presents an alarming "*WARNING*" message (**Figure 3**), alongside a wallet address and amount, both of which are dynamically retrieved from the command-and-control server. The requisite HTML content for displaying this on the victim's screen is embedded within the APK itself. This behavior is remotely initiated when the malware receives the **ransome** command from the C2. Furthermore, the attacker possesses the capability to remotely dismiss the overlay from the victim’s screen by issuing a "delete_ransome" command.

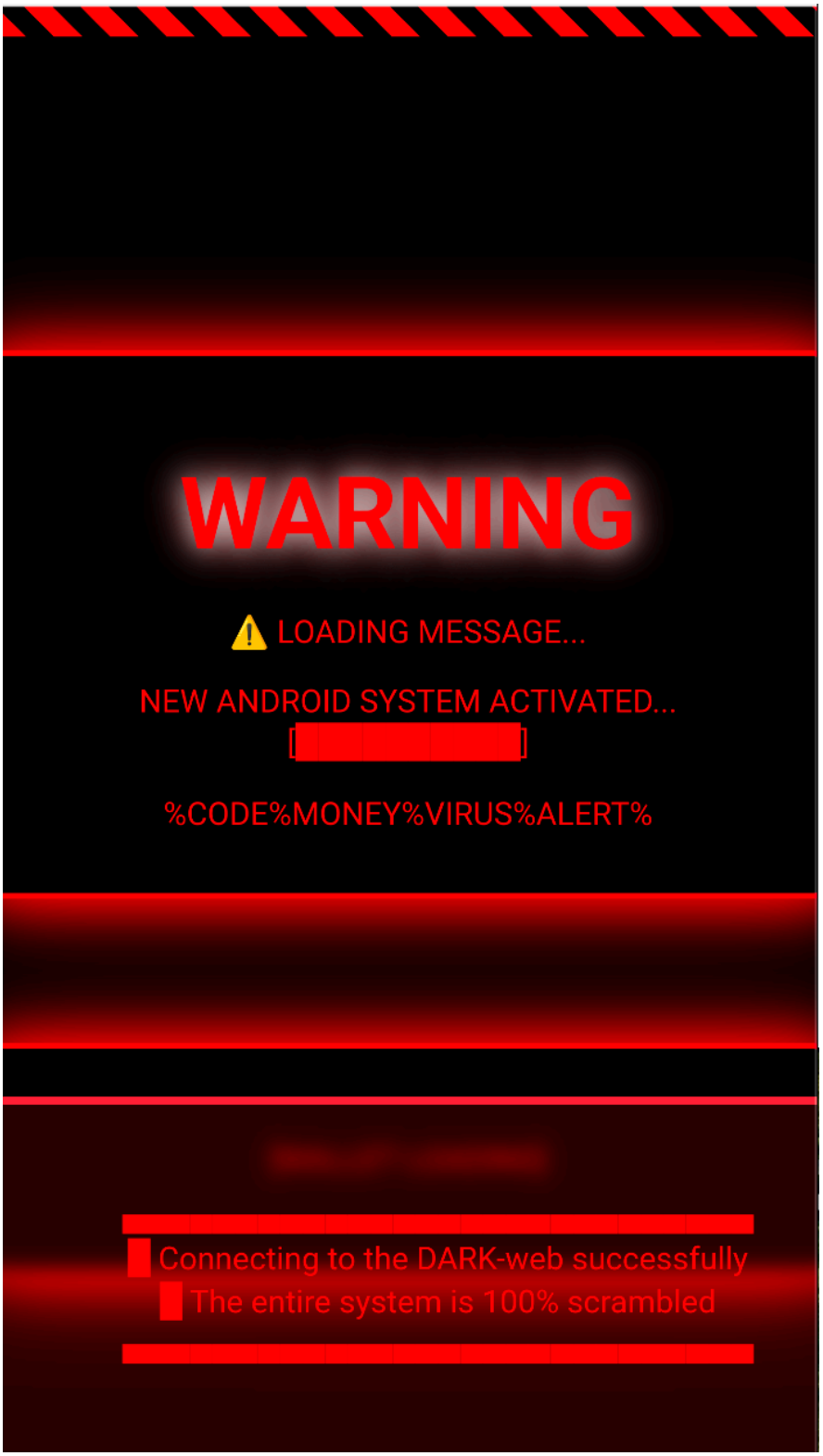

Fig.3: Ransomware style overlay

Fake NFC Overlay

The **takenfc** command is used by Android malware to display a fake NFC (**Figure 4)** scanning screen using a fullscreen WebView overlay. While the code sets up a JavaScript interface to capture user input, the current

HTML does not include the injected JavaScript needed to collect and send sensitive data to the attacker. This shows how attackers are planning to keep adding capabilities to the malware.

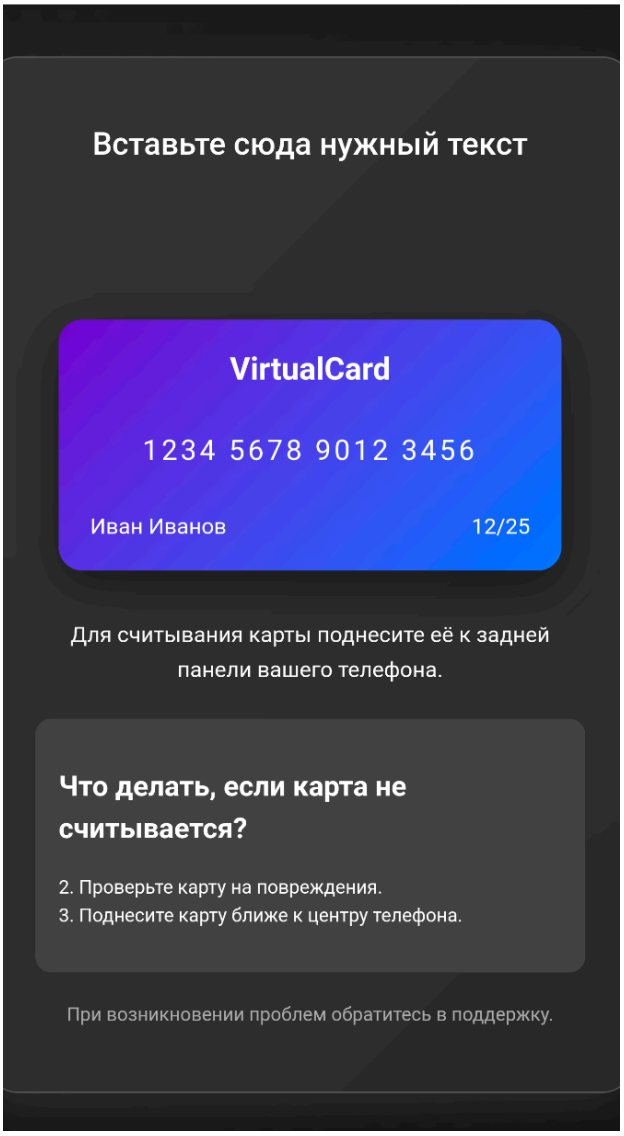

Fig.4: Fake NFC overlay

Stealing Device Lock Screen and Automating Pin Unlocking

The malware leverages an overlay technique that places a deceptive interface over the device's lock screen. This overlay mimics the legitimate unlock pattern or PIN (**Figure 5)** entry screen, tricking users into entering their credentials. By capturing the unlock pattern or PIN, the attackers gain unauthorized access to the device, effectively bypassing the lock screen security and taking full control.

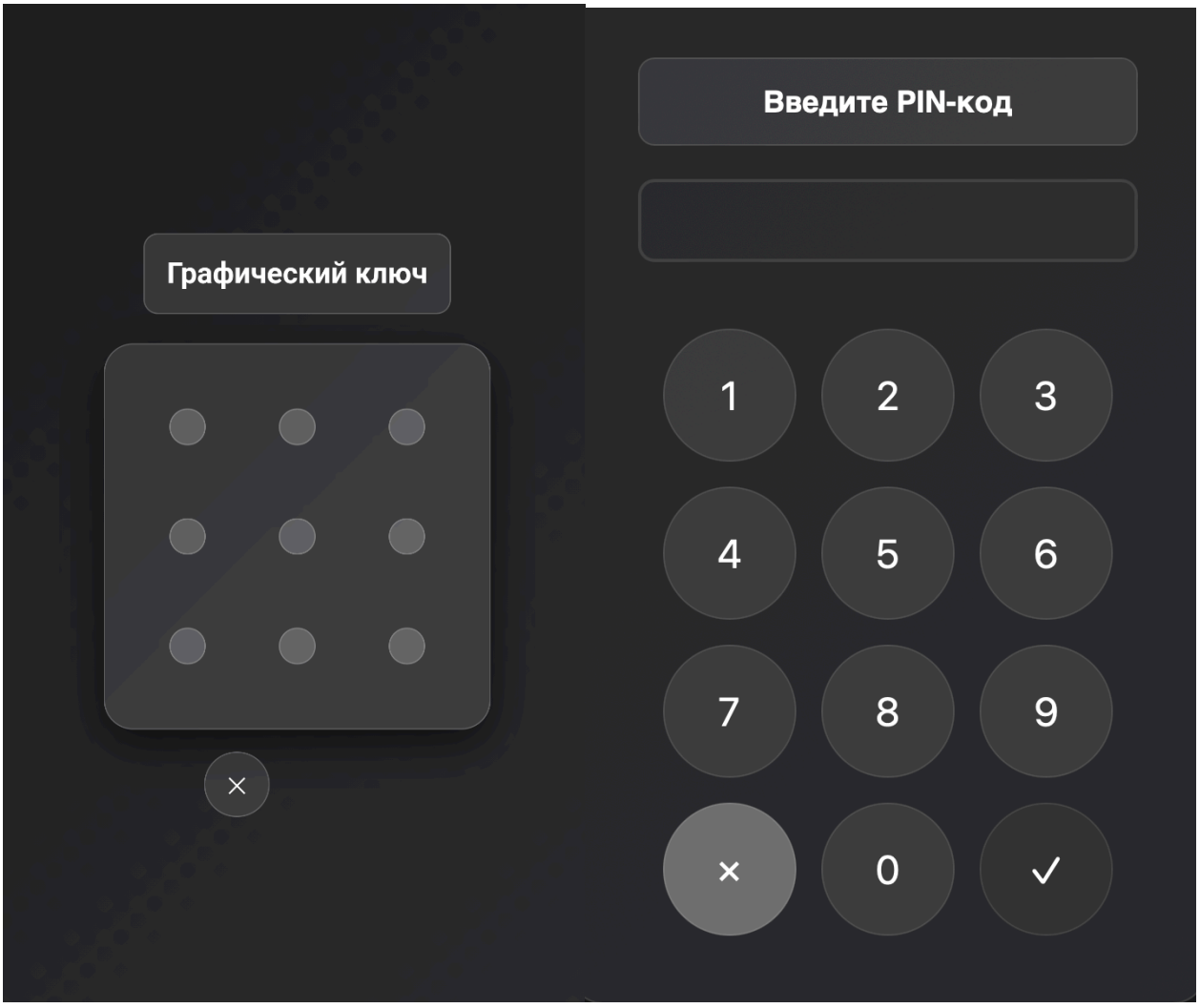

Fig.5: Overlays for stealing device lock screen

The **unlock_pin** command can programmatically unlock the device by simulating user interaction. It first acquires a WakeLock to wake the device, performs a swipe-up gesture to reveal the lock screen, and then inputs a PIN received from the payload. Each digit is clicked individually, followed by simulated taps on various confirmation buttons (e.g., "OK", "Enter", "Submit", including variants in different languages and symbols)

Fraudulent Phishing Overlay Used to Steal Card Information

The malware displays an overlay to steal credit card information whenever a **takencard** command is received from the server. It creates a full-screen WebView overlay (**Figure 6**) that mimics a legitimate interface and loads a fake HTML form.This HTML file mimics Google Pay to capture sensitive user input like card details or PIN entered in the form, then sends that data back to the server.

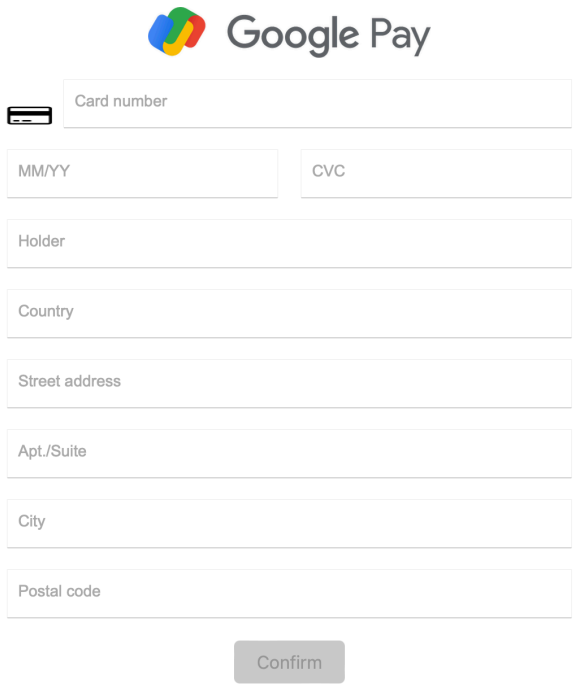

Fig.6: Phishing overlay page mimicking Google Pay

Still Cooking: Hints of Wider Plans?

The first version of Hook was published by ThreatFabric (**Figure 7**), with the malware's name explicitly present in the code. Later, NCC Group released a comparison between Hook and Ermac and shared details on a newer variant. In this updated version, the threat actors had modified the logging strings (**Figure 8**).

During our analysis of the latest banker variant, we identified several noteworthy strings being initialized, including RABBITMQ_SERVER (**Figure 9**) along with hardcoded usernames and passwords. RabbitMQ is a dedicated message broker that manages queues and messages between clients and servers, offering a more reliable and flexible C2 channel compared to basic HTTP or WebSocket communication.

Although the current build does not actively leverage RabbitMQ, its presence suggests that future versions of the malware could be configured to utilize this infrastructure, potentially enhancing resilience and scalability in C2 operations.

```
static {
    c.a = j.a("%debug1%", "%debug%");
    c.b = j.a("%blockCIS1%", "%blockCIS%");
    c.c = "http://5.42.199.22:3434";
    c.d = "1A1zP1eP5QGefi2DMPTfTL5SLmv7Divf";
    c.e = "Hook";
    c.f = "Google Chrome ";
    c.g = "Google Chrome ";
    c.h = "%Enable_Accessibility_Service%";
    c.i = new String[]{"android.permission.WRITE_EXTERNAL_STORAGE", "android.permission.READ_EXTERNAL_STORAGE", "andı
}
```

Fig.7: Hook1

```
    static {
        j.a = i.a("%debug1%", "%debug%");
        j.b = i.a("%blockCIS1%", "%blockCIS%");
        j.c = i.a("%addWaitView1%", "%addWaitView%");
        j.d = "http://91.215.85.223:3434";
        j.e = "1A1zP1eP5QGefi2DMPTfTL5SLmv7Divf";
        j.f = "0123456789abcdef";
        j.g = "Money7";
        j.h = "CaixaBankNow";
        j.i = "CaixaBankNow";
        j.j = "%Enable_Accessibility_Service%";
        String[] arr_s = {"android.permission.WRITE_EXTERNAL_STORAGE", "android.permission.READ_EXTERNAL_STORAGE", "an
        j.k = arr_s;
        String[] arr_s1 = {"android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"};
        j.l = arr_s1;
        String[] arr_s2 = {"android.permission.SYSTEM_ALERT_WINDOW"};
        j.m = arr_s2;
        String[] arr_s3 = (String[])b.H0(b.H0(arr_s, arr_s1), arr_s2);
    }

    public static String[] a() {
        return j.k;
    }
}
```

Fig8: Hook2

```
    static {
        Constantsfd.INSTANCE = new Constantsfd();
        Constantsfd.debug = Intrinsics.areEqual("%debug1%", "%debug%");
        Constantsfd.blockCIS = Intrinsics.areEqual("%blockCIS1%", "%blockCIS%");
        Constantsfd.addWaitView = Intrinsics.areEqual("true", "true");
        Constantsfd.addPush = Intrinsics.areEqual("true", "true");
        Constantsfd.RABBITMQ_SERVER = "%INSERT_RABBIT_HERE%";
        Constantsfd.RABBITMQ_USER = "android";
        Constantsfd.RABBITMQ_PASSWORD = "qwerty12345";
        Constantsfd.DEVELOPMENT_SERVER = "https://ws.asedkop.world";
        Constantsfd.BOT_SERVER = "https://coc.asedkop.world";
        Constantsfd.k = "1A1zP1eP5QGefi2DMPTfTL5SLmv7Divf";
        Constantsfd.IV = "0123456789abcdef";
        Constantsfd.tag = "fakfak";
        Constantsfd.telegrambot = "%INSERT_TELEGRAMBOT_HERE%";
        Constantsfd.telegramchat = "%INSERT_TELEGRAMCHAT_HERE%";
        Constantsfd.access1 = "FakFakService";
        Constantsfd.access2 = "FakFakService";
        Constantsfd.acname = "FakFakService";
        String[] arr_s = {"android.permission.READ_SMS", "android.permission.SEND_SMS", "android.permission.RECEIVE_SMS",
        Constantsfd.PERMISSIONS = arr_s;
        Constantsfd.PERMISSIONS14 = new String[]{"android.permission.INTERNET"};
        String[] arr_s1 = {"android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"};
        Constantsfd.PERMISSIONS2 = arr_s1;
        Constantsfd.PERMISSIONS_samsung = new String[]{"android.permission.ACCESS_COARSE_LOCATION"};
        Constantsfd.REQUEST_PERMISSIONS_samsung = new String[]{"android.permission.READ_SMS", "android.permission.SEND_SM
        Constantsfd.PERMISSIONSNFC = new String[]{"android.permission.NFC"};
        Constantsfd.PERMISSIONSNtest = new String[]{"android.permission.SYSTEM_ALERT_WINDOW"};
        String[] arr_s2 = {"android.permission.CALL_PHONE"};
        Constantsfd.PERMISSIONS3 = arr_s2;
        Constantsfd.PERMISSIONSA = (String[])ArraysKt.plus(ArraysKt.plus(arr_s, arr_s1), arr_s2);
    }
```

Fig.9: Hook3

Use of Telegram?

The malware seems to be still developing a few more features which includes the use of telegram for C2 communication (**Figure 9)**, although we have seen the use of telegram in an instance to send injection **(Figure 10**) type and injection data but we did not see any traces of chatid or bot token which strongly suggests that the malware is still developing few more features.

🆕 New inject+++++ | 🆔 UID: #<device_uid> | 📲 Application: <application_name> | 🔑 Type: <type_injects> | 📌 Field1: value1 | 📌 Field2: value2

Fig.10: Fields that are used to send to telegram

Zimperium vs. Hook

Zimperium's Mobile Threat Defense (MTD) and Mobile Runtime Protection (zDefend) protects against Hook and other advanced banking trojans through **on-device dynamic detection engine**, even if malware is sideloaded from phishing sites or GitHub.

In addition to providing protection for our customers, Zimperium collaborated with industry stakeholders to help remove the malicious repository from which Hook was being distributed. This takedown significantly reduced the threat actor's operational capabilities.

Why This Matters

The evolution of Hook illustrates how banking trojans are **rapidly converging with spyware and ransomware tactics**, blurring threat categories. With continuous feature expansion and broad distribution, these families pose a growing risk to financial institutions, enterprises, and end users alike.

Zimperium customers are protected against Hook and its variants through on-device detection and behavioral analysis.

MITRE ATT&CK Techniques

Tactic	ID	Name	Description
Initial Access	T1660	Phishing	Adversaries host phishing websites or host apk's in github
Persistance	T1624.001	Event Triggered Execution: Broadcast Receivers	It creates a broadcast receiver to receive SMS events
Privilege Escalation	T1626.001	Abuse Elevation Control Mechanism: Device Administrator Permissions	Malware is capable of factory reset, reset device pin/password, Disable lockscreen, Can watch login attempts from victim
Defense Evasion	T1655.001	Masquerading: Match Legitimate Name or Location	Malware pretending to be google chrome and many other legit applications

	T1630.001	Indicator Removal on Host: Uninstall Malicious Application	Malware can uninstall itself
	T1629.002	Device Lockout	Malware can lockout victim through the device by DevicePolicyManager.lockNow()
	T1516	Input Injection	Malware can mimic user interaction, perform clicks and various gestures, and input data
	T1406.002	Obfuscated Files or Information: Software Packing	It is using obfuscation and packers (JSONPacker) to conceal its code.
Credential Access	T1517	Access Notifications	The malware leverages Android NotificationListenerService to intercept OTPs and sensitive data from notifications, dismissing or manipulating them to avoid user detection.
	T1414	Clipboard Data	It extracts data stored on the clipboard.
	T1417.001	Input Capture: Keylogging	It has a keylogger feature
	T1417.002	Input Capture: GUI Input Capture	It is able to get the shown UI.
Discovery	T1420	File and Directory Discovery	lists the files at a specified path (additional parameter “ls”), or downloads a file from the specified path (additional parameter “dl”)
	T1430	Location Tracking	Malware can track victim’s location

	T1418	Software Discovery	Malware collects installed application package list
	T1421	System Network Connections Discovery	Adversaries may attempt to get a listing of network connections to or from the compromised device
	T1426	System Information Discovery	The malware collects basic device info.
Collection	T1517	Access Notifications	It registers a receiver to monitor incoming SMS messages
	T1513	Screen Capture	Malware can record screen content
	T1533	Data from Local System	Malware can access photos from the device
	T1512	Capture Camera	Malware opens camera and takes pictures
	T1429	Audio Capture	Malware captures Audio recordings
	T1616	Call Control	Malware can make calls
	T1636.002	Protected User Data: Call Log	Malware steals call logs
	T1636.003	Protected User Data: Contact List	It exports the device’s contacts.
	T1636.004	Protected User Data: SMS Messages	Steals SMSs from the infected device

	T1409	Stored Application Data	Hook can request the GET_ACCOUNTS permission to get the list of accounts on the device,
	T1417.001	Input Capture: Keylogging	Malware can capture keystrokes
	T1417.002	Input Capture: GUI Input Capture	It is able to get the shown UI.
	T414	Clipboard Data	It has the ability to steal data from the clipboard.
	T1616	Call Control	TA can forward call from the device
Command and Control	T1616	Call Control	TA can forward call from the device
	T1637	Dynamic Resolution	It receives the injected HTML payload endpoint dynamically from the server.
	T1481.002	Web Service: Bidirectional Communication	It uses websocket communication to poll the TA's server and get the commands to execute.
Exfiltration	T1646	Exfiltration Over C2 Channel	Sending exfiltrated data over C&C server
Impact	T1616	Call Control	TA can make and block call in the device
	T1516	Input Injection	It displays inject payloads like pattern lock and mimics banking apps login screen through overlay and steal credentials.

	T1582	SMS Control	It can read and send SMS.

Indicators of Compromise

The full list of IOCs can be found in this repository.

Hook Command List

Command	Description
action_recorded_gesture	**Executes remote gesture commands via AccessibilityService to simulate user actions on the device.**
start_vnc	Starts capturing the victim's screen constantly (streaming)
startussd	Executes a given USSD code on the victim's device
get_unlockpass	resets the unlock password status to false.
send_sms_many	Sends an SMS message to multiple phone numbers
swipeup	Perform a swipe up gesture
takescreenshot	Takes a screenshot of the victim's device
bitcoincom	Launches the Bitcoin Wallet app
clickatcontaintext	Clicks on the UI element that contains the payload text
start_hvnc	starts an HVNC session by simulating a swipe gesture and sends device/app info to the attacker's server.

start_perm	Requests necessary permissions and logs of all, some, or none are granted
startadmin	Sets the “start_admin” shared preference key to value 1, which is probably used as a check before attempting to gain Device Admin privileges
delete_pincodep	Removes PIN input overlay from top of the screen
takenfc	Places NFC overlay on top of the screen
start_record_gesture	Starts recording user gesture by displaying a transparent full screen overlay
removewaitview	Removes the “wait / loading” view that is displayed on the victim’s device because of the “addwaitview” command
cookie	Steals session cookies (targets victim’s Google account)
exodus	Starts the Exodus Wallet application (and steals seed phrases as a result of starting this application, as observed during analysis of the accessibility service)
clearcash	Sets the “autoClickCache” shared preference key to value 1, and launches the “Application Details” setting for the specified app (probably to clear the cache)
stop_textview	Triggers action to stop text view
updateinjectandlistapps	Gets a list of the currently installed apps on the victim’s device, and downloads the injection target lists
logaccounts	Gets a list of the accounts on the victim’s device by their name and account type
metamask	Launches the Metamask Wallet app

pincodep	Places an overly for Pincode
scrollup	Performs a scroll up gesture
getlocation	Gets the geographic coordinates (latitude and longitude) of the victim
stop_record_gesture	Stops the gesture recording and removes the overlay, packages recorded data into json and resets it again
mycelium	Launches the Mycelium Wallet app
swipePattern	Parses a list of points from json which are received from the server and converts them into integer coordinate pairs representing a swipe pattern
restart3	Restarts the accessibility services
restart4	Same as restart3
getinstallapps	Gets a list of the installed apps on the victim’s device
getaccounts	Gets a list of the accounts on the victim’s device by their name and account type
onpointerevent	Sets X and Y coordinates and performs an action based on the payload text provided. Three options: “down”, “continue”, and “up”. It looks like these payload texts work together, as in: it first sets the starting coordinates where it should press down, then it sets the coordinates where it should draw a line to from the previous starting coordinates, then it performs a stroke gesture using this information
deleteapplication	Uninstalls a specified application received from the server

tap	Dispatches a tap gesture at the specified coordinates
kill	kills the current running process of the app
piuk	Launches the Blockchain Wallet app
push	Displays a push notification with app name,title,text from the server
downloadimage	Downloads an image from the victim’s device
makecall	Calls the number specified from the payload received from the server
openwhatsapp	Sends a message through Whatsapp to the specified number
scrolldown	Performs a scroll down gesture
swipe	Performs a swipe gesture with the specified 4 coordinates
toshi	Launches the Coinbase Wallet app
trust	Launches the Trust Wallet app
width	Extracts “width” value from the payload then converts it to integer and saves it to “image_width” in the sharedprefs
delete_patternp	Removes overlay of pattern
longpress	Dispatches a long press gesture at the specified coordinates

addviewhvnc	Displays a transparent overlay on screen with a message “please wait”
swiperight	Performs a swipe right gesture
calling	Calls the number specified in the “number” payload, tries to lock the device and attempts to hide and mute the application
forwardsms	Sets up an SMS forwarder to forward the received and sent SMS messages from the victim device to the specified number in the payload
quality	Sets and saves the image quality settings for the VNC
getcallhistory	Gets a log of the calls that the victim made
clickat	Clicks at a specific UI element
clicker	Simulates a gesture(tap or series of taps) on the screen with specified points and duration
ransome	Shows Ransomware overlay on top of the device
settransperet	requests needed permissions on startup and closes itself immediately after, logging the permission results.
getgmailmessage	Sets the “gm_mes_command” shared preference key to the value “start” and starts the Gmail app
restart	Restarts accessibility just like restart3 and restart 4

removeview	Removes the view with the black background that was added by the “addview” command
getvktitles	Launches the VKontakte app
cuttext	Replaces the clipboard on the victim’s device with the payload text
addcontact	Adds a new contact to the victim’s device
delete_ransome	Removes the ransomware overlay
startauthenticator2	Starts the Google Authenticator app
patternp	Places overlay for pattern
startapp	Starts the app specified in the payload
fpslimit	Updates the stored image quality setting
sendsmsall	Sends a specified SMS message to all contacts on the victim’s device. If the SMS message is too large, it will send the message in multiple parts
getimages	Gets list of all images on the victim’s device
getcontacts	Gets list of all contacts on the victim’s device
takencard	Places card overlay on top of the screen
takephoto	Takes a photo of the victim using the front facing camera

swipedown	Performs a swipe down gesture
swipeleft	Performs a swipe left gesture
stop_hvnc	Sets the running status of hvnc to false
forwardcall	Sets up a call forwarder to forward all calls to the specified number in the payload
stop_vnc	Stops capturing the victims screen
clickattext	Clicks on the UI element with a specific text value
delete_nfc	Removes the fake nfc overlay
safepal	Starts the Safepal Wallet application
samourai	Launches the Samourai Wallet app
sendsms	Send a specified SMS message to a specified number. If the SMS message is too large, it will send the message in multiple parts
settext	Sets a specified UI element to the specified text
getphone	Sends the device manufacturer and model to the server
start_vnc_socket	immediately starts the screen streaming activity with minimal setup, skipping overlays and wake locks. It’s designed for a quick, direct launch of the VNC session.

fmmanager	Either lists the files at a specified path (additional parameter “ls”), or downloads a file from the specified path (additional parameter “dl”)
openapp	Opens a specified app
openurl	Opens the specified URL
getsim	Gets a sim operator and sends to server
getsms	Steals all SMS messages
startinject	Performs a phishing overlay attack against the given application
height	Sets the image height for the VNC stream based on the value received in the payload.
addview	Adds a new view with a black background that covers the entire screen
flash_set	Adjusts screen brightness to maximum if system write permission is granted; otherwise logs and flags permission denial.
killme	Stores the package name of the malicious app in the “killApplication” shared preference key, in order to uninstall it.
delete_card	Removed the card overlay
onkeyevent	Performs a certain action depending on the specified key payload (POWER DIALOG, BACK, HOME, LOCK SCREEN, or RECENTS
imagesize	Sets the image size received from the server

unlock_pin	Remotely unlocks the device by simulating swipe, PIN entry, and confirmation taps using AccessibilityService and wake lock control
unlock	Unlocks device
addwaitview	Displays a “wait / loading” view with a progress bar, custom background colour, text colour, and text to be displayed
gmailtitle	Sets the “gm_list” shared preference key to the value “start” and starts the Gmail app
clearcache	Sets the “autoClickCache” shared preference key to value 1, and launches the “Application Details” setting for the specified app

---