

The Rage of Android Banking Trojans

threatfabric.com/blogs/the-rage-of-android-banking-trojans.html

April 2021





Introduction

In Greek mythology Achilles was quite simply invincible during the Trojan War; he was also rather proud and bad-tempered for his own good and his rage would cost both his countrymen and the enemy dearly. In the past 7 years, ThreatFabric has discovered many new Android banking trojans, all with one common trait: an insatiable rage against Android banking apps. In this blog we will discuss what have been the underlying catalysts behind this rage and what new weapons are currently filling the virtual Trojan Horses. The second part of the blog is focused on new on-device fraud capabilities utilised by malware families to perform fraud in an automated way using the victims own Android banking app.

Catalysts

Catalysts

behind The Rage



Increase
Mobile Banking
Users



From MaaS to
Private Trojans
Pays out



Leaked src
Anubis 2.5
Cerberus



Open OS
Accessibility
Notification
Clipboard



Dropper as a Service
Google Play Store
Professionalisation



One of the most obvious catalysts that played an important role in The Rage we are experiencing are the source code leaks of two very effective bots, namely Anubis 2.5 and Cerberus: these leaks resulted in multiple private trojan versions actively targeting regions such as Poland, Spain, Turkey, and Italy (local actors).

We also noticed a very clear new trend adopted by Android banking families in the way they advertise themselves. From 2018 to mid 2020 Android banking trojans from families like Red Alert or Cerberus, had all adopted the Malware as a Service (MaaS) model: actors would rent their malware services on a subscription basis and would aggressively advertise their service on multiple dark web forums. What's noticeable is that the MaaS strategy for most adversaries has resulted in financial gain on short term but has not been very sustainable over time (Maza-in's Anubis and Red Alert as MaaS are good examples). However, recent malware families, including Alien or Medusa among others, adopted a more reserved approach, limiting their exposure on public forums and using side-channels for the customers to communicate directly with the vendor.

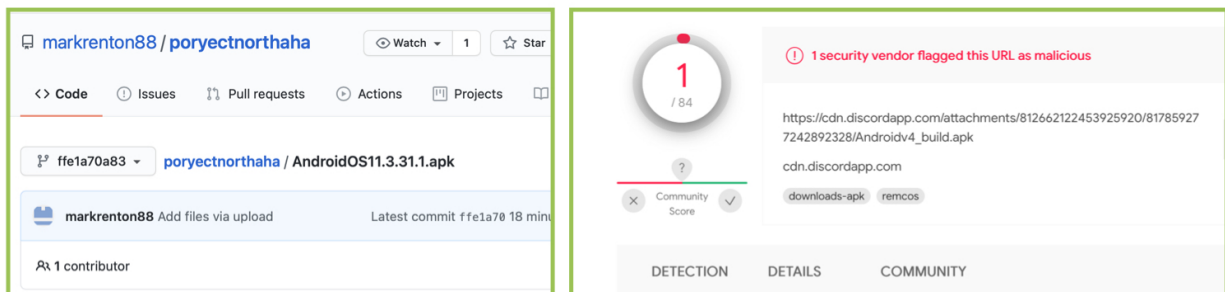
Looking at the current successful infections rates of the Trojans families we are tracking, we can only conclude that the new private strategy is paying out and seems to be a more sustainable business model.

The last catalyst that endorses The Rage is the professionalization in malware distribution campaigns. Within the Android banking trojans ecosystem, we observed an increase in the number of advisories providing so-called dedicated trojan distribution services (DaaS). These services usually consist of dropper/loader Android apps (masquerading as legitimate apps) in different app stores, including the Google Play Store. The Rage does not stop at abusing

trusted apps stores. Recently we have seen a considerable number of distribution campaigns utilizing GitHub, Discord and other social media channels as main storage and spreading tactic.

GitHub & Discord

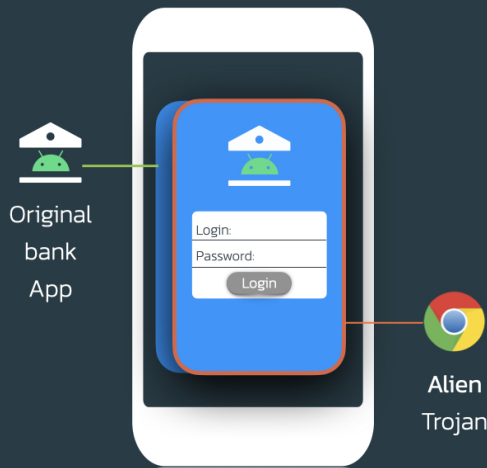
Abusing CDN's and Repositories



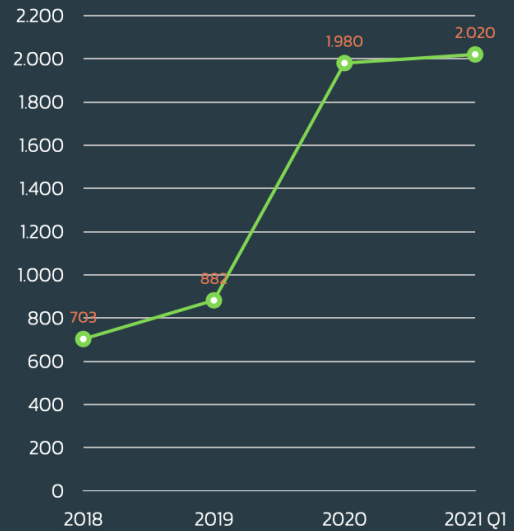
Statistics

Our Mobile Threat Intel (MTI) platform, responsible for classifying Android banking malware samples, cataloguing their technical malware capabilities and extracting the so-called overlay targets, has observed a 129% increase in the list of apps targeted with overlays attacks since 2019. The largest increase has taken place the past year and The Rage is continuing in 2021.

Overlay Attack



129% increase targeted apps



The most targeted apps are related to crypto currency, but the top 20 also includes many apps from banks.

Top 20 Overlay targets Q1-2021

#	Overlay targets	Malware samples
1	piuk.blockchain.android	14.049
2	com.coinbase.android	14.025
3	au.com.nab.mobile	12.289
4	com.commbank.netbank	12.250
5	org.banksa.bank	12.192
6	org.stgeorge.bank	12.191
7	com.grppl.android.shell.halifax	12.074
8	com.grppl.android.shell.cmbllloydstsb73	12.068
9	es.cm.android	12.050
10	org.bom.bank	11.799
11	uk.co.santander.santanderuk	11.762
12	com.paypal.android.p2pmobile	11.614
13	com.bbva.bbvacontigo	11.387
14	com.bankinter.launcher	11.380
15	com.kutxabank.android	11.359
16	es.evobanco.bancamovil	11.251
17	com.tecnocom.cajalaboral	11.244
18	es.bancosantander.apps	11.007
19	es.lacaixa.mobile.android.newwapicon	10.797
20	com.bbva.netcash	10.664

With many different crypto-currencies hitting their highest market value in 2021, populating newsfeeds all over the world and now more than ever being discussed extensively in mainstream media, it is not a surprise that crypto-currency wallets are the most common targets for this new wave of banking trojans. Another important fact to consider is that, while

banking apps tend to have different versions of their APK based on the country they serve, crypto-wallets tend to have one unique APK, making it easier for malicious actors to target them.

New capabilities & trends

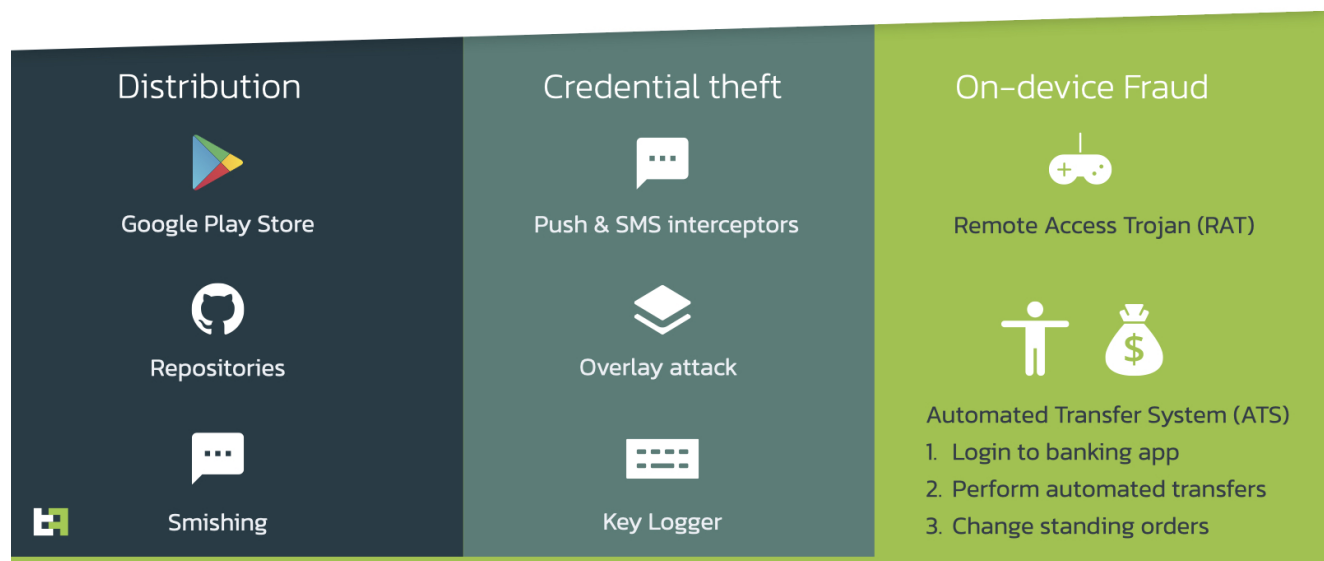
A clear new trend in Android banking trojans families is the focus on developing malware capabilities that allow actors to perform fraud on a victims in an automated way using the victims own banking or Bitcoin wallet apps, overview:

- Scaling on-device fraud attempts by developing Automated Transfer System modules powered by AccessibilityService.
- New ways to start a Remote Access sessions (RAT) relaying on Android native code to avoid additional installations (VNC/Teamvier)
- Logging all (secret) content inside apps, including OTP apps like Authenticator (Google/Microsoft).
- Manipulating the beneficiary input fields of Android banking apps while the victim is in the flow of performing payments (very successful attack).

Entering a new ERA shifting focus from credential stealing capabilities to on-device fraud automation.

Android Banking Trojans

New ERA – From credential theft to on-device fraud tactics



For the past 5 years the main way to steal mobile banking login credentials and verification codes (OTP) has been the use of so-called overlay attacks. With this attack pattern (MITRE TTP: T1411) attackers harvest login credentials with a fake login window on top of the original banking app. In the past year malicious actors mainly used these stolen credentials

to register a new device to perform fraud or steal the currency in a crypto wallet using a different channel, for example through the web interface. This attack is also known as device registrations fraud, which results in financial loss on a separate device or channel.

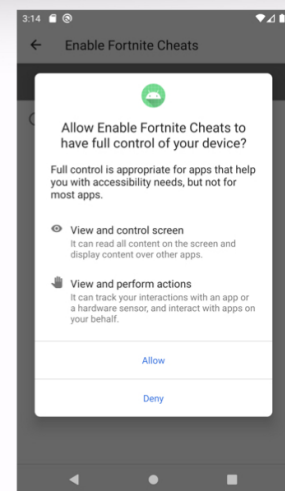
However, as covered in our 2020 blog [“Year of the RAT”](#), actors moved to execute financial fraud scenarios directly on the victim’s device by installing additional services such as a back-connect proxy and remote access software, such as VNC or TeamViewer, to control the victim’s device remotely.

This year actors have taken the so called on-device fraud strategy to the next level by performing actions in the targeted banking app on behalf of the victim and even automating fraudulent transfers.

Automated Transfer System (ATS)



- Retrieve UI information (including notifications)
- Perform actions on the user’s behalf
- Perform Key logging
- Read any text displayed on the screen (such as OTP)
- Listen to touch events
- Draw over other apps (API level >= 22)



Android’s Accessibility Service’s main purpose is to assist users with disabilities. However, when a victim is lured by Android banking trojans into enabling this service with enticing and repeating fake messages, the (malicious) AccessibilityService can read anything a normal user can see and recreate any action an user can do on an Android device.

Let’s dig a bit deeper on how this works by analyzing the trojans that have mastered this attack vector this year: Gustuff and Medusa. Let’s take a transfer activity from a demo banking app as an example: from an Accessibility perspective, all the input fields have a so called `@Android:id` label which can be read and controlled by any AccessibilityService running on the victim’s device. In this example, by providing the bot command `setText(TEXT)`, an attacker can, in a fully automated way, change the beneficiary account

number to anything he/she wants in order to transfer funds to a money mule. In general, the malware's Accessibility script first reads the balance information of a victim (also through an automated process) before they perform this attack.

Android UI labels

AccessibilityService scripting

The diagram illustrates the connection between an AccessibilityService script and a mobile banking app interface. On the left, a red spider icon represents the trojan. Below it, the text "Trojan Accessibility setText" is shown. A code block contains the following script:

```
"type": "money_mule",  
"id": "com.your.bank.id/  
input_IBAN_account_number_UI",  
"setText": "CZ8050513596529899771545"
```

 Above the code, the Android ID `@android:id=input_IBAN_account_number_UI` is highlighted in a green box. A green arrow points from this box to the IBAN input field in the app screenshot on the right. The screenshot shows a "Sent" transaction screen for "Fintory GmbH" with a value of "- 5.720,30 €". The "Details" section shows a "Bank transfer" with an IBAN of "DE56 3902 0000 1203 2339 39" and a BIC of "DUISDE33XX".

To provide a bit more context, the Accessibility script below is used by the Android Banking trojan Gustuff for the St.George Android banking app: it performs a login on behalf of the victim to start a session in a timely fashion (by using some sleep cycles to look more legitimate) and uses this active apps session to script against the transfer screen of the mobile banking app to perform a payment to a mule on behalf of the victim, successfully completing a full ATS attack.


```

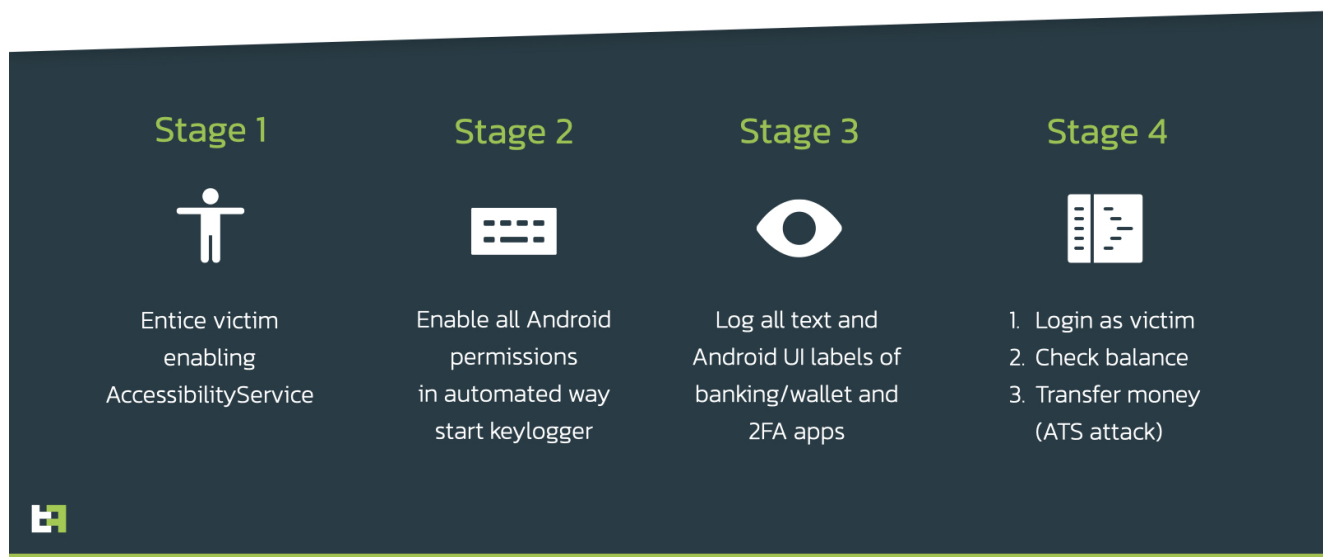
function stgeorge(info) {
    let actions = [{"type": "open", "open": "launch", "value": "org.stgeorge.bank"},
{"type": "delay", "time": 1000}, {"type": "windows", "root": true},
    {"type": "interactive", "viewId": "org.stgeorge.bank:id/continue_button",
"click": true}, {"type": "delay", "time": 1000},
    {"type": "interactive", "viewId": "org.stgeorge.bank:id/btn_logon", "click":
true}, {"type": "delay", "time": 1000},
    {"type": "interactive", "viewId": "org.stgeorge.bank:id/logon_button",
"click": true}, {"type": "delay", "time": 8000}
    ];
    if (info.securityNum) {
        actions = actions.concat([
            {"type": "interactive",
            "viewId": "org.stgeorge.bank:id/pin_editor",
            "setText": info.securityNum
            }]);
    } else if (info.pass) {
        actions = actions.concat([
            {"type": "interactive", "viewId":
"org.stgeorge.bank:id/internet_password_ET", "setText": info.pass }, {"type":
"delay", "time": 500},
            {"type": "interactive", "viewId": "org.stgeorge.bank:id/login_Button",
"click": true }
        ]);
    }
    return utils.buildCommand("array", {"actions": actions});
}

```

This capability adds a new layer of danger to the Android banking malware ecosystem, by making large scale campaigns more automated and easier to manage for threat actors. To summarize the on-device fraud MO:

Gustuff: Automated Transfer System (ATS)

Accessibility malware script to automate login, balance check and transaction



Accessibility Event Logging

Another incredibly powerful feature of multiple families, including Medusa and Gustuff, is event logging. If the bot receives the command from the C2, it starts to recursively collect the information about the active window starting from the root node, which means it is able to collect information about everything that is displayed on the screen. Information of interest includes, but is not limited to, the following:

- Node bounds in screen coordinates (position of elements in the UI)
- Text of the node (the text inside an element)
- Whether this node is password (if the element is a field of type “password”)

The following snippet from Anatsa shows the code that collects the information of active Node and all its children that are matching a specific string:

```
public static List getAllNodes(AccessibilityNodeInfo arg6, String arg7) {
    String v0 = arg7.toLowerCase();
    ArrayList v1 = new ArrayList();
    if(arg6 == null) {
        return v1;
    }

    int v2 = arg6.getChildCount();
    int v3;
    for(v3 = 0; v3 < v2; ++v3) {
        AccessibilityNodeInfo v4 = arg6.getChild(v3);
        if(v4 != null) {
            if(v4.getClassName() != null &&
v4.getClassName().toString().toLowerCase().contains(v0)) {
                v1.add(v4);
            }
            else {
                v1.addAll(Utils.getAllNodes(v4, arg7));
            }
        }
    }
    return v1;
}
```

By collecting all this data, the actor can get a better understanding of the interface of different applications and therefore implement relevant scenarios for accessibility scripting feature. Moreover, it allows actors to have deeper insights on the applications the victim uses, its typical usage and it allows actors to intercept some of its data.

Replacing account number in input fields

Another Accessibility trick has been introduced by Medusa Android banking trojan, and it is triggered by the command `fillfocus`. This feature allows the actor to change the content of the focused input field with some text specified by the attacker. This feature can be used

to invisibly substitute the victim's input with the one set by the actor(s). This is done by abusing the AccessibilityService. The following snippet shows the code that sets the focused input field with text received from the C2:

```
public void fillfocus(int cmdId, String t_text) {
    if (WorkerAccessibilityService.accessibilityService != null) {
        Bundle bundle = new Bundle();
        bundle.putCharSequence("ACTION_ARGUMENT_SET_TEXT_CHARSEQUENCE", t_text);
//find FOCUS_INPUT field
        AccessibilityNodeInfo accessNodeInfo =
WorkerAccessibilityService.accessibilityService.findFocus(1);
        if (accessNodeInfo != null) { //perform ACTION_SET_TEXT
            accessNodeInfo.performAction(0x200000, bundle);
        }
        this.sendCmdExecuted(cmdId);
        return;
    }
    throw null;
}
```

With this feature, actors can for example modify the bank account number that the victim selected with one controlled by the attacker, effectively tricking the victim into transferring money to a money mule.

Clipper

Another newly introduced feature is the capability to change the clipboard content to some text specified by the actor(s). The Medusa Trojan can receive the command "copyclip" with as parameter text to be set. This is a common MO for so called "clippers", a type of malware that steals or substitutes the clipboard data. Similar in concept to the previous technique, it is usually used in order to invisibly substitute some sensitive data such as, IBAN or cryptocurrency wallet address, tricking the victim into performing an operation, such as a transaction, to a beneficiary which was not the original one. The following snippet shows the code that sets the clipboard data with text received from the C2:

```
private void copyclip(int cmdId, String textFromC2) {
    Context ctx = this.mContext.getApplicationContext();
    try {
        ((ClipboardManager)ctx.getSystemService("clipboard"))
            .setPrimaryClip(ClipData.newPlainText("Copied Text", textFromC2));
    }
    catch(Exception unused_ex) {}
    this.sendCmdExecuted(cmdId);
}
```

Screen casting using integrated solutions

In the past 2 years Android banking trojans actors have focused on adding so called Remote Access Trojan (RAT) capabilities by installing and configuring an additional VNC and Team services on the victims. This is a very loud activity from malware detection perspective. It

seems that new actors behind trojans such as Medusa, have figured out that the Android OS itself can natively support the hidden RAT objective. Many new families are using Accessibility services to perform actions on the victims' behalf in combination with audio and video streaming using RTSP (Real Time Streaming Protocol) giving an incredibly powerful feature to the RAT without the need to install additional apps such as VNC/TeamViewer:

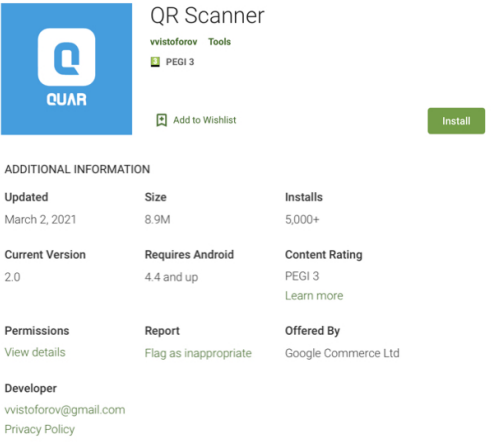
```
public static String lA(int arg3, String arg4, String arg5, String arg6) {
    StringBuilder v0 = new StringBuilder().insert(0, "m=video 0 RTP/AVP
96\r\na=rtpmap:96 H265/90000\r\na=fmtp:96 sprop-sps=");
    v0.append(arg4);
    v0.append("; sprop-pps=");
    v0.append(arg5);
    v0.append("; sprop-vps=");
    v0.append(arg6);
    v0.append(";\r\na=control:trackID=");
    v0.append(arg3);
    v0.append("\r\n");
    return v0.toString();
}
```

Distribution

New Google Play Store banking malware campaign

Distribution via Google Play Store

QR Scanners dropping Anubis (private) Banking Trojan








QR Scanner
vvistorov Tools
PEGI 3
Install

ADDITIONAL INFORMATION

Updated March 2, 2021	Size 8.9M	Installs 5,000+
Current Version 2.0	Requires Android 4.4 and up	Content Rating PEGI 3 Learn more
Permissions View details	Report Flag as inappropriate	Offered By Google Commerce Ltd
Developer vvistorov@gmail.com Privacy Policy		

5.000 – 10.000+ installs per app

- 
QR Scanner (com.quar.qrscanner)
998ba967bb23e6324c8f689ca0e1b5f28434d1ffdd52eac751f0649f037328c1
- 
QR & Barcode Scanner (com.globalcorp.qrcodebarcode)
7c19e93a31452cb354d5e4627f9bcfdbf7b4b2dd134624e4cc236f0d9360e7a3
- 
QR & Bar Code Scanner (com.solvercompany.qrbarcodescanner)
63321393c42fc6b4337922bd9d6d63d3ecbab71f790a4131136e1afb01be89a
- 
QR Code Scanner (com.tasklog.qrscodescanner)
22a5bbcace98cc2aed99a90b767db2062cdacbab20f84361b466881ea57dee23
- 
QR Scanner (com.qrtask.qrscanner)
2ac5c2dbbc2fa75431f8f0d7a9340c2e913d2b4843c1e56d0f4927f2465455c4

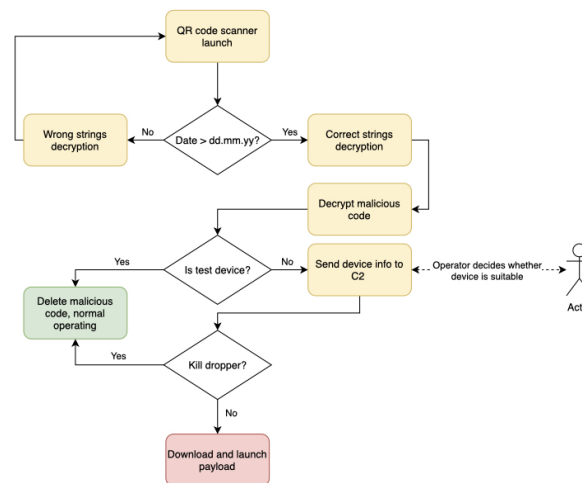
ThreatFabric has been tracking a strong group that has been very successful in spreading trojans on the Google Play Store using apps masquerading as “QR Scanners”. The main purpose of these so-called malware dropper apps is to spread a private/customized version of the Anubis Banking Trojan targeting over 1200 banking and cryptocurrency wallet apps.

The dropper apps have been successfully reappearing in the Google Play Store over a period of 13 months, regardless of our strong efforts in reporting these apps as malicious to Google.

The first Google Play dropper app masquerading as “QR Scanner” appeared in February 2020 (com.tasklog.qrscanner) and one of the latest (com.quar.qrscanner) was uploaded to the Play Store on March 2021. This malware distribution campaign has resulted in at least 30.000 infected devices, and the actors behind them are preparing a new dropper app at the time of writing.

Dropper Evasion tactics

Google Play Store



The dropper apps are only active for a short time with long-time pauses between active periods. To stay under the radar, it also implements several evasion techniques to bypass static and dynamic analysis during the Google Play Store evaluation period, as well as to make further analysis by security researchers and AV products more complicated. For example, the string decryption routine will be performed correctly only if a datetime check will be passed: if the date is earlier than stated in the code, the decryption will be performed in a wrong way, which will prevent the decryption, and therefore the launch of the malicious dropper code.

```
public static byte[] decrypt_string(byte[] arg1) {
    return qk.is_later_then_05_03_2021() ?
        ra.xor_int(arg1, rq.int_52) : ra.xor_int(arg1, 0);
}
```

After the deadline passed, the malicious dropper is decrypted and launched. Nevertheless, this stage will also perform several checks to define if the device is suitable to download the actual payload. The dropper will collect the information about the device and send it to the

C2: hardware information, list of all system and third-party packages installed on the device, is device being used to debug applications via USB, etc. On the C2 side the actor(s) will decide whether to continue with downloading the payload or not.

The C2 will respond, whether the dropper should download a payload or kill itself. If “kill” command is received, the malicious code launched earlier will be deleted from the device and never be launched again.

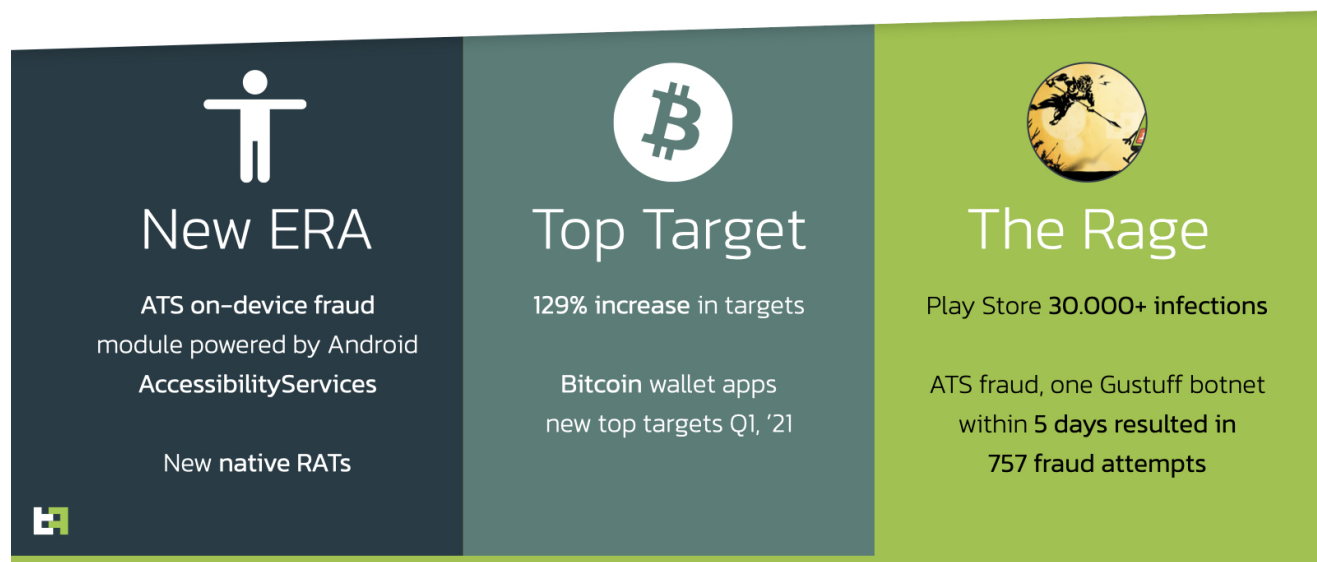
The whole process is highly manually maintained by the actor(s), making it difficult to detect from an automated perspective.

The actual payload seen by ThreatFabric analysts is a private and customized variant of the Anubis Trojan that is packed with a commercial tool (Dexprotector). This campaign shows that actor(s) behind banking Trojans are highly skilled and inventive to stay under the radar and deliver the malicious applications on users’ devices.

Conclusion

The Rage of Android Banking Trojans

Conclusion



There has been a 129% increase in targeted banking apps in the span of only one year. A noticeable addition are cryptocurrency wallets apps, that are now part of every new Android banking trojan family.

Existing families like Gustuff and new Android Banking Trojans like Medusa have fully adapted to performing on-device fraud attacks by automating the login sequence, checking balance of the victim, and creating payments to money mules using ATS (Automated Transfer System) modules. This attack vector is achieved by abusing Accessibility features

of the Android operating system. To continue fortifying the on-device fraud strategy, adversaries also discovered that they can use native Android code (instead of TeamViewer/VNC) to achieve screen streaming capabilities (using RTSP), making the attack less noticeable on the victim's device. Chaining this native screen streaming feature of Android with Accessibility controls, such as performing actions (clicks) on the victims' behalf, results in a full hidden Remote Access Trojan (RAT). We can consider these developments a significant threat to mobile payments on the Android platform.

This overview shows a brief recap of the main ATS capabilities from different families:



The success rate of one Gustuff botnet (4 active botnets at the time of writing) in only one week time consisted in 757 harvested credentials and ATS fraud attempts in countries such as the UK, Canada and Australia.

The Rage does not end here, adversaries have also proven to continuously bypass Google Play Store malware protection controls with apps masquerading as "QR-Reader/Scanner" over a year time resulting in a strong private botnet of at least 30.000 infections targeting over 200 banking apps with a private version of Anubis that is obfuscated with a commercial tool (DexProtector).

Most of the new strains of these malware families are now raging as privately run projects, switching from the very loud MaaS (Malware as a Service) trend that we observed last year. This could also be a result of the increasing success/attempts of law enforcement efforts to catch and punish the people behind these threats and boldly disturbing underground forms.

More than ever, a clear overview and understanding of the mobile banking threat landscape is crucial for mobile payments, and tools to detect the attack behavior such as ATS from Android banking malware on devices have become invaluable to avoid fraud.

Appendix

IOCs

Name	SHA256 Hash
Anubis QRcode Dropper	998ba967bb23e6324c8f689ca0e1b5f28434d1ffdd52eac751f0649f037328c1
Anubis.C	617f3969267477d9c50e089139ea7627f1916259fc9b8c5028e2257a7ab7077a
Anatsa.A	a20f6c19ef20213df5b8e277d21dd70fe1cf99215ab42c39d69cce2396e72972
Medusa.B	05c8fc94e6f08bb0600fe7d8177a17ad65f01ec34fe749ea4981994dd890b1c8
Gustuff.C	3d196d954a2ea68c5ea65901fb7905b4773ead3fdb6967400beb370580e6f4a5

Capabilities

Medusa.B

Name	Description
Clipboard interaction	The malware can extract data from or insert data into the device's clipboard
App auto-start at device boot	The malware starts automatically when the device is turned on or restarted
App termination	The malware can terminate apps
Preventing removal	The malware can prevent it's removal
Hiding the app icon	The malware can hide it's icon from the application drawer
Screen streaming	The malware can stream what is displayed on the device's screen
Alerts	The malware can issue Android alerts with arbitrary text

Name	Description
Push notifications	The malware can show push notifications
Screen locking	The malware can lock the screen of the infected device
(Partial) Automated Transfer System	The malware uses a AccessibilityService to control the infected device and perform automated payments using the targeted banking apps (still requires interaction from the C2 to initiate the process)
Web pages	The malware can show arbitrary web pages on the infected device
App removal	The malware can remove applications
App starting	The malware can start applications
App installing	The malware can install applications
SMS spamming	The malware can perform SMS spam campaigns
SMS sending	The malware can send SMS messages
Target list update	Actors can configure targets for overlay phishing attack dynamically
Application listing	The malware can access list of all installed applications and send it to the C2
Contact list collection	The malware can read the contact list of the infected device and send it to the C2
Device info collection	The malware can access device related information(SIM, build info, settings) and send it to the C2
Accessibility event logging	The malware uses a AccessibilityService to get a stream of events happening on the device and send it to the C2
SMS forwarding	The malware can forward all incoming SMS messages to a phone number controlled by actors
SMS listing	The malware can access the content of SMS messages and send it to the C2
Keylogging	The malware can log victim's keystrokes and send them to the C2

Name	Description
Dynamic overlaying	The malware can show phishing screens to steal information. Phishing screens are retrieved from the C2

Anatsa.A

Name	Description
Clipboard interaction	The malware can extract data from or insert data into the device's clipboard
App auto-start at device boot	The malware starts automatically when the device is turned on or restarted
App termination	The malware can terminate apps
Preventing removal	The malware can prevent it's removal
Hiding the app icon	The malware can hide it's icon from the application drawer
Files/pictures collection	The malware can access the file system of the infected device and upload it's content to the C2
Alerts	The malware can issue Android alerts with arbitrary text
Push notifications	The malware can show push notifications
Screen locking	The malware can lock the screen of the infected device
(Partial) Automated Transfer System	The malware uses a AccessibilityService to control the infected device and perform automated payments using the targeted banking apps (still requires interaction from the C2 to initiate the process)
Web pages	The malware can show arbitrary web pages on the infected device
App removal	The malware can remove applications
App starting	The malware can start applications
App installing	The malware can install applications

Name	Description
SMS spamming	The malware can perform SMS spam campaigns
SMS sending	The malware can send SMS messages
Target list update	Actors can configure targets for overlay phishing attack dynamically
Application listing	The malware can access list of all installed applications and send it to the C2
Contact list collection	The malware can read the contact list of the infected device and send it to the C2
Device info collection	The malware can access device related information(SIM, build info, settings) and send it to the C2
Accessibility event logging	The malware uses a AccessibilityService to get a stream of events happening on the device and send it to the C2
SMS forwarding	The malware can forward all incoming SMS messages to a phone number controlled by actors
SMS listing	The malware can access the content of SMS messages and send it to the C2
Keylogging	The malware can log victim's keystrokes and send them to the C2
Dynamic overlaying	The malware can show phishing screens to steal information. Phishing screens are retrieved from the C2

Gustuff.C

Name	Description
Clipboard interaction	The malware can extract data from or insert data into the device's clipboard
App auto-start at device boot	The malware starts automatically when the device is turned on or restarted
Updatable	The malware can update itself
Emulation detection	The malware can detect whether or not it is running on the real device

Name	Description
App termination	The malware can terminate apps
Preventing removal	The malware can prevent it's removal
Hiding the app icon	The malware can hide it's icon from the application drawer
SMS C2	The malware is able to receive commands using incoming text messages
C2 update primary-channel	The malware can update the C2 using a new value/list of values received from the original C2
Alerts	The malware can issue Android alerts with arbitrary text
Push notifications	The malware can show push notifications
Screen locking	The malware can lock the screen of the infected device
Automated Transfer System	The malware uses a AccessibilityService to control the infected device and perform automated payments using the targeted banking apps
Web pages	The malware can show arbitrary web pages on the infected device
App removal	The malware can remove applications
App starting	The malware can start applications
App installing	The malware can install applications
SMS spamming	The malware can perform SMS spam campaigns
SMS sending	The malware can send SMS messages
Target list update	Actors can configure targets for overlay phishing attack dynamically
Files/pictures collection	The malware can access the file system of the infected device and upload it's content to the C2
Application listing	The malware can access list of all installed applications and send it to the C2

Name	Description
Contact list collection	The malware can read the contact list of the infected device and send it to the C2
Device info collection	The malware can access device related information(SIM, build info, settings) and send it to the C2
Accessibility event logging	The malware uses a AccessibilityService to get a stream of events happening on the device and send it to the C2
SMS forwarding	The malware can forward all incoming SMS messages to a phone number controlled by actors
SMS listing	The malware can access the content of SMS messages and send it to the C2
Keylogging	The malware can log victim's keystrokes and send them to the C2
Dynamic overlaying	The malware can show phishing screens to steal information. Phishing screens are retrieved from the C2

Targets

Anatsa.A

Package name

com.db.pwcc.dbmobile

com.db.pbc.miabanca

de.fiducia.smartphone.android.banking.vr

es.ibercaja.ibercajaapp

com.bbva.bbvacontigo

com.mobileloft.alpha.droid

de.commerzbanking.mobil

com.cajasur.android

net.inverline.bancosabadell.officelocator.android

es.lacaixa.mobile.android.newwapicon

com.rsi

Package name

eu.unicreditgroup.hvbapptan

com.binance.dev

es.bancosantander.apps

de.sdv rz.ihb.mobile.secureapp.sparda.produktion

piuk.blockchain.android

de.postbank.finanzassistent

es.openbank.mobile

es.cm.android

es.liberbank.cajasturapp

de.ingdiba.bankingapp

es.univia.unicajamovil

com.grupocajamar.wefferent

de.santander.presentation

de.comdirect.android

app.wizink.es

com.coinbase.android

com.starfinanz.smob.android.sfinanzstatus

com.kutxabank.android

vivid.money

de.traktorpool

www.ingdirect.nativeframe

Gustuff.C

Package name

au.com.bankwest.mobile

au.com.ingdirect.android

Package name

au.com.nab.mobile

au.com.suncorp.SuncorpBank

au.com.ubank.internetbanking

bcc.org.freewallet.app

bcn.org.freewallet.app

btc.org.freewallet.app

btg.org.freewallet.app

o.edgesecure.app

com.airbitz

com.android.vending

com.anz.android

com.anz.android.gomoney

com.arcbit.arcbit

com.barclays.android.barclaysmobilebanking

com.barclays.bca

com.bitcoin.mwallet

com.bitcoin.wallet

com.bitpay.copay

com.bitpay.wallet

com.bitpie

com.btcontract.wallet

com.circle.android

com.citibank.mobile.au

com.coinbase.android

com.coincorner.app.crypt

Package name

com.coinspace.app

com.commbank.netbank

com.cooperativebank.bank

com.grppl.android.shell.BOS

com.grppl.android.shell.CMBllloydsTSB73

com.grppl.android.shell.halifax

com.hashengineering.bitcoincash.wallet

com.kibou.bitcoin

com.kryptokit.jaxx

com.lloydsbank.businessmobile

com.moneybookers.skrillpayments

com.monitise.client.android.yorkshire

com.nearform.ptsb

com.plutus.wallet

com.qcan.mobile.bitcoin.wallet

com.rbs.mobile.android.natwest

com.rbs.mobile.android.rbs

com.westernunion.android.mtapp

com.wirex

com.xapo

de.schildbach.wallet_test

distributedlab.wallet

eth.org.freewallet.app

It.spectrofinance.spectrocoin.android.wallet

me.cryptopay.android

Package name

net.bither

org.banksa.bankß

org.bom.bank

org.electrum.electrum

org.stgeorge.bank

org.vikulin.etherwallet

org.westpac.bank

piuk.blockchain.android

tsb.mobilebanking

uk.co.hsbc.hsbcukbusinessbanking

uk.co.hsbc.hsbcukmobilebanking

uk.co.mbna.cardservices.android

uk.co.metrobankonline.mobile.android.production

uk.co.santander.businessUK.bb

uk.co.santander.santanderUK

uk.co.tescomobile.android

uk.co.tsb.newmobilebank