BugDrop: the first malware trying to circumvent Google's security Controls

threatfabric.com/blogs/bugdrop-new-dropper-bypassing-google-security-measures.html

research

16 August 2022

Intro

Between 2020 and 2021, the Android malware epidemic took over the banking threat landscape. Families like Anatsa and Cabassous spread to thousands of victims by international SMiShing campaigns, coupled with various kinds of web phishing pages to trick users into downloading malicious APKs.

In 2022, this trend was confirmed, with a heavy switch in distribution techniques in favour of **droppers**, Android applications whose sole purpose is to bypass security measures used in official market places like **Google Play Store**, and deploy they payload, which is usually an Android banking trojan.

Android Droppers are becoming the most reliable and preferred way to deploy malware on victim's devices. Google has taken action to restrict the amount of privileges that sideloaded applications can obtain starting from Android 13, but, as we covered in our last blog, this solution seems to not be enough to stop criminals from downloading and enabling all the necessary privileges and permissions on their malware.

Recently, ThreatFabric's predictions became true, when we discovered an in-development dropper, which we named **BugDrop**, that criminals have been working on to **circumvent this security feature** that will become part of the next release of the mobile OS from Mountain View.

Once finished, this product would become another weapon in criminal's already dangerous arsenal, potentially rendering Google's solution obsolete even before its deployment.

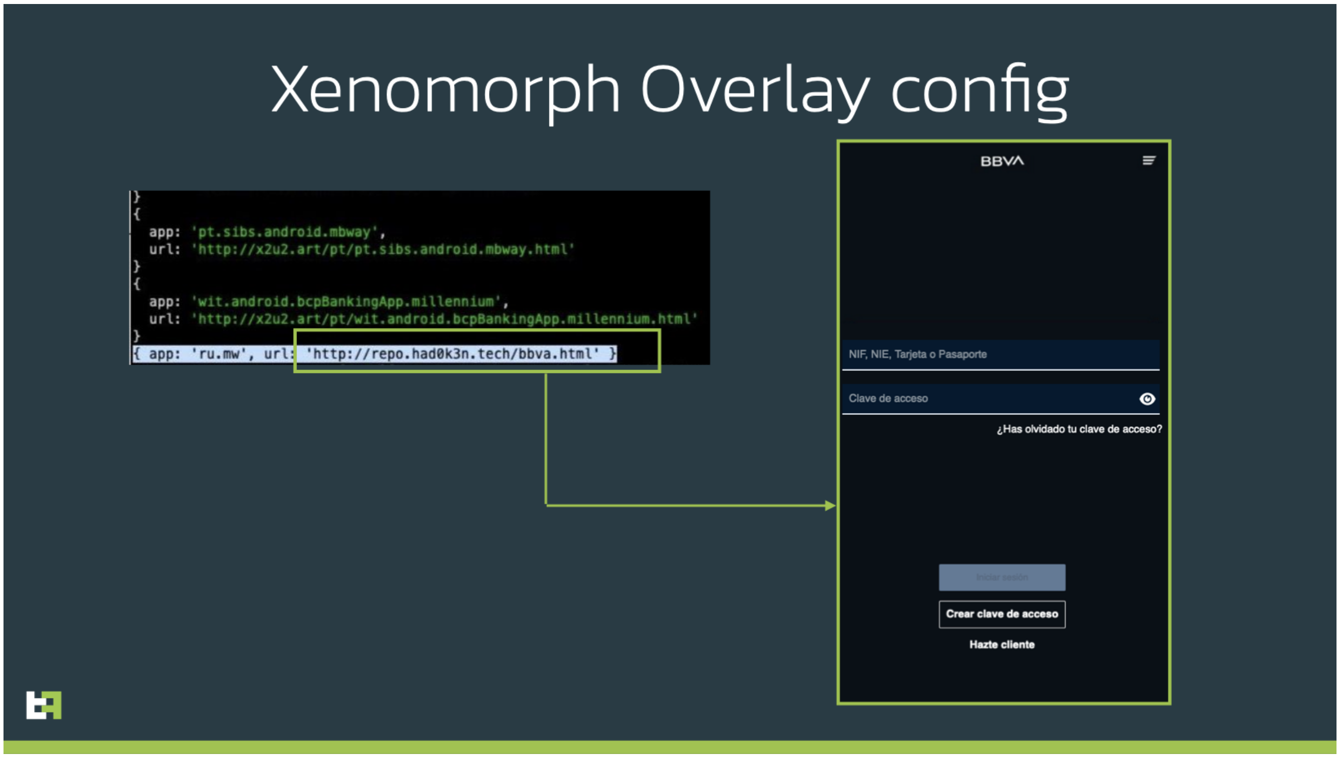
Context

Recently, in their daily monitoring of Android malware activity, our researchers noticed something unusual in the latest sample of the malware family Xenomorph.

This malware family has been resurfacing in the last couple of months with a few campaigns, mostly appearing on Google Play Store. These new campaigns feature a new and improved version of Xenomorph, which added RAT capabilities thanks to the addition of a handful of, as they are named in the code itself, “Runtime modules”. These modules give the malware the capability of performing gestures, touches, and much more on an infected device.

What sparked our researchers’ curiosity was the presence, among the targets returned by the C2, of a Russian application. Usually it is very rare for banking malware to target CIS countries, mostly due some sort of twisted code of honor among criminals from that region. For this reason ThreatFabric decided to investigate further.

In the case of Xenomorph, after installation, the bot will always request overlays from the C2, which will send back an encrypted JSON configuration, with the URLs where the overlays are hosted. Below you can see the decrypted result returned from the server, together with the overlay itself.



Xenomorph Overlay config

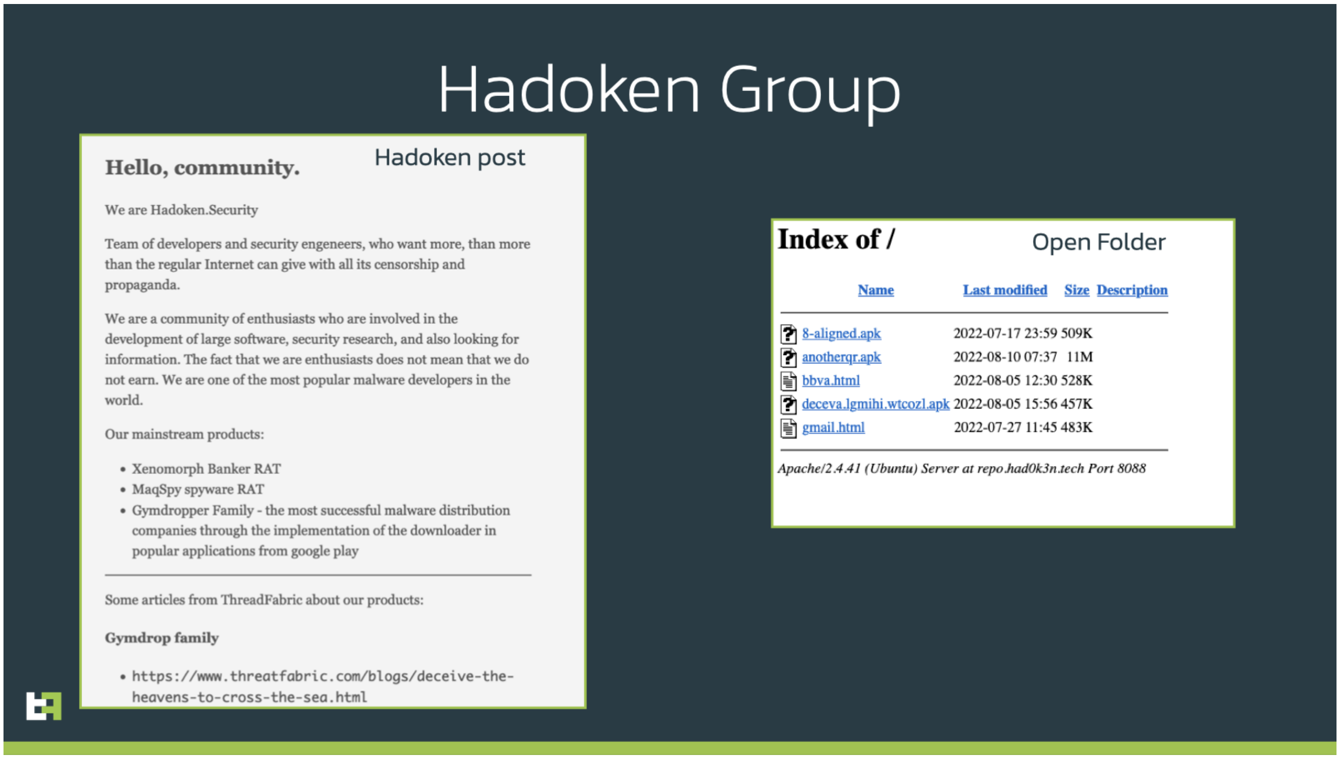
```
}
{
  app: 'pt.sibs.android.mbway',
  url: 'http://x2u2.art/pt/pt.sibs.android.mbway.html'
}
{
  app: 'wit.android.bcpBankingApp.millennium',
  url: 'http://x2u2.art/pt/wit.android.bcpBankingApp.millennium.html'
}
{ app: 'ru.mw', url: 'http://repo.had0k3n.tech/bbva.html' }
```

As you can see, the highlighted overlay is the one corresponding to the Russian organization. However, the overlay itself corresponds to a famous Spanish institution. What was very interesting and started a more detailed investigation, was the fact that the server hosting this specific overlay was different from the one hosting all the others. We simply tried to connect to the server, and were welcomed with a nice open folder.

The “Hadoken Security” Group

Before analyzing the content of this folder, it is worth mentioning where it is hosted. The main domain redirects to the following page, containing a post, apparently posted on the 31st of May 2022.

In this post, which seems to be a self-advertisement message, the “**Hadoken Security**” group claims the ownership of multiple malware families, including the Android Banking trojan Xenomorph and the Dropper Gymdrop, and unfortunately even uses **ThreatFabric own blogs** as reference:



Hadoken Group

Hadoken post

Hello, community.

We are Hadoken.Security

Team of developers and security engeneers, who want more, than more than the regular Internet can give with all its censorship and propaganda.

We are a community of enthusiasts who are involved in the development of large software, security research, and also looking for information. The fact that we are enthusiasts does not mean that we do not earn. We are one of the most popular malware developers in the world.

Our mainstream products:

- Xenomorph Banker RAT
- MaqSpy spyware RAT
- Gymdropper Family - the most successful malware distribution companies through the implementation of the downloader in popular applications from google play

Some articles from ThreadFabric about our products:

Gymdrop family

- `https://www.threatfabric.com/blogs/deceive-the-heavens-to-cross-the-sea.html`

Open Folder

Index of /

Name	Last modified	Size	Description
8-aligned.apk	2022-07-17 23:59	509K	
anotherqr.apk	2022-08-10 07:37	11M	
bbva.html	2022-08-05 12:30	528K	
deceva.lgmihi.wtcozl.apk	2022-08-05 15:56	457K	
gmail.html	2022-07-27 11:45	483K	

Apache/2.4.41 (Ubuntu) Server at repo.had0k3n.tech Port 8088

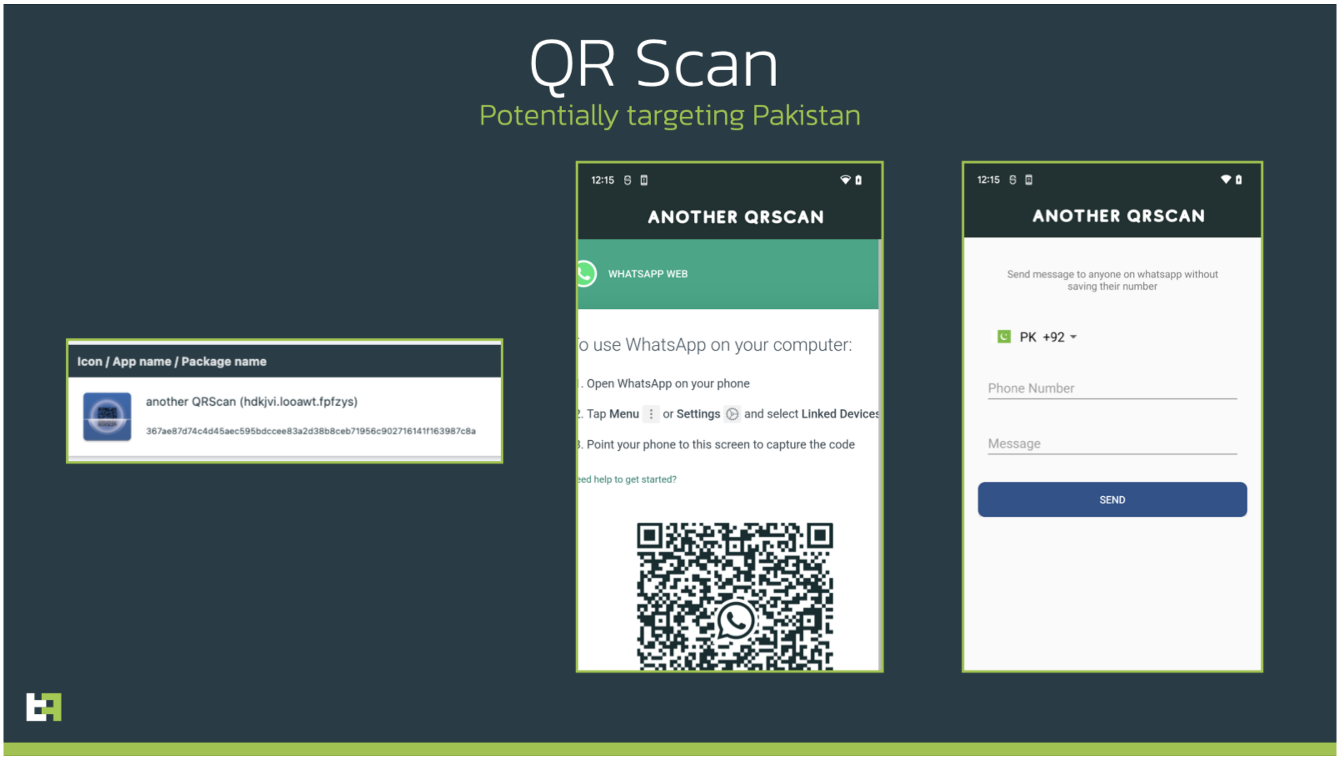
Coming back to the investigation, the open folder contained a few different interesting files. In addition to the previously mentioned Spanish overlay, and another overlay targeting Gmail, three APKs were available for download. Two of these were different Xenomorph samples belonging to two different campaigns (the “Task Scheduler” campaign and the “Android Settings” campaign).

However, the third APK sample, which poses as a QR code reader, looked different from the others. Based on Xenomorph known Modus Operandi, our best guess was for this to be a dropper, likely of the Gymdrop family, which has been previously associated with this Banking malware and to the Threat Actor group behind it.

BugDrop: A Dropper Trying to Bypass Google Security

The application poses as a QR code reader, much like many other droppers that ThreatFabric has seen and reported over the past couple of years.

It is interesting to see that in one of the fake activity used by the dropper, specifically the one that should be used to send messages via the social messaging app WhatsApp, the default country code is set up to be +92, corresponding to **Pakistan**. This information could give an indication of the possible target area for the future of this dropper, but currently we do not possess enough information to substantiate this claim.



QR Scan

Potentially targeting Pakistan

Icon / App name / Package name
another QRScan (hdkjvi.looawt.fpfzys) 367ae87d74c4d45aec595bdccee83a2d38b8ceb71956c902716141f163987c8a

o use WhatsApp on your computer:

. Open WhatsApp on your phone

. Tap **Menu** ⋮ or **Settings** and select **Linked Devices**

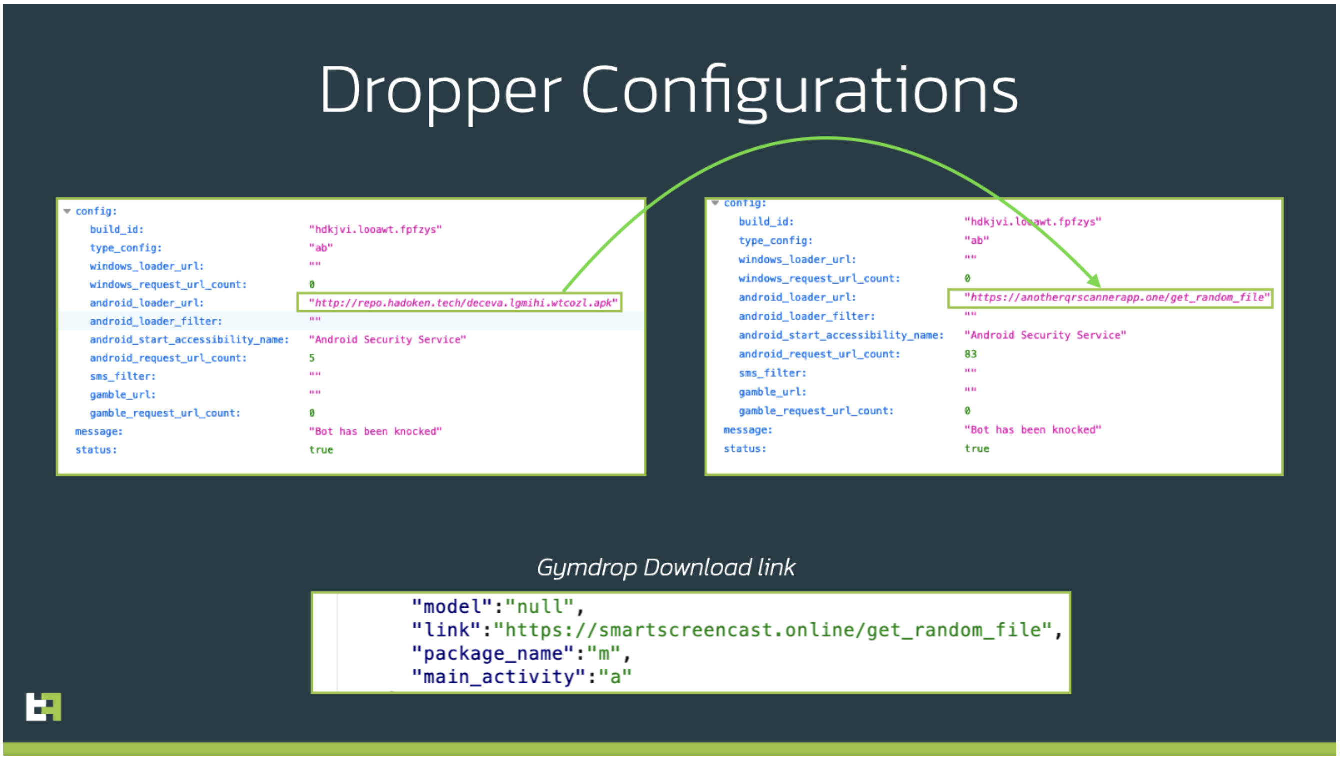
. Point your phone to this screen to capture the code

ed help to get started?

Send message to anyone on whatsapp without saving their number

Once started, the application immediately requests the **Accessibility Services** access to the user. This is already a red flag, as this kind of services grant applications the ability to perform gestures and touches in the user’s stead. No QR code reader should require these kind of priviliges. In addition, it bars out the possibility of this sample being part of the Gymdrop family, as this dropper family does not rely on Accessibility Services to install malware on victim’s devices.

Once granted, while showing a loading screen, the dropper initiates a connection with its **onion.ws** C2, which relies on the **TOR protocol**, obtaining back its configuration and the URL of the payload to download and install. Throughout the course of our investigation, this URL changed from being one of the samples in the open folder, to an external URL again referring to QR code scanners functionalities, which used a endpoint very similar to what was used by Gymdrop samples that we observed in the wild in the last few months.



Dropper Configurations

```
config:
  build_id:                           "hdkjvi.looawt.fpfzys"
  type_config:                        "ab"
  windows_loader_url:                 ""
  windows_request_url_count:          0
  android_loader_url:                 "http://repo.hadoken.tech/deceva.lgmihi.wtcozl.apk"
  android_loader_filter:              ""
  android_start_accessibility_name:   "Android Security Service"
  android_request_url_count:          5
  sms_filter:                         ""
  gamble_url:                         ""
  gamble_request_url_count:           0
message:                              "Bot has been knocked"
status:                               true
```

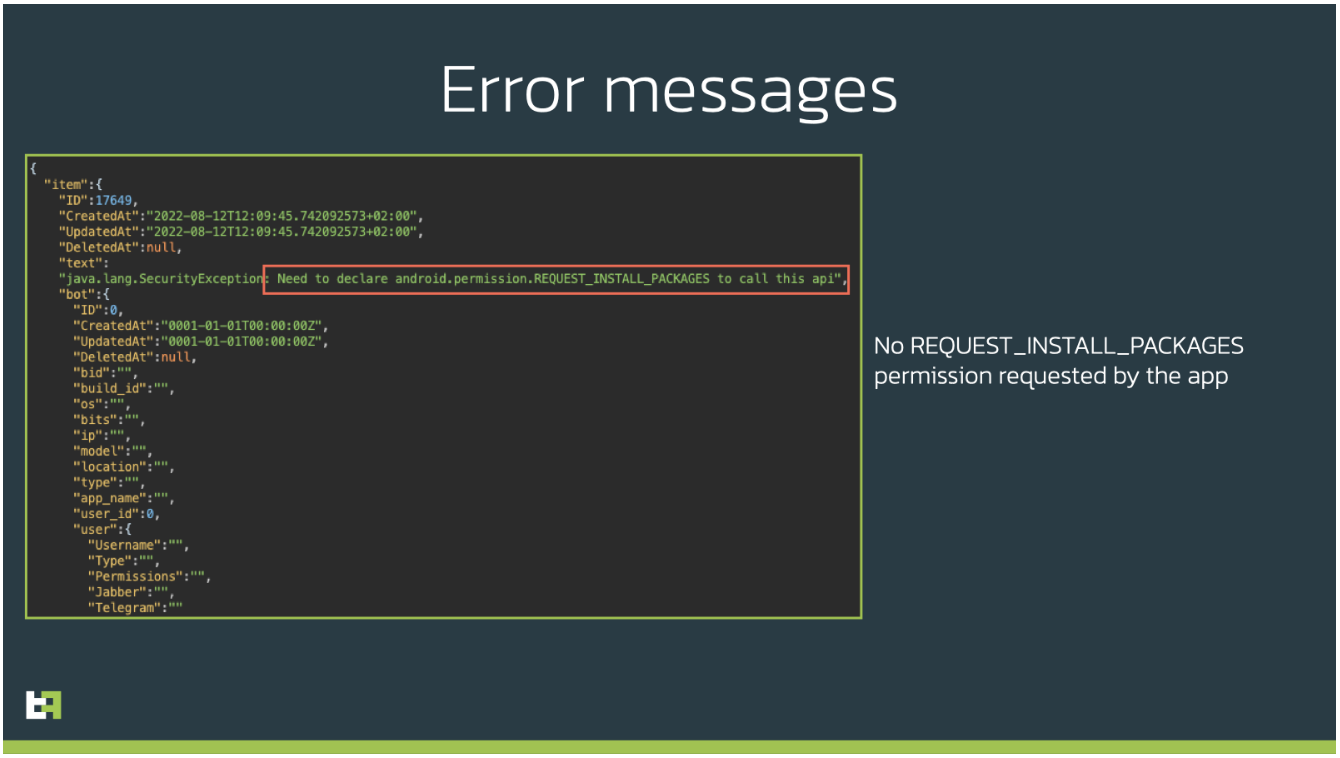
```
config:
  build_id:                           "hdkjvi.looawt.fpfzys"
  type_config:                        "ab"
  windows_loader_url:                 ""
  windows_request_url_count:          0
  android_loader_url:                 "https://anotherqrscannerapp.one/get_random_file"
  android_loader_filter:              ""
  android_start_accessibility_name:   "Android Security Service"
  android_request_url_count:          83
  sms_filter:                         ""
  gamble_url:                         ""
  gamble_request_url_count:           0
message:                              "Bot has been knocked"
status:                               true
```

Gymdrop Download link

```
"model":"null",
"link":"https://smartscreencast.online/get_random_file",
"package_name":"m",
"main_activity":"a"
```

The file downloaded in both cases was still belonging to the Xenomorph family.

This is where it becomes clear that this dropper and the actors behind it are still in deep development phase. In the image below you can see the error message sent back to the C2 by the bot:



Error messages

```
{
  "item":{
    "ID":17649,
    "CreatedAt":"2022-08-12T12:09:45.742092573+02:00",
    "UpdatedAt":"2022-08-12T12:09:45.742092573+02:00",
    "DeletedAt":null,
    "text":
    "java.lang.SecurityException: Need to declare android.permission.REQUEST_INSTALL_PACKAGES to call this api",
    "bot":{
      "ID":0,
      "CreatedAt":"0001-01-01T00:00:00Z",
      "UpdatedAt":"0001-01-01T00:00:00Z",
      "DeletedAt":null,
      "bid":"",
      "build_id":"",
      "os":"",
      "bits":"",
      "ip":"",
      "model":"",
      "location":"",
      "type":"",
      "app_name":"",
      "user_id":0,
      "user":{
        "Username":"",
        "Type":"",
        "Permissions":"",
        "Jabber":"",
        "Telegram":""
```

No REQUEST_INSTALL_PACKAGES permission requested by the app

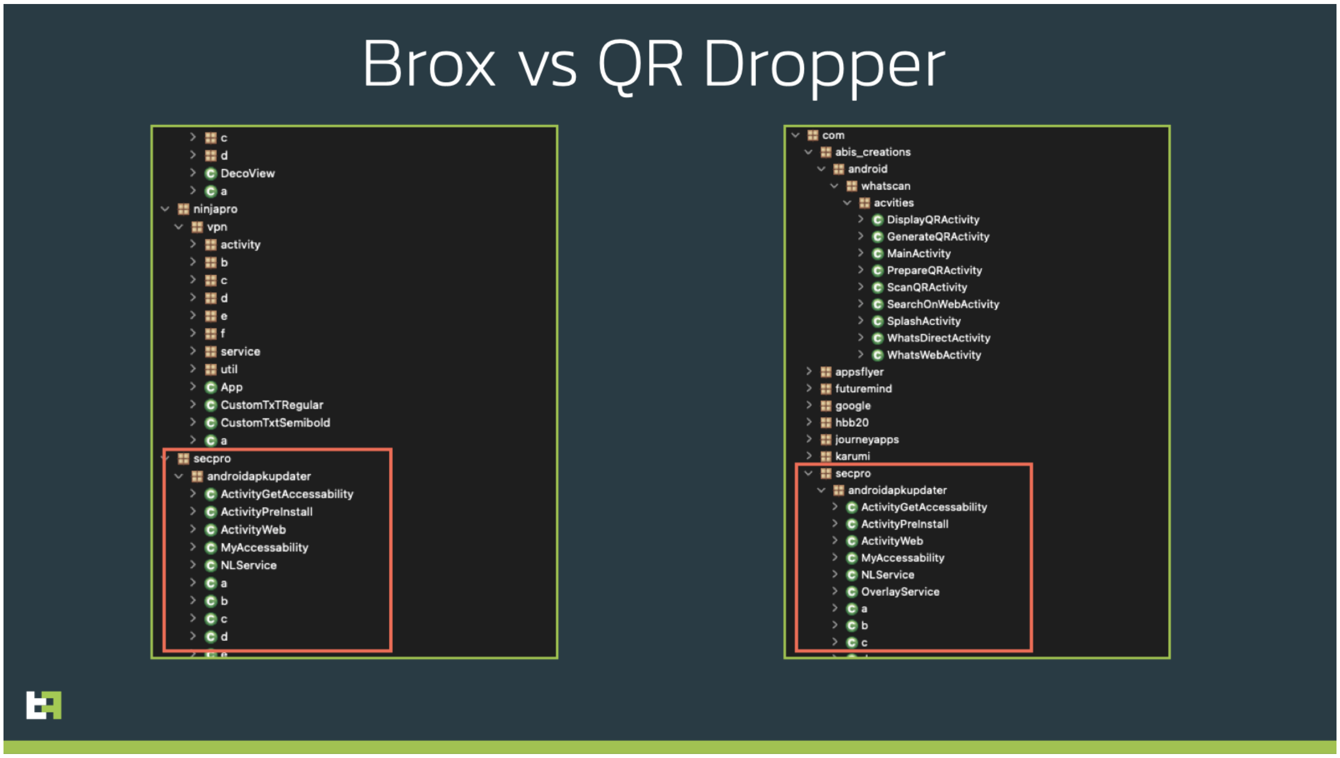
The clear issue here is that the APK is not requesting the **"REQUEST_INSTALL_PACKAGES"** permission to the OS, without which it is impossible for a application to install anything on the device (even with accessibility services privileges), as it is even pointed out by the error message sent back to the Onion C2 by the bot.

A few hours after the discovery of the open folder, criminals noticed the issue, also due to a tweet thread that drew the attention of a few researchers, and blocked the access to the data.

A page out of Brox tutorial manual

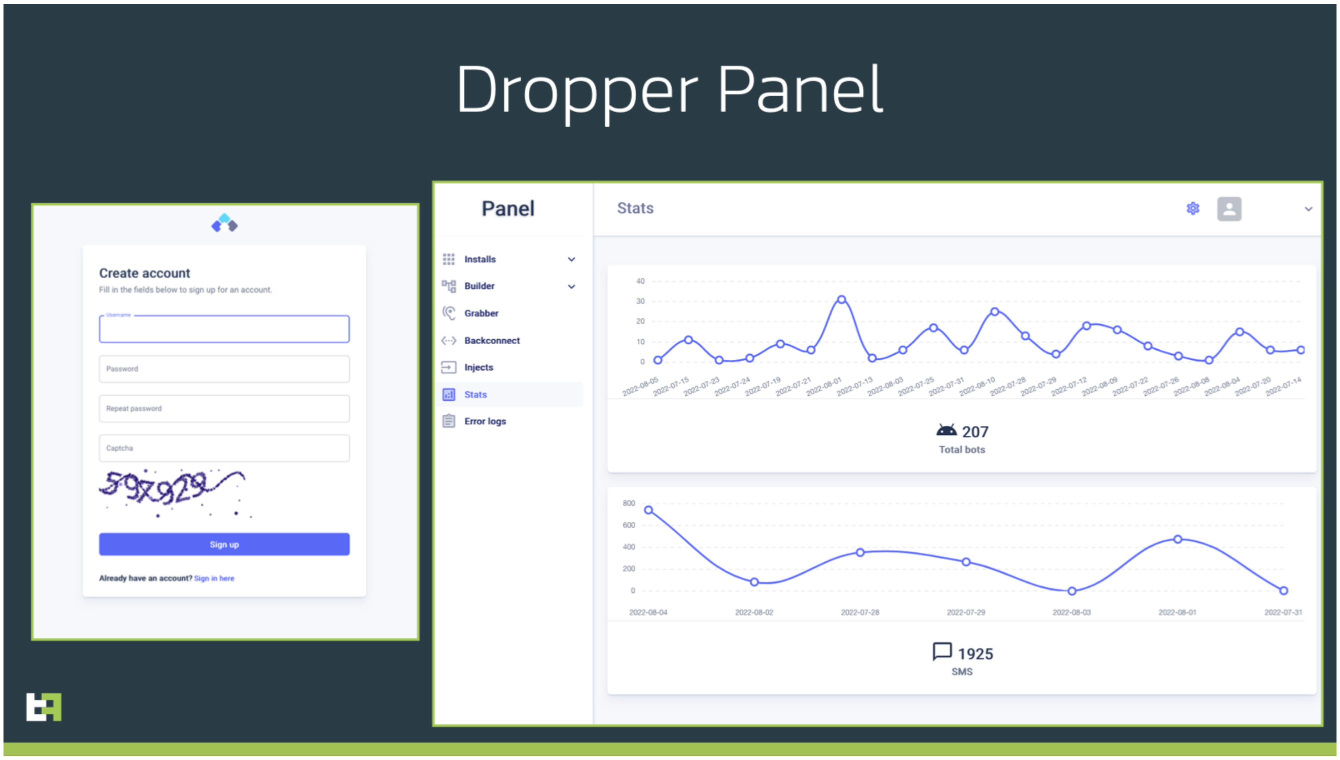
As it is clear, this dropper-in-development is a new product from the actors behind Xenomorph. Most of its malicious behaviour is contained in a package named **com.secpro.androidapkupdater**. If this activity name rings a bell, it is because it is borrowed from as a lesser known malware, named **Brox**. Brox (also known as MasterFred) is a Android malware family that was created and distributed on hacking forums as a mean to teach malware development to criminals. The actors behind it offered a full paid course to teach the basics of malware development, including access to panels and support in the management and abuse of the botnet. This malware family briefly resurfaced towards the end of 2021, where it was also named MasterFred, but with just a handful of samples.

This QR code reader seems to be a slightly modified version of the original Brox code, with a just a slightly tweaked communication protocol.



Brox vs QR Dropper

In this case, during the setup, criminals forgot to hide the sign-up page to the **panel**, which, in good tutorial fashion, does not have any specific requirement for access, which allowed us to have a nice view of the panel itself, together with some information on the amount of infections for this in-development dropper. We assume that this number is mostly made of testing devices (both from criminals as well as researchers alike).



It is possible that this specific panel is simply automatically set-up from the tutorial code of Brox, as there does not seem to be much activity or capabilities enabled from this panel.

This malware family was known previously as an overlaying Android malware, but had always the capability of installing other APKs on the device. It seems like this is the primary purpose of the malware when deploying Xenomorph, while relying on a C2 server structure very similar to the one used by Gymdrop, which is as discussed another product of the Hadoken group.

Currently we do not see any banking malware activity coming from these samples, despite the fact that they do feature code capable of performing overlay attacks as well.

A brand new technique to bypass Google's Security Measures

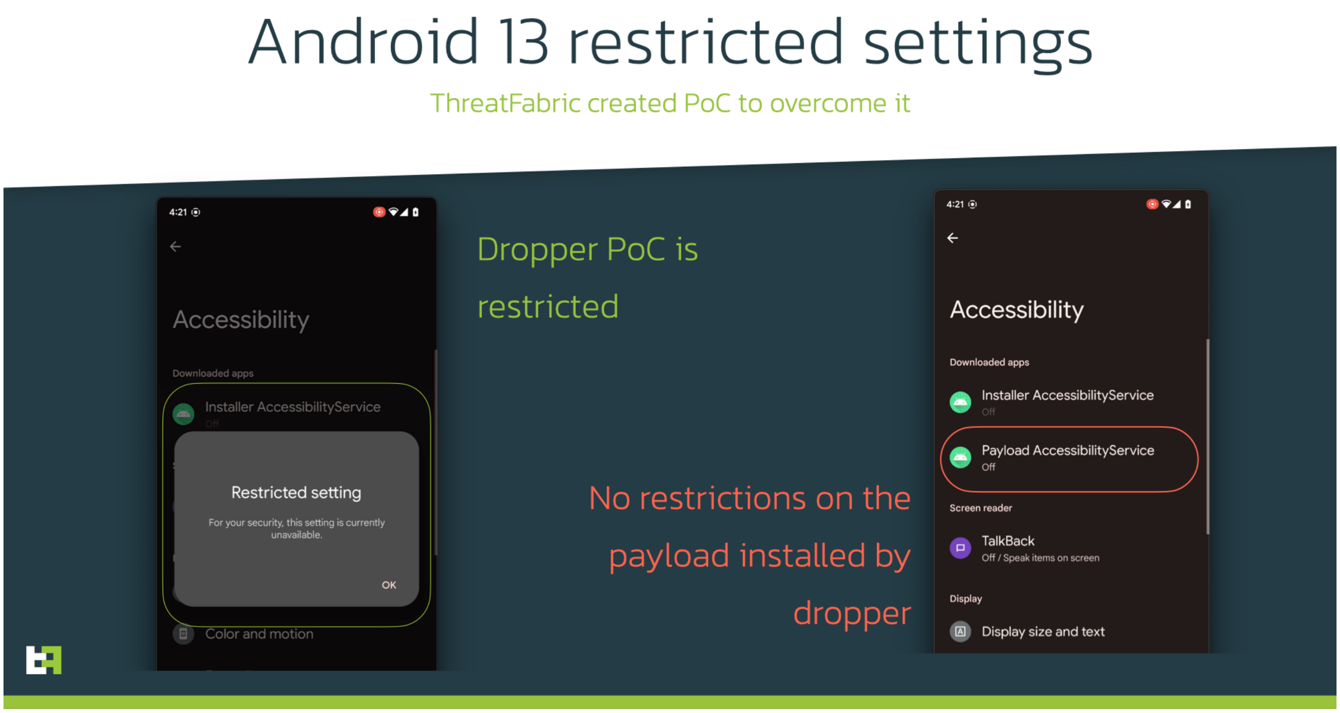
The most interesting fact about this new Brox sample is the methodology used to install the APK downloaded from the server. The code is not fully functional, and some of the references seem to be missing, but one string in the installer function stands out among others. You can find the mentioned function in the following code snippet:

```
.method public constructor <init>(Context, String, c)V
          .registers 6
invoke-direct       Object-><init>()V, p0
const-string        v0, "com.example.android.apis.content.SESSION_API_PACKAGE_INSTALLED"
invoke-static       h->a(String)String, v0
const-string        v0, "~~~"
const-string        v1, "4.Update"
invoke-static       Log->d(String, String)I, v0, v1
iput-object         p1, p0, g->a:Context
iput-object         p2, p0, g->b:String
iput-object         p3, p0, g->c:c
new-instance        p1, StringBuilder
invoke-direct       StringBuilder-><init>()V, p1
const-string        p3, "using url "
invoke-virtual      StringBuilder->append(String)StringBuilder, p1, p3
invoke-virtual      StringBuilder->append(String)StringBuilder, p1, p2
invoke-virtual      StringBuilder->toString()String, p1
move-result-object  p1
invoke-static       Log->d(String, String)I, v0, p1
new-instance        p1, g$b
const/4             p2, 0
invoke-direct       g$b-><init>(g, g$a)V, p1, p0, p2
const/4             p2, 0
new-array           p2, p2, [Void
invoke-virtual      AsyncTask->execute([Object)AsyncTask, p1, p2
return-void
```

What drew our attention is the presence in the Smali code of the string **“com.example.android.apis.content.SESSION_API_PACKAGE_INSTALLED”**. This string, which is not present in the original Brox code, corresponds to the action required by intents to create an installation process by session.

In this context, it is important to remind the new security features of Android 13, which will be released in fall of 2022. With this new release, Google introduced the **“restricted setting”** feauture, which blocks sideloaded applications from requesting Accessibility Services privileges, limiting this kind of request to applications installed with a session-based API (which is the method usually used by app stores).

With this in mind, it is clear what criminals are trying to achieve. What is likely happening is that actors are using an already built malware, capable of installing new APKs on an infected device, to test a session based installation method, which would then later be incorporated in a more elaborate and refined dropper.



Android 13 restricted settings

ThreatFabric created PoC to overcome it

Dropper PoC is restricted

No restrictions on the payload installed by dropper

4:21
Accessibility
Downloaded apps
Installer AccessibilityService
Off
Restricted setting
For your security, this setting is currently unavailable.
OK
Color and motion
4:21
Accessibility
Downloaded apps
Installer AccessibilityService
Off
Payload AccessibilityService
Off
Screen reader
TalkBack
Off / Speak items on screen
Display
Display size and text

This is very dangerous, and in line with what ThreatFabric predicted in our previous blog “2022 Mobile Threat Landscape update”. When fully implemented, this slight modification would circumvent fully Googles new security measures, even before they are effectively in place.

Considering that the Hadoken Group has very kindly adopted our naming for all of their products, we decided to name this dropper “**BugDrop**”, in honor of all the issues that are present in its codebase.

Conclusion

The Hadoken group has been active since at least the end of 2021, starting with their dropper product, Gymdrop, and in early 2022 they introduced their first Android banking malware, Xenomorph. Both these malware families have been proven to be high threats to banking institutions and banking customers alike, the first being used by multiple malware families as a mean of distribution, while the second being a very advanced Banking trojan with On-Device Fraud capabilities.

With the completion and resolution of all the issues currently present in BugDrop, criminals will have another efficient weapon in the war against security teams and banking institutions, defeating solutions that are currently being adopted by Google, which are clearly not sufficient to deter criminals.

ThreatFabric expects the Hadoken group to continue working on this dropper family, or more generally to develop and start distributing a dropper family abusing the Session driven installation, to bypass Google new Security features.

ThreatFabric Fraud Risk Suite

With our Fraud Risk Suite we are helping financial institutions to gain visibility on fraud attempts by mobile (banking) malware. If you would like to know more about how we use our fraud detection SDK to detect any type of fraud on mobile devices, feel free to reach out to sales@threatfabric.com.

Appendix

Xenomorph Samples

App name	Package name	SHA-256
Android Security Service	deceva.lgmihi.wtcozl	ab345951a3e673aec99f80d39fa8f9cdb0d1ac07e0322dae3497c237f7b37277
Task Scheduler	wyrkpv.slyffg.berykl	65c655663b9bd756864591a605ab935e52e5295735cb8d31d16e1a6bc2c19c28

Gymdrop Samples

App name	Package name	SHA-256
Gym and Fitness Trainer	com.gym.trainer.jeux	30ee6f4ea71958c2b8d3c98a73408979f8179159acccc01b6fd53ccb20579b6b
Document Scanner	com.portus.docscan	3484a3e8743d65510de60b7bc91ee87da57573e22294fc36f731b3e1096adf15

BugDrop Samples

App name	Package name	SHA-256
another QRScan	hdkjvi.looawt.fpfzys	214a576b46241bdf76bb4dbeacc7a456905eacd345fc515e0b38d6976c271168
another QRScan	hdkjvi.looawt.fpfzys	367ae87d74c4d45aec595bdccee83a2d38b8ceb71956c902716141f163987c8a

Brox Samples

App name	Package name	SHA-256
Master	master.com	1284d9e44fa5ac5b645c26c5e941cc392d77ab24ebfa91948688ce769ff71667
Test	com.test.com	8d9facf6319339cfaf0de3e2da5727bd25a933b34b5f0b0029459d6d7e22689a