

The Android Malware's Journey: From Google Play to banking fraud

By Francesco Iubatti, Alessandro Strino

Archived: 2026-04-06 00:27:08 UTC

Background and Key points

- In the last two months, we observed, through our telemetries, an increase in the number of **Vultur** infections among our customers.
- At the beginning of October 2022, the Cleafy Threat Intelligence Team discovered and reported to Google a **dropper of Vultur**, a known Android banking trojan, on the official **Play Store** with **100.000+ downloads**. Recently, other researchers publicly disclosed the same malicious dropper application.
- This dropper hides behind a fake utility application. Since its relatively small amount of permissions and **small footprint**, it appeared as a legitimate app and was able to elude the antivirus solutions and the Google Play analysis.
- After the installation phase, the dropper uses **advanced evasion techniques**, such as **steganography**, **file deletion**, and **code obfuscation**, in addition with multiple checks before the malware download.
- Once the banking trojan (Vultur) has been downloaded and installed through a fake update, Threat Actors (TAs) can observe everything that happens on the infected devices and carry out bank fraud through **ATO (Account Takeover) attacks**.

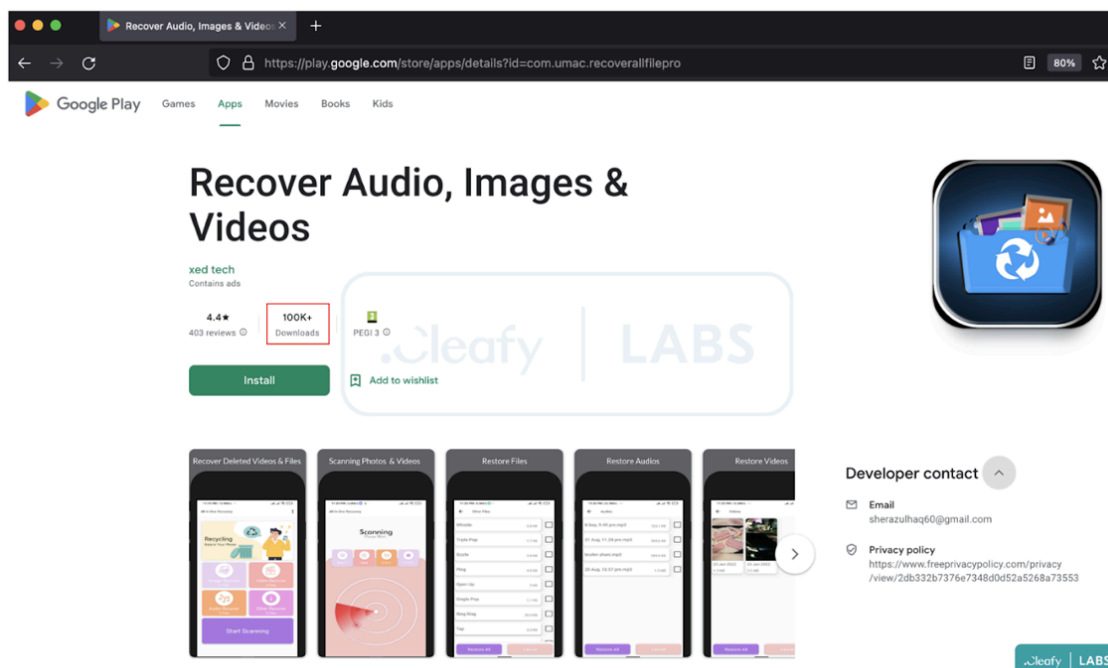


Figure 1 - Malicious dropper on Google Play Store

Analysis of the malicious dropper

During the last years, the number of Android banking trojans has increased, and new techniques to perform banking fraud have been developed. Although most of the banking trojans are distributed via *ishing campaigns, TAs also use official app stores to deliver their malware using **dropper applications**, namely an application designed to download malware into the target device.

One of the main reasons behind this choice is the possibility of reaching a more significant number of potential victims and, thus, a greater likelihood of completing fraud. Furthermore, since these droppers hide behind utility apps and come from a trusted source, they can mislead even “experienced” users.

The application, discovered in October by the Cleafy TIR team on the Google Play Store, appears like a legitimate recovery tool with a relatively small amount of permissions and a small footprint.

```
<manifest android:compileSdkVersion="30" android:compileSdkVersionCodename="11"
  <uses-sdk android:minSdkVersion="26" android:targetSdkVersion="30"/>
  <meta-data android:name="android.support.VERSION" android:value="25.3.1"/>
  <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.QUERY_ALL_PACKAGES"/>
  <uses-permission android:name="android.permission.INTERNET"/>
  <uses-permission android:name="android.permission.WAKE_LOCK"/>
  <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
  <uses-permission android:name="android.permission.REQUEST_INSTALL_PACKAGES"/>
  <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
  <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
  <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
  <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
  <uses-permission android:name="android.permission.FOREGROUND_SERVICE"/>
  <uses-permission android:name="com.android.vending.BILLING"/>
```

Figure 2 - Snippet of AndroidManifest file

The combination of these elements, plus the use of multiple evasion techniques, makes the application very difficult to detect with automatic sandboxes or machine learning methodology. It goes, then, undetected by antivirus solutions and Google Protects.

This explains why, even though an overview of this dropper was already described in the last article of Threat Fabric, we decided to publish this report and analyze in detail how this application ended up in the Play Store and attempted to commit bank fraud.

The application found on the Play Store belongs to the [Brunhilda dropper](#) service since it shares multiple behaviors with the same past related samples, such as:

- The request of Android 8.0 or above versions;
- Some code similarities;
- The information sent to the C2 server and his configuration.



Figure 3 - Antivirus detection of the Vultur’s dropper application

The application code has changed compared to previous variants and a piece of interesting evidence is the use of multiple evasion techniques, used to stay undetectable and slow down the analysis. Some techniques are listed below:

- **Steganography** (a technique used to hide secret data within an ordinary file to avoid detection): as shown in Figure 4, inside the application asset directory, there is a PNG file that hides an encrypted payload that, once decrypted, becomes a zip file that contains a dex file. The following dex file contains the code used to communicate with the C2 server and download the “real” malware on the user devices (currently, the Android banking trojan called Vultur).

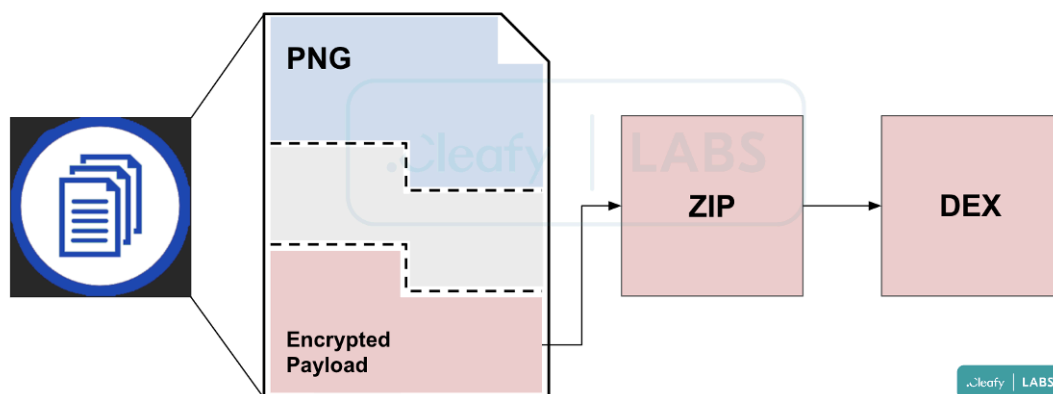


Figure 4 - Extraction of encrypted payload from PNG image

- **File Deletion:** After the payload is decrypted and uncompressed, both the zip and dex files are removed from their directories, as shown in Figure 5.
- **Code obfuscation and Anti-emulation:** All the code is obfuscated, and the strings are encrypted with the AES algorithm. Moreover, the dropper performs multiple checks to control if it is running on an emulator/sandbox device or a legitimate one to stop the attack or proceed.

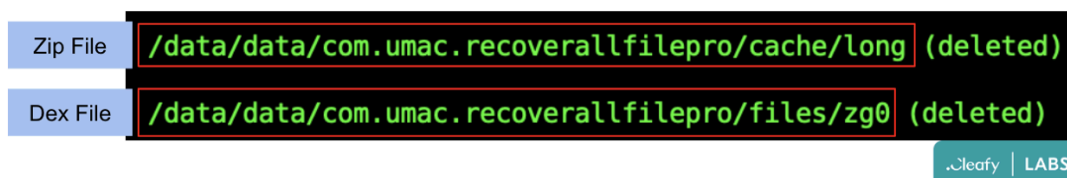


Figure 5 - Deletion of zip and dex files from memory

Once the victim downloads and installs the application from the Google Play Store to complete the attack chain, the dropper displays to the user a persistent update request to download a new application (Figure 6) that represents the actual malware, namely the Android banking trojan belonging to the Vultur family.

Although in that way, the user has to accept the Android permission to download and install the application from a different source than the official Google Store, this technique allows TAs to not upload the malicious application directly to the official store, making the dropper application undetectable.

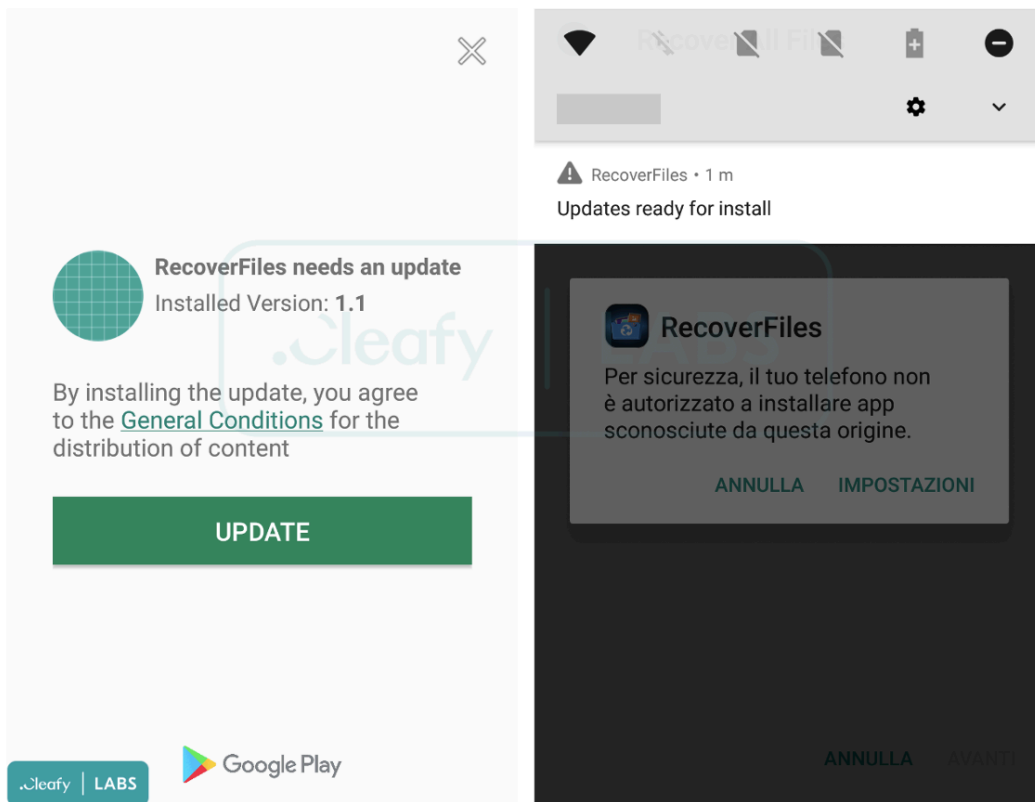


Figure 6 - Fake update requested to download the banking trojan

```
public static void b() {
    Context context0 = cy.e();
    File file0 = new File(context0.getFilesDir(), yr.e);
    Intent intent0 = new Intent();
    intent0.setAction("android.intent.action.INSTALL_PACKAGE");
    intent0.setFlags(1);
    intent0.setData(n.a(context0, context0.getApplicationContext().getPackageName() + ".provider", file0));
    intent0.addFlags(0x10800000);
    try {
        context0.getApplicationContext().startActivity(intent0);
    }
    catch(Exception unused_ex) {
    }
}

static {
    yr.a = "frappucinos.shop";
    yr.b = "p:state";
    yr.c = "APP-TOKEN";
    yr.d = "p:started";
    yr.e = "1.apk";
    yr.class.getPackage().getName();
    yr.f = "fall";
    yr.g = "vallue";
}
```

Figure 7 - Configuration file of the malicious dropper

Overview of Vultur

When the user installs the application requested in the fake update, a new popup appears to the user (Figure 8). Notably, the malware needs the “notorious” Accessibility Services to control the user’s device.

At this point, Vultur uses multiple techniques to try to remain unsuspected in the eyes of the user, in particular:

- A fake alert is displayed to the user, saying, “This update is incompatible with the current version of the Play Store and will be removed,” followed by a toast (a small Android popup) saying that the application has been removed;
- The icon of the downloaded application is not displayed on the phone;
- With the accessibility services, the malware does not allow the infected user to open the settings app to see which applications are installed on their device

Furthermore, if an analyst tries to install Vultur directly on a device, to analyze it, the malware does not start, and it does not communicate with the C2 server. The malware must be installed and launched through the dropper application.

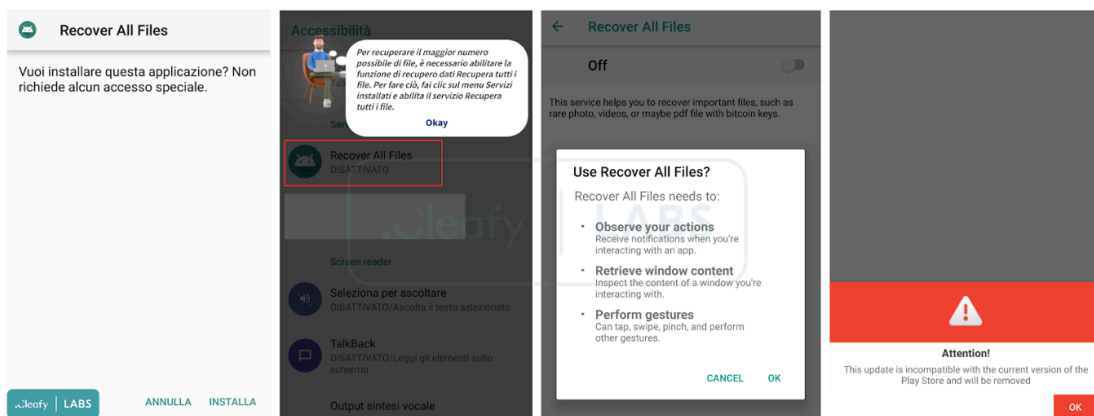


Figure 8 - Installation phases of Vultur

Using keylogging and screen recording capabilities, TAs can obtain all the information they need to carry out their fraudulent activities. During our investigations, we noticed that the usual modus operandi of Vultur’s TAs is to try to carry out bank fraud during the night hours.

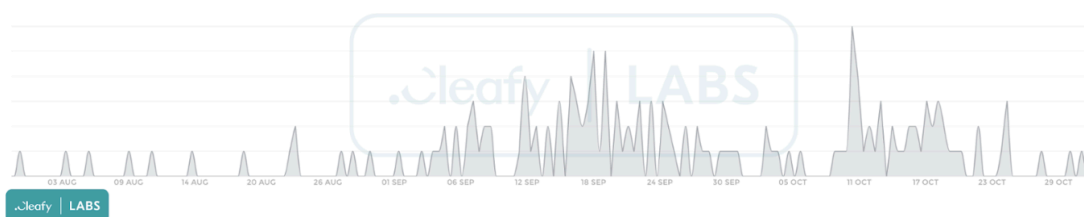


Figure 9 - Vultur infection during the months on our customers

Final Considerations

This research aims to show how TAs are constantly improving their techniques to stay undetected using advanced evasion techniques such as steganography, file deletion, and code obfuscation. And at the same time, the use of

official app stores to deliver banking trojans to reach a more significant number of potential victims is a new trend that is gaining strength.

According to our findings, we expect to see new sophisticated banking droppers campaigns on the official stores in the next months.

Appendix 1:IOCs

IoC	Description
com.umac.recoverallfilepro	Package name of the dropper on Google Play Store (currently removed)
89a5ebab2b9458e0d31dca80c2cd3e02	MD5 of the dropper
frappucinos[.]shop	C2 of the dropper
95c8f5879f6d83d7c98a8d737cf2783e	MD5 of Vultur
flipstageparty[.]club	C2 of Vultur

Source: <https://www.cleafy.com/cleafy-labs/the-android-malwares-journey-from-google-play-to-banking-fraud>