# MOSCOW FORENSICS DAY '25



## NATALIA KOTOVA

"SPYNOTE IN ACTION: HOW A MOBILE SPYWARE TROJAN IS CREATED, DEPLOYED AND EXAMINED"

MFD

# {SpyNote} in action

## how a mobile spyware trojan is created, deployed and examined

NATALIA KOTOVA

MOSCOW FORENSICS DAY '25



# Nice to meet you!

- **expert of the computer and radio engineering examinations and research division, EKC UMVD of Russia for the Yaroslavl Region**
- **Kikot Moscow University of the MVD of Russia, 2024. 10.05.05 IT Security in Law Enforcement. Specialization: Computer Forensic Examination**
- **one year of forensic practice**

So who is this SpyNote?




## RAT trojan

Remote Administration Tool – remote administration or control software

## Top 10

According to Kaspersky's 2024 report, the verdict is among the 10 most common and steadily growing ones.

## Since 2016

2016 – tools for building it appeared online.

2022 – the source code of one of the versions leaked

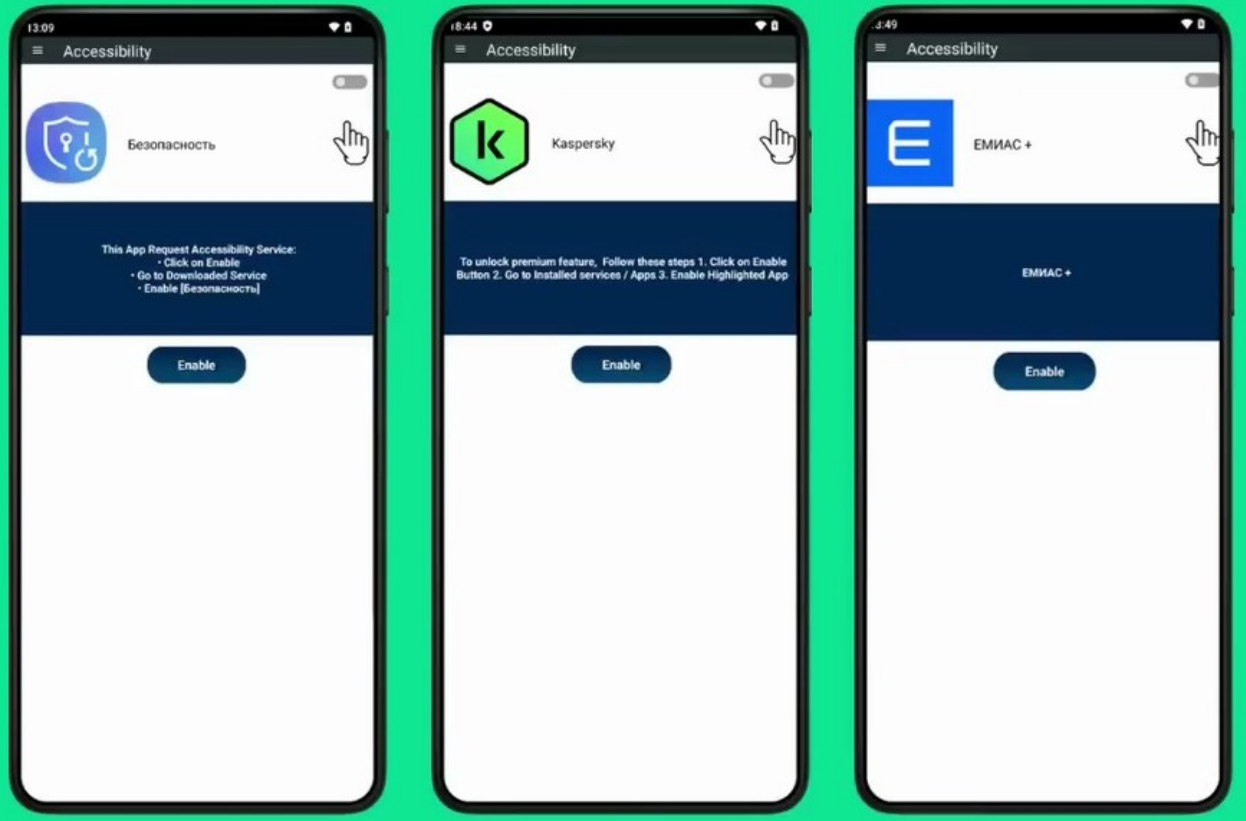
So who is this SpyNote?

## Main attack vector

- bypassing two-factor authentication by intercepting SMS messages

## Disguise methods

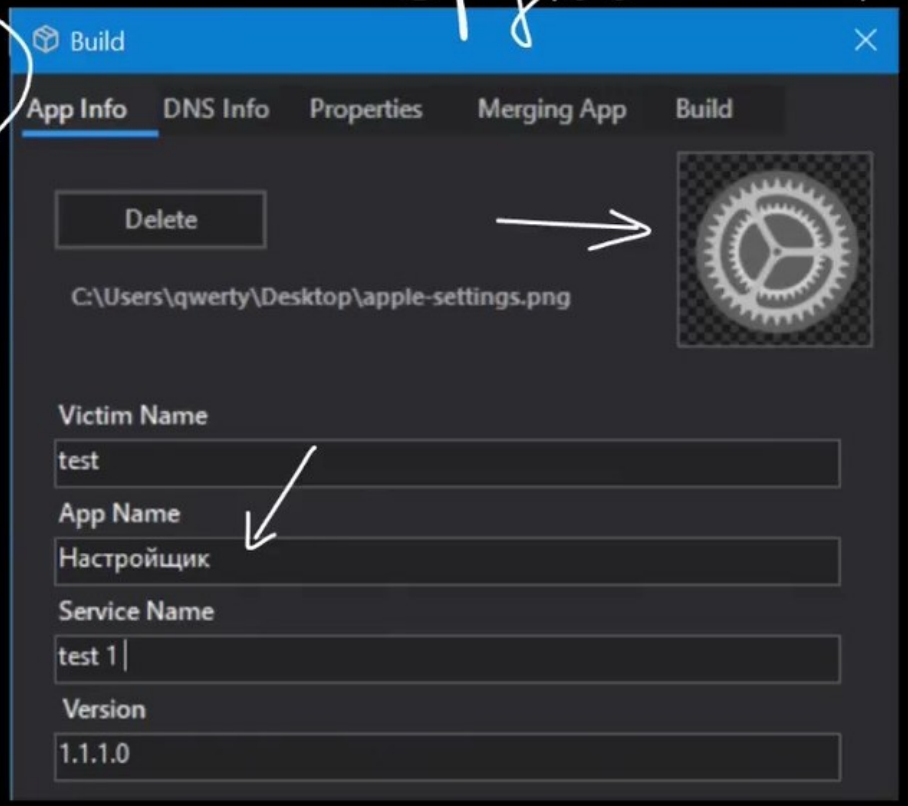
- using the names and icons of legitimate apps
- hiding the app in the list of installed apps



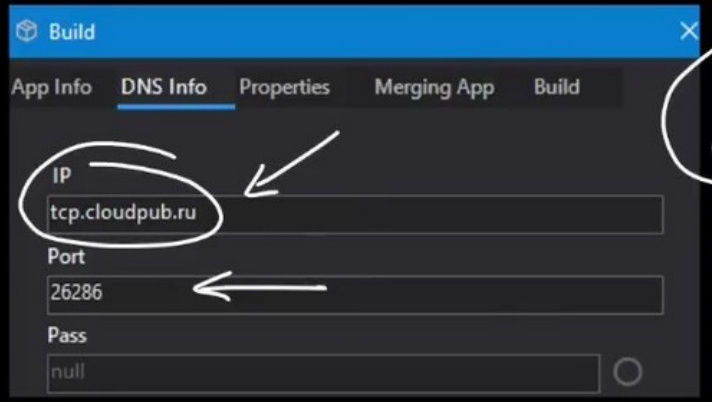
# The apk file builder

SpyNote 6.4

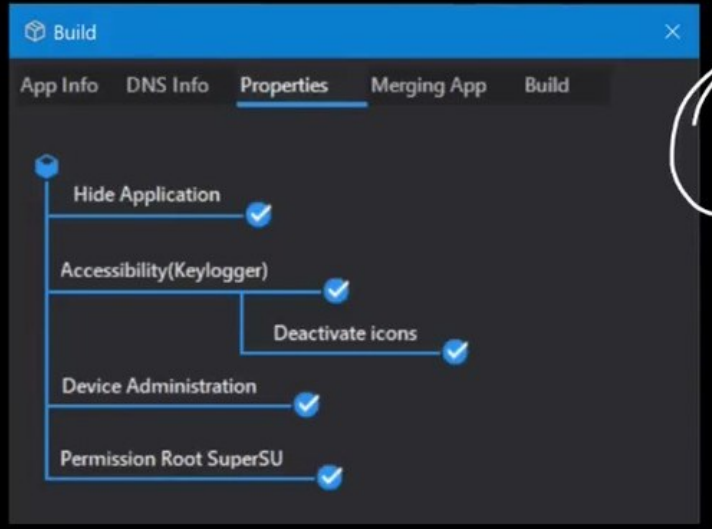
1



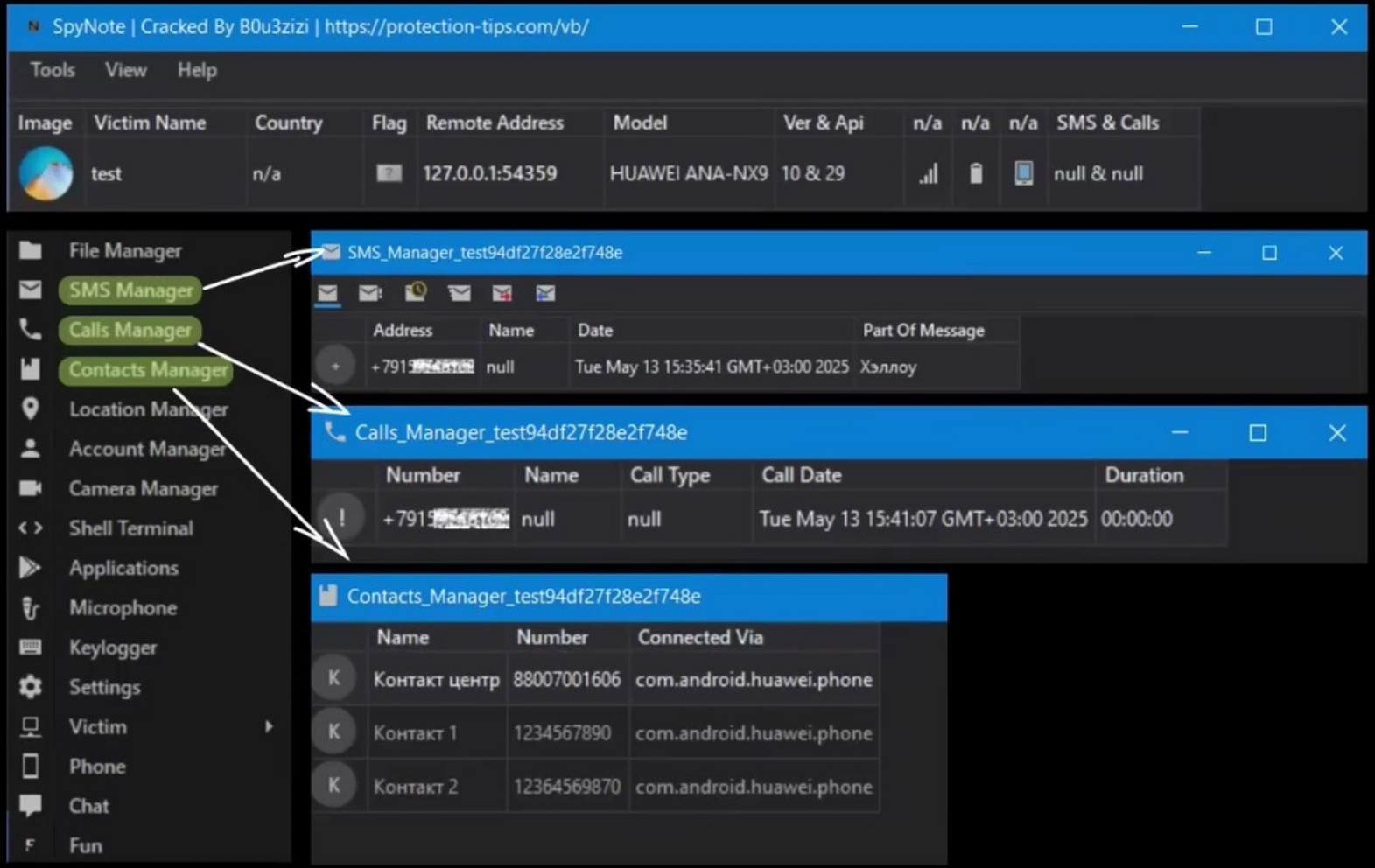

2



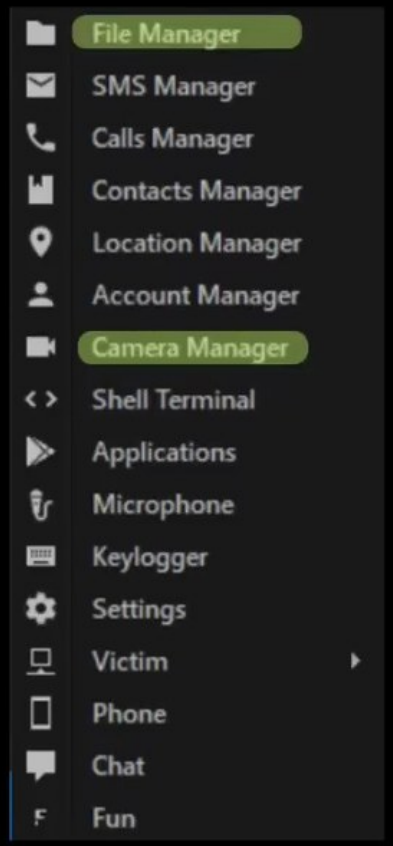

3

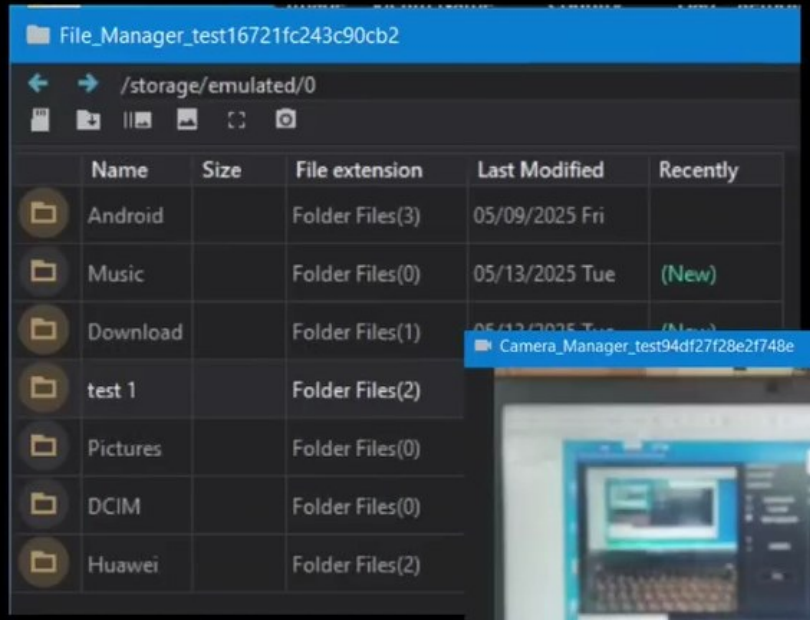


# Actions on an infected device



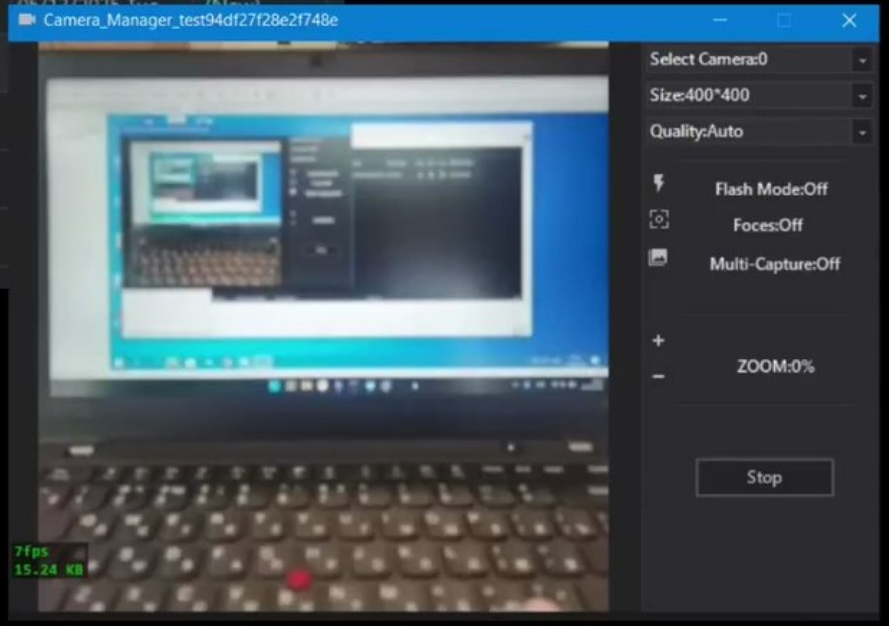
# Actions on an infected device



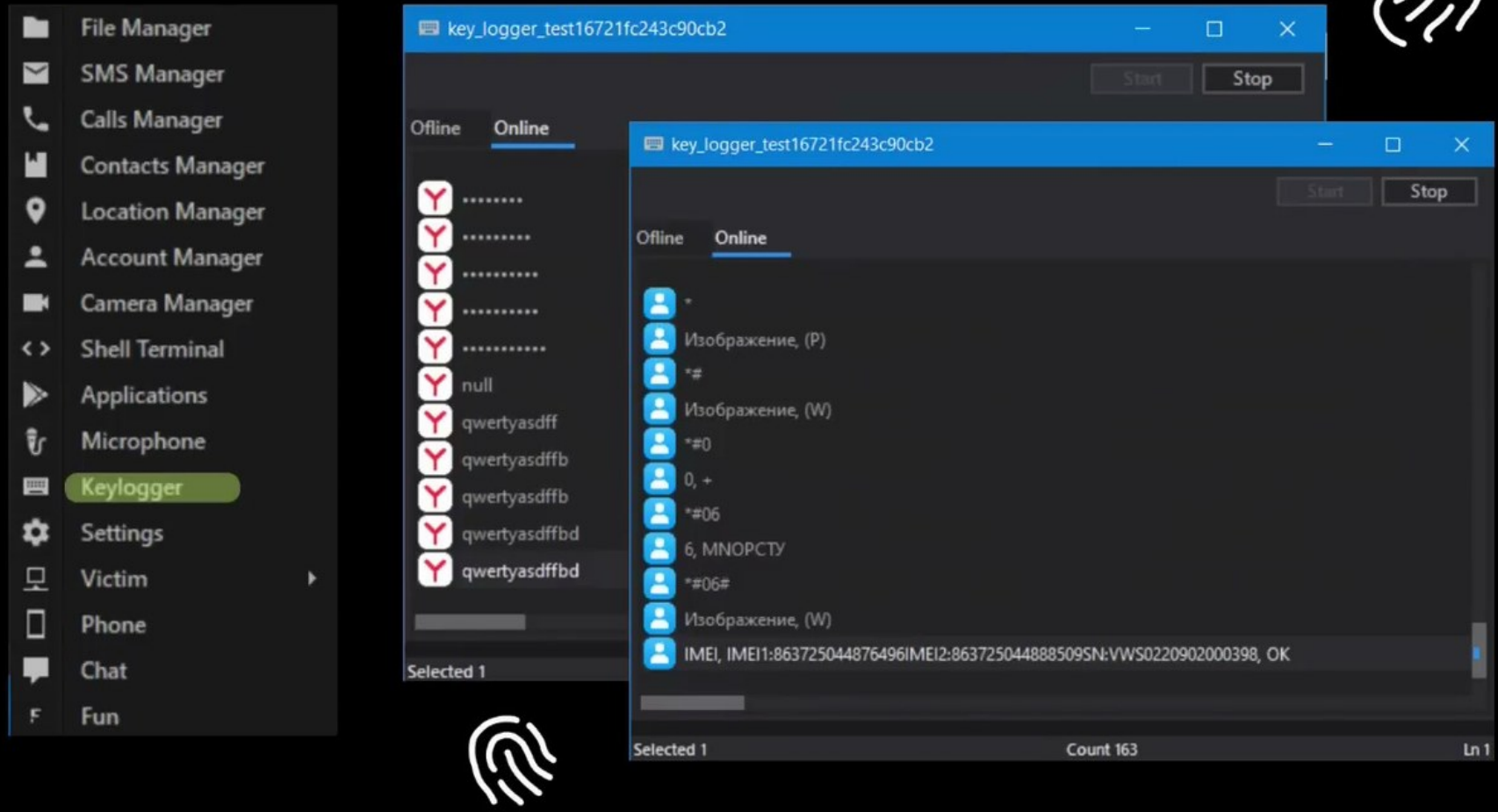





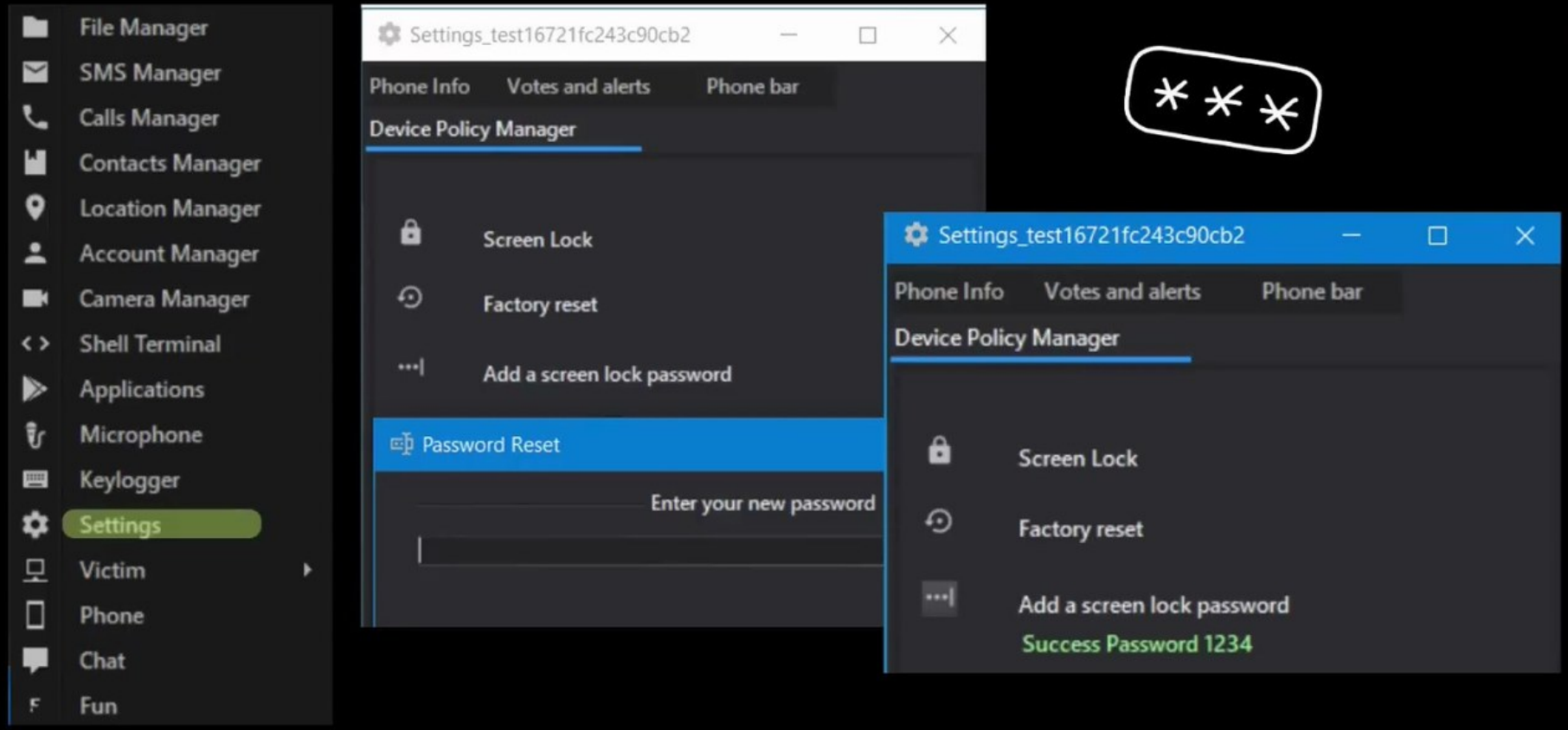


# Actions on an infected device



# Actions on an infected device



# Other SpyNote capabilities

**Location tracking**



**Ability to make calls**



**Microphone access**



**Chat with the user**



**Shell terminal**



**Access to account details**



# Standard questions

1. Are there any files on the mobile phone submitted for examination that are detected by antivirus software?

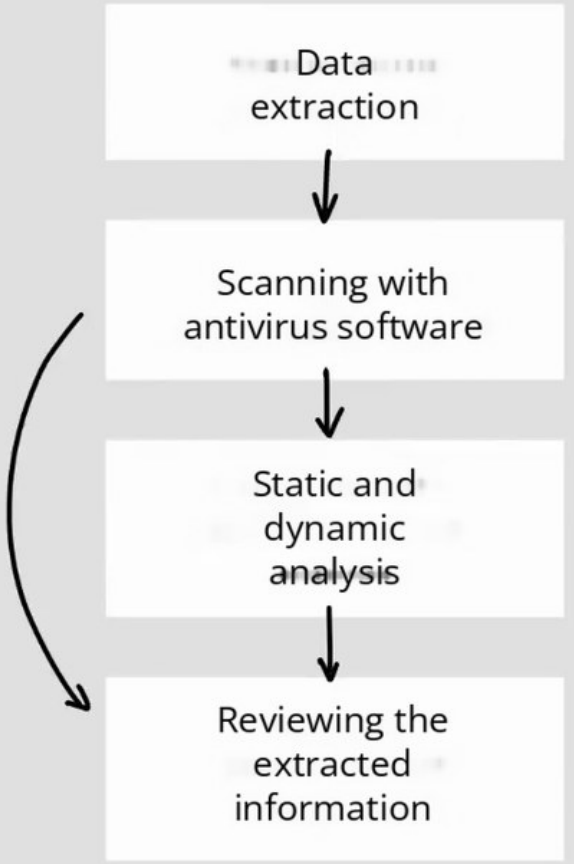
2. Does the mobile phone submitted for examination hold information about installed remote access software?

3. Does the mobile phone submitted for examination hold call and SMS history, present or deleted?

4. Does the mobile phone submitted for examination hold a history of messaging via messengers and social networks?

5. Does the mobile phone submitted for examination hold a history of web pages visited via browsers?

# Stages of the examination



**Tasks of static and dynamic analysis**

## Static analysis

**1. Extracting and decompiling the APK**

2. Examining the metadata (AndroidManifest.xml)
   - package name, version, minSdk/targetSdk.

**3. Analyzing the app structure (AndroidManifest.xml)**
   - **activities, services, receivers**

**4. Identifying protection methods (APKiD)**
   - obfuscation, packing

**5. Source code analysis**
   - list of Android APIs used
   - behavioral analysis

**6. Analysis of resources and strings**
   - searching for URLs, IPs, domains, e-mail

## Dynamic analysis

**1. Installing and running in an isolated environment**

**2. Monitoring the app's behavior**
   - **requested permissions**
   - **interaction with the user**
   - **behavior under different conditions**

**3. Network activity analysis**
   - network connections (TCP/UDP, HTTP/HTTPS)
   - **analysis of network packet payloads**

**4. File activity analysis**
   - **files created and modified**
   - app data

# Tools

## Static analysis

### Decompilation

- apktool
- JADX / JADX-GUI
- dex2jar
- Androguard

### Search and analysis

- strings
- grep
- APKiD

## Dynamic analysis

### Emulators

- AndroidStudio (AVD)
- Genymotion
- Nox, LD player

### Traffic analysis

- Burp Suite
- ZAP
- Wireshark

## Static + dynamic analysis

https://mobsf.live

### Mobile Security Framework (MobSF)

Repo: https://github.com/MobSF/Mobile-Security-Framework-MobSF

Doc: https://mobsf.github.io/docs/#/

docker

- **automated static analysis**
- **dynamic analysis with the ability to track app activity**
- report generation

# AndroidManifest.xml

<manifest package=package name versionName= version>

<uses-permission>
the app's permissions (rights) to access various features and data on the device
</uses-permission>

<application>
<activity>
the app's graphical forms (screens)
<intent-filter> intent type </intent-filter>
</activity>
<service>
defines the app's background services
</service>
<receiver>
defines the receivers of broadcasts
</receiver>
</application>

</manifest>

https://developer.android.com/guide/topics/manifest/manifest-intro

```
<uses-permission android:name="android.permission.WRITE_CONTACTS" />
<uses-permission android:name="android.permission.READ_CONTACTS" />
<uses-permission android:name="android.permission.RECORD_AUDIO" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE" />
<uses-permission android:name="android.permission.READ_CALL_LOG" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.CALL_PHONE" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.GET_TASKS" />
```

```
<activity android:label="@string/app_name" android:name="cmf0.c3b5bm90zq.patch.C7" android:
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>
```

MAIN+LAUNCHER → main activity

```
<service android:name="cmf0.c3b5bm90zq.patch.C1" android:permission="android.permission.B
    <intent-filter>
        <action android:name="android.accessibilityservice.AccessibilityService" />
    </intent-filter>
</service>
```

background service with access to accessibility features

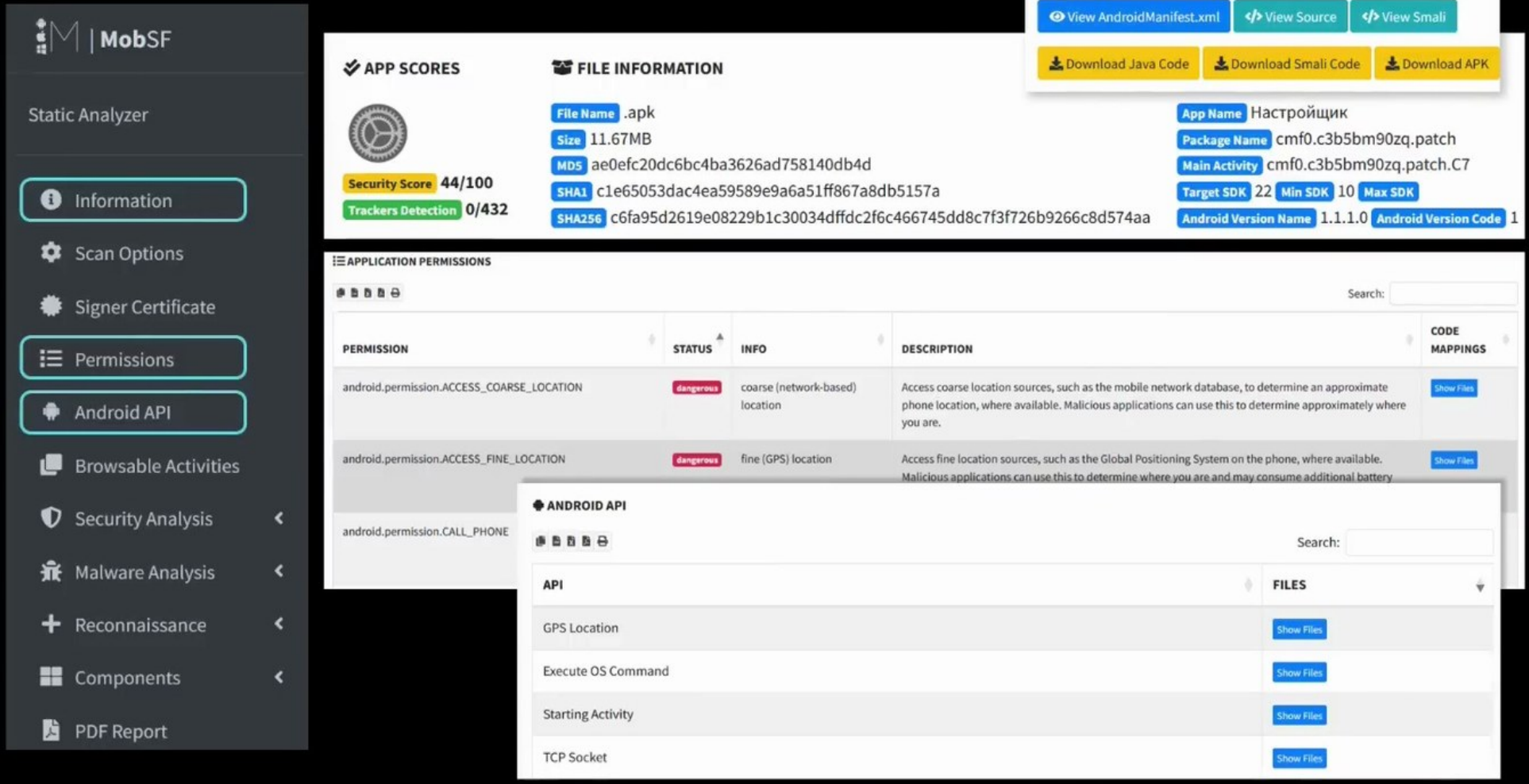
```
<receiver
    <intent-filter>
        <action android:name="android.provider.Telephony.SMS_RECEIVED" />
    </intent-filter>
</receiver>
```

monitoring incoming SMS

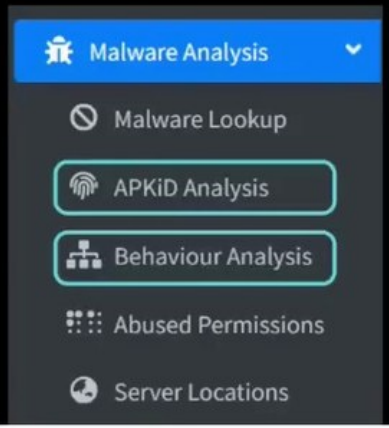
```
<receiver android:name="cmf0.c3b5bm90zq.patch.C4" android:enabled="true" android:exp
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED" />
    </intent-filter>
</receiver>
```

monitoring device startup

# Mobile Security Framework – static analysis



# Mobile Security Framework – static analysis




**APKiD ANALYSIS**

Search:

| DEX | DETECTIONS |
|---|---|
| classes.dex | Search: |

| FINDINGS | DETAILS |
|---|---|
| Anti-VM Code | Build.FINGERPRINT check<br>Build.MANUFACTURER check<br>Build.BOARD check<br>possible Build.SERIAL check |
| Compiler | dexlib 2.x |

**BEHAVIOUR ANALYSIS**

| RULE ID | BEHAVIOUR | LABEL | |
|---|---|---|---|
| 00188 | Get the address of a SMS message | sms | |
| 00191 | Get messages in the SMS inbox | sms | |
| 00052 | Deletes media specified by a content URI(SMS, CALL_LOG, File, etc.) | sms | |
| 00011 | Query data from URI (SMS, CALLLOGS) | sms calllog collection | cmf0/c3b5bm90zq/patch/C11.java |
| 00049 | Query the phone number from SMS sender | sms collection | cmf0/c3b5bm90zq/patch/C10.java |
| 00163 | Create new Socket and connecting to it | socket | cmf0/c3b5bm90zq/patch/C11.java<br>cmf0/c3b5bm90zq/patch/C5.java |

**URLS**

| URL | FILE |
|---|---|
| https://www.google.com | cmf0/c3 |

**STRINGS**

**From APK Resource**

▸ Show all **1365** strings

**From Code**

▸ Show all **428** strings

# Static analysis – searching for network addresses

Search sources:

- **resources (AndroidManifest.xml, <res>, <assets>)**
- **source code**

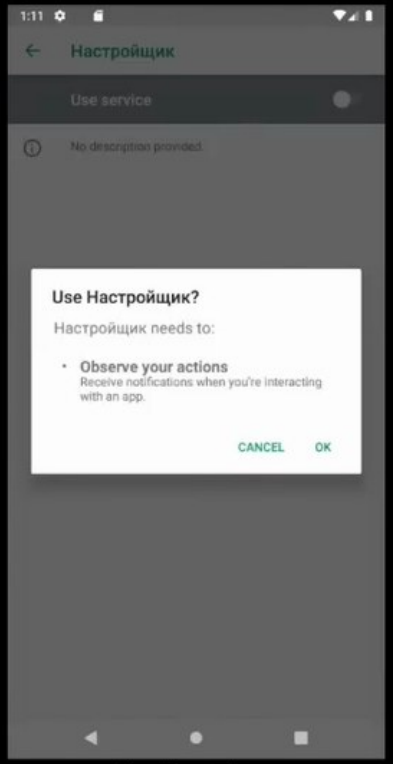
Search methods:

- **searching for the strings “host”, “port”, “http://”, etc.**
- searching by regular expression (ip address)
- searching by Java methods, e.g. “Socket”, “HttpURLConnection”, “OkHttpClient”
- searching for decoding/decryption methods in the code (e.g. Base64 / XOR / AES)

```
<string name="app_name">Настройщик</string>
<string name="gp">01011</string>
<string name="h">tcp.cloudpub.ru</string>
<string name="n">test</string>
<string name="p">26268</string>
<string name="ps">null</string>
<string name="s">test 1</string>
<string name="search_menu_title">Search</string>
<string name="status_bar_notification_info_overflow"
<string name="v">1.1.1.0</string>
```

In the test SpyNote v.6.4 apk file the address of the C2 server sits in the resources (res/values/strings.xml)

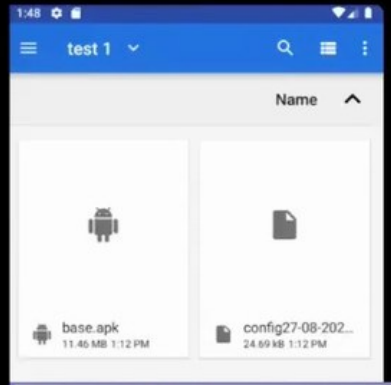
# MobSF + AVD (Android 10) + Wireshark




Activating Accessibility Service




Activating Device Admin App




Creating the directory /data/media/0/test 1/

1. APK of the app used for binding*
2. SpyNote log file (keylogger records)

Text data + Base64

2DlLKVSOHdEUBRCKh3SqOj+wukQudAbiYimTdFlfPUOx/tN/1/dS1b/X7x6OYAAAAASUVORK5CYII=10251Activate this device admin app10251com.android.settings1025113:12 PM10250-110251Apps list10251com.google.android.apps.nexuslauncher1025113:12 PM10250-110251Calculator, Calendar, Camera, Chrome, Clock, Contacts, Drive, Files, Gmail, Google, Maps, Messages, Phone, Photos, Play Movies & TV, Play Music, Settings, WebView Browser Tester, YouTube, Настройщик10251com.google.android.apps.nexuslauncher1025113:12 PM10250-110251Apps list10251com.google.android.apps.nexuslauncher1025113:12 PM10250-110251Package installer10251com.google.android.packageinstaller1025113:12 PM10250-110251Настройщик, For your security, your phone is not allowed to install unknown apps from this source., CANCEL, SETTINGS10251com.google.android.packageinstaller1025113:12 PM10250-110251Package installer10251com.google.android.packageinstaller1025113:12 PM10250-110251SETTINGS10251com.google.android.packageinstaller1025113:12 PM10250iVBORw0KGgoAAAANSUhEUgAAABgAAAAYCAYAAADgdz34AAAABHNCSVQICAgIfAhkiAAAA1hJREFUSImNlU9sG0UUxn9vdr12mig0TQE1XJBoJcqB3jlRcUFIXCrEBQ7lUiGhXhLqTZzUcVLbat1W6gEJFSEUCYEKFw4IceHMvSABErTc+kegOiUhxPbuPA673j9eJ82TLNvfvPm+7828mRGeFJcuvIEx340d0/AMK9e+2W+67DnS8neAiScaiJR6LHcqBxO4+uEkfWf7YMQj4XCYxSuPs5DJJbT882PJVRFrkTCMPtaCalEgZJNW9eL4ChrnDlGa+XecMQkt2wuNHDZ1vYE6Zlw6GDnC0uVuvoI9yIfuC6J7VQFg9VGiBUDLT8lV8VRxbURsQkt3xD3A5kIjWjZrcaxSQfKCTb8H4MZ/D2WddRfWxjvLhGMM2yN5U9dWUdeJifCir1b1NZAfInLl2zPvcPr54wXC+S++5PPbPwHW1skX+fi9s4Wc2/fu8cqtT8CY4WqcNUPyGBhLPusvsvHLrwTlMkGlzK07d5n1Fwt5p+bmcLdIl0pkI98GAk+1a/SDIHY+1dfYksfR/qW6l9borq/x+5KPLYm8

# Dynamic analysis – network traffic analysis

## Without internet access

DNS query for the domain name “tcp.cloudpub.ru”, no response

```
7 15.965001000  192.168.232.2  10.0.2.3  DNS  77 Standard query 0x486c AAAA tcp.cloudpub.ru
```

The address of our C2 server

## With internet access

DNS query for the domain name “tcp.cloudpub.ru”, with a response

```
40 35.752435000  192.168.232.2  10.0.2.3        DNS  77 Standard query 0x4fab AAAA tcp.cloudpub.ru
41 35.803319000  10.0.2.3        192.168.232.2  DNS  132 Standard query response 0x4fab AAAA tcp.cloudpub.ru SOA ns1.reg.ru
42 35.804194000  192.168.232.2  10.0.2.3        DNS  77 Standard query 0x0fa8 A tcp.cloudpub.ru
43 35.862414000  10.0.2.3        192.168.232.2  DNS  93 Standard query response 0x0fa8 A tcp.cloudpub.ru A 87.242.106.13
```

Establishing a TCP connection with “87.242.106.13” and exchanging data

```
44 35.864728000  192.168.232.2  87.242.106.13  TCP   76 55016 → 26268 [SYN] Seq=0 Win=65535 Len=0 MSS=1360 SACK_PERM TSval=4294962891 TSecr=0 WS=64
45 35.909881000  87.242.106.13  192.168.232.2  TCP   60 26268 → 55016 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460
46 35.910227000  192.168.232.2  87.242.106.13  TCP   56 55016 → 26268 [ACK] Seq=1 Ack=1 Win=65535 Len=0
47 36.942962000  192.168.232.2  87.242.106.13  TCP  1416 55016 → 26268 [ACK] Seq=1 Ack=1 Win=65535 Len=1360
48 36.943365000  192.168.232.2  87.242.106.13  TCP  184 55016 → 26268 [PSH, ACK] Seq=1361 Ack=1 Win=65535 Len=128
49 36.944045000  87.242.106.13  192.168.232.2  TCP   56 26268 → 55016 [ACK] Seq=1 Ack=1361 Win=8760 Len=0
50 36.944117000  87.242.106.13  192.168.232.2  TCP   56 26268 → 55016 [ACK] Seq=1 Ack=1489 Win=8760 Len=0
51 38.112548000  87.242.106.13  192.168.232.2  TCP   74 26268 → 55016 [PSH, ACK] Seq=1 Ack=1489 Win=8760 Len=18
52 38.112843000  192.168.232.2  87.242.106.13  TCP   56 55016 → 26268 [ACK] Seq=1489 Ack=19 Win=65535 Len=0
53 38.553716000  87.242.106.13  192.168.232.2  TCP   63 26268 → 55016 [PSH, ACK] Seq=19 Ack=1489 Win=8760 Len=7
54 38.553985000  192.168.232.2  87.242.106.13  TCP   56 55016 → 26268 [ACK] Seq=1489 Ack=26 Win=65535 Len=0
```

Packet payload

Gzip signature

```
30  50 10 ff ff e3 df 00 00  31 34 38 33 00 1f 8b 08  P······ 1483····
40  00 00 00 00 00 00 00 ad  56 c9 d2 ab b8 0e 7e 95  ········ V·····~·
50  b3 ea 0d 8b 30 84 00 8b  5e 18 30 33 01 33 85 b0  ····0··· ^·03·3··
60  23 cc 43 80 30 87 a7 bf  9c bf bb ba ba ee ba e5  #·C·0··· ········
```

→ unzip → Decode Base64 →



.jpeg

| Image | Victim Name | Country | Flag | Remote Address | Model |
|---|---|---|---|---|---|
| | test | n/a | ? | 127.0.0.1:61018 | GOOGLE |

The C2 server admin panel

# Installation details

**«frosting.db»**
/data/data/com.android.vending/databases/

**«verify_apps.db»**
/data/data/com.android.vending/databases/

**«packages.xml»**
/data/system/

- package name
- path to the app file
- installation time
- installation initiator
- app installer

/data/data/<package name> – the app data directory

**Установленные приложения\История установок приложений**

| Время последнего обновления (Москва) | ID приложения | Путь к APK |
|---|---|---|
| 17.12.2024 16:32:50 (UTC+3) | com.whh.premium | base.apk |

**Установленные приложения\Верифицированные приложения** 1/1

| Название | ID приложения |
|---|---|
| Энергосбыт | com.whh.premium |

```
package
  name = com.whh.premium
  codePath = /data/app/~~5fmOrSm3aKS8E9SKjK9l9w==/com.whh.premium-ITqA8cmV9JJIomuiF-zZYw==
  nativeLibraryPath = /data/app/~~5fmOrSm3aKS8E9SKjK9l9w==/com.whh.premium-ITqA8cmV9JJIomuiF-zZYw==/lib
  publicFlags = 945339972
  privateFlags = -1409282048
  ft = 193d4d23fe8      Unix Milliseconds (Java Time)   2024-12-17 16:32:49.0000000 +03:00
  ut = 193d4d24462      Unix Milliseconds (Java Time)   2024-12-17 16:32:50.1460000 +03:00
  version = 3000
  userId = 10366
  installer = com.google.android.packageinstaller
  packageSource = 3
  installInitiator = com.google.android.packageinstaller
  installOriginator = org.telegram.messenger
  loadingProgress = 1
  domainSetId = aa070b91-1d0d-4671-874c-7b6e13391cc4
```

- ft – time of last modification
- it – time of first installation

• ut – time of last update

# Searching by package name

**newbatterystats**
/data/log/batterystats/
Samsung devices

```
+1h20m36s334ms (2) 095 pkginst=57401:"com.iproxy.android"
+1h20m36s503ms (2) 095 pkginst=57401:"com.iproxy.android"
+1h20m36s594ms (2) 095 pkginst=57401:"com.iproxy.android"
+1h20m36s639ms (2) 095 pkginst=57401:"com.iproxy.android"
```

```
+9h42m55s610ms (2) 060 pkgunin=0:"com.iproxy.android"
+9h42m55s666ms (2) 060 pkgunin=0:"com.iproxy.android"
```

**pm_debug_info.txt**
/data/log/

```
2025-06-13 18:02:41.662: START INSTALL PACKAGE: observer{229554387}
          stagedDir{/data/app/vmdl53600148.tmp}
          stagedCid{null}
          pkg{com.example.application}
          Request from{com.google.android.packageinstaller}
```

```
2025-06-13 19:49:11.541: START DELETE PACKAGE: observer{216981792}
pkg{com.example.application}, user{0}, caller{10085} flags{2}
```

# Other artifacts

The built-in antivirus of the “Sberbank” app

**«threats.db»**

/data/data/ru.sberbankmobile/databases

«{"virusName":"HEUR:Trojan-Spy.AndroidOS.SpyNote.bz","objectName":"/data/ app/~~aUtq8Y3Z3RGSyUUJ9fuMuA==/ate.asking.humanity-J32gBHsvrskliOSBlHL_Kw==/base.apk","fileFullPath":"/data/app/~~aUtq8Y3Z3RGSyUUJ9fuMuA==/ate.asking.humanity-J32gBHsvrskliOSBlHL_Kw==/base.apk","fileHash":null,"package Name":"ate.asking.humanity","severityLevel":"HIGH","application":true,"adminThreat":false,"cloudCheckFailed":false}»

SpyNote log

**«log-yyyy-mm-dd.txt»**

/data/media/0/Config/sys/apps/log

decode from Base64

Графический интерфейс системы#[Панель уведомлений, RT-10, Развернуть, Bluetooth, Развернуть, . Нажмите для установки режима вибрации. При этом могут также выключиться звуки в службе специальных возможностей., 19:59, чт, 21 нояб., Открыть настройки., NFC, Вибровызов., Bluetooth disconnected., VoLTE для карт SIM1 и SIM2, 90 %, Удалить все уведомления]#5