

SpyNote continues to attack financial institutions

cleafy.com/cleafy-labs/spynote-continues-to-attack-financial-institutions

Francesco Iubatti,

Download your PDF guide to TeaBot

Get your free copy to your inbox now

[Download PDF Version](#)

Key points

- Starting from the end of 2022, an Android Spyware called **SpyNote** was observed to carry out bank fraud due to its many features.
- SpyNote abuses Accessibility services and other Android permissions in order to:
 - Collects SMS messages and contacts list;
 - Record audio and screen;
 - Keylogging activities;
 - Bypass 2FA;
 - Tracking GPS locations.
- The spyware is distributed through email phishing or smishing campaigns and the fraudulent activities are executed with a combination of remote access trojan (RAT) capabilities and vishing attack.
- During the months of June and July 2023, we have observed an extensive campaign against multiple European customers of different banks.

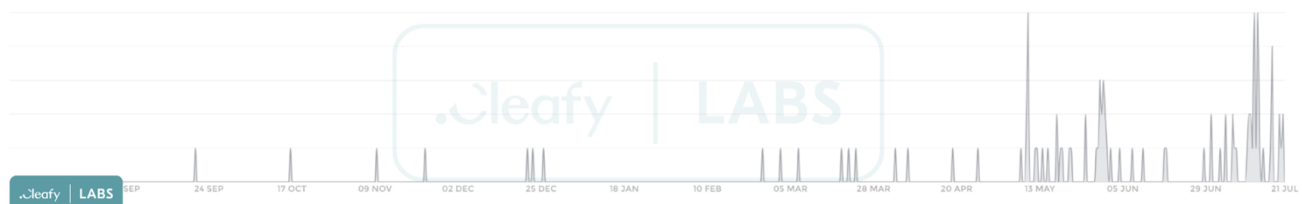


Figure 1 – SpyNote infection based on Cleafy telemetries

Introduction

During the last years, Cleafy Threat Intelligence Team has discovered and analyzed multiple Android banking trojans (e.g Sharkbot, Teabot etc), namely malicious applications used to carry out bank frauds through ATO or ATS techniques.

However, in recent months, we have observed an increase in spyware infections, particularly **SpyNote** (Figure 1). Although spyware is usually used to collect user data (and profit from them) or conduct espionage campaigns, SpyNote is currently also used to perform bank fraud. Similar campaigns were also reported by other researchers during the current year.

By analyzing these recent campaigns, we observed that the chain of infection usually starts with a fake SMS message (smishing) where the user is asked to install the “new certified banking app”. A second message follows, redirecting the user to the legitimate app of TeamViewer, an app used to receive technical remote support. The right image of Figure 2 shows how the link redirects the user to the official app of TeamViewer QuickSupport on the Google Play Store.

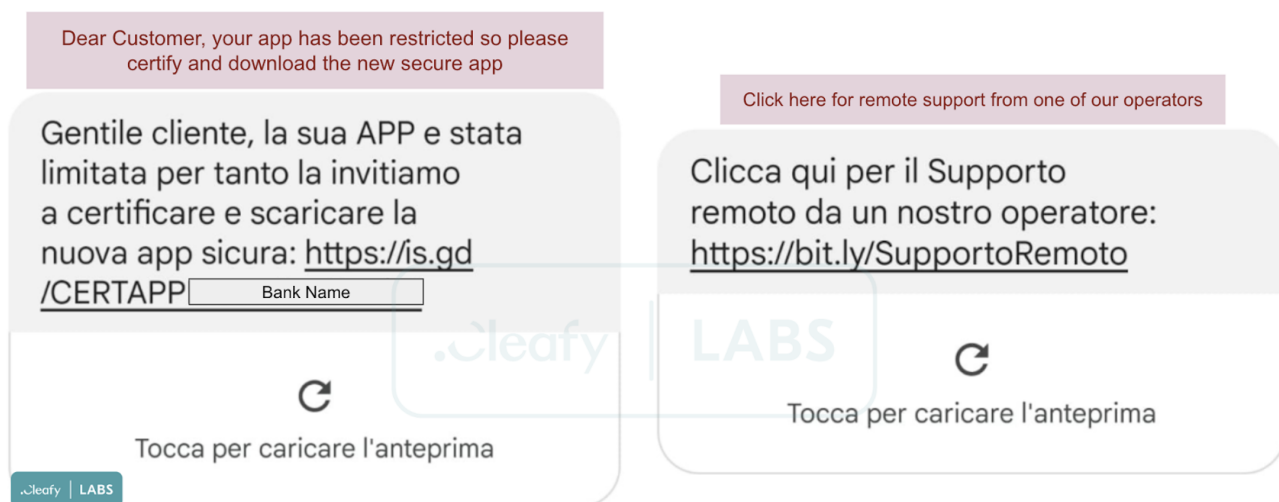


Figure 2 – Examples of sms messages used during the recent SpyNote campaign. According to our analysis, Teamviewer has been adopted by several TAs to execute fraud operations through social engineering attacks. In particular, the attacker calls the victim, impersonating bank operators, and performs fraudulent transactions directly on the victim’s device.

During our analysis, we have intercepted multiple samples masquerading behind various applications, such as security apps, bank names or Android updates, as shown in Figure 3.

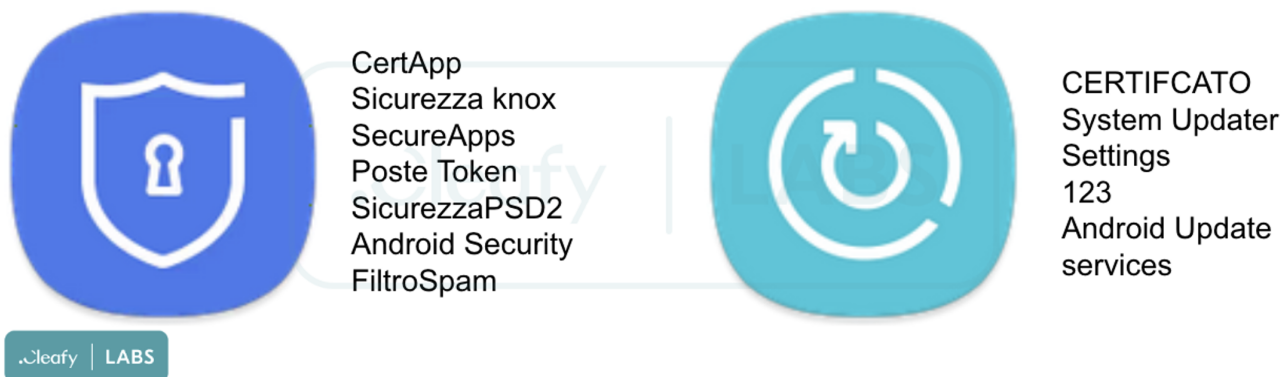


Figure 3 – Examples of icons/names used by SpyNote

Main features

Similar to other Android banking trojans, SpyNote abuses the Accessibility services granted by the victim during the installation of the app. The spyware uses this permission to accept other permissions popups automatically (Figure 4) and perform keylogging activities.

SpyNote has lots of capabilities (e.g., access to the camera or microphone of the infected device, GPS tracking etc.), but in this article we will explain only the main features used to perform banking fraud.

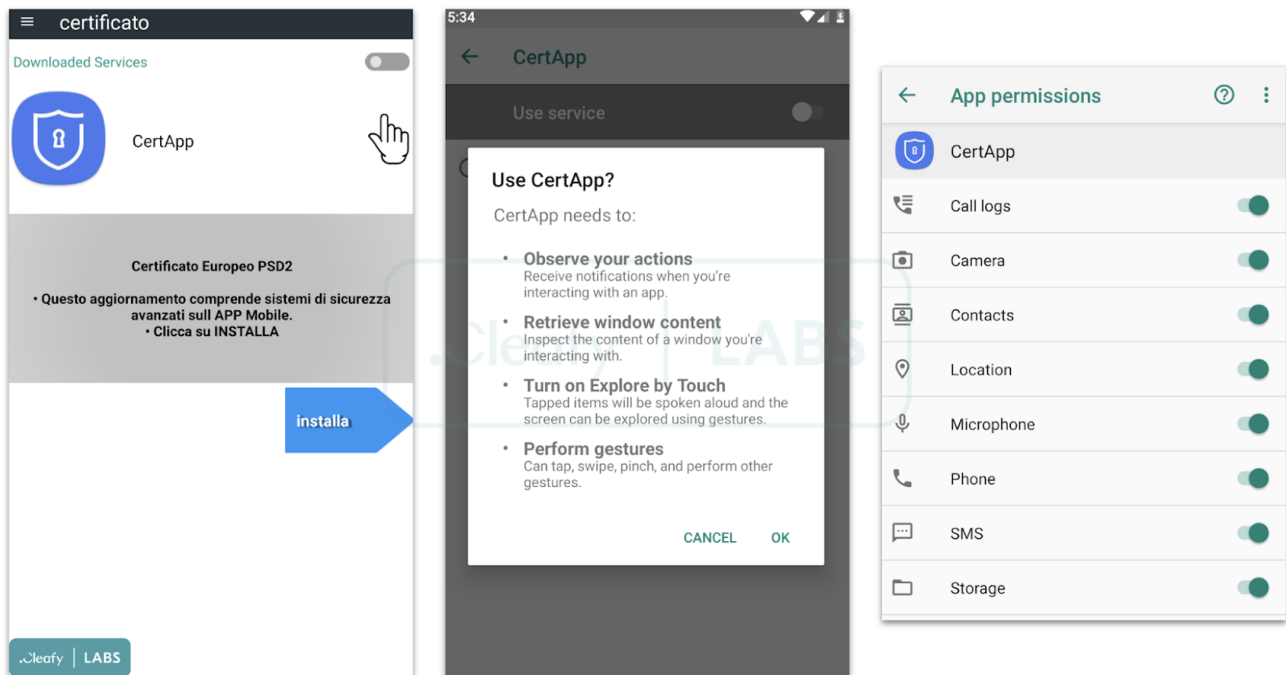


Figure 4 – SpyNote installation phases and permissions automatically accepted

Keylogger

Once the user accepts the Accessibility popup, it allows SpyNote to see every activity done by the user on the compromised device. In particular, the spyware tracks:

- The list of applications installed on the infected device;
- Which application is using the user and, in particular, some specific app properties such as package name, name, label etc.;
- Any text written by the user.

To keep track of the above information, SpyNote saves everything (encoded in Base64) inside a “log-yyyy-mm-dd.txt” file, in a directory created by the spyware, named: “/Config/sys/apps/log”.

```
void H(String s) {
    try {
        String s1 = DateFormat.format("yyyy-MM-dd", new Date()).toString();
        File file0 = Environment.getExternalStorageDirectory();
        File file1 = new File(file0, "/Config/sys/apps/log");
        File file2 = new File(file0, "/Config/sys/apps/log/log-" + s1 + ".txt");
        if(!file1.exists()) {
            file1.mkdirs();
        }
    }
}
```

Figure 5 - SpyNote keylogger file

The following feature could be used by TAs to identify the bank(s) application(s) used by the user and then to steal the credentials (as shown in Figure 6), credit card information, or other essential data.

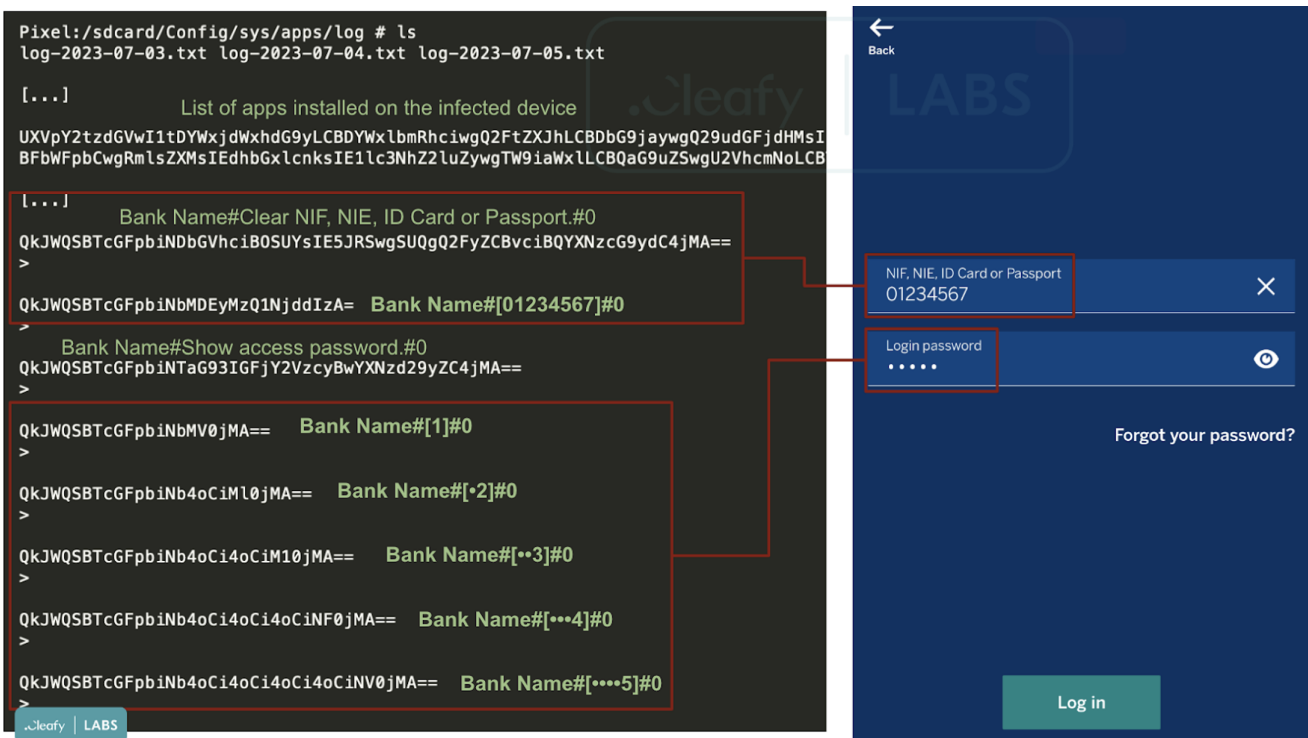


Figure 6 - Example of how SpyNote is able to steal bank credentials

SMS Collection & 2FA Bypass

Multiple apps (e.g., emails, social networks, etc) allow to use two-factor authentication (2FA) codes to add an extra layer of security. This means that, in addition to the password, the user must also enter a code to log into the account; this code can be generated by apps like Google Authenticator or sent via SMS message or email. For banks, as established by the EU's Payment Services Directive 2 (PSD2), it is necessary to use strong customer authentication (SCA) to confirm a money transaction, such as through a pin sent by the bank to the user's device or fingerprint.

SpyNote can gather SMS messages received by the user and transmit them to the C2 server (Figure 7) and it can also gain access to the temporary codes generated by the Google Authenticator app, exploiting the Accessibility services.

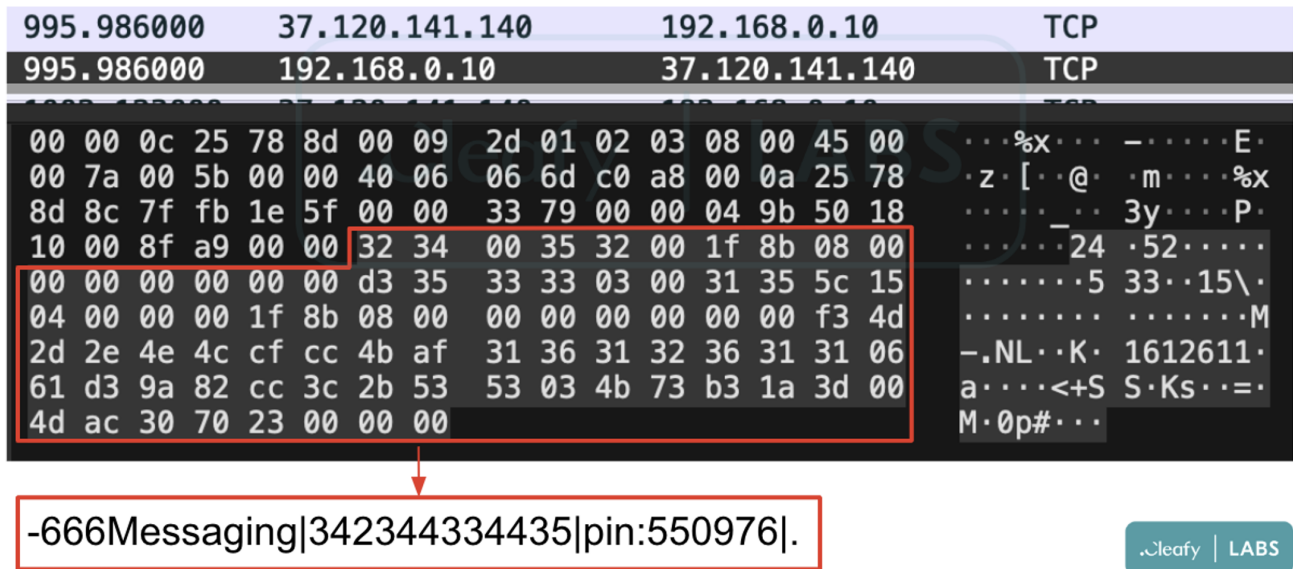


Figure 7 - Example of SMS message stolen by SpyNote

C2 Communications

Once installed, SpyNote contacts the C2 via socket communication using a hardcoded IP address and port within the application code, both encoded in Base64.

By analyzing multiple samples, we observed that a characteristic of SpyNote is the use of different uncommon ports (in the following sample, it uses the 7771 port) to communicate with the C2 server.

The data exchanged between the spyware and the C2 server are packaged with a custom scheme (Figure 8), where the first bytes represent the length of the data, followed by a null byte, and then the compressed data using the GZip algorithm.

```

4 1.107000 192.168.0.10 37.120.141.144 TCP 174 32763 → 7771 [PSH, ACK] Seq=1 Ack=1 Win=4096 Len=120
5 1.264000 37.120.141.144 192.168.0.10 TCP 168 7771 → 32763 [PSH, ACK] Seq=1 Ack=121 Win=4096 Len=114
6 1.629000 192.168.0.10 37.120.141.144 TCP 1454 32763 → 7771 [PSH, ACK] Seq=121 Ack=115 Win=4096 Len=1400
7 1.629000 37.120.141.144 192.168.0.10 TCP 54 7771 → 32763 [ACK] Seq=115 Ack=1521 Win=4096 Len=0
8 1.629000 192.168.0.10 37.120.141.144 TCP 1221 32763 → 7771 [PSH, ACK] Seq=1521 Ack=115 Win=4096 Len=1167
9 1.773000 37.120.141.144 192.168.0.10 TCP 150 7771 → 32763 [PSH, ACK] Seq=115 Ack=2688 Win=4096 Len=96
10 2.474000 192.168.0.10 37.120.141.144 TCP 116 32763 → 7771 [PSH, ACK] Seq=2688 Ack=211 Win=4096 Len=62
11 17.629000 37.120.141.144 192.168.0.10 TCP 54 7771 → 32763 [ACK] Seq=211 Ack=2750 Win=4096 Len=0
12 17.629000 192.168.0.10 37.120.141.144 TCP 281 32763 → 7771 [PSH, ACK] Seq=2750 Ack=211 Win=4096 Len=227
13 34.085000 37.120.141.144 192.168.0.10 TCP 54 7771 → 32763 [ACK] Seq=211 Ack=2977 Win=4096 Len=0

```

```

Time to Live: 64
Protocol: TCP (6)
Header Checksum: 0x069c [validation disabled]
[Header checksum status: Unverified]
Source Address: 192.168.0.10
Destination Address: 37.120.141.144
Transmission Control Protocol, Src Port: 32763, Dst Port: 7771, Seq: 1, Ack: 1, Len: 120
Source Port: 32763
Destination Port: 7771
[Stream index: 0]

```

```

0000 00 00 0c 25 78 8d 00 09 2d 01 02 03 08 00 45 00 ...%x... ..E
0001 00 a0 00 02 00 00 40 06 06 9c c0 a8 00 0a 25 78 ...@... ..%x
0002 8d 90 7f fb 1e 5b 00 00 00 01 00 00 00 01 50 18 ...[... ..P
0003 10 00 c7 93 00 00 32 31 00 39 33 00 1f 8b 08 00 ...21.93...
0004 00 00 00 00 00 00 b3 00 00 13 57 00 fa 01 00 00 ...W... ..W
0005 00 1f 8b 00 00 00 00 00 00 00 00 33 36 d7 33 34 ...-36-34
0006 32 40 33 34 31 04 02 13 2b 73 73 73 43 2b c7 02 2-341-b+sssC...
0007 02 97 c4 92 44 ab e2 ca e2 92 d4 dc f8 cc bc b4 ...D... ..D
0008 7c 18 3b 39 3f 2f 2d 33 dd 2a 37 b5 24 31 3e 05 |-;97/-3 *7-$1->
0009 a4 c6 18 55 bf a1 a1 01 10 18 1a 5a 45 3b 07 c5 ...U... ..ZE:
00a0 5a 85 99 58 01 00 8b 35 30 ea 60 00 00 00 Z-X...5 0...

```

3636003234001f8b08000000000000400330ca900c2829cd2f4d4bc62bdc4bcf4d41c3d38afa0a0182c9d9b5a92919f0266ea1a82a99cfcc4940a0317838a3c00687753f6410000001f8b080000000000400cb2bcdc901004ffccb2504000000



1TxTxTplugens.angel.pplugens.appsTxTxTmethodTxTxT-1TxTxTloadx0D0xnnull

Figure 8 – Example of SpyNote communication with the C2

Screen Recording and Defense Evasion

Another interesting technique adopted by TAs to observe user actions and collect more information is the Media Projection APIs. This Android feature allows capturing the screen content of the device display. As shown in Figure 9, the user can see, in the notification panel, that an application, in that case “CERTIFICATO”, is projecting his screen.



Figure 9 – Screen recording in action

Defense Evasion

SpyNote uses different defense evasion techniques, such as the obfuscation of all class names (Figure 10), the use of junk code to slow down the static analysis of the code, and anti-emulator controls to prevent it from being launched and analyzed within an emulator or sandbox by security analysts. It is also capable of downloading additional files from the C2 server (Figure 11).

```
<activity android:name="glasgow.pl.ಶಿಸ್ತುಗಳ ಸ್ವತಂತ್ರತೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ದೃಶ್ಯ ಅನುಕರಣೆಗಳನ್ನು ಒದಗಿಸುವ ಉದ್ದೇಶದೊಂದಿಗೆ" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<activity android:name="glasgow.pl.ಶಿಸ್ತುಗಳ ಸ್ವತಂತ್ರತೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ದೃಶ್ಯ ಅನುಕರಣೆಗಳನ್ನು ಒದಗಿಸುವ ಉದ್ದೇಶದೊಂದಿಗೆ" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<activity android:launchMode="singleInstance" android:name="glasgow.pl.ಶಿಸ್ತುಗಳ ಸ್ವತಂತ್ರತೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ದೃಶ್ಯ ಅನುಕರಣೆಗಳನ್ನು ಒದಗಿಸುವ ಉದ್ದೇಶದೊಂದಿಗೆ" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<activity android:name="glasgow.pl.ಶಿಸ್ತುಗಳ ಸ್ವತಂತ್ರತೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ದೃಶ್ಯ ಅನುಕರಣೆಗಳನ್ನು ಒದಗಿಸುವ ಉದ್ದೇಶದೊಂದಿಗೆ" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<activity android:name="glasgow.pl.ಶಿಸ್ತುಗಳ ಸ್ವತಂತ್ರತೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ದೃಶ್ಯ ಅನುಕರಣೆಗಳನ್ನು ಒದಗಿಸುವ ಉದ್ದೇಶದೊಂದಿಗೆ" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<activity android:name="glasgow.pl.ಶಿಸ್ತುಗಳ ಸ್ವತಂತ್ರತೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ದೃಶ್ಯ ಅನುಕರಣೆಗಳನ್ನು ಒದಗಿಸುವ ಉದ್ದೇಶದೊಂದಿಗೆ" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<activity android:name="glasgow.pl.ಶಿಸ್ತುಗಳ ಸ್ವತಂತ್ರತೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ದೃಶ್ಯ ಅನುಕರಣೆಗಳನ್ನು ಒದಗಿಸುವ ಉದ್ದೇಶದೊಂದಿಗೆ" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
<activity android:name="glasgow.pl.ಶಿಸ್ತುಗಳ ಸ್ವತಂತ್ರತೆಯ ಅನುಷ್ಠಾನಕ್ಕೆ ಸಂಬಂಧಿಸಿದಂತೆ ದೃಶ್ಯ ಅನುಕರಣೆಗಳನ್ನು ಒದಗಿಸುವ ಉದ್ದೇಶದೊಂದಿಗೆ" android:theme="@android:style/Theme.Translucent.NoTitleBar"/>
```

Figure 10 – Example of SpyNote code obfuscation

Furthermore, after the installation, the application icon is not shown on the device display, and it prevents the user from manually removing the application via settings.

The image displays network traffic analysis. The top section shows a list of packets with their sequence numbers, acknowledgment numbers, window sizes, and lengths. The bottom section shows a hex dump of a packet, with a red box highlighting the word 'dex' and a blue callout box containing a string of obfuscated characters.

Conclusion

Although this is not the first time that spyware has been used to carry out bank fraud (e.g, Revive: from spyware to Android banking trojan), this SpyNote campaign is certainly one of the most aggressive in recent times.

This research aims to show some new details about how TAs are using SpyNote and social engineering techniques to perform Account Takeover attacks (ATO) and on-device fraud (ODF) against customers of several banks in Europe.

Finally, by observing the aggressiveness and extension of this recent SpyNote campaign, we assume that TAs will continue to use this spyware to carry out bank fraud due to the multiple functionalities.

Appendix 1: IOCs

IoC	Description
9e185dd6d7137357b61941525e935124	Md5 SpyNote (CERTIFCATO)
291c24d9b3f4a5793a2600610671eb42	Md5 SpyNote (CertApp)
37.120.141.]144:7771	SpyNote C2 Server
37.120.141.]140:7775	SpyNote C2 Server