Exobot (Marcher) - Android banking Trojan on the rise

Published: 2024-10-01 · Archived: 2026-04-06 03:21:03 UTC

Introduction

The past months many different banking Trojans for the Android platform have received media attention. One of these, called Marcher (aka Exobot), seems to be especially active with different samples appearing on a daily basis. This malware variant also appears to be technically superior to many other banking Trojans being able to use its overlay attack even on Android 6, which has technical improvements compared to the previous Android versions to prevent such attacks.

The main infection vector is a phishing attack using SMS/MMS. The social engineering message includes a link that leads to a fake version of a popular app, using names like Runtastic, WhatsApp or Netflix. On installation, the app requests the user to provide SMS storage access and high Android privileges such as Device Admin. Other infection vectors include pornographic websites serving apps called Adobe Flash or YouPorn.

The Marcher banking malware uses two main attack vectors. The first attack vector is to compromise the out of band authentication for online banks that rely on SMS using SMS forwarding. The second attack vector, the overlay attack, shows a customized phishing window whenever a targeted application is started on the device. The overlay window is often indistinguishable from the expected screen (such as a login screen for a banking app) and is used to steal the victim’s banking credentials. The target list and bank specific fake login pages can be dynamically updated via their C2 panel (dashboard back-end) which significantly increases the adaptability and scalability of this attack. In addition, this type of Android banking malware does not require the device to be rooted or the app to have any specific Android permission (besides android.permission.INTERNET to retrieve the overlay contents and send its captured data).

The many changes we see in the way the attacks are performed show that attackers are heavily experimenting to find the best way of infecting a mobile device and abusing existing functionality to perform successful phishing attacks. The next stage in device infection could be the use of exploit kits and malvertising, which would be quite effective due the many Android vulnerabilities and consumers with unpatched devices. In addition future Trojans could leverage root exploits to make them almost impossible to remove and give malicious actors the ability to hook generic low level API’s that are used by all (banking) applications, just like the attack vector as has been used on the desktop platform for years.

Technical Analysis

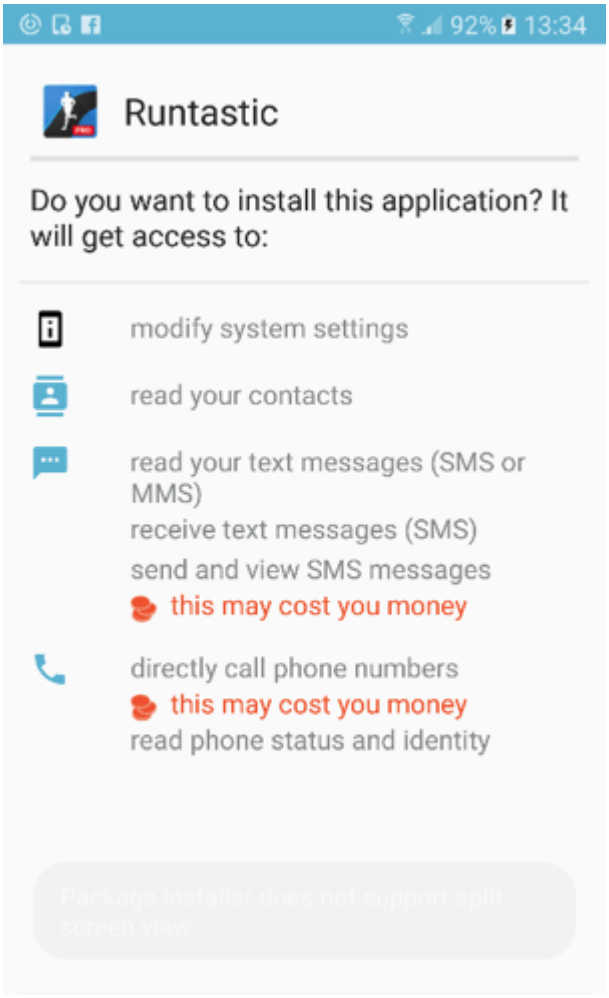
Permissions

Marcher’s APK size is fairly small (only 683KB for sample eb8f02fc30ec49e4af1560e54b53d1a7), much smaller than most legitimate apps and other popular mobile malware samples. This sample only includes Dalvik bytecode and resources without any native libraries. The package name (vyn.hhsdzgvoexobmkygffzwuewrbikzud) and its

many activities and services have randomized names, probably to make it a bit more difficult to detect the package using blacklisting. The set of permissions required by Marcher according to the manifest is as follows:

* android.permission.CHANGE_NETWORK_STATE (change network connectivity state)
* android.permission.SEND_SMS (send SMS messages)
* android.permission.USES_POLICY_FORCE_LOCK (lock the device)
* android.permission.RECEIVE_BOOT_COMPLETED (start malware when device boots)
* android.permission.INTERNET (communicate with the internet)
* android.permission.VIBRATE (control the vibrator)
* android.permission.ACCESS_WIFI_STATE (view information about the status of Wi-Fi)
* android.permission.WRITE_SMS (edit/delete SMS)
* android.permission.ACCESS_NETWORK_STATE (view the status of all networks)
* android.permission.WAKE_LOCK (prevent the phone from going to sleep)
* android.permission.GET_TASKS (retrieve running applications)
* android.permission.CALL_PHONE (call phone numbers)
* android.permission.WRITE_SETTINGS (read/write global system settings)
* android.permission.RECEIVE_SMS (intercept SMS messages)
* android.permission.READ_PHONE_STATE (read phone details of the device such as phone number and serial number)
* android.permission.CHANGE_WIFI_STATE (connect to and disconnect from Wi-Fi networks and make changes to configured networks)
* android.permission.READ_CONTACTS (read all contact data)
* android.permission.READ_SMS (read SMS messages)

Obviously a fairly significant list of permissions of which many are suspicious, especially when combined.



Runtastic sample permission prompt

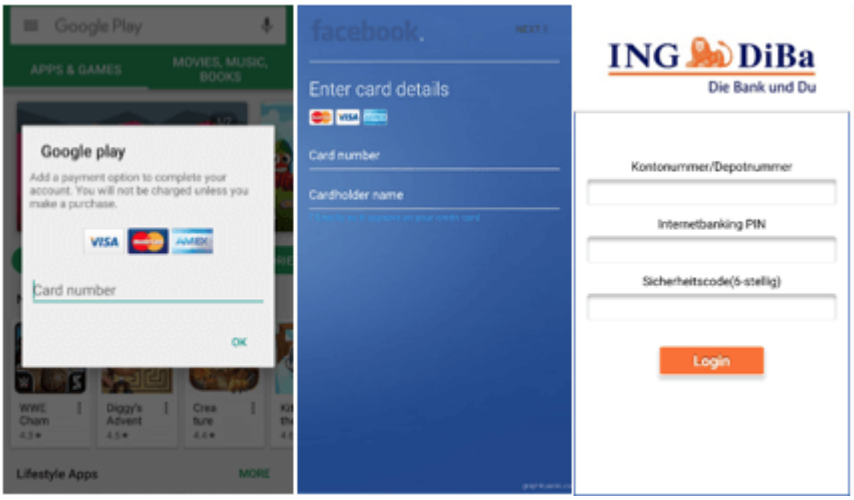
Checking foreground app

Marcher is one of the few Android banking Trojans to use the AndroidProcesses library, which enables the application to obtain the name of the Android package that is currently running in the foreground. This library is used because it uses the only (publicly known) way to retrieve this information on Android 6 (using the process OOM score read from the /proc directory). When the current app on the foreground matches with an app targeted by the malware, the Trojan will show the corresponding phishing overlay, making the user think it is the app that was just started.

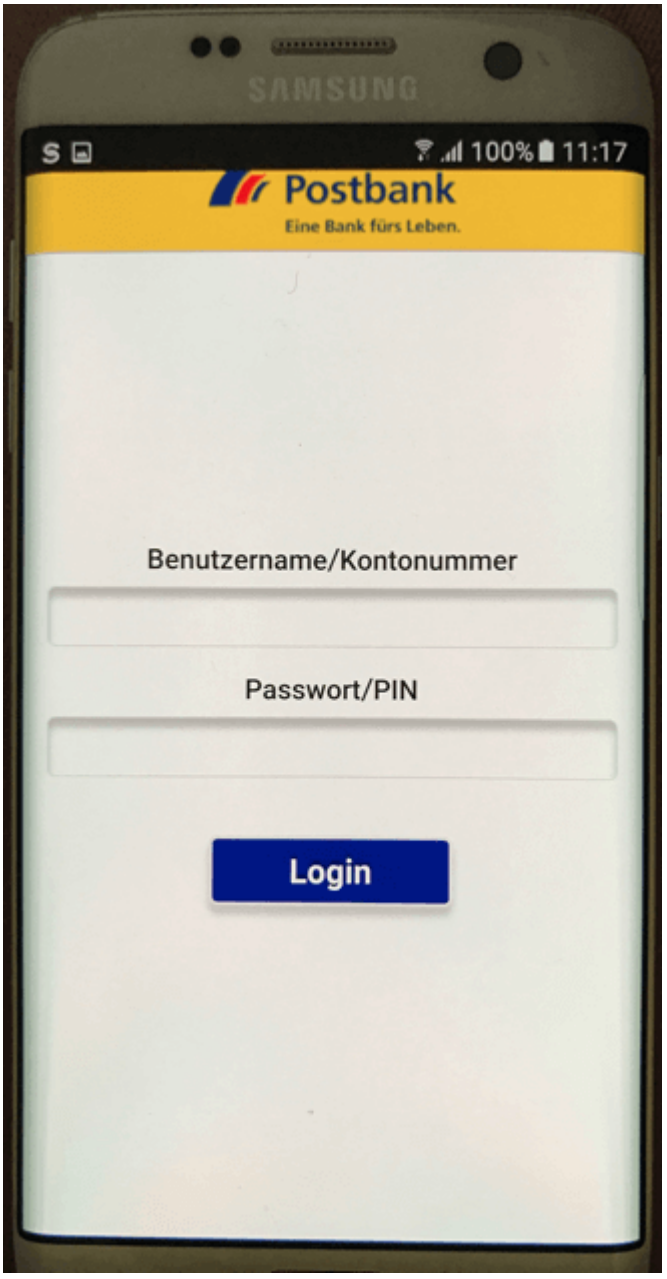
Dynamic overlays

When victims open up a targeted app, Marcher smoothly displays an overlay, a customized WebView, looks in its application preferences (main_prefs.xml) and decides which specified URL is needed for the targeted app. The complete list of apps can be seen below. The phishing pages shown in the overlay use Ajax calls to communicate with a PHP back-end which stores all user input. The C2 backend url looks like this: https://evilhost/c2folder/njs2/?fields[]. There is no way to access the original app again even if victims terminate the overlay process and reopen app, until credit card (name, number, expiry date, security code) and/or bank

information (PIN, VBV passcode, date of birth, etc.) are filled in and verified. The information is then stored in local app database as well as sent to the backend.



Overlays for phishing Google Play, Facebook and ING-DiBa



Overlay for Postbank Finanzassistent

Targeted banking apps * at.bawag.mbanking (BAWAG P.S.K.)
* at.easybank.mbanking (easybank)
* at.spardat.netbanking (ErsteBank/Sparkasse netbanking)
* at.volksbank.volksbankmobile (Volksbank Banking)
* com.bankaustria.android.olb (Bank Austria MobileBanking)
* com.db.mm.deutschebank (Meine Bank)
* com.ing.diba.mbbr2 (ING-DiBa Banking + Brokerage)
* com.isis_papyrus.raiffeisen_pay_eyewdg (Raiffeisen ELBA)
* com.starfinanz.smob.android.sfinanzstatus (Sparkasse)
* de.comdirect.android (comdirect mobile App)
* de.commerzbanking.mobil (Banking)

* de.consorsbank (Consorsbank)
* de.dkb.portalapp (DKB-Banking)
* de.fiducia.smartphone.android.banking.vr (VR-Banking)
* de.postbank.finanzassistent (Postbank Finanzassistent)
* mobile.santander.de (Santander MobileBanking)
* com.barclays.android.barclaysmobilebanking (Barclays Mobile Banking)
* com.grppl.android.shell.BOS (Bank of Scotland Mobile Bank)
* com.grppl.android.shell.CMBlloydsTSB73 (Lloyds Bank Mobile Banking)
* com.grppl.android.shell.halifax (Halifax Mobile Banking app)
* com.htsu.hsbcpersonalbanking (HSBC Mobile Banking)
* com.rbs.mobile.android.natwest (NatWest)
* com.rbs.mobile.android.rbs (Royal Bank, RBS)
* com.rbs.mobile.android.ubr (Ulster Bank ROI)
* uk.co.santander.santanderUK (Personal Banking)
* uk.co.tsb.mobilebank (TSB Mobile Banking)
* com.bbl.mobilebanking (Bualuang mBanking)
* com.kasikornbank.retail.kmerchant (K-PowerPay (mPOS))
* com.scb.phone (SCB EASY)
* ktbcs.netbank (KTB netbank)
* ar.com.santander.rio.mbanking (Santander Río)
* br.com.bb.android (Banco do Brasil)
* cl.santander.smartphone (Banca Personas)
* co.com.bbva.mb (BBVA Colombia)
* com.bancodebogota.bancamovil (Banco de Bogotá)
* com.bancomer.mbanking (Bancomer móvil)
* com.bapro.movil (Banco Provincia)
* com.bbva.nxt_argentina (BBVA Francés | Banca Móvil AR)
* com.bbva.nxt_peru (BBVA Continental - Banca Móvil)
* com.bcp.bank.bcp (Banca Móvil BCP)
* com.citibanamex.banamexmobile (Citibanamex Móvil)
* com.grupoavalav1.bancamovil (AV Villas App)
* com.itau (Banco Itaú)
* com.mosync.app_Banco_Galicia (Banco Galicia)
* com.santander.app (Santander Brasil)
* com.todo1.davivienda.mobileapp (Davivienda Móvil)
* com.todo1.mobile (Bancolombia App Personas)
* mx.bancosantander.supermovil (Supermóvil)
* org.banelco (Banelco MÓVIL)
* org.microemu.android.model.common.VTUserApplicationLINKMB (Link Celular)
* pe.com.interbank.mobilebanking (Interbank APP)
* se.accumulate.me.core.androidclient.csb (Bancoomeva Móvil)
* se.accumulate.me.core.androidclient.occidente (Banco de Occidente B.P)

* au.com.bankwest.mobile (Bankwest)
* au.com.ingdirect.android (ING DIRECT Australia Banking)
* au.com.nab.mobile (NAB)
* com.commbank.netbank (CommBank)
* org.banksa.bank (BankSA Mobile Banking)
* org.stgeorge.bank (St.George Mobile Banking)
* org.westpac.bank (Westpac Mobile Banking)
* com.chase.sig.android (Chase Mobile)
* com.citi.citimobile (Citi Mobile®)
* com.schwab.mobile (Schwab Mobile)
* com.wf.wellsfargomobile (Wells Fargo Mobile)
* de.ing_diba.kontostand (ING-DiBa Kontostand)
* de.adesso.mobile.android.gadfints (Online-Filiale+)
* com.starfinanz.mobile.android.dkbpushtan (DKB-pushTAN)
* com.starfinanz.smob.android.sbanking (Sparkasse+)
* com.kasikorn.retail.mbanking.wap (K-Mobile Banking PLUS)
* com.scb.tablet (SCB EASY for Tablet)
* com.SCBBizNet (SCB Business Net)
* com.scbup2me (SCB UP2ME)
* th.co.ktam.ktampvd (KTAM PVD)
* com.ktb.bizgrowing (KTB Biz Growing)
* com.ing.mobile (ING Bankieren)
* com.caisseepargne.android.mobilebanking (Banque)
* fr.lcl.android.customerarea (Mes Comptes - LCL pour mobile)
* net.bnpparibas.mescomptes (Mes Comptes BNP Paribas)
* com.cic_prod.bad (CIC)
* com.fullsix.android.labanquepostale.accountaccess (La Banque Postale)
* fr.banquepopulaire.cyberplus (Cyberplus)
* fr.creditagricole.androidapp (Ma Banque)
* mobi.societegenerale.mobile.lappli (L’Appli Société Générale)
* pt.santandertotta.mobileparticulares (Santander Totta)
* wit.android.bcpBankingApp.millennium (Millenniumbcp)
* com.IngDirectAndroid (ING Direct France)
* fr.bred.fr (BRED)
* fr.lcl.android.entreprise (Pro & Entreprises LCL)
* mobi.societegenerale.mobile.lapplipro (L’Appli Pro Société Générale)
* com.axabanque.fr (AXA Banque France)
* com.fpe.comptenickel (Mon Compte-Nickel)
* com.carrefour.bank (Carrefour Banque)
* com.bnpp.easybanking (Easy Banking)
* com.paypal.android.p2pmobile (PayPal)
* com.westernunion.moneytransferr3app.eu (Western Union International)

* fr.banquepopulaire.cyberplus.pro (Cyberplus PRO)
* com.akbank.android.apps.akbank_direkt (Akbank Direkt)
* com.akbank.softotp (Akbank Direkt Şifreci)
* com.teb (CEPTETEB)
* com.finansbank.mobile.cepsube (QNB Finansbank Cep Şubesi)
* com.garanti.cepbank (Garanti CepBank)
* biz.mobinex.android.apps.cep_sifrematik (Garanti Cep Şifrematik)
* com.garanti.cepsubesi (Garanti Mobile Banking)
* com.tmobtech.halkbank (Halkbank Mobil)
* com.ingbanktr.ingmobil (ING Mobil)
* com.pozitron.iscep (İşCep)
* com.intertech.mobilemoneytransfer.activity (fastPay)
* com.tmob.denizbank (MobilDeniz)
* tr.com.sekerbilisim.mbank (ŞEKER MOBİL ŞUBE)
* com.vakifbank.mobile (VakıfBank Mobil Bankacılık)
* com.ykb.android.mobilonay (Yapı Kredi Kurumsal Mobil Şube)
* com.ykb.androidtablet (Yapı Kredi Mobil Şube)
* com.ykb.android (Yapı Kredi Mobile)
* com.ziraat.ziraatmobil (Ziraat Mobil)
* com.akbank.android.apps.akbank_direkt_tablet (Akbank Direkt Tablet)
* com.zentity.sbank.csobsk (SmartBanking SK)
* cz.csob.smartbanking (ČSOB SmartBanking)

Other targeted apps (credit card overlay) * com.instagram.android (Instagram)
* com.android.vending (Play Store)
* com.facebook.katana (Facebook)
* com.skype.raider (Skype)
* com.viber.voip (Viber)
* com.whatsapp (WhatsApp Messenger)
* com.google.android.gm (Gmail)
* com.amazon.mShop.android.shopping (Amazon Shopping)

Antivirus “evasion”

In addition to the list of apps that are targeted for phishing the app contains a list of antivirus applications for which it prevents removal of the malware. The technique used is quite simpe: look for any AV app in the list and if it is running, the malware will force the phone back to home screen. Even the AV program detects the malware, it will still wait and ask for permission from users before starting the removal process, but because the user can’t give the permission, the malware will not be removed.

```
if(vyn.hhsdzgvoexobmkygffzwuewrbikzud.a.a.d.length > 0) {
    int v0 = 0;
    while(v0 < vyn.hhsdzgvoexobmkygffzwuewrbikzud.a.a.d.length) {
        if(c.a(vyn.hhsdzgvoexobmkygffzwuewrbikzud.a.a.d[v0], v3, v6)) {
            Intent v0 = new Intent("android.intent.action.MAIN");
            v0.addCategory("android.intent.category.HOME");
            v0.setFlags(268435456);
            arg11.startActivity(v0);
            return;
        }
        else {
```

Check if AV running

Go to homescreen

ode snippet showing the malware forcing the device back to the home screen

The following antivirus apps are targeted with this technique: * com.clean.booster.security.battery.memory (Booster Cleaner)
* com.qihoo.security.lite (360 Security Lite)
* com.piriform.ccleaner (CCleaner)
* com.antivirus.tablet (Tablet AntiVirus FREE 2017)
* com.dianxinos.optimizer.duplay (DU Speed Booster & Cleaner)
* com.womboidsystems.antivirus.security.android (Antivirus Go Next for Android™)
* com.trustlook.antivirus (Free Antivirus & Security)
* com.avast.android.mobilesecurity (Mobile Security & Antivirus)
* com.cleanmaster.mguard (Clean Master (Boost&Antivirus))
* com.qihoo.security (360 Security - Antivirus)
* com.symantec.mobilesecurity (Norton Security and Antivirus)
* com.cleanmaster.security (CM Security AppLock AntiVirus)
* com.duapps.antivirus (DU Antivirus - Lock app, video)
* com.antivirus (AVG AntiVirus FREE for Android)
* com.cleanmaster.boost (CM Speed Booster | Cache Cleaner)
* com.zrgiu.antivirus (Antivirus Free - Virus Cleaner)
* com.kms.free (Kaspersky Antivirus & Security)
* com.nqmobile.antivirus20 (NQ Mobile Security & Antivirus)
* com.cleanmaster.mguard (Clean Master (Boost&Antivirus))
* com.drweb (Anti-virus Dr.Web Light)
* com.bitdefender.antivirus (Bitdefender Antivirus Free)
* com.avira.android (Avira Antivirus Security)
* com.ikarus.mobile.security (IKARUS mobile.security)

SMS harvesting

At startup, Marcher will ask for read/write permissions for both SMS and MMS messages if it doesn’t have the permissions already. Then, whenever the client received command ‘load_sms’ from the C2 server, it will grab all SMS messages from the device and send them back to the backend. In the same way, this method also is used to invoke ‘processIncomingMessages’ to intercept incoming messages.

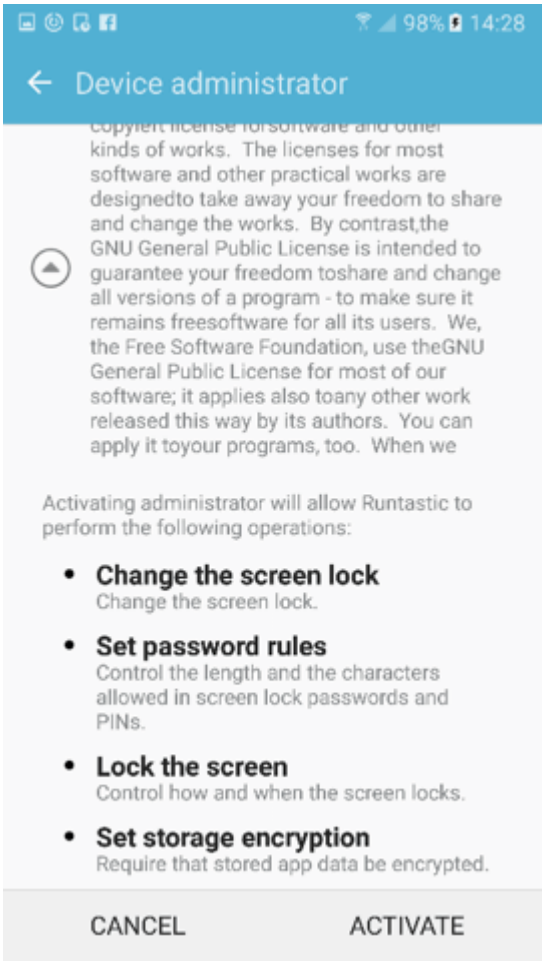
```
@Keep public void processIncomingMessages(Context arg12, Intent arg13) {
    String v10 = null;
    if(this.b.b()) {
        Object v0 = arg13.getExtras().get(vyn.hhsdzgvoexobmkygffzwuewrbikzud.a.b.bw);
        SmsMessage[] v4 = new SmsMessage[v0.length];
        int v2;
        for(v2 = 0; v2 < v0.length; ++v2) {
            try {
                v4[v2] = SmsMessage.class.getMethod(vyn.hhsdzgvoexobmkygffzwuewrbikzud.a.b.bx, byte[].class).invoke(null, v0[v2]);
            }
```

PDUs

createFromPdu

Smartly using permissions

When the malware first runs, it will ask for device administrative rights, even when users deny or kill the process it will come up again, until they accept the request. Having this permission enables malware to lock and mute the phone, even reset the password and make a permanent phishing WebView. This malicious activity works similar to ransomware, but no files are encrypted.



Device admin “nagging” screen

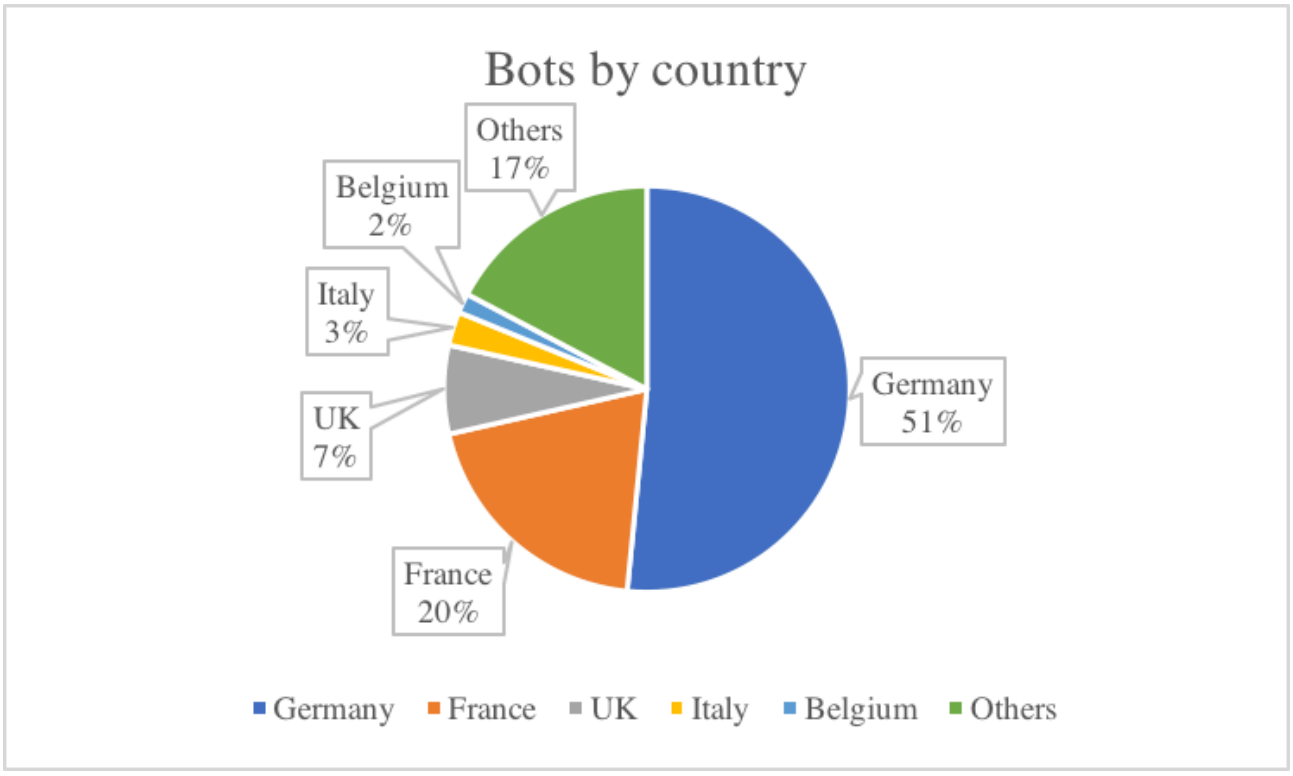
```
public static void a(Context arg9) {
    Object v0 = arg9.getSystemService("audio");
    int v1 = ((AudioManager)v0).getStreamVolume(5);
    ((AudioManager)v0).setStreamVolume(5, 0, 0);
    int v2 = ((AudioManager)v0).getStreamVolume(0);
    ((AudioManager)v0).setStreamVolume(0, 0, 0);
    int v3 = ((AudioManager)v0).getStreamVolume(1);
    ((AudioManager)v0).setStreamVolume(1, 0, 0);
    int v4 = ((AudioManager)v0).getStreamVolume(2);
    ((AudioManager)v0).setStreamVolume(2, 0, 0);
    int v5 = ((AudioManager)v0).getStreamVolume(3);
    ((AudioManager)v0).setStreamVolume(3, 0, 0);
    int v6 = ((AudioManager)v0).getStreamVolume(4);
    ((AudioManager)v0).setStreamVolume(4, 0, 0);
    d.c(arg9, v1 + "," + v2 + "," + v3 + "," + v4 + "," + v5 + "," + v6);
    ((AudioManager)v0).setRingerMode(0);
}

@TargetApi(value=8) public static void a(Context arg3, String arg4) {
    Object v0 = arg3.getSystemService("device_policy");
    if(((DevicePolicyManager)v0).isAdminActive(new ComponentName(arg3, p060s.class))) {
        ((DevicePolicyManager)v0).resetPassword(arg4, 0);
    }
}


@TargetApi(value=8) public static void c(Context arg3) {
    Object v0 = arg3.getSystemService("device_policy");
    if(((DevicePolicyManager)v0).isAdminActive(new ComponentName(arg3, p060s.class))) {
        ((DevicePolicyManager)v0).lockNow();
    }
```

Code snippet showing code to reset the password or lock the device

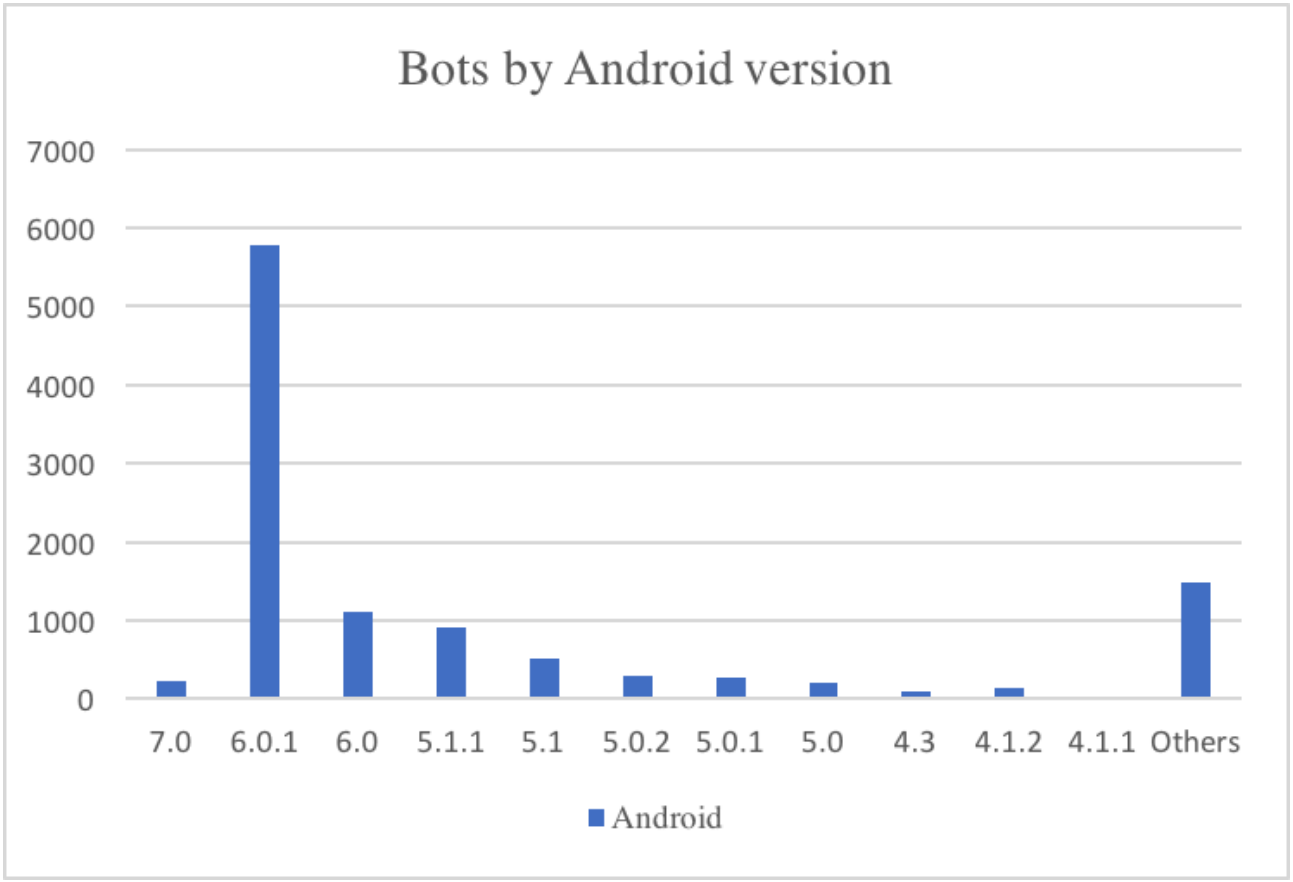
Different botnets

We have researched various Marcher actors the last 6 months. Many of them targeting financials in Germany, France, UK and the United States. The latest samples are mainly targeting banks from Germany, Austria and France. Based on their own Trojan user manual we know that there are at least 9 Marcher actors with their own botnets supported by the original creators of the Trojan with new modules and targeted banks/webinjects (HTML overlay files) every week. The following botnets where observed by our team: * flexdeonblake
* angelkelly
* MUCHTHENWERESTO
* balls51
* CHECKPIECEUNTIL
* crystalknight
* jadafire
* sinnamonlove
* CONTAINSURE

The details in this blog are based on an assessment of only one Marcher actor/botnet. Based on statistics of the backend we know that their campaign has successfully infected 5696 German and 2198 French mobile devices over total of 11049 affected mobile devices. While assessing their C2 server, we found that most infected devices are running Android 6.0.1. The C2 server at the time of investigation contained at least 1300 credit card numbers and other bank information (username/password + SMS tan).



Bot amount by country



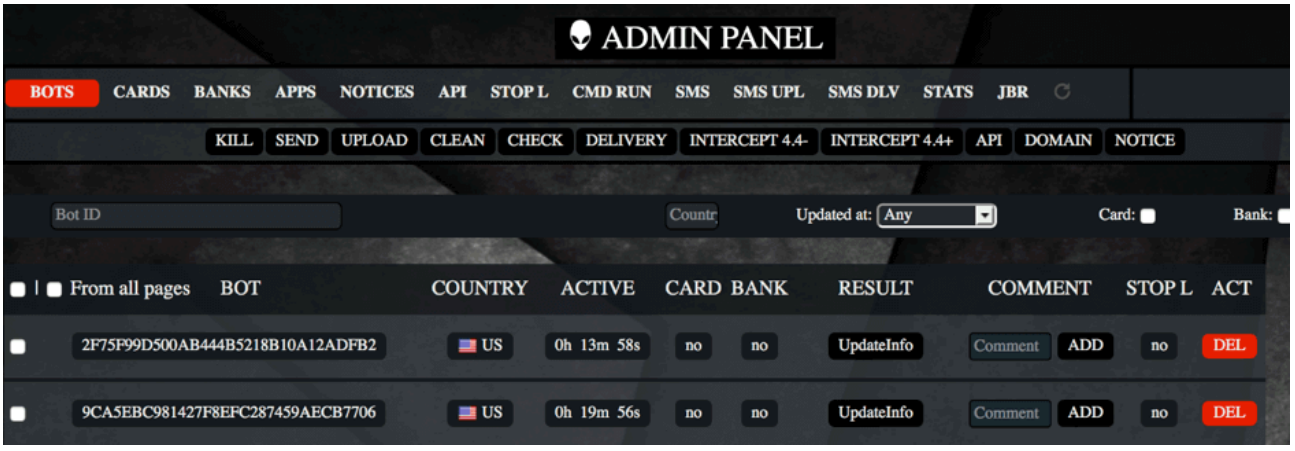
Bot amount by Android version

C2 panel features

Besides information obtained through phishing, the C2 server collects the following information from infected devices: * IMEI
* phone number
* IP address
* carrier name
* SMS messages
* contact phone numbers
* installed packages

It can also instruct devices to send an SMS message, lock the screen showing a webpage and run USSD commands for call-forwarding. The panel also has a feature to control bot via SMS messages using commands like: * rent&&& (intercept and forward SMS)
* ussd&&& (call USSD code)
* sent&&& (send SMS from bot (e.g: sent&&&+31000000#sms_body))
* killStart (lock phone with password, disable screen permanently)
* killStop (undo the killStart changes)

Other options the panel has are chaning the backend URL and creating, enabling and disabling web injects which allows for a lot of flexibility concerning the targeted apps and displayed screens. The phishing screens are hosted on the C2 server and are loaded from there at the time the screen is displayed.



Panel menu:

- Stats — statistics of bots by country/Android version
- Bots — list of bots with information for each bot:
  - is CC data collected (icon will be green)
  - is bank apps (webinject) data collected (icon will be green)
  - is bot in black list (red alert icon)
  - IMEI
  - country flag
  - carrier name
  - last connect time
  - result of last command
  - is SMS intercept enabled
  - set phone number (as note)
  - set comment (as note)
- SMS — all sms from bots
- Cards — CC data from grabber (card number, expire, CVC)
- Banks — data from webinjects (all that user entered in bank apps fake login forms)
- Contacts — phones and names grabbed from bots. Paid extra feature
- Apps — apps installed on bots. Total count means number of bots having this app
- Settings:
  - Apps black list — bots contains these apps will be marked as Blacklisted. For example, API Server will be changed only on Blacklisted bots.
  - Run commands — list of commands that will be executed on each new bot.
  - App mass — list of file names or URLs for "Appmass" command in Bots menu. Each file contains phones list. Then it will send lists to bots. 1 bot = 1 list. Names of used files will be deleted from this field.
  - Change admin panel URL — change URL of admin panel.
  - Show regular notifications — send notifications regularly (title, description, custom package name)
  - Show notify each N minutes — show notifies on bots each N minutes
- Refresh — reload content of the current page

Guest stats:

Link to guest stats placed in "Stats" section.
You need only basic-authorization to see guest statistics.
You can clear statistics (with Clear link) — to show only new bots, or restore (Restore link) previous values — to show all bots existing in panel.

Bots menu:

- Send SMS — send SMS from a bot
- Screenlock — lock screen with a webpage
- Mass SMS Spam — send SMS to all contacts
- Intercept — SMS intercept
- Set webinjects — packageName of the app / url of webpage to show above the app
- Appmass — run commands from Settings/Appmass field on the bot
- Send Command — send custom command to the bot
  - Enable sms intercept
  - Disable sms intercept

    - Update bot info
    - Disable screen lock
    - Set new admin phone
    - Kill — lock screen, no sound, change password
    - Disable kill — unfreeze app, remove password
    - Repeat filled inject — set active inject that was already submitted by user
- Notification — send notification (title, description, custom package name)
- Admin phone — set admin phone number (for forwarding messages, etc)
- Admin URL — set new admin panel url; "As BL alert" — mark bot as blacklisted in the current panel

Selected bot menu:

- Send USSD - run USSD command
- Block webinject - stop showing a web window above the specified app
- Unblock webinject - start showing web window that was disabled early
- Set webinjects - set pairs App-URL for web injects
- Update info - force bot to send its info to admin panel
- Show SMS - switch panel to SMS page filtered by current bot ID
- Show cards - switch to Cards page filtered by current bot ID
- Show banks - switch to Banks page filtered by current bot ID

Control bot by SMS (from admin phone):

- @DELETE - disable SMS intercept
- rent&&& - set SMS intercept (SMS goes to admin phone)
- ussd&&& - execute USSD code
- sent&&& - send SMS from bot. Format: sent&&&+31000000#sms_body
- killStart - lock phone with password, disable screen permanently
- killStop - unlock for previous command

How to redirect/Forward calls:

- You can forward calls to a target phone number by simply sending USSD command.
  How to compose USSD command read here — https://en.wikipedia.org/wiki/Call_forwarding

How to set new webinjects into existing bots:

- Get list of your active injects from support
- It will looks like: com.package:333, where com.package - name of the app package and 333 - unique index of the inject
- Open "Set webinjects" command in bot menu
- Make inject URL. Example: https://your-domain/YOUR_FOLDER/njs2/?m=333 where 333 is an unique index of the inject
- Put package name (com.package) to first field of the command and inject URL to second

Marcher documentation

The source code of the Marcher C2 server indicate that they successfully implemented a SOCKS feature for bots and are selling this as a separate module. Socks enables the attackers to perform malicious transactions using the victim's device and IP. This feature could be enabled to circumvent detection of financial institutions that relies on device binding and the IP address of the customer's Android device.

Source code snippet showing SOCKS functionality

```
 replace: 1111 - > last_ip(ip of bot) $SOCKS_status = "Offline";
if ($row["SOCKS_status"]) {
    $SOCKS_status = $row["SOCKS_status"];
    if (isset($_SERVER["HTTP_X_FORWARDED_HOST"])) $SOCKS_status = str_replace("", $_SERVER["HTTP_X_FORWARDED_HOS
    else$SOCKS_status = str_replace("", $_SERVER["REMOTE_ADDR"], $SOCKS_status);
}
if (!isset($client_cfg['mod_SOCKS']) || !$client_cfg['mod_SOCKS']) {
    $data = "<div style='padding: 20px'>      <h2>SOCKS: <b style='color: red'>disabled</b></h2>      <br />
    return $data;
}
```

Samples

Below are the most recent Marcher samples we have come across. There are however many more out there. Samples can be obtained from for example Koodous.

Netflix BETA
* Package name: iyq.bmjhaqtqndshhxmrzeyxaaepaxxahy
* SHA256: b087728f732ebb11c4a0f06e02c6f8748d621b776522e8c1ed3fb59a3af69729

Postbank
* Package name: jihpynmjnsftqlslbg.iraqakpzzdzspqbneq
* SHA256: 5bb9b9173496d8b70093ef202ed0ddddd48ad323e594345a563a427c1b2ebc22

Youporn
* Package name: cisfm.rygkfxpsyyldznvjufubiacoriibbx
* SHA256: c8f753904c14ecee5d693ce454353b70e010bdaf89b2d80c824de22bd11147d5

Android Update
* Package name: mor.yehoeiphksbxbwfigcopschkhfxpkj
* SHA256: c172567ccb51582804e589afbfe5d9ef4bc833b99b887e70916b45e3a113afb8

DHL Express Mobile
* Package name: ijrtc.jwieuvxpjavuklczxdqecvhrjcvuho
* SHA256: fcd18a2b174a9ef22cd74bb3b727a11b4c072fcef316aefbb989267d21d8bf7d

Mobilfunknetz Update
* Package name: com.tpvxjnxophkekmritrhjyeyrbnfsyl
* SHA256: a1258e57c013385401d29b75cf4dc1559691d1b2a9afdab804f07718d1ba9116

Bloomberg PRO
* Package name: djgd.zvnnpjllwxrnqcvdonprixxpizlfzg
* SHA256: a1258e57c013385401d29b75cf4dc1559691d1b2a9afdab804f07718d1ba9116

Alzashop.com
* Package name: atlk.ussdpifhzgedqrysfygranbxmffhck

* SHA256: ed2b26c9cf4bc458c2fa89476742e9b0d598b0c300ab45e5211f29dfd9ddd67b

Super Mario Run

* Package name: vlhtc.hsicifsgxehymvdvajyzyckijyatpo
* SHA256: be6c8a4afbd4b31841b2d925079963f3bd5422a5ee5f248c5ed5013093c21cf9

Runtastic

* Package name: zwhp.nbneaijecxwskcxtlkvmnqkryxgdgq
* SHA256: ec4d182b0743dbdedb989d4f4cb2d607034ee1364c30103b2415ea8b90df8775

Whatsapp Security

* Package name: com.wood
* SHA256: 5a9e3d2c2ef29b76c628e70a91575dc4be3999b60f34cab35ee70867faaff4a0

Postbank Sicherheitszertifikat

* Package name: zcdr.kmvxvlidqpezvegypetddrutebanrp
* SHA256: 5df132235eccd1e75474deca5b95e59e430e23a22f68b6b27c2c3a4aeb748857

Pošta Online

* Package name: nkl.gewpfvqsnxehngqtzjlhrcqivqsqhw
* SHA256: 25e07c50707c77c8656088a9a7ff3fdd9552b5b8022d8c154f73dca1e631db4f

360 Security

* Package name: com.p360courv
* SHA256: f7743a01fc80484242d59868938ec64990c19bea983fb58b653822c9ee3306a1

Volksbank Sicherheitszertifikat

* Package name: amise.syrwhshjopuvyqhrucvvosjjcnrbrz
* SHA256: 6f8b7aa6293238d23b1c5236d1c10cecc54ec8407007887e99ea76f9fce51075

ING Beveiligingsupdate

* Package name: com.ingbvupdd
* SHA256: 7f08cc20aa6e1256f6a8db3966ac71ad209db6dff14a6dde0fd7b2407c2c23e7

Google Play Services

* Package name: cosmetiq.fl
* SHA256: b4e5affbc3ea94eb771614550bc83fde85f90caddcca90d25704c9a556f523da

C2s

hxxps://loupeacara.net/flexdeonblake/
hxxps://sarahtame.at/flexdeonblake/
hxxps://loupeahak.com/flexdeonblake/
hxxps://chudresex.at/flexdeonblake/
hxxps://chudresex.cc/flexdeonblake/
hxxps://memosigla.su/flexdeonblake/

hxxps://rockybalboa.at/angelkelly/
hxxps://storegoogle.at/angelkelly/
hxxps://trackgoogle.at/angelkelly/
hxxps://track-google.at/angelkelly/
hxxps://coupon-online.fr/angelkelly/
hxxps://inovea-engineering.com/angelkelly/
hxxps://lingerieathome.eu/angelkelly/
hxxps://playgoogle.at/angelkelly/
hxxps://i-app5.online/MUCHTHENWERESTO/
hxxps://i-app4.online/MUCHTHENWERESTO/
hxxps://i-app1.online/MUCHTHENWERESTO/
hxxps://176.119.28.74/balls51/
hxxps://soulreaver.at/balls51/
hxxps://olimpogods.at/balls51/
hxxps://divingforpearls.at/balls51/
hxxps://fhfhhhrjtfg3637fgjd.at/CHECKPIECEUNTIL/
hxxps://dfjdgxm3753u744h.at/CHECKPIECEUNTIL/
hxxps://dndzh457thdhjk.at/CHECKPIECEUNTIL/
hxxps://playsstore.mobi/QUESTIONROADFAR/
hxxps://secure-ingdirect.top/CHECKPIECEUNTIL /
hxxps://playsstore.net/QUESTIONROADFAR/
hxxps://compoz.at/crystalknight/
hxxps://cpsxz1.at/crystalknight/
hxxps://securitybitches3.at/jadafire/
hxxps://wqetwertwertwerxcvbxcv.at/jadafire/
hxxps://securitybitches1.at/jadafire/
hxxps://ldfghvcxsadfgr.at/jadafire/
hxxps://weituweritoiwetzer.at/jadafire/
hxxps://wellscoastink.biz/jadafire/
hxxps://deereebee.info/jadafire/
hxxps://ssnoways.info/jadafire/
hxxps://elitbizopa.info/jadafire/
hxxps://filllfoll.biz/jadafire/
hxxps://bizlikebiz.biz/jadafire/
hxxps://barberink.biz/jadafire/
hxxps://nowayright.biz/jadafire/
hxxps://messviiqqq.info/jadafire/
hxxps://qqqright.info/jadafire/
hxxps://sudopsuedo1.su/sinnamonlove/
hxxps://sudopsuedo2.su/sinnamonlove/
hxxps://sudopsuedo3.su/sinnamonlove/

hxxps://androidpt01.asia/CONTAINSURE/
hxxps://androidpt02.asia/CONTAINSURE/

Preventing infection

Users should avoid downloading apps from a third-party and only use Google Play Store (so do not enable installation from unknown sources). Take note however that even in the Google Play Store apps are not necessarily malware free. Check if the requested privileges correspond with the expected privileges of the app you want to install. Also, never click on a suspicious link in SMS and email messages even it is from trusted contacts.

Conclusion

Marcher is growing into a mature Trojan with solid organization behind it like many of the banking malware variants we have seen over the years on the Windows platform (Sinowal/Torpig, Dyre, Dridex, Gozi, etc.). Development of new features and support for newer Android versions is ongoing and we will be keeping an eye on it to see where things are going. The main actors of Marcher appear to not only make money off the stolen credentials but also from providing their Trojan to other groups and selling new capabilities such as the SOCKS module and new injects.

Based on the statistics we found on this one C2 panel we researched and the amount of different C2 panels out there, we believe that the potential financial losses due to Android banking Trojans are, or will soon be, bigger than the current losses from desktop malware like Gozi and Dridex, especially since hardly any of the banking apps seem to detect the attack.

If you yourself want to remain safe from malware, be vigilant when installing new applications on your device and try to keep your device up-to-date. From Google's side we are seeing some improvements on the Android platform itself to combat techniques used by malware. The current techniques used to retrieve the foreground app for example are no longer working on Android 7. However, more can be done to improve platform security, especially around the "Unknown Sources" setting. To quote Yorick Koster, who has been testing mobile app security on all platforms for many years: "all these MDM solutions require you to enable Untrusted Sources. I don't know why Google doesn't have a solution for this yet, like Apple does with enterprise certificates".

We have been worrying about security on desktop computers for decades. Now, with mobile malware on the rise, it's about time everyone starts worrying about mobile device security, especially considering that for many targeted financials most transactions these days take place on mobile devices.

Securify's Client Side Detection for Android solution

For organizations interested in detecting Android banking malware on customer devices, please contact us to learn how our CSD solution can adaptively detect merging mobile and web banking Trojan threats.
