Will 2018 Be Another Year of Bankbots?

The malware family continues to frustrate efforts to keep it off Google Play
By Jagadeesh Chandraiah

Last year saw continued growth of malware attempting to cheat Google’s own security service, Play Protect, in order to make its way into the Google Play store.

Of all the active Android malware families over the last year, Bankbot has proven to be especially resilient, given its repeated appearances on Google Play and its ability to infect thousands of users. It’s very likely that Bankbot will continue to be a force to be reckoned with throughout 2018.

As reported by CERT Polska in January 2018, Bankbot operators are still very much active, stubbornly continuing to push this persistent malware pest into the public domain.

Since we expect to see much more of this threat, we set out to understand its inner workings more clearly. To that end, this paper looks into its evolutionary path, infection cycle, and the tricks it hides up its sleeves.
The Timeline

The Bankbot family consists of many ‘relatives’ – slight variations of the code that basically do the same thing, stealing credentials of online banking users.

Over the last year, Bankbot showed up on Google Play several times, disguised under different ‘identities’ as demonstrated with the timeline below:

As the chart below demonstrates, most Sophos-reported Bankbot infections are concentrated across Europe. Some of the infections were also reported in China.

The top 10 countries affected by Bankbot over the last six months are:

- Bulgaria 3%
- Poland 4%
- Brazil 4%
- France 4%
- Belgium 4%
- China 11%
- Japan 24%
- Germany 24%
- Britain 4%
- China 11%
Leaked Source Code

One of the main reasons behind the wide proliferation of this family of malware is that Bankbot’s source code was leaked in late 2016.

Soon after it was published, beginning in early 2017, Bankbot variants started to appear in the wild. One of the newbie authors, nicknamed “maza-in,” published source code that is similar to the original leaked code.

Building on top of the Bankbot foundation, he published on such forums as Android app source code and PHP code for the admin panel. A Google translation of his post is provided below:

![Android BOT from scratch](https://i.imgur.com/3QX5y.png)

The post, titled “Android BOT from scratch,” enlists various features, such as Admin rights request, SMS handling, and web injections support.

The admin panel provides a detailed view on all bots, such as International Mobile Equipment Identity (IMEI), phone number, Android operating system version, APK version, device location (country) and model name, what online banking app the victim is using, whether the device is rooted or not, whether it’s powered on or off, date of infection, and all the additionally collected logs, as shown below:

![Admin panel](https://i.imgur.com/3QX5y.png)

---

1. [https://securityintelligence.com/after-big-takedown-efforts-20-more-bankbot-mobile-malware-apps-make-it-into-google-play/](https://securityintelligence.com/after-big-takedown-efforts-20-more-bankbot-mobile-malware-apps-make-it-into-google-play/)
2. [https://vms.drweb.com/virus/?i=14895561&lng=en](https://vms.drweb.com/virus/?i=14895561&lng=en)
3. [https://github.com/bemre/bankbot-mazain](https://github.com/bemre/bankbot-mazain)
The PHP code serves the main admin panel page and provides interface for remote commands and configurations. Android application’s source code includes prototype code to obtain device admin rights, harvest banking data, and manage SMS in order to evade a bank’s two-factor authentication. Such code is pretty much common for the entire family of the Bankbots.

### Rise of the Bankbots

As we learned from the past, any malware source code leak, just as it happened to ZeuS, bears a risk of spawning a large wave of replicas. Looking at Bankbot from this perspective, it can be thought of as a ZeuS equivalent for the world of Android. Soon after initial source code publishing, Bankbot-derived samples and C2 servers start popping up both in the wild and on Google Play store.

### Bankbot: the 1st Generation

The first generation of the Bankbots, born in early 2017, used fake overlays along with obtained device administrator privileges to mount attacks against users.

The flowchart below is typical for the first generation of the Bankbots:
A typical Bankbot from this generation appears as a flash utility, a system update, or some other utility app. When the user installs the app, it immediately asks for device admin privileges.

At the same time, working in the background, the app also submits a beacon request containing root/screen status to the remote C2 server. This request is submitted into a remote PHP script called tuk_tuk.php (‘tuk-tuk’ is translated from Russian as ‘knock-knock’).

Next, it collects device location and checks what known banking applications are installed on the device. For that, it uses a hard-coded list of banking apps:

```
com.kuveytturk.mobil
com.magiclick.odeabank
com.isis_papyrus.raiffeisen_pay_eyewdg
com.spardat.netbanking
com.bawag.mbanking
com.volksbank.volksbankmobile
com.bankaustria.android.obl
com.easybank.mbanking
```

The collected data is submitted to C2’s set data.php script.

Following that, Bankbot waits for the user to launch the banking app. Sooner or later, this will happen, and when it happens, the malware fetches a fake page from C2 that mimics the banking application page. This page is then laid over the original page (hence the name ‘overlay’) so that the original page is hidden behind the fake one.

The user then enters his or her banking credentials into the fake overlay page, unaware that now the entered credentials are intercepted by the cyber criminals.

The stolen credentials are delivered to the attackers so that they can wire money into money mule accounts. This step, however, is protected against by the bank’s two-factor authentication (2FA), typically based on SMS.

**SMS Interception Module**

In order to defeat a bank’s 2FA, the Bankbot sets itself up as a default application that handles all SMS messages. This way, it monitors all incoming message and is able to delete any message it wants.

When the attackers attempt to wire the funds from a compromised banking account, the bank will send SMS message, and the app will intercept that message. The app deletes the message from the device (so the user never sees it), and forwards it instead to the attackers by sending in back to C2. Once the attackers receive intercepted message, then can complete the fraudulent transaction by using the intercepted code for 2FA.
Bankbot even mutes the device’s ability to ring or vibrate in order to intercept SMS stealthily:

![Code snippet](https://i.imgur.com/3Q2Q5Q.png)

**Bankbot: the 2nd Generation**

First generation bankbots needed the victim to enable device’s admin privileges. Many users soon became suspicious, lessening the impact of this tactic.

To overcome this obstacle, a new variant⁴ of Bankbot, found on Google Play store, resorted to a social engineering trick based on accessibility service abuse.

First, it pretended to be an innocent looking gaming application:

![Gaming app](https://i.imgur.com/3Q2Q5Q.png)

Once started, it would wait for a few minutes, hoping to evade sandbox analysis. Then, it would pop up a dialog box, asking the user to enable “Google Service”:

![Dialog box](https://i.imgur.com/3Q2Q5Q.png)

⁴ [https://www.zscaler.com/blogs/research/malware-google-play-abusing-accessibility-service](https://www.zscaler.com/blogs/research/malware-google-play-abusing-accessibility-service)
After clicking ‘ok’, the user is prompted to enable accessibility service, pretending to be Google copyright access notice:

Accessibility service is designed to assist users with special needs.

If the user enables this permission for malware, a fake Google Update view is displayed. In the background, the malware receives the Bankbot payload and authorizes itself to be a device admin with the highest level of privileges, allowing itself to use overlays.

Will 2018 Be Another Year of Bankbots?

Foreground Applications
To put the overlay over an existing banking app, Bankbot first needs to know when that app is active. Initially, Bankbot relied on `getRunningTasks()` and `getRunningAppProcesses()`, as shown below:

```java
for (int t = 0; t < -1; t++) {
    try {
        TimeUnit.SECONDS.sleep(1);
    } catch (InterruptedException e) {
        e.printStackTrace();
    }
    String massivedata_s = /*some code*/
    try {
        for (int ttt = 0; ttt < getRunningAppProcesses().length; ttt++) {
            Log.e("Def", "" + getRunningAppInLollipopAndMarshallow(massivedata_s)[ttt]);
        }
    } catch (Exception e) {Log.e("ERROR", "ERROR");}
    try {Thread.sleep(3000);}
    catch (InterruptedException e) {e.printStackTrace();
}
```

Later, the attackers switched to accessibility service. By design, the accessibility service coupled with the SYSTEM_ALERT_WINDOW permission class (now defined by Google as ‘above dangerous’ permission) is a very powerful mechanism, as it can provide apps with higher privileges. Accessibility service allows apps to simulate clicks on buttons from the displayed screen, see what foreground applications are running, and even receive the callbacks on such events like clicks and focus change events. If malware is abusing this service, it can effectively listen to these events and construct the whole context about the device, all the while doing pretty much what it wants in the background.

The code snippet below demonstrates how the malware abuses accessibility service with the `OnAccessibilityEvent` callback and `findNodeInfoByViewId`:

```
public void onAccessibilityEvent(AccessibilityEvent arg12) {
    Object v1;3;
    MyAccessibilityService v0;
    c v1;1;
    int v0 = -2;
    int v0 = 0x10;
    c v2 = new c(this.getSystemService());
    AccessibilityNodeInfo v4 = this.getRootInActiveWindow();
    Log.d("DEMO", "ALLOTRIX-DEMO(\"AA\")");
}
```

Cloak-and-Dagger.org’s team of researchers have shown different types of attacks that are made using SYSTEM_ALERT_WINDOW and accessibility service.\(^6\)

Increase in Accessibility Service Abuse by Malware

Beyond Bankbots, many other malware families are also known to abuse accessibility service. They use it to obtain additional privileges, read data displayed on the device, and change the device pin. Recently, Skygo Free malware abused accessibility service, and ransomware is known to rely on it to gain access to device admin privileges and change device pins.

Over the last six months Sophos has found over 13,000 malicious Android applications abusing accessibility service.

Monetization

The ultimate purpose for cybercriminals is to make money. To be effective, deployed malware targets online banking apps, and this targeting is region-based. That means that, depending on the market, the list of targeted banks can be different.

Apart from online banking, malware authors also target credit card data. After a successful infiltration, the stolen data is sold in underground forums or is used to purchase goods.

The example below demonstrates fake login pages overlaid over the region-based banking apps and an overlay that harvests credit card data.

---

Conclusion
Given that Google Play store is protected with Play Protect, the consistency of Bankbot penetration into the store means that Play Protect alone cannot effectively protect its users from Bankbots.

To be absolutely fair, each time it happened, Google was very prompt in removing all the infected applications from the Play store. Nevertheless, such response is reactive in nature as it always follows an actual store penetration.

Because of extensive abuse of the accessibility service by malware, Google started cracking down on apps that rely on accessibility service in Play store\(^8\).

It’s worth noting that apart from malware, accessibility service is also used by many legitimate applications, such as password managers and security software.

SYSTEM\_ALERT\_WINDOW is used by such popular applications as Skype and Facebook. Hence, uprooting this privilege might not necessarily be as straightforward as it seems.

Fortunately, no Bankbot can change the fundamental principle of staying safe: install apps only from reputable vendors and rely on an antivirus and other IT security solutions you can trust.