

Android'de Kanca Atmak

written by Mert SARICA | 1 January 2021

If you are looking for an English version of this article, please visit [here](#).

Konumuz Android dünyası olsa da kanca atma denilince nedense aklıma ilk olarak enerji nakil hattına kanca atılarak hanelere çekilen kaçak elektrik gelir. Android dünyasında da aslında uygulamaları dinamik olarak analiz etmek veya müdahale etmek istediğimizde de benzer bir yöntem izleriz. Peki buna neden ihtiyaç duyarız ? Kimi zaman Android uygulamaları ile ilgili bir güvenlik araştırması yapmak istediğimizde veya sızma testi esnasında güvenlik zafiyeti bulmak için hedef Android uygulamasını analiz etme ihtiyacı duyarız. Bunun için genellikle ilk olarak hedef Android uygulamasını kaynak koduna çevirip statik kod analizi yapmakla işe başlarız. Fakat günümüzde çoğu Android uygulamasında kodlar gizlenerek (obfuscation) anlaşılması güçleştirildiği için uygulamayı Genymotion gibi öykünücüler (emulator) üzerinde dinamik olarak analiz etmeye çalışırız. Çalışırız dememin bir sebebi ise yine günümüzde Snapchat gibi mobil uygulamaların statik analizin yanı sıra dinamik analizi de engelleme adına çok sayıda yöntemle başvurduğu görülebilmektedir.

2019 yılının Aralık ayında gerçekleştirdiğim Stockholm seyahatim esnasında şehri gezerken etrafımdan sık şıkırdım insanların vızır vızır elektrikli scooterlar ile geçip gittiğini gördüm. Avrupa'da kullanımının son derece yaygın olması sebebiyle elektrikli scooter uygulamalarının zaman içinde güvenlik araştırmacılarının radarına girmesi ile bazılarında güvenlik zafiyetlerinin tespit edildiğini hatırladım. Ülkemizde de yaygınlaşmaya başlayan elektrikli scooterları ve uygulamalarını kullanmaya başlamadan önce Android uygulamalarından bir tanesinin haberleşmesini, merakımı gidermek amacıyla güvenlik araştırmacısı gözüyle incelemeye karar verdim. Tabii ki her zaman olduğu gibi evdeki hesap çarşıya uymadı ve karşıma çıkan engeller sayesinde ortaya bu blog yazısı çıkmış oldu. :)

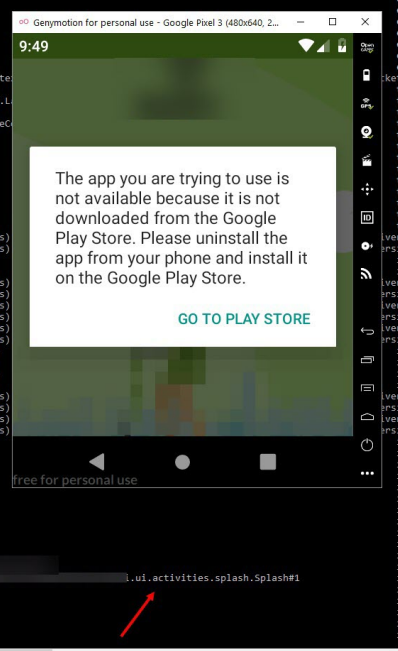
Zamanı kısıtlı bir güvenlik araştırmacısı olarak statik kod analizi ile vakit kaybetmek istemediğim için APK dosyasını APK Downloader web uygulaması ile indirip Geny Motion öykünücüsüne yükledim. Uygulamayı çalıştırdıktan sonra karşıma APK dosyasının Google Play üzerinden indirilmemesi sebebiyle sürpriz bir uyarı mesajı çıktı. :)

The app you are trying to use is not available because it is not downloaded from the Google Play Store. Please uninstall the app from your phone and install it on the Google Play Store.

GO TO PLAY STORE

Bunun üzerine gözü kapalı statik kaynak kodu analizine girmeden önce komut satırında sistem mesajlarını görüntülemek için adb logcat komutunu çalıştırıp ardından da uygulamayı tekrar çalıştırdım. Mesajlar arasından ekrana gelen uyarı mesajı ile ilişkili olduğunu düşündüğüm fonksiyonu bulmak için jadx aracı ile APK dosyasını kaynak koduna çevirdikten sonra activities.splash.Splash dosyasını incelemeye başladım. Dosyanın sonunda yer alan verifyInstallerId fonksiyonu hemen dikkatimi çekti. Bu fonksiyon adını Google arama motorunda arattığımda benzer bir fonksiyonun tam da bu amaçla kullanıldığını gördüm. Bu fonksiyon ile Android'in getInstallerPackageName fonksiyonundan faydalanarak uygulamayı Android'e yükleyen uygulamanın Google Play olup olmadığı kontrol ediliyordu. Şayet uygulama Google Play tarafından işletim sistemine yüklendiyse installerPackagename değişkeni sıfırdan farklı bir değer oluyordu.

```
Command Prompt - adb logcat
11-16 09:49:08.060 4633 4669 D Fabric : Using AdvertisingInfo from Reflection Provider
11-16 09:49:08.073 502 527 I ActivityManager: Displayed /ui.activities.splash.Splash: +1603ms
11-16 09:49:08.101 360 360 I health@2.0-serv: type=1400 audit(0.0:1461): avc: denied ( read ) for name="capacity" dev="fuse" ino=8 scontext=u:r:hal_health_default:s0 tcontext=u:object_r:fuse:s0 tclass=file perm
11-16 09:49:08.101 360 360 I health@2.0-serv: type=1400 audit(0.0:1462): avc: denied ( open ) for path="/dev/pipe/battery/BAT0/capacity" dev="fuse" ino=8 scontext=u:r:hal_health_default:s0 tcontext=u:object_r:fuse:s0 tclass=file perm
11-16 09:49:08.105 360 360 I health@2.0-serv: type=1400 audit(0.0:1463): avc: denied ( getattr ) for path="/dev/pipe/battery/BAT0/capacity" dev="fuse" ino=8 scontext=u:r:hal_health_default:s0 tcontext=u:object_r:fuse:s0 tclass=file perm
11-16 09:49:08.240 372 451 W SurfaceFlinger: Attempting to set client state on removed layer: Splash Screen com.
11-16 09:49:08.240 372 451 W SurfaceFlinger: Attempting to destroy on removed layer: Splash Screen com.
11-16 09:49:08.484 4633 4667 D Analytics : Sending 1 analytics files to https://e.crashlytics.com/spl/v2/events
11-16 09:49:08.544 4633 4667 D Answers : Response code for analytics file send is 200
11-16 09:49:08.988 4633 4691 D CrashlyticsCore: Checking for crash reports...
11-16 09:49:08.989 4633 4691 D CrashlyticsCore: No reports found.
11-16 09:49:23.340 1192 1201 W System : A resource failed to call close.
11-16 09:49:24.473 316 316 I Local opening: type=1400 audit(0.0:1464): avc: denied ( write ) for laddr=192.168.68.103 lport=22468 faddr=192.168.68.2 fport=1772 scontext=
11-16 09:49:24.501 502 525 D AutofillManagerService: Close system dialogs
11-16 09:49:24.513 502 4100 I ActivityManager: START w0 (act-android.intent.action.MAIN cat:[android.intent.category.HOME] flg=0x10000000 cmp=com.android.launcher3/L
11-16 09:49:24.523 640 640 I vol.Events: writeEvent dismiss_dialog volume_controller
11-16 09:49:24.526 440 640 V StatusBar: sStatusBarWindow: com.android.systemui.statusbar.phone.StatusBarWindowView{b4a3852 V.ED..... 0,0-480,36} canPanelBeC
11-16 09:49:24.526 4633 D Answers : Logged lifecycle event: PAUSE
11-16 09:49:24.575 4633 4633 D Answers : Logged lifecycle event: STOP
11-16 09:49:24.624 1302 1307 E EGL_emulation: tid 1307: eglSurfaceAttrib(1354): error 0x3089 (EGL_BAD_MATCH)
11-16 09:49:24.635 1302 1307 W OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xd67f5560, error=EGL_BAD_MATCH
11-16 09:49:24.635 502 2412 I ConfigStore: android:hardware:configstore:vl_0::ISurfaceFlingerConfigs:hasWideColorDisplay retrieved: 0
11-16 09:49:24.636 502 2412 I ConfigStore: android:hardware:configstore:vl_0::ISurfaceFlingerConfigs:hasHDRDisplay retrieved: 0
11-16 09:49:24.636 502 2412 I OpenGLRenderer: Initialized EGL, version 1.4
11-16 09:49:24.636 502 2412 D OpenGLRenderer: Swap behavior 1
11-16 09:49:24.650 502 2412 D EGL_emulation: eglCreateContext: 0xd6dad380: maj 2 min 0 rcv 2
11-16 09:49:24.972 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:24.985 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:24.994 372 451 W SurfaceFlinger: Attempting to set client state on removed layer: Dim Layer for - Task=44#0
11-16 09:49:24.994 372 451 W SurfaceFlinger: Attempting to destroy on removed layer: Dim Layer for - Task=44#0
11-16 09:49:25.011 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.011 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.025 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.025 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.033 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.033 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.049 372 451 W SurfaceFlinger: Attempting to set client state on remove
11-16 09:49:25.049 372 451 W SurfaceFlinger: Attempting to destroy on removed layer:
11-16 09:49:25.052 372 451 W SurfaceFlinger: Attempting to set client state on remove
11-16 09:49:25.052 372 451 W SurfaceFlinger: Attempting to destroy on removed layer:
11-16 09:49:25.121 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.121 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.146 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.147 502 517 W BroadcastQueue: Background execution not allowed: receiving Intent { act=android.intent.action.DROPPBOX_ENTRY_ADDED flg=0x18 (has extras)
11-16 09:49:25.327 1011 1047 I : gms-persistent: Waiting for a blocking GC Profilesaver
11-16 09:49:25.328 1011 1047 W : Empty content. This might mean that the settings are not synced down.
11-16 09:49:25.350 1011 1047 W : Empty content. This might mean that the settings are not synced down.
11-16 09:49:25.368 1011 1047 I : gms-persistent: WaitForGCtoComplete blocked Profilesaver on Profilesaver for 40.860ms
11-16 09:49:25.420 1011 1047 W : Empty content. This might mean that the settings are not synced down.
11-16 09:49:25.511 1011 1047 I chatty : uid=10071(com.google.android.gms) highpool[3] identical 8 lines
11-16 09:49:25.522 1011 1047 W : Empty content. This might mean that the settings are not synced down.
11-16 09:49:25.004 4633 4680 E EGL_emulation: tid 4680: eglSurfaceAttrib(1354): error 0x3089 (EGL_BAD_MATCH)
11-16 09:49:25.005 4633 D Answers : Logged lifecycle event: START
11-16 09:49:25.030 4633 D Answers : Logged lifecycle event: RESUME
11-16 09:49:25.034 502 1220 W ActivityManager: Receiver with filter android.content.IntentFilter@e3667d already registered for pid 4633, callerPackage is c
11-16 09:49:25.080 372 449 D SurfaceFlinger: duplicate layer name: changing
11-16 09:49:25.080 372 449 W SurfaceFlinger: Attempting to destroy on removed layer:
11-16 09:49:25.119 4633 4680 E EGL_emulation: tid 4680: eglSurfaceAttrib(1354): error 0x3089 (EGL_BAD_MATCH)
11-16 09:49:25.119 4633 W OpenGLRenderer: Failed to set EGL_SWAP_BEHAVIOR on surface 0xec5cc320, error=EGL_BAD_MATCH
11-16 09:49:25.346 372 496 W SurfaceFlinger: Attempting to set client state on removed layer: SnapshotStartingWindow for taskId=44#0
11-16 09:49:25.346 372 496 W SurfaceFlinger: Attempting to destroy on removed layer: SnapshotStartingWindow for taskId=44#0
11-16 09:30:10.534 4444 4475 D Answers : Analytics collection enabled
```



```
124 public int getContentView() {
125     return R.layout.activity_splash;
126 }
127
128
129 public Context getContext() {
130     return this;
131 }
132
133 public void initView() {
134     if (this.presenter == null) {
135         this.presenter = new SplashPresenter(this);
136     }
137     if (verifyInstallerId(this)) {
138         showWrongAppVersion();
139         return;
140     }
141     this.presenter.getConfig();
142     setSnackBarView(findViewById(R.id.rootlayout), true);
143     this.versionCode = 75;
144 }
145
146 public void onError(String str) {
147     if (!"inprogress".equals(str)) {
148         hideProgress();
149         if (!str.equals("") && !str.equals(Constants.EXCEPTION)) {
150             showAlert(str);
151         }
152     } else if (progressIsShown()) {
153         setProgressMessage();
154     }
155 }
156
157 public void onHasRide(boolean z) {
158     if (z) {
159         gotoActiveRide();
160     } else {
161         gotoHomePage();
162     }
163 }
164
165 public void onLoadConfig() {
166     if (LocalDataManager.getInstance().getConfig().getAndroidVersion() > this.versionCode) {
167         showUpdateAler();
168     } else {
169         checkLogin();
170     }
171 }
172
173 public boolean verifyInstallerId(Context context) {
174     ArrayList arrayList = new ArrayList(Arrays.asList(new String[]{"com.android.vending", "com.google.android.feedback"}));
175     String installerPackageName = context.getPackageManager().getInstallerPackageName(context.getPackageName());
176     return installerPackageName != null && arrayList.contains(installerPackageName);
177 }
178 }
```

Code Small

JADX memory usage: 0,65 GB of 4,00 GB

android - Detect if an app is installed - X

stackoverflow.com/questions/37539949/detect-if-an-app-is-installed-from-play-store

Hack 4 Career: Inform... LinkedIn Mert SARICA (merts... Inbox - mert.sarica...

stackoverflow

Products Customers Use cases Search... Log in Sign up

Detect if an app is installed from Play store

Asked 3 years, 5 months ago Active 3 months ago Viewed 6k times

I want to check and allow the use of my app just if it has been downloaded from the Play store, and it has not been shared by other user or from any other source. How can I prevent a user to use the app if it has not been downloaded from the Google Play store?

android

share improve this question

edited May 31 '16 at 10:14 Javier S 293 +3 +9

asked May 31 '16 at 7:53 Manthan Patel 1,564 +6 +16

Possible duplicate of [How to know an application is installed from google play or side-load?](#) - Julien Lopez May 31 '16 at 8:04

add a comment

2 Answers

active oldest votes

23

This method will check if your app has been installed from the Play Store.

```
boolean verifyInstallerId(Context context) {
    // A list with valid installers package name
    List<String> validInstallers = new ArrayList<>(Arrays.asList("com.android.vending", "com.
    // The package name of the app that has installed your app
    final String installer = context.getPackageManager().getInstallerPackageName(context.getPack
    // true if your app has been downloaded from Play Store
    return installer != null && validInstallers.contains(installer);
}
```

Some days ago I released an Android library, [PiracyChecker](#), that protects your app using some techniques, such as Google Play Licensing (LVL), APK signature protection and installer ID (this one).

share improve this answer

edited May 31 '16 at 8:10 answered May 31 '16 at 8:04 Javier S.

Blog

- We're Rewarding the Question Askers
- Why is the Migration to Python 3 Taking So Long?

Featured on Meta

- Feedback post: Moderator review and reinstatement processes
- Post for clarifications on the updated pronouns FAQ
- New Post Notices (Closed/On Hold/etc.) rolling out on Stack Overflow
- Upvotes on questions will now be worth the same as upvotes on answers

Remote jobs

- Senior DevOps Engineer (Remote) X-Team No office location REMOTE amazon-web-services docker
- Senior Full Stack Engineer (Node.js, React) Namaste Technologies No office location \$45K - \$90K REMOTE javascript node.js
- Senior Software Engineer, Backend

Bu fonksiyona kaynak kodu seviyesinde müdahale edip, installerPackageName değişkenini sıfırdan farklı bir değer yapıp ardından derleyip Android işletim sistemi üzerinde çalıştırabilirdim fakat Bill Gates'in bir röportajında dediği gibi "Her zaman en tembel insanları işe alırım çünkü tembeller çok karışık işleri bile en kısa yoldan yaparlar" ben de tembellik yapıp kısa bir yol aramaya karar verdim. :)

Eminim bu gibi bir durumla karşılaşan güvenlik arařtırmacılarının çoęu Frida araç kiti ile ilerlemeyi tercih ederler fakat hayatın Frida'dan ibaret olmaması gerektięine inanarak Frida'ya alternatif bir araç, farklı bir yol aramaya karar verdim. Google arama motorunda kısa bir arařtırma yaptıktan sonra daha önce sızma testlerinde özellikle SSL Pinning'i atlatmak için kullandığım ve 1400'den fazla eklentiye sahip olan Xposed Framework aklıma geldi.

Geny Motion üzerinde bulunan Android Oreo işletim sistemine Xposed Framework'u kurduktan sonra eklentilerine göz atmaya başladım. Eklentiler arasında XPrivaclyLua isimli eklenti hemen dikkatimi çekti. Adından da anlaşılabilirce üzere bu eklenti, Android üzerinde yüklü olan uygulamaları sahte bilgilerle (sahte konum bilgisi gibi) besleyerek mahremiyetinizi korumaya yardımcı olmaktadır. Çalışma yöntemi olarak kabaca bu bilgileri toplamaya çalışan fonksiyonlara kanca atarak gerçek bilgiler yerine sahte bilgiler vermektedir. Eklentiye ihtiyaçlarınız doğrultusunda şekillendirmek için ise Pro sürümünü yükleyerek Lua programlama dili ile betikler oluşturmanız gerekmektedir.

Xposed Installer

Xposed Status 🔴



Xposed Framework version 90-beta3 is active.





INSTALL/UPDATE

Version 90-beta3 🔒

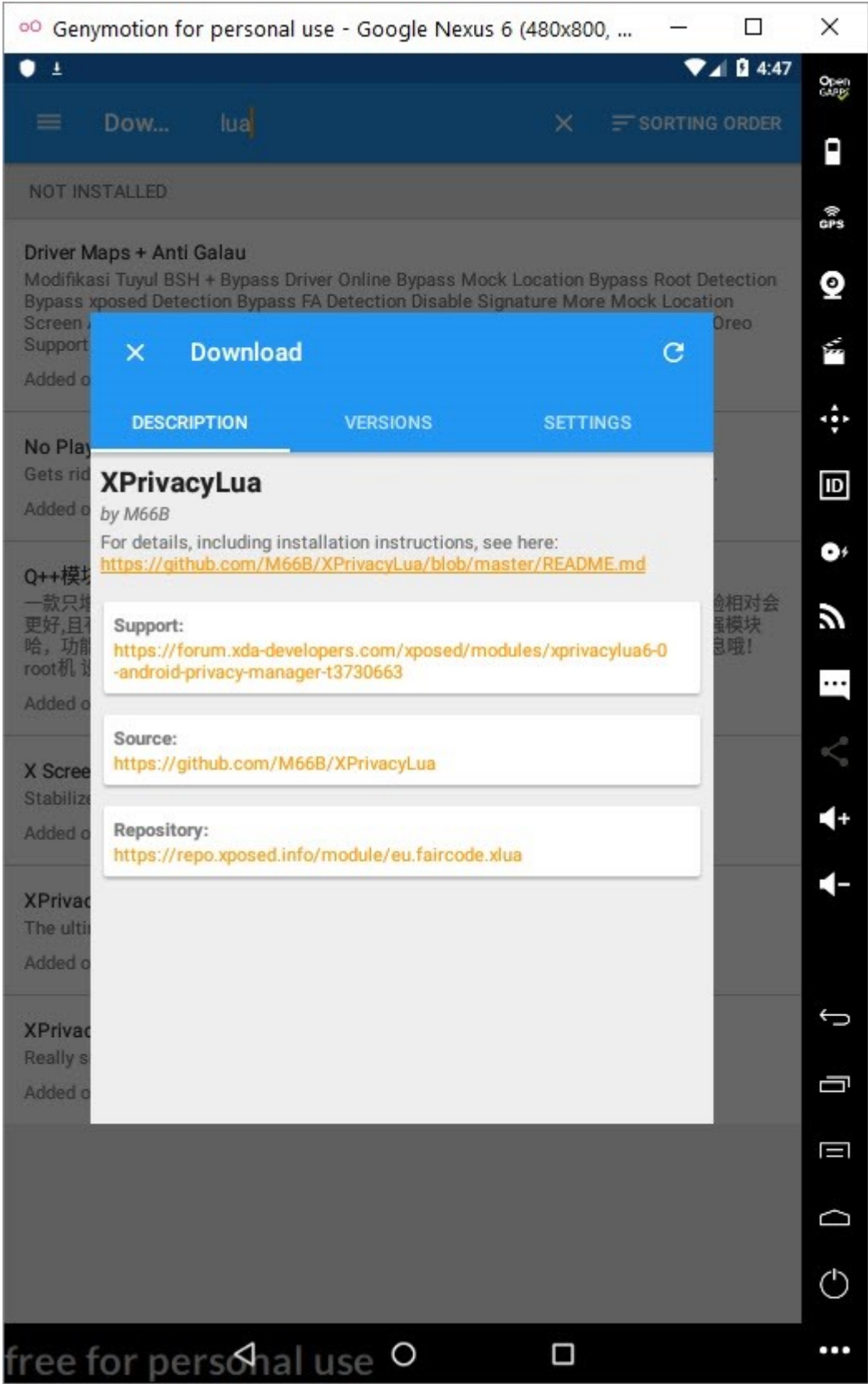
UNINSTALL

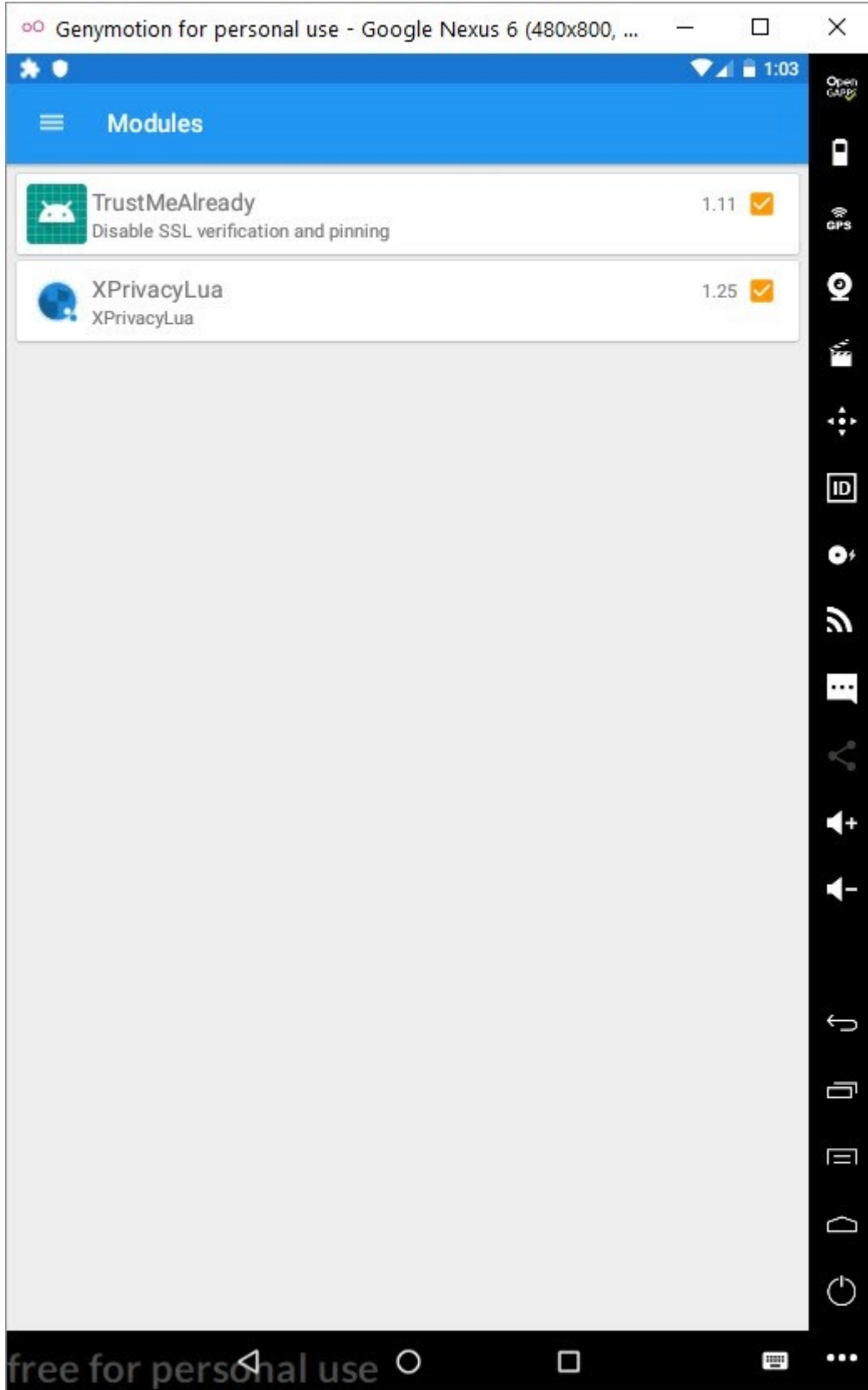
Uninstaller (20180117) ☁

YOUR DEVICE

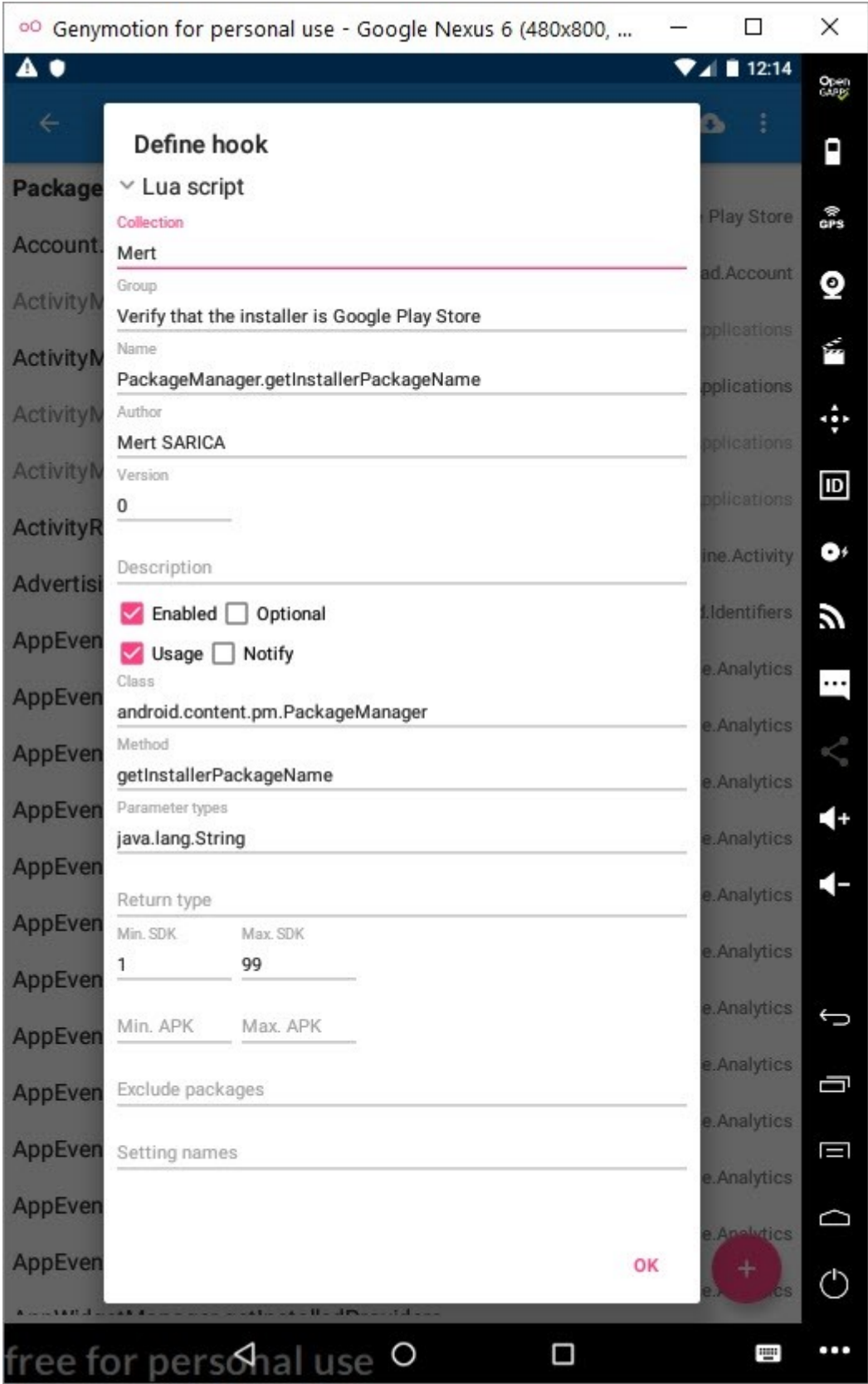
-  Android 8.0.0 (Oreo, API 26)
-  Genymotion Android Google Nexus 6
-  x86
-  Verified Boot is deactivated







Lua ile `installerPackagename` deęişkenini deęiřtiren ufak bir betik hazırlayıp aktif hale getirdikten sonra uygulamayı alıřtırdıęımda uygulamanın artık daha nceki uyarı mesajını ıkarmadıęını ve Charles Proxy ile web trafięini grntleyebildięimi grerek mutlu sona ulařmıř oldum.



PackageManager.getInstallerPackageName

```

Mert Verify that the installer is Google Play Store
-- Mert.PackageManager.getInstallerPackageName is a Lua
hook definition
-- designed to work with XPrivacyLua.

-- Mert.PackageManager.getInstallerPackageName is free
software: you can redistribute it and/or modify
-- it under the terms of the GNU General Public License
as published by
-- the Free Software Foundation, either version 3 of the
License, or
-- (at your option) any later version.

-- Mert.PackageManager.getInstallerPackageName is
distributed in the hope that it will be useful,
-- but WITHOUT ANY WARRANTY; without even the implied
warranty of
-- MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
See the
-- GNU General Public License for more details.

-- You should have received a copy of the GNU General
Public License
-- along with XPrivacyLua. If not, see
<http://www.gnu.org/licenses/>.

-- Copyright (C) 2019 Mert SARICA (www.mertsarica.com)

function after(hook, param)
    local result = param:getResult()
    local fake = 'com.android.vending'
    param:setResult(fake)
    return true, result, fake
end

```

Define hook

^ Lua script

OK

The screenshot displays the Charles Proxy interface. The left sidebar shows a sequence of requests from 'https://onesignal.com'. The main pane shows the response body for a successful request:

```
{
  "isSuccess": true,
  "message": "SUCCESS",
  "data": {
    "iosVersion": 102,
    "androidVersion": 89,
    "fineAmount": 40,
    "provisionPrice": "5",
    "kvkkRead": false,
    "pricePerMinute": "0.75",
    "msgEn": "",
    "msgTr": "",
    "bgColor": "",
    "imgUrl": "",
    "icon": "",
    "opAvailable": false,
    "startingPrice": "3"
  }
}
```

On the right, a Genymotion emulator window is visible, showing a 'New Version on the Store' dialog with the text 'A new version is available. Please update it now!' and buttons for 'CANCEL' and 'UPDATE NOW'.

Bu yazının güvenlik arařtırmalarında Frida'ya alternatif araç ve yöntem arayanlara ışık tutacağını ümit ederek bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.