

# Deep dive into mobile app

---

scopri se un'app ti spia

pasquale 'sid' fiorillo



```
> whoami
```



- 🏠 Penetration Tester @ ISGroup
- 👤 IT Consultant @ rooters >\_
- 👥 Teacher @ IFTS - Scuola la Tecnica
- 🔗 Independent Security Researcher
- ⚙️ Reverse Engineering addicted
- ⚙️ Hacking enthusiast



```
> cat /underground
```



**MEMOTEL**  
Hack

Idea by: mR\_bIs0n    Coded by: ^SiD^

http://www.setek.tk - #Setek @AzzurraNet



**SPAGHETTIPHREAKERS**  
REBIRTH

**8200 e Omitel...**

From: "Red" <red@unibn.it>  
To: <spaghettiphreak@gnat.com>, <spaghettiphreak@groups.com>  
Subject: POSSIBILE BUG DI MOTOROLA 8200 ?  
Date: Mon, 17 May 1999 20:27:25 +0200

Ciao, sono sempre RedRebel, credo di avere trovato un bug in Motorola 8200, utilizzo non insalvabile una comune scheda GSM TIM ma, ho riscontrato (postando alle mie TIM corduna scheda OMNITEL (con credito zero!) che posso chiamare GRATUITI e gratis

(CC) BY-SA

# THP

NUMERO [0]

```
<<<----->>>
>>> Italian Hard Phreaking <<<
<<< L.I.P. Magazine >>>
>>> Log In Progress Project <<<
<<< http://www.lipforever.tk >>>
>>>-----<<<
```

INDICE:

```
+-- [ INTRO ] -----+
+ .....+
+ .....+
+-- [ GREEN ] -----+
+ .....+
+....'Guida alle numerazioni'.....(lsdmaster)....+
+....'Niente di che... ma piuttosto interessante!'.....(mR_bIs0n)....+
+ .....+
+-- [ CRASH AND PLAY ] -----+
+ .....+
+....'La beige box dei tuoi sogni - Parte I'.....(mR_bIs0n)....+
+....'SAM DIGITO - Episodio 1'.....(^SiD^)....+
```

```
cat /security-researcher
```

The vulnerability allows any local user, such as “httpdusr” used to run web application, to escalate to Domain Administrator if the NAS is a domain member.

Pasquale ‘sid’ Fiorillo from ISGroup ([www.isgroup.biz](http://www.isgroup.biz)), an Italian Security Company, and Guido ‘go’ Oricchio of PCego ([www.pcego.com](http://www.pcego.com)), a System Integrator, have just released a critical security advisory for any version of QNAP NAS prior to 4.2.4 Build 20170313 ([https://www.qnap.com/en/support/con\\_show.php?cid=113](https://www.qnap.com/en/support/con_show.php?cid=113)).



The vulnerability allows a local unprivileged user of a Windows guest to gain Local and/or Domain Administrator access when VeeamVixProxy is active, the de-facto default in VMWare and Hyper-V environments.

Pasquale ‘sid’ Fiorillo, Francesco ‘ascii’ Ongaro from ISGroup, an Italian Security firm, and Antonio ‘s4tan’ Parata from ush team, have just released a critical security advisory for any version of Veeam Backup & Replication prior to 8 Update 3 (released today, October 8th, 2015).

Veeam Software provides backup, disaster recovery and virtualization management software for the VMware and Hyper-V environments. The ISGroup team has discovered this Oday in the Veeam Software while performing a Penetration Test for a customer.

*“The vulnerability allows a local unprivileged user of a Windows guest to gain Local and/or Domain Administrator access when VeeamVixProxy is active, the de-facto default in VMWare and Hyper-V environments.” states the advisory.*

The issue potentially involves 157,000 customers and 9.1 million Virtual Machines worldwide and could lead to full Domain Administrator compromise of the affected infrastructures.

```
sid@zen:~/veeam$ cat VeeamVixProxy_16072015.log | grep "01/07/2015 1.33.42" | cut
-d ' ' -f 6 | base64 -d | hexdump -C | lolcat
base64: invalid input
00000000 23 00 00 00 0a 00 00 00 50 00 65 00 65 00 01 00 [#.....V.e.e.e.]
00000010 6d 00 55 00 73 00 65 00 72 00 10 00 00 00 55 00 [m.U.s.e.r.....U.]
00000020 32 00 56 00 6a 00 63 00 6d 00 56 00 30 00 [2.V.j.c.m.V.0.]
0000002e
sid@zen:~/veeam$ echo -en "U2VjcmV0" | base64 -d | xargs -I {} echo {} | lolcat
Secret
sid@zen:~/veeam$
```

[OBJ]  
[OBJ]

```
> cat /app/index
```



```
> cat /app/anatomy
```



CLIENT-SIDE











SERVER-SIDE



```
> cat /app/anatomy
```

CLIENT-SIDE



-  foreground code
-  background code
-  internal storage
-  keychain
-  network access
-  sensors, bluetooth, etc
-  camera, microphone
-  external storage

```
> cat /app/anatomy
```

CLIENT-SIDE



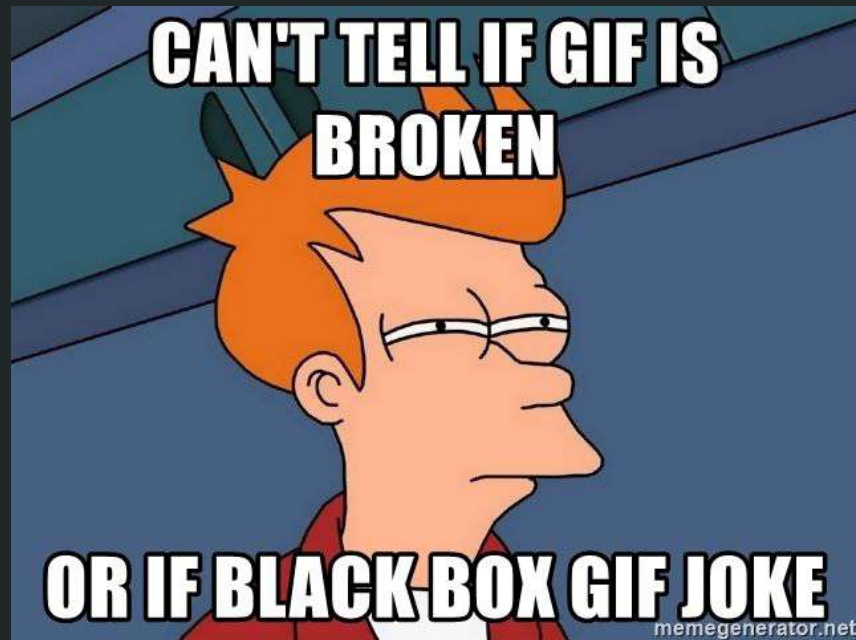
SERVER-SIDE









```
> cat /app/anatomy
```



SERVER-SIDE



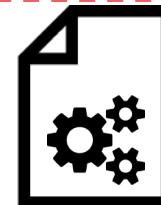
```
> cat /app/package
```



-  source code & assets
-  build process
-  digital signature
-  publish

-  ipa
-  apk/aab

# REVERSE ENGINEER

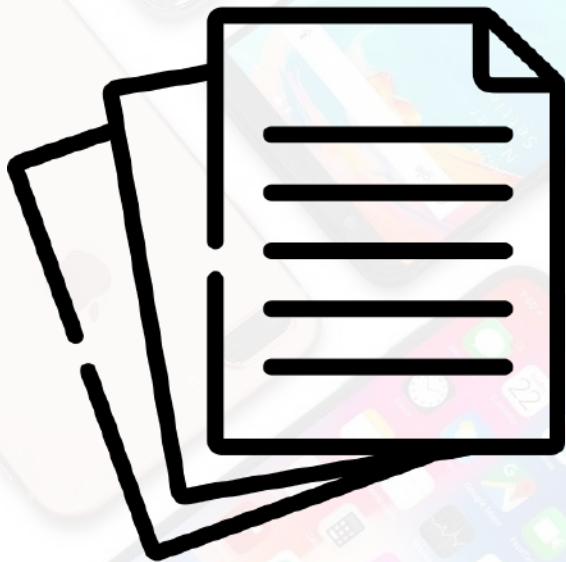


# ALL THE THINGS

```
> cat /hack/index
```

STATIC ANALYSIS

DYNAMIC ANALYSIS





reverse engineering







static analysis

```
> cat /hack/static
```

## STATIC ANALYSIS

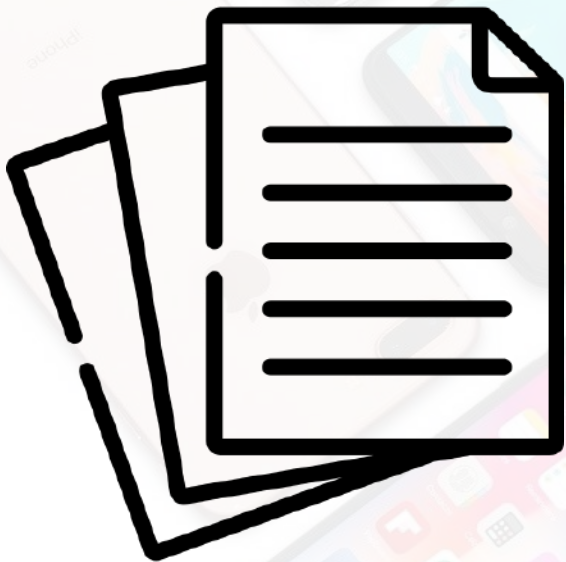





-  download/dump the apk
-  unpack it
-  decode manifest XML
-  decompile the code

 Android: apk/aab

```
> cat /hack/static
```

DOWNLOAD/DUMP THE APK



 py googleplay-api  
 online downloader  
 adb

1

obtain the binary package



**EXAMPLE:** dump the APK from the device with adb

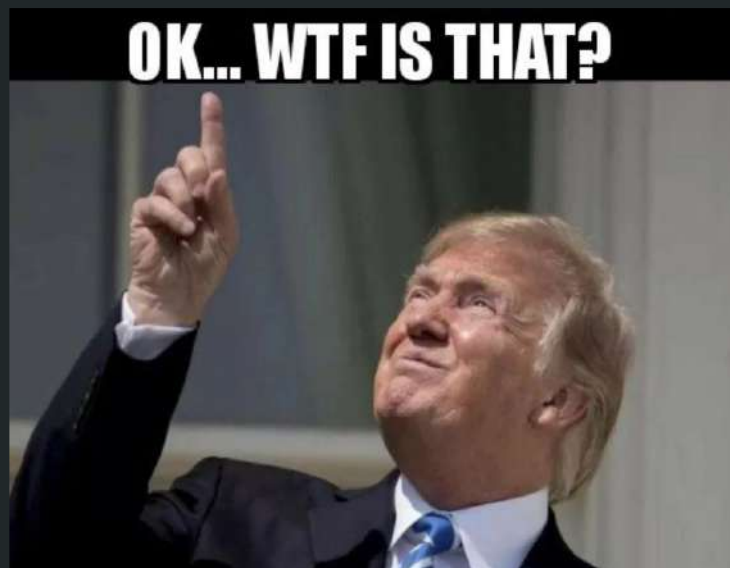
```
$ adb shell pm list packages
```

```
$ adb shell pm path com.facebook.katana
```

```
$ adb pull /data/app/com.facebook.katana.apk ./
```

```
[...]
```

```
5185 KB/s (15106048 bytes in 2.844s)
```



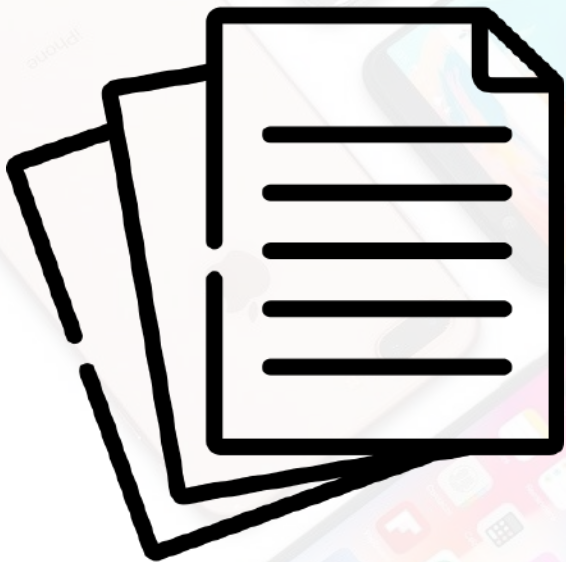


## EXAMPLE: getting the package name from Google Play

The screenshot shows the Google Play Store interface. The address bar at the top contains the URL `https://play.google.com/store/apps/details?id=com.facebook.katana`, which is circled in red. The main content area displays the app page for Facebook, including the logo, name, social category, and a green 'Installata' button. The left sidebar shows navigation options like 'Le mie app', 'Acquista', and 'Account'. The right sidebar shows 'Simili' (Similar) apps, including Facebook Lite and Messenger Lite.

```
> cat /hack/static
```

(UN)PACK THE APK



 apktool

 APK Studio

1

obtain the SMALI code



2

obtain the assets



3

obtain AndroidManifest.xml



## EXAMPLE: unpack and pack an apk

```
$ apktool d com.facebook.katana.apk
```

```
[...]
```

```
I: Decoding AndroidManifest.xml with resources...
```

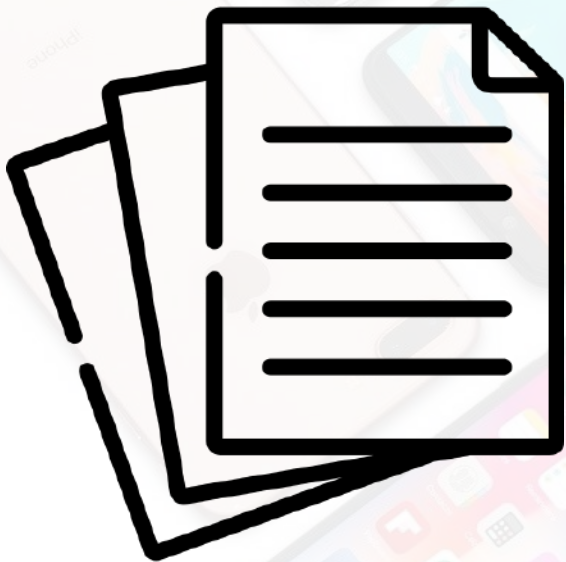
```
$ apktool b facebook
```

```
[...]
```

```
I: Building apk file...
```

```
> cat /hack/static
```

AndroidManifest.xml



✓ just a text editor

1

main Activity



2

services



3

permissions



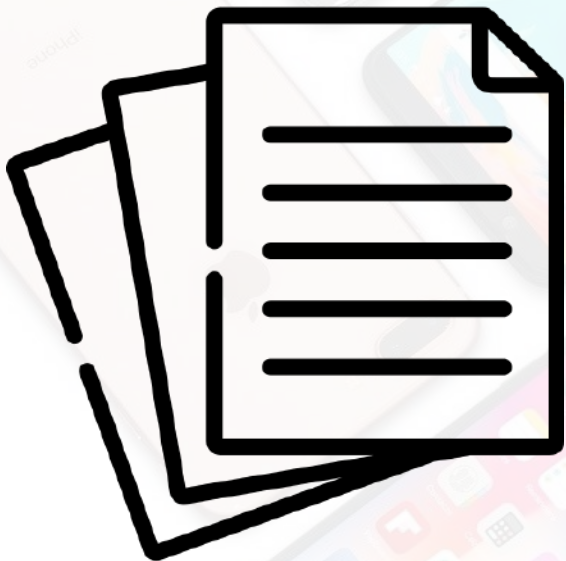
4




...



```
> cat /hack/static
```

DECOMPILE THE SMALI CODE



-  jadx (Java native)
-  dnSpy / ilSpy (Xamarin C#)
-  **NOTHING** (JS framework)

1

obtain the source code



## EXAMPLE: how decompiled code looks like

```
14
15     // Token: 0x0600166D RID: 5741 RVA: 0x00081194 File Offset:
16     // 0x0007F394
17     public Position(string deviceId, Location location)
18     {
19         if (location == null)
20         {
21             return;
22         }
23         this.deviceId = deviceId;
24         this.latitude = location.Latitude;
25         this.longitude = location.Longitude;
26         this.altitude = location.Altitude;
27         this.speed = (double)location.Speed;
28         this.course = (double)location.Bearing;
29         if (location.Provider != null && location.Provider != "gps")
30         {
31             this.accuracy = (double)location.Accuracy;
32         }
33         if (Build.VERSION.SdkInt >= 18)
34         {
35             this.mock = location.IsFromMockProvider;
36         }
37     }
```





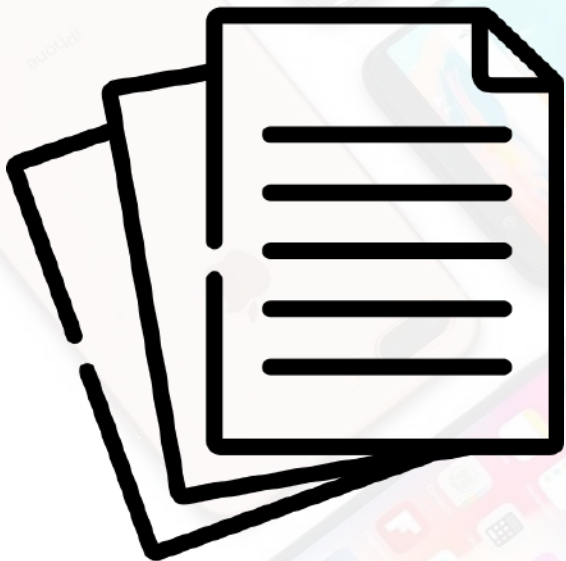
# reverse engineering





---

static analysis

```
> cat /hack/static
```

## STATIC ANALYSIS



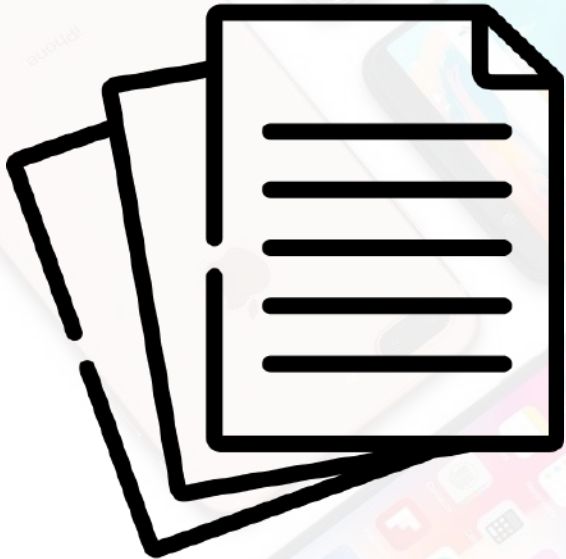
-  download/extract the ipa
-  unpack it
-  decode info.plist
-  decompile the code



 iOS: ipa



```
> cat /hack/static
```

DOWNLOAD/DUMP THE IPA



 [apple configurator 2](#)  
 [frida-ios-dump](#)

**EXAMPLE:** dump the IPA from the device with frida-ios-dump

```
$ ./dump.py Facebook
```

```
Start the target app Facebook
```

```
Dumping Facebook to ./
```

```
[...]
```

```
Generating Facebook.ipa
```

```
Done.
```



```
> cat /hack/static
```

UNPACK THE IPA



✓ just a zip file

1

obtain the binary code



2

obtain the assets



3

obtain info.plist



```
> cat /hack/static
```

info.plist



✓ just a text editor

1

some useful app informations



2

supported devices



3

permissions



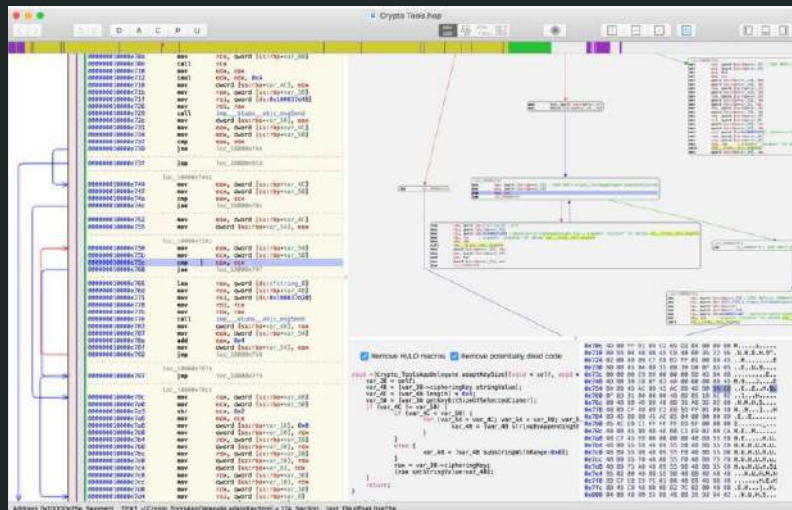
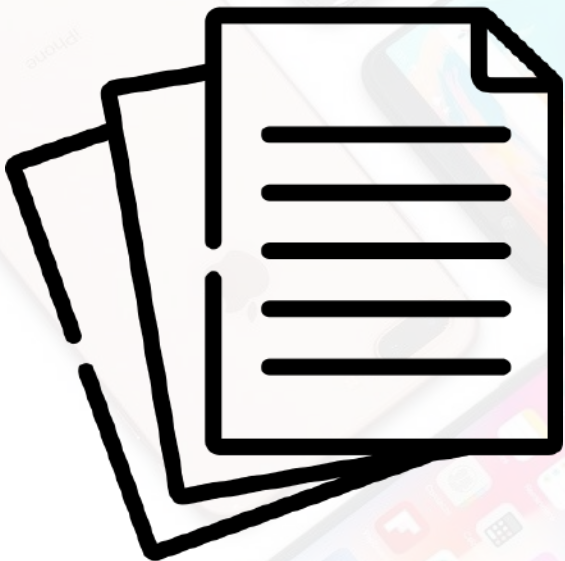
4

...



```
> cat /hack/static
```

## DECOMPILE THE BINARY CODE



\$ Hopper

\$ IDA Pro

1

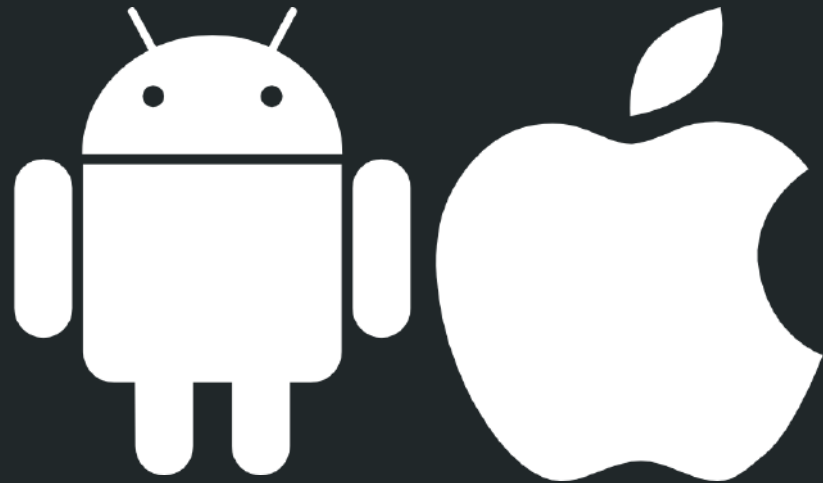
obtain the assembly code



2

obtain dirty object-c code





reverse engineering

dynamic analysis

```
> cat /hack/dynamic
```

CODE INJECTION

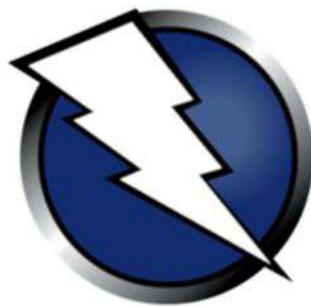


# FRIDA



```
> cat /hack/dynamic
```

NETWORK MitM



**OWASP**  
Zed Attack Proxy





```
> cat /hack/dynamic
```

NETWORK MitM



✓ with an interceptor proxy

you can:

1

sniffing TLS communication



2

intercept and modify data



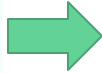
3

attack the back-end



## EXAMPLE: how network communication looks like

```
Request  Response
Pretty  Raw  Hex  In  ≡
1 POST /admin/Device/SaveDevice HTTP/2
2 Host: ██████████
3 Accept: application/json, text/json, text/x-json, text/javascript, application/xml, text/xml
4 User-Agent: Dalvik/2.1.0 (Linux; U; Android 9; ██████████)
5 Content-Type: application/json; charset=utf-8
6 Content-Length: 656
7 Expect: 100-continue
8 Connection: keep-alive
9 Cookie: ASP.NET_SessionId=██████████; NID=██████████
10 Accept-Encoding: gzip, deflate
11
12 {
  "UserAgent": "Dalvik/2.1.0 (Linux; U; Android 9; ██████████)",
  "Size": {
    "WidthPixels": 0,
    "HeightPixels": 0
  },
  "oid": 0,
  "Identifier": "android-██████████",
  "Name": null,
  "Model": null,
  "Serial": null,
  "Latitude": 41.14██████████,
  "Longitude": 14.7██████████,
  "Altitude": 228.37009742271337,
  "DeviceLocation": {
    "Latitude": 41.14██████████,
    "Longitude": 14.7██████████,
    "Altitude": 228.37009742271337,
    "Address": "",
    "City": "Benevento",
    "Province": "Provincia di Benevento",
    "Zip": "82100",
    "Region": "Campania",
    "Nation": "Italia",
    "NationCode": "IT"
  }
}
```



```
"Identifier": "android-██████████",
"Name": null,
"Model": null,
"Serial": null,
"Latitude": 41.14██████████,
"Longitude": 14.7██████████,
"Altitude": 228.37009742271337,
"DeviceLocation": {
  "Latitude": 41.14██████████,
  "Longitude": 14.7██████████,
  "Altitude": 228.37009742271337,
  "Address": "",
  "City": "Benevento",
  "Province": "Provincia di Benevento",
  "Zip": "82100",
  "Region": "Campania",
  "Nation": "Italia",
  "NationCode": "IT"
}
```



A close-up, high-angle shot of Morpheus from the movie The Matrix. He is bald, has a serious expression, and is wearing dark sunglasses. The reflection in the sunglasses shows a scene from the movie with Neo, Trinity, and Morpheus. The background is a blurred greenish-grey.

**What if i told you**

**This is the end of my presentation**