



# L'histoire de la découverte d'une backdoor signée OnePlus

DevFest Toulouse - Jeudi 3 Octobre, 2019



/ Elliot Alderson  
@fs0c131y

A screenshot of a mobile Google search interface. The search bar at the top contains the text "Baptiste Robert". Below the search bar is a list of ten search suggestions, each preceded by a magnifying glass icon. The suggestions are:

- baptiste robert
- baptiste robert **faceapp**
- baptiste robert **hacker**
- baptiste robert **security**
- baptiste robert **elliot alderson**
- baptiste robert **Irem**
- baptiste robert **toulouse**
- baptiste robert **twitter**
- baptiste robert **linkedin**
- baptiste robert **chercheur**

The background of the search interface is white, and there is a vertical scroll bar on the right side of the suggestion list.

Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Whoami

- French security researcher known under the pseudonym Elliot Alderson on Twitter
- I love to break things, especially Android apps
- Sometimes, I find some cool stuff
- I have a lot of enemies in India



Il était une fois...



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Extraction du Firmware

# Extraction du Firmware

[Version officielle](#)

[Version beta](#)



OnePlus 5

[Modifier](#)

Version

9.0.8

Mise à jour sur

August 8, 2019

Taille

1.77 GB

MD5

905c024f35dd70ca5a0b2d73b95c3f68

[Download](#)

Pensez à sauvegarder vos données avant d'envoyer votre appareil.



Elliot Alderson  
@fs0c131y

DevFest  
Toulouse 2019

# Extraction du Firmware

```
fs0c131y@Elliots-MacBook-Pro:~$ unzip OnePlus50xygen_23_OTA_019_all_1710311604_11573682a759c.zip -d contents
Archive: OnePlus50xygen_23_OTA_019_all_1710311604_11573682a759c.zip
signed by SignApk
extracting: contents/system.patch.dat
inflating: contents/META-INF/com/android/metadata
inflating: contents/META-INF/com/google/android/update-binary
inflating: contents/META-INF/com/google/android/updater-script
inflating: contents/RADIO/static_nvbk.bin
inflating: contents/boot.img
inflating: contents/firmware-update/BTFM.bin
inflating: contents/firmware-update/NON-HLOS.bin
inflating: contents/firmware-update/abl.elf
inflating: contents/firmware-update/adspso.bin
inflating: contents/firmware-update/cmnlib.mbn
inflating: contents/firmware-update/cmnlib64.mbn
inflating: contents/firmware-update/devcfg.mbn
inflating: contents/firmware-update/hyp.mbn
inflating: contents/firmware-update/keymaster.mbn
inflating: contents/firmware-update/logo.bin
inflating: contents/firmware-update/pmic.elf
inflating: contents/firmware-update/rpm.mbn
inflating: contents/firmware-update/tz.mbn
inflating: contents/firmware-update/xbl.elf
inflating: contents/system.new.dat
inflating: contents/system.transfer.list
inflating: contents/META-INF/com/android/otacert
inflating: contents/META-INF/MANIFEST.MF
inflating: contents/META-INF/CERT.SF
inflating: contents/META-INF/CERT.RSA
```



# Extraction du Firmware

```
fs0c131y@Elliots-MacBook-Pro:~/contents$ ls -Slhr
total 6055248
-rw-r--r--@ 1 fs0c131y staff      0B Jan  1  2009 system.patch.dat
drwxr-xr-x@ 3 fs0c131y staff    96B Sep 11 22:27 RADIO/
drwxr-xr-x@ 6 fs0c131y staff   192B Sep 11 22:28 META-INF/
drwxr-xr-x@ 16 fs0c131y staff   512B Sep 11 22:27 firmware-update/
-rw-r--r--@ 1 fs0c131y staff   136K Jan  1  2009 system.transfer.list
-rw-r--r--@ 1 fs0c131y staff   18M Jan  1  2009 boot.img
-rw-r--r--@ 1 fs0c131y staff  2.9G Jan  1  2009 system.new.dat
```



# Extraction du Firmware

xpirt / sdat2img

Watch 40

Star 481

Fork 294

Code

Issues 1

Pull requests 1

Projects 0

Wiki

Security

Insights

Convert sparse Android data image to filesystem ext4 image

android sdat2img python

42 commits

1 branch

0 releases

9 contributors

Branch: master ▾

New pull request

Create new file

Upload files

Find File

Clone or download ▾



xpirt sdat2img.py: revert to write access mode only ...

Latest commit 1b08432 on 30 Oct 2018

README.md

add note for google brotli format

11 months ago

sdat2img.py

sdat2img.py: revert to write access mode only

11 months ago

README.md

## sdat2img

Convert sparse Android data image (.dat) into filesystem ext4 image (.img)



Elliot Alderson  
@fs0c131y

DevFest  
Toulouse 2019

# Extraction du Firmware

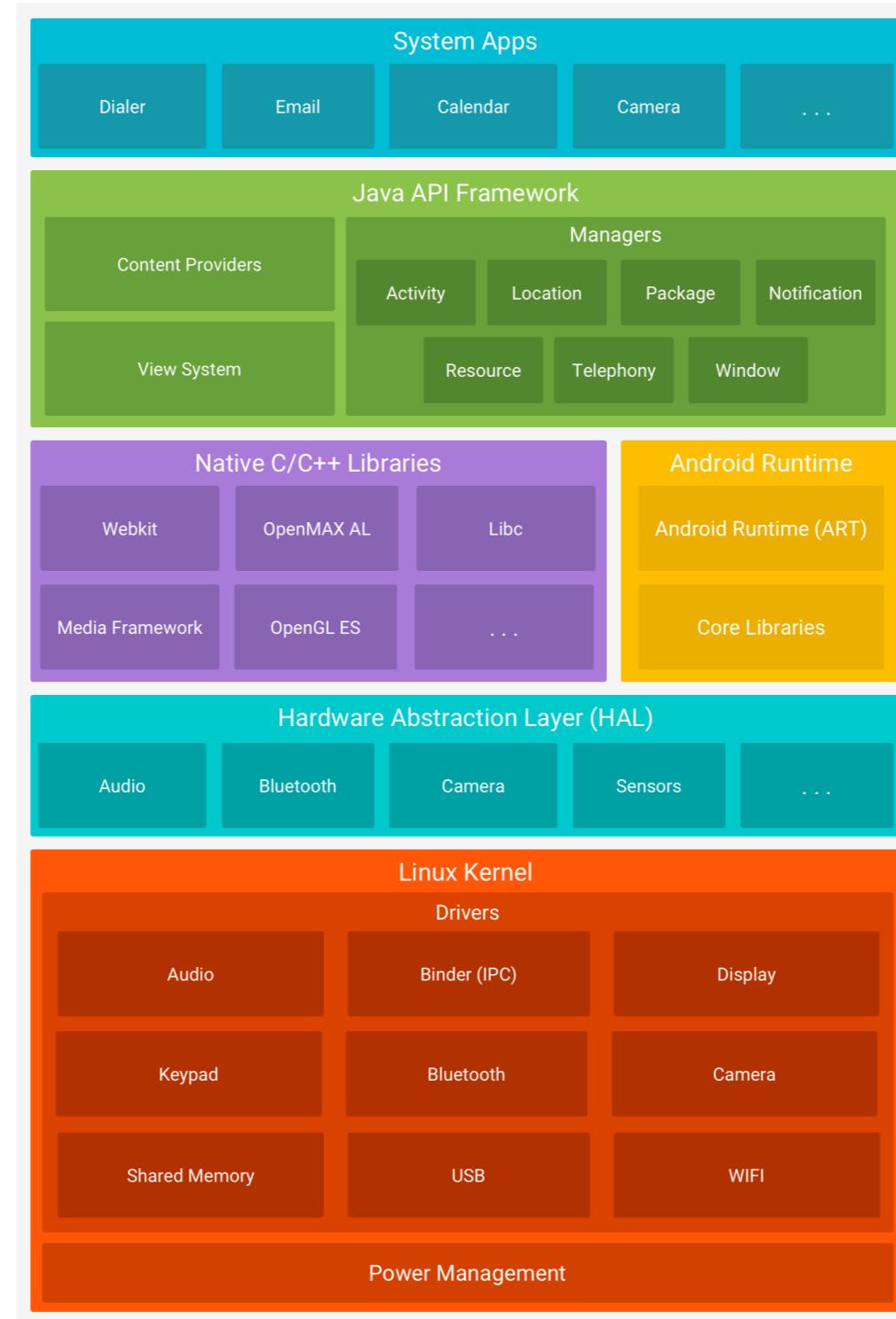
```
fs0c131y@Elliots-MacBook-Pro:/Volumes/Linux$ ll
total 9544
dr-xr-xr-x@ 21 root wheel 4.0K Jan  1 1970 .
drwxr-xr-x@  6 root wheel 192B Sep 11 23:35 ../
dr-xr-xr-x  95 root wheel 4.0K Dec 31 2008 app/
dr-xr-xr-x   3 root 2000 8.0K Dec 31 2008 bin/
dr-xr-xr-x   2 root wheel 4.0K Dec 31 2008 bpm/
-r--r--r--   1 root wheel 11K Dec 31 2008 build.prop
dr-xr-xr-x  22 root wheel 8.0K Dec 31 2008 etc/
dr-xr-xr-x   2 root wheel 4.0K Dec 31 2008 fake-libs/
dr-xr-xr-x   2 root wheel 4.0K Dec 31 2008 fake-libs64/
dr-xr-xr-x   2 root wheel 8.0K Dec 31 2008 fonts/
dr-xr-xr-x   5 root wheel 4.0K Dec 31 2008 framework/
dr-xr-xr-x   8 root wheel 12K Dec 31 2008 lib/
dr-xr-xr-x   6 root wheel 12K Dec 31 2008 lib64/
dr-x-----  2 root wheel 4.0K Jan  1 1970 lost+found/
dr-xr-xr-x   5 root wheel 4.0K Dec 31 2008 media/
dr-xr-xr-x  77 root wheel 4.0K Dec 31 2008 priv-app/
-r--r--r--   1 root wheel 4.5M Dec 31 2008 recovery-from-boot.p
dr-xr-xr-x   6 root wheel 4.0K Dec 31 2008 reserve/
dr-xr-xr-x   5 root wheel 4.0K Dec 31 2008 rfs/
dr-xr-xr-x   3 root wheel 4.0K Dec 31 2008 tts/
dr-xr-xr-x   9 root wheel 4.0K Dec 31 2008 usr/
dr-xr-xr-x  11 root 2000 4.0K Dec 31 2008 vendor/
dr-xr-xr-x   2 root 2000 4.0K Dec 31 2008 xbin/
```



Elliot Alderson  
@fs0c131y

# Surface D'attaque

# Surface D'attaque



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Surface D'attaque

## Applications Systèmes

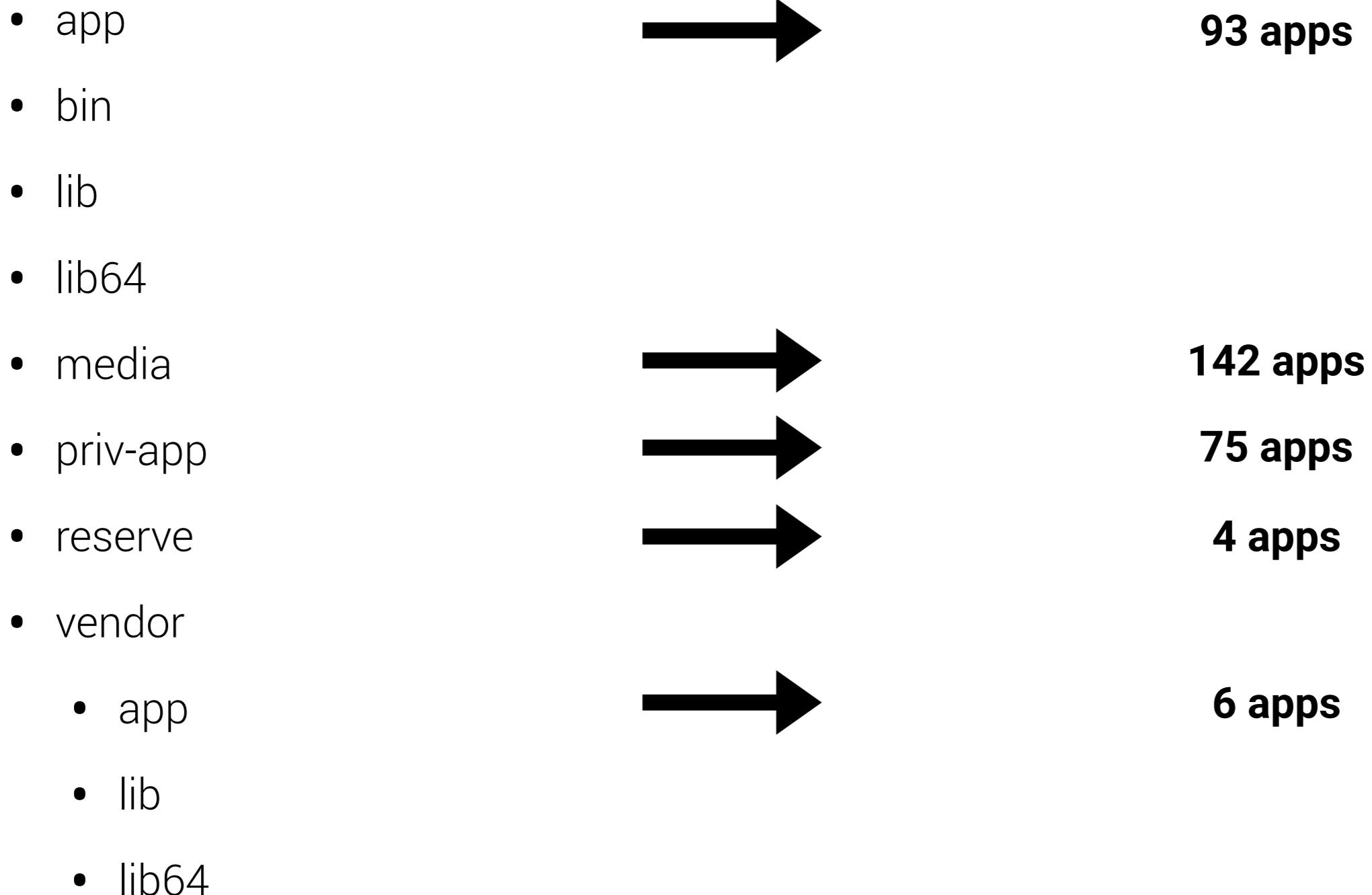
- Situé dans la partition /system/app et /system/priv-app
- Préinstallés par les fabricants de téléphones
- Ne peut pas être désinstallé
- Tous les droits!



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Surface D'attaque



# Surface D'attaque

320 applications!



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# AppAnalyser

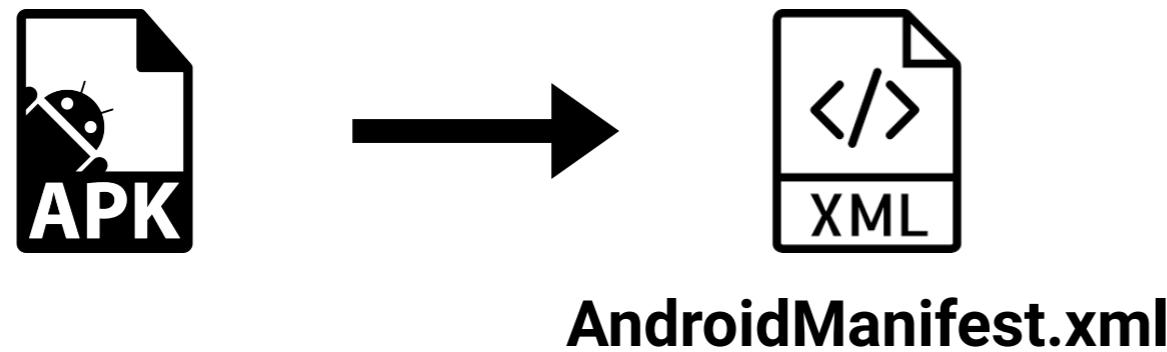
# AppAnalyser



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

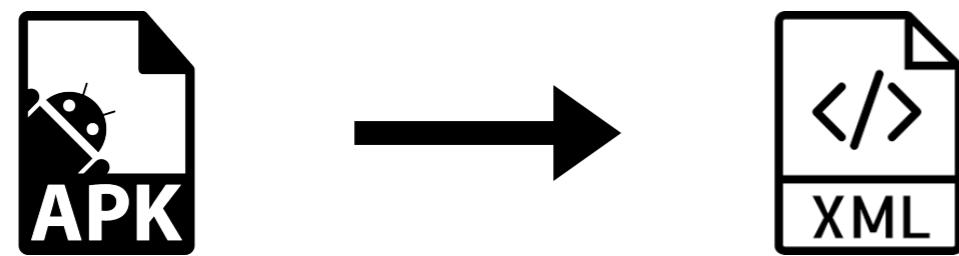
# AppAnalyser



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# AppAnalyser



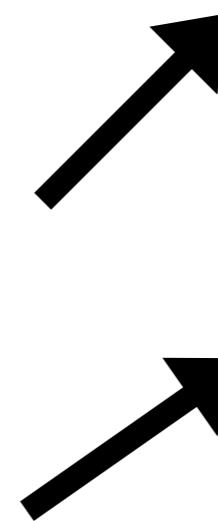
Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# AppAnalyser



**AndroidManifest.xml**



**activity**

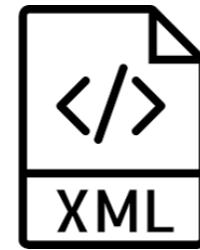
**receiver**



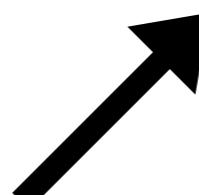
Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

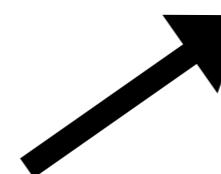
# AppAnalyser



**AndroidManifest.xml**



**activity**



**receiver**



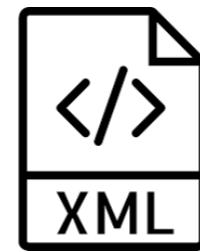
**service**



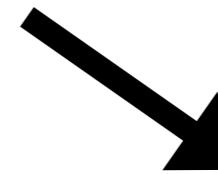
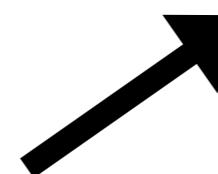
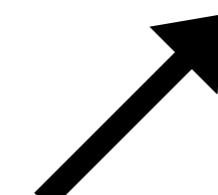
Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# AppAnalyser



**AndroidManifest.xml**



**activity**

**receiver**

**service**

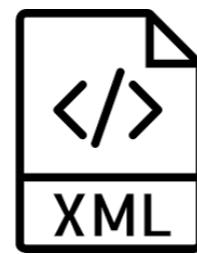
**provider**



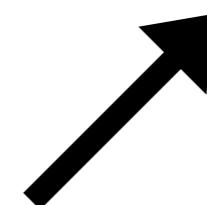
Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

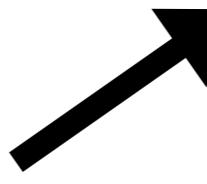
# AppAnalyser



**AndroidManifest.xml**



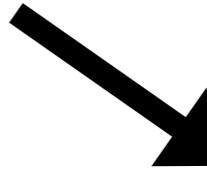
**activity**



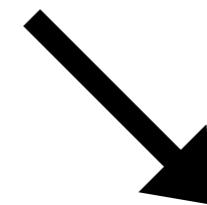
**receiver**



**service**



**provider**



**activity\_alias**



Elliot Alderson  
@fs0c131y

DevFest  
Toulouse 2019

# AppAnalyser

## VictimApp



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# AppAnalyser

**VictimApp**



**MyActivity**



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# AppAnalyser

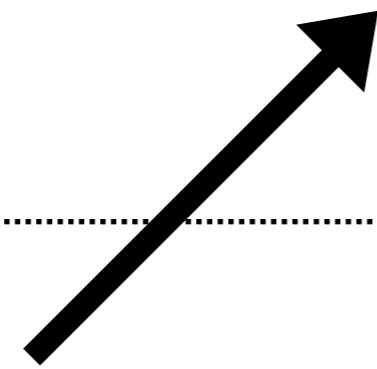
**VictimApp**



**MyActivity**



**AttackerApp**



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# AppAnalyser

## AppAnalyser

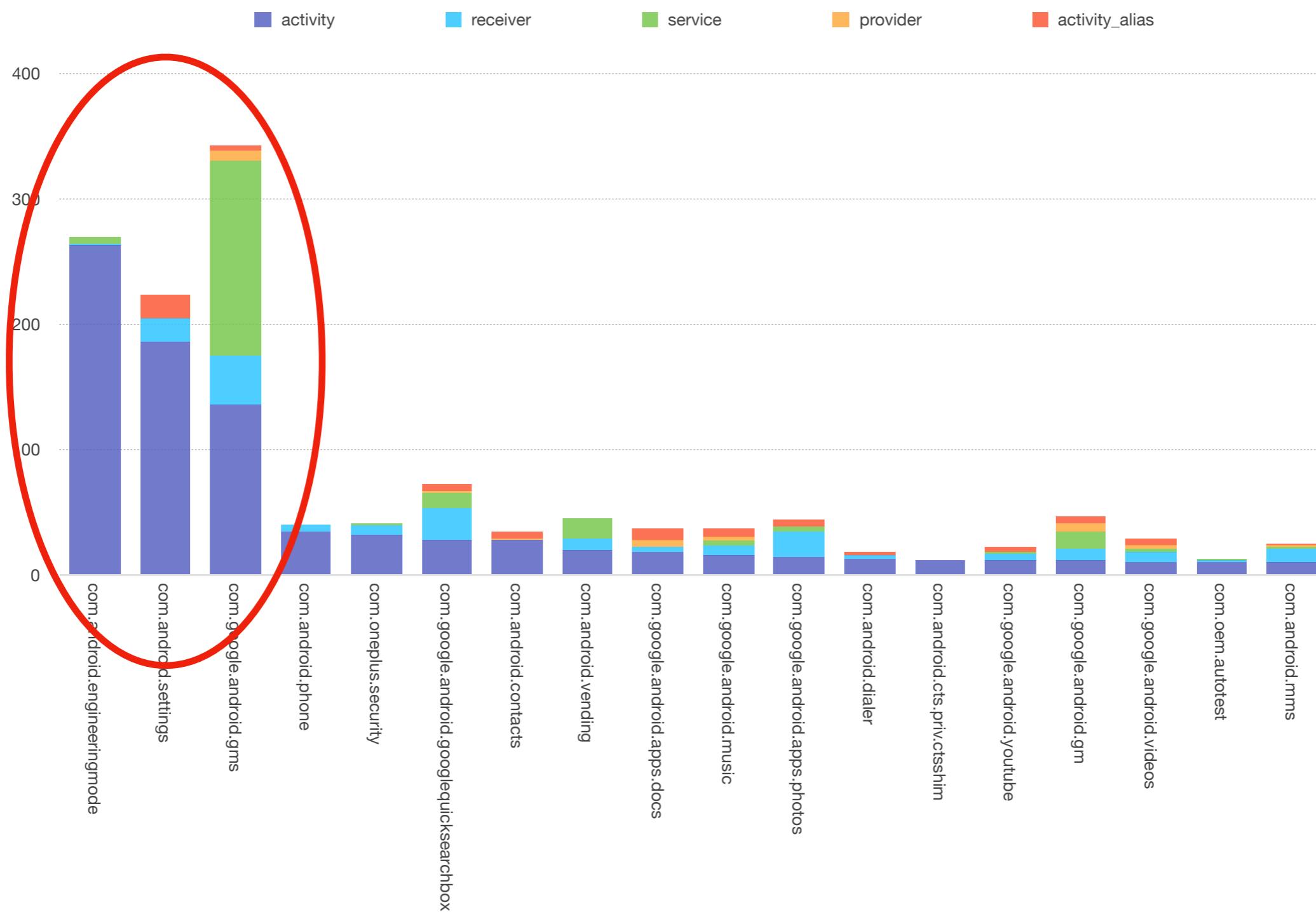
Un outil pour trouver les composants  
« exportés »



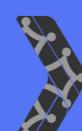
Elliot Alderson  
@fs0c131y

DevFest  
Toulouse 2019

# AppAnalyser

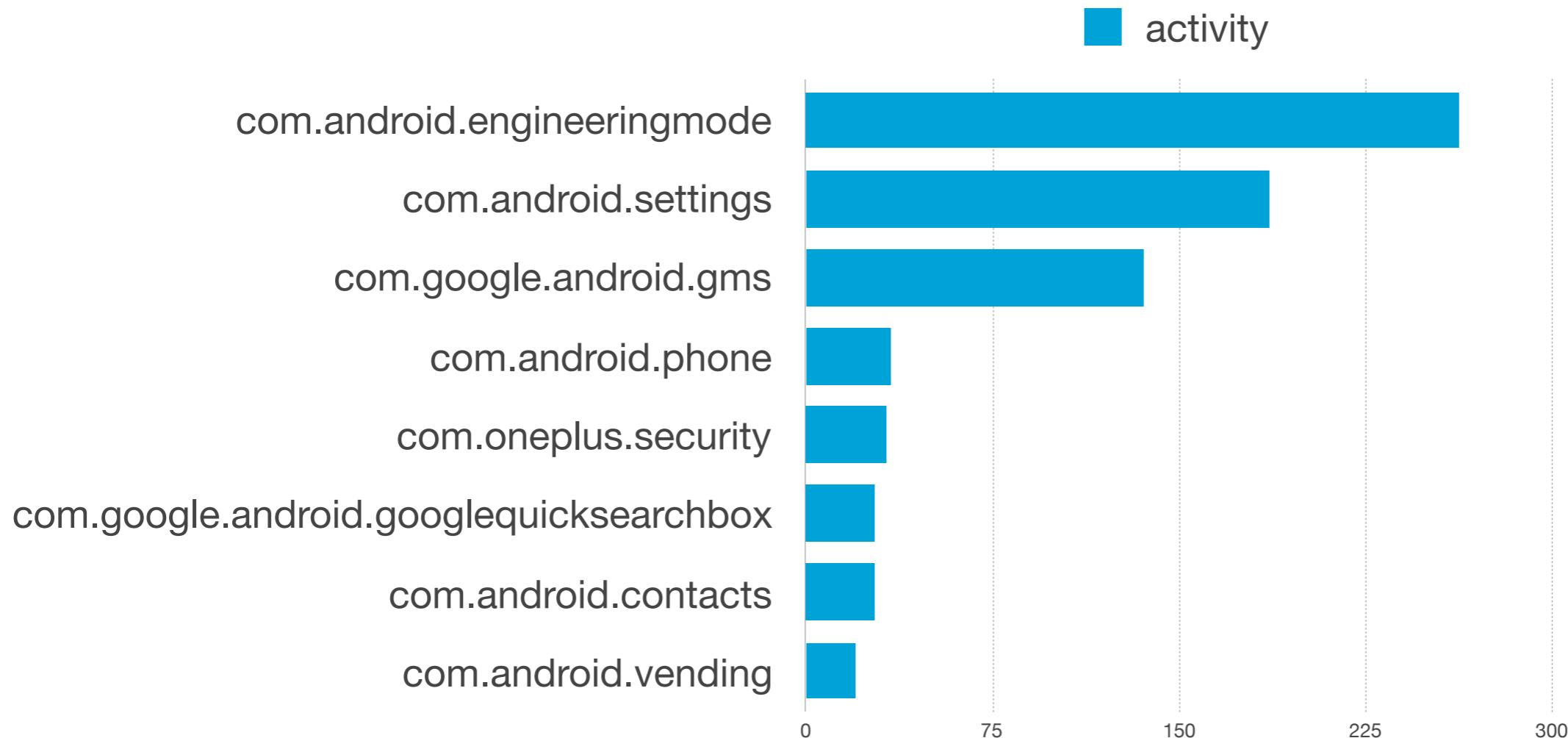


Elliot Alderson  
@fs0c131y



# DevFest Toulouse 2019

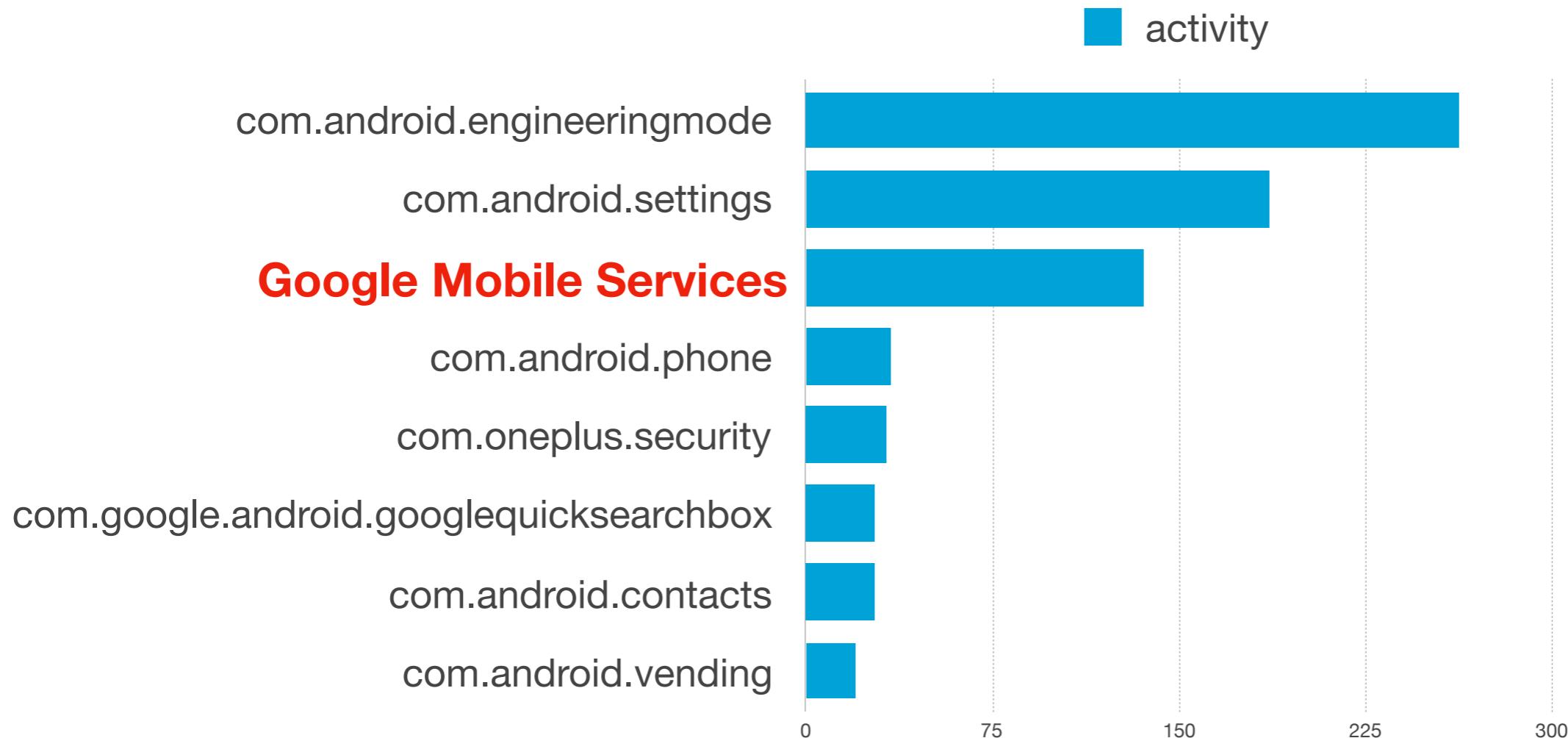
# AppAnalyser



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

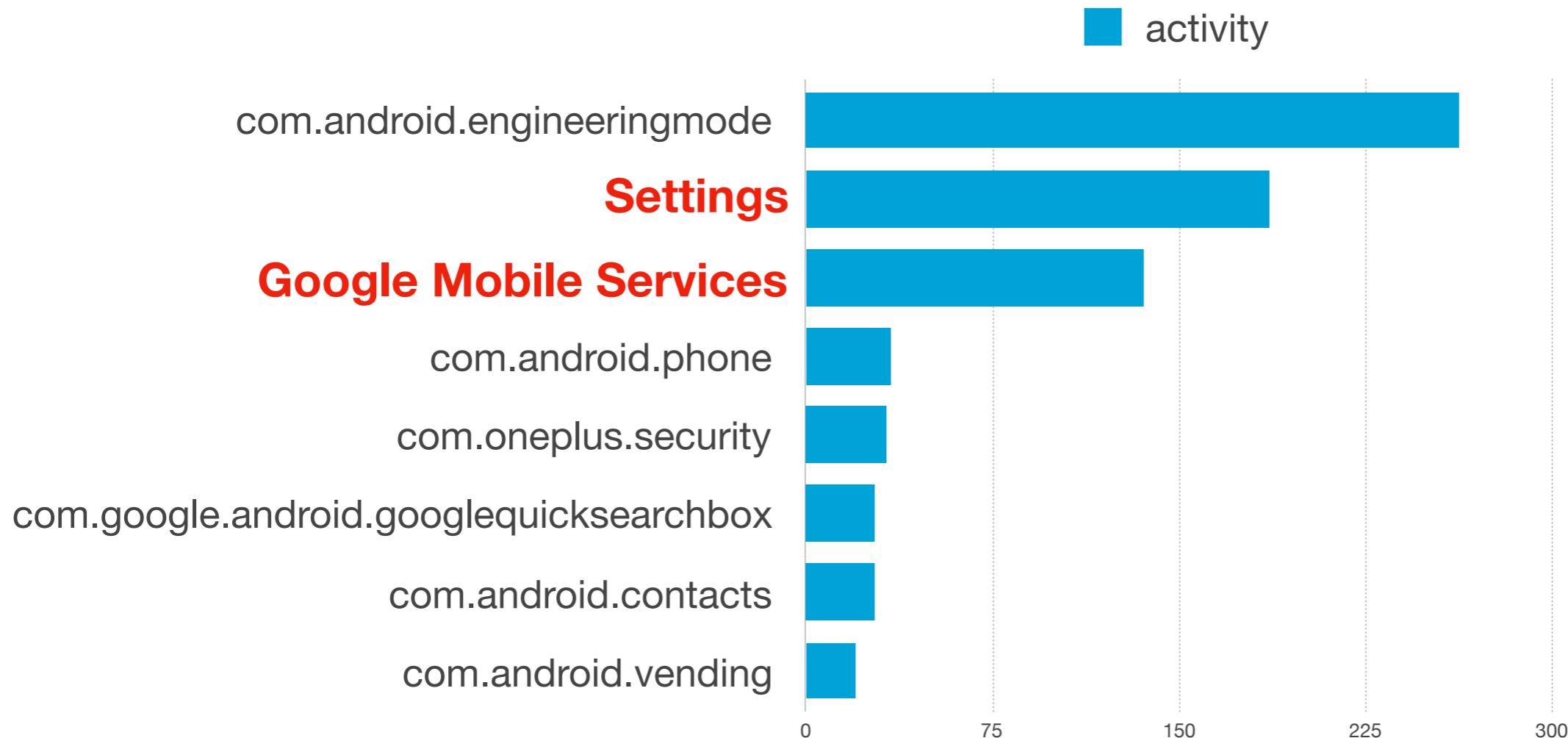
# AppAnalyser



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

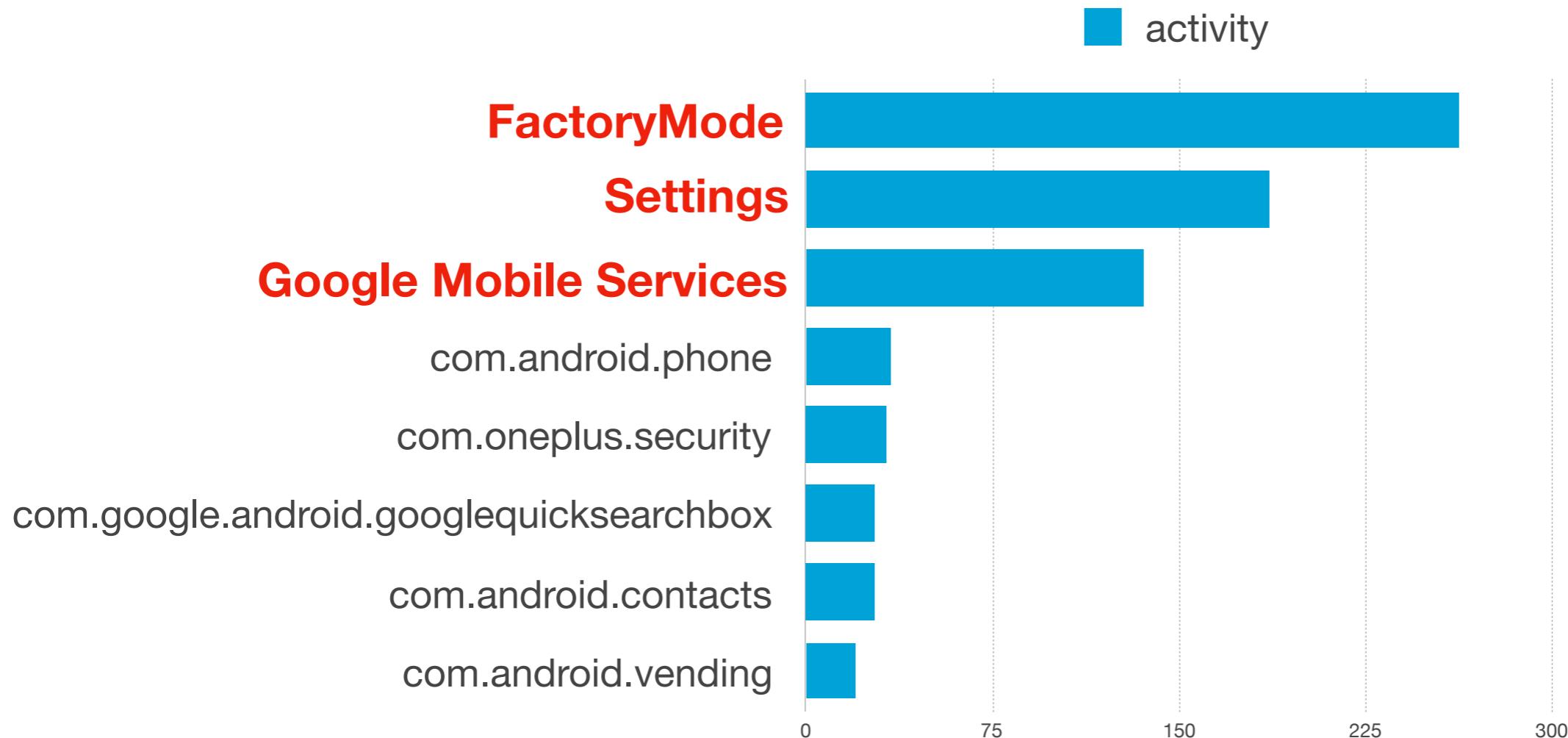
# AppAnalyser



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# AppAnalyser



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

1er cible:  
FactoryMode



# FactoryMode

# FactoryMode

← Infos appli      

 FactoryMode  
Installée

**DÉSACTIVER**      **FORCER L'ARRÊT**

**Notifications**  
Activé

**Autorisations**  
Aucune autorisation demandée

**Stockage**  
36,43 Mo utilisés dans stockage interne

**Conso. des données**  
6,16 Mo utilisés depuis 21 juin

**Batterie**  
Utilisation depuis la dernière charge complète : 3 %

**Ouvrir par défaut**  
Aucun paramètre par défaut défini



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# FactoryMode

## FactoryMode

Network set  
Protocol test switch  
GPRS Act  
check and encrypt IMEI  
Software version  
Audio  
automatic test  
manual test  
bluetooth test  
Wi-Fi test  
NFC continue transmit  
NFC SE test

## FactoryMode

all clear  
Erase all data in the phone including its internal storage  
auto aging  
Fill SMS and contacts  
QualComm  
charging settings  
Dolby Effect  
RF Cable Config   
Commonly used commands  
Enter Fastboot  
RF\_VERSION  
TDD\_FDD\_Eu\_Am\_All  
LightSensor Offset  
Light Sensor Offset Test



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Analyse Technique

# Analyse Technique

```
<string name="root_status_test">root status test</string>
<string name="root_status_success">root successful!</string>
<string name="root_status_fail">root failed!</string>
```

**strings.xml**



# Analyse Technique

```
private boolean checkAngelaRoot() {  
    boolean isAngelaRoot = SystemProperties.get("persist.sys.adbroot", "").equals("1");  
    Log.i("CheckRootStatusActivity", "my device has been angela root :" + isAngelaRoot);  
    return isAngelaRoot;  
}
```

## CheckRootStatusActivity.java





Elliot Alderson  
@fs0c131y

 DevFest  
Toulouse 2019

# Analyse Technique

```
private boolean checkAngelaRoot() {  
    boolean isAngelaRoot = SystemProperties.get('persist.sys.adbroot', "").equals("1");  
    Log.i("CheckRootStatusActivity", "my device has been angela root :" + isAngelaRoot);  
    return isAngelaRoot;  
}
```

## CheckRootStatusActivity.java



# Analyse Technique

```
protected void onCreate(Bundle savedInstanceState) {  
    super.onCreate(savedInstanceState);  
    setContentView(R.layout.diag_enabled);  
    if (getIntent() != null) {  
        escalatedUp(true, getIntent().getStringExtra("code"))  
    }  
}
```

**DiagEnabled.java**



# Analyse Technique

```
public boolean escalatedUp(boolean enable, String password) {  
    boolean ret = true;  
    if (enable) {  
        if (password != null) {  
            enable = Privilege.escalate(password);  
            if (enable) {  
                SystemProperties.set("persist.sys.adbroot", "1");  
                SystemProperties.set("oem.selinux.reload_policy", "1");  
            }  
            Log.d("DiagEnabled", "privilege escalate " + (enable ? "success" : "failed"));  
        } else {  
            enable = false;  
        }  
        ret = enable;  
    } else {  
        SystemProperties.set("persist.sys.adbroot", "0");  
        Privilege.recover();  
    }  
}
```

## DiagEnabled.java



Elliot Alderson  
@fs0c131y

# Analyse Technique

```
public boolean escalatedUp(boolean enable, String password) {
    boolean ret = true;
    if (enable) {
        if (password != null) {
            enable = Privilege.escalate(password);
            if (enable) {
                SystemProperties.set("persist.sys.adbroot", "1");
                SystemProperties.set("oem.selinux.reload_policy", "1");
            }
            Log.d("DiagEnabled", "privilege escalate " + (enable ? "success" : "failed"));
        } else {
            enable = false;
        }
        ret = enable;
    } else {
        SystemProperties.set("persist.sys.adbroot", "0");
        Privilege.recover();
    }
}
```

**DiagEnabled.java**



# Analyse Technique

```
public boolean escalatedUp(boolean enable, String password) {
    boolean ret = true;
    if (enable) {
        if (password != null) {           1
            enable = Privilege.escalate(password)
            if (enable) {
                SystemProperties.set("persist.sys.adbroot", "1");
                SystemProperties.set("oem.selinux.reload_policy", "1");
            }
            Log.d("DiagEnabled", "privilege escalate " + (enable ? "success" : "failed"));
        } else {
            enable = false;
        }
        ret = enable;
    } else {
        SystemProperties.set("persist.sys.adbroot", "0");
        Privilege.recover();
    }
}
```

**DiagEnabled.java**



# Analyse Technique

```
public boolean escalatedUp(boolean enable, String password) {
    boolean ret = true;
    if (enable) {
        if (password != null) {           1
            enable = Privilege.escalate(password)           2
            if (enable) {
                SystemProperties.set("persist.sys.adbroot", "1");
                SystemProperties.set("oem.selinux.reload_policy", "1");           3
            }
            Log.d("DiagEnabled", "privilege escalate " + (enable ? "success" : "failed"));
        } else {
            enable = false;
        }
        ret = enable;
    } else {
        SystemProperties.set("persist.sys.adbroot", "0");
        Privilege.recover();
    }
}
```

**DiagEnabled.java**



# Analyse Technique

```
public class Privilege {  
    public static native boolean escalate(String str);  
  
    public static native boolean isEscalated();  
  
    public static native void recover();  
  
    static {  
        System.loadLibrary("door");  
    }  
}
```

**Privilege.java**



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Analyse Technique

```
public class Privilege {  
    public static native boolean escalate(String str);  
  
    public static native boolean isEscalated();  
  
    public static native void recover();  
  
    static {  
        System.loadLibrary("door");  
    }  
}
```

**Privilege.java**



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Analyse Technique

Address	Length	Type	String
's' .rodata:000...	00000018	C	password verify passed\n
's' .rodata:000...	00000018	C	password verify failed\n
's' .rodata:000...	00000024	C	crypto path, password: %s, key: %s\n

**libdoor.so**



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Analyse Technique

```
ADD      R4, SP, #0x2D8+var_1A8
MOV      R0, R4
BLX      SHA256_Init
MOV      R0, R4
MOV      R1, R7
MOV      R2, R5
BLX      SHA256_Update
ADD      R5, SP, #0x2D8+var_2C8
MOV      R1, R4
MOV      R0, R5
BLX      SHA256_Final
LDR      R1, =(unk_5028 - 0x1000)
MOV      R0, R5 ; s1
MOVS   R2, #0x20 ; ' ' ; n
ADD      R1, PC ; unk_5028 ; s2
BLX      memcmp
CBZ      R0, loc_1048
```

```
loc_1048
LDR      R1, =(aDoor - 0x1052)
MOVS   R0, #3
LDR      R2, =(aPasswordVerify_0 - 0x1054)
ADD      R1, PC ; "door"
ADD      R2, PC ; "password verify passed\n"
BLX      __android_log_print
MOVS   R0, #0x6F ; 'o'
MOVS   R1, #0
STRB.W R0, [SP,#0x2D8+var_2D0]
ADD      R0, SP, #0x2D8+var_2D0
ORR.W  R0, R0, #1
```

libdoor.so



# Analyse Technique

The image shows two windows from a debugger, likely Immunity Debugger, displaying assembly code for the library `libdoor.so`.

**Top Window:**

```
ADD      R4, SP, #0x2D8+var_1A8
MOV      R0, R4
BLX    SHA256_Init 1
MOV      R0, R4
MOV      R1, R7
MOV      R2, R5
BLX    SHA256_Update
ADD      R5, SP, #0x2D8+var_2C8
MOV      R1, R4
MOV      R0, R5
BLX    SHA256_Final
LDR      R1, =(unk_5028 - 0x1000)
MOV      R0, R5 ; s1
MOVS   R2, #0x20 ; ' ' ; n
ADD      R1, PC ; unk_5028 ; s2
BLX    memcmp
CBZ      R0, loc_1048
```

**Bottom Window:**

```
loc_1048
LDR      R1, =(aDoor - 0x1052)
MOVS   R0, #3
LDR      R2, =(aPasswordVerify_0 - 0x1054)
ADD      R1, PC ; "door"
ADD      R2, PC ; "password verify passed\n"
BLX    __android_log_print
MOVS   R0, #0x6F ; 'o'
MOVS   R1, #0
STRB.W R0, [SP,#0x2D8+var_2D0]
ADD      R0, SP, #0x2D8+var_2D0
ORR.W  R0, R0, #1
```

`libdoor.so`



# Analyse Technique

The image shows two screenshots of a debugger interface, likely Immunity Debugger, displaying assembly code for the `libdoor.so` library. The code implements a SHA256 hashing process and logs success via `__android_log_print`.

**Top Window (SHA256 Hashing Implementation):**

```
ADD R4, SP, #0x2D8+var_1A8
MOV R0, R4
BLX SHA256_Init 1
MOV R0, R4
MOV R1, R7
MOV R2, R5
BLX SHA256_Update 2
ADD R5, SP, #0x2D8+var_2C8
MOV R1, R4
MOV R0, R5
BLX SHA256_Final
LDR R1, =(unk_5028 - 0x1000)
MOV R0, R5 ; s1
MOVS R2, #0x20 ; ' '
ADD R1, PC ; unk_5028 ; s2
BLX memcmp
CBZ R0, loc_1048
```

**Bottom Window (Success Logging):**

```
loc_1048
LDR R1, =(aDoor - 0x1052)
MOVS R0, #3
LDR R2, =(aPasswordVerify_0 - 0x1054)
ADD R1, PC ; "door"
ADD R2, PC ; "password verify passed\n"
BLX __android_log_print
MOVS R0, #0x6F ; 'o'
MOVS R1, #0
STRB.W R0, [SP,#0x2D8+var_2D0]
ADD R0, SP, #0x2D8+var_2D0
ORR.W R0, R0, #1
```

`libdoor.so`



# Analyse Technique

```
ADD      R4, SP, #0x2D8+var_1A8
MOV      R0, R4
BLX    SHA256_Init 1
MOV      R0, R4
MOV      R1, R7
MOV      R2, R5
BLX    SHA256_Update 2
ADD      R5, SP, #0x2D8+var_2C8
MOV      R1, R4
MOV      R0, R5
BLX    SHA256_Final 3
LDR      R1, -(unk_5028 - 0x1000)
MOV      R0, R5 ; s1
MOVS   R2, #0x20 ; ' ' ; n
ADD      R1, PC ; unk_5028 ; s2
BLX    memcmp
CBZ      R0, loc_1048

loc_1048
LDR      R1, =(aDoor - 0x1052)
MOVS   R0, #3
LDR      R2, =(aPasswordVerify_0 - 0x1054)
ADD      R1, PC ; "door"
ADD      R2, PC ; "password verify passed\n"
BLX    __android_log_print
MOVS   R0, #0x6F ; 'o'
MOVS   R1, #0
STRB.W R0, [SP,#0x2D8+var_2D0]
ADD      R0, SP, #0x2D8+var_2D0
ORR.W  R0, R0, #1
```

libdoor.so



# Analyse Technique

The image shows two screenshots of a debugger interface, likely Immunity Debugger, displaying assembly code for the library `libdoor.so`.

**Top Window (Main Function):**

```
ADD R4, SP, #0x2D8+var_1A8
MOV R0, R4
BLX SHA256_Init 1
MOV R0, R4
MOV R1, R7
MOV R2, R5
BLX SHA256_Update 2
ADD R5, SP, #0x2D8+var_2C8
MOV R1, R4
MOV R0, R5
BLX SHA256_Final 3
LDR R1, -(unk_5028 - 0x1000)
MOV R0, R5 ; s1
MOVS R2, #0x20 ; ' '
ADD R1, PC ; unk_5028 ; s2
BLX memcmp
CBZ R0, loc_1048
```

**Bottom Window (loc\_1048):**

```
loc_1048
LDR R1, =(aDoor - 0x1052)
MOVS R0, #3
LDR R2, =(aPasswordVerify_0 - 0x1054)
ADD R1, PC ; "door"
ADD R2, PC ; "password verify passed\n"
BLX android_log_print
MOVS R0, #0x6F ; 'o'
MOVS R1, #0
STRB.W R0, [SP,#0x2D8+var_2D0]
ADD R0, SP, #0x2D8+var_2D0
ORR.W R0, R0, #1
```

A red box highlights the `SHA256_Init`, `SHA256_Update`, `SHA256_Final`, and `memcmp` instructions. A red number **4** points to the `memcmp` instruction. A green arrow points from the main code to the `loc_1048` block.

`libdoor.so`



# Analyse Technique

```
ADD R4, SP, #0x2D8+var_1A8
MOV R0, R4
BLX SHA256_Init 1
MOV R0, R4
MOV R1, R7
MOV R2, R5
BLX SHA256_Update 2
ADD R5, SP, #0x2D8+var_2C8
MOV R1, R4
MOV R0, R5
BLX SHA256_Final 3
LDR R1, -(unk_5028 - 0x1000)
MOV R0, R5 ; s1
MOVS R2, #0x20 ; s2
ADD R1, PC ; unk_5028 ; s2
BLX memcmp
CBZ R0, loc_1048
```

4

```
loc_1048
LDR R1, =(aDoor - 0x1052)
MOVS R0, #3
LDR R2, =(aPasswordVerify_0 - 0x1054)
ADD R1, PC ; "door"
ADD R2, PC ; "password verify passed\n"
BLX android_log_print
MOVS R0, #0x6F ; 'o'
MOVS R1, #0
STRB.W R0, [SP,#0x2D8+var_2D0]
ADD R0, SP, #0x2D8+var_2D0
ORR.W R0, R0, #1
```

5

libdoor.so



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Analyse Technique

```
v3 = a1;
v4 = a3;
v5 = 0;
v6 = (*(int (__fastcall **)(int, int, _DWORD))(*(_DWORD *)a1 + 676))(a1, a3, 0);
(*(void (__fastcall **)(int, int))(*(_DWORD *)v3 + 672))(v3, v4);
v25 = 302976772;
v24 = 235212815;
if ( v6 )
{
    v7 = -1;
    v8 = _strlen_chk(v6, -1);
    if ( !v8 )
        goto LABEL_6;
    SHA256_Init(&v28);
    SHA256_Update(&v28, v6, v8);
    SHA256_Final(&v26, &v28);
    if ( !memcmp(&v26, &unk_5028, 0x20u) )
        goto LABEL_10;
    ((void (__fastcall *)(signed int, const char *, const char *))_android_log_print)(
        3,
        "door",
        "password verify failed\n");
}
```

**libdoor.so**



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Analyse Technique

```
unk_5028    DCB 0x79 ; y
              DCB 0xA6
              DCB 0xA9
              DCB 0x33 ; 3
              DCB 0xDF
              DCB 0xC9
              DCB 0xB1
              DCB 0x97
              DCB 0x5E ; ^
              DCB 0x44 ; D
              DCB 0x4D ; M
              DCB 0x4E ; N
              DCB 0x84
              DCB 0x81
              DCB 0xC6
              DCB 0x4C ; L
              DCB 0x77 ; W
              DCB 0x1D
              DCB 0x8A
              DCB 0xB4
              DCB 0xB
              DCB 0x7A ; z
              DCB 0xC7
              DCB 0x2F ; /
              DCB 0x8B
              DCB 0xC1
              DCB 0xA1
              DCB 0xBC
              DCB 0xA1
              DCB 0x71 ; q
              DCB 0x8B
              DCB 0xEF
ends
; .data
```

**libdoor.so**



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Analyse Technique

## Cracker Results:

```
79A6A933DFC9B1975E444D4E8481C64C771D8AB40B7AC72F8BC1A1BCA1718BEF SHA-256 angela
```

## SHA256 Decryption



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Analyse Technique

```
adb shell am start  
    -n com.android.engineeringmode/.qualcomm.DiagEnabled  
    -es "code" "angela"
```



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Conséquences

# Consequences

**Elliot Alderson** @fs0c131y · Nov 13, 2017  
<Thread> Hey @OnePlus! I don't think this EngineerMode APK must be in an user build... 🤔  
This app is a system app made by @Qualcomm and customised by @OnePlus. It's used by the operator in the factory to test the devices.

**Permissions**  
No permissions requested

**Notifications**

**Open by default**  
No defaults set

**Battery**  
7% use since last full charge

Mobile Security and 5 others

72 704 902



Elliot Alderson  
@fs0c131y

# Consequences

Carl Pei @getpeid

Replying to @fs0c131y @oneplus and 16 others

Thanks for the heads up, we're looking into it.

11:14 PM · Nov 13, 2017 · Twitter for Android

---

13 Retweets 89 Likes



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Consequences

---

## OCT 2017 SUMMARY

Tweet impressions

**212K**

---

New followers

**204**

---

## NOV 2017 SUMMARY

Tweet impressions

**11.9M**

---

New followers

**6,994**

---



Elliot Alderson  
@fs0c131y

DevFest  
Toulouse 2019



DATA CENTRE

SOFTWARE

SECURITY

DEVOPS

BUSINESS

PERSONAL TECH

SCIENCE

Personal Tech

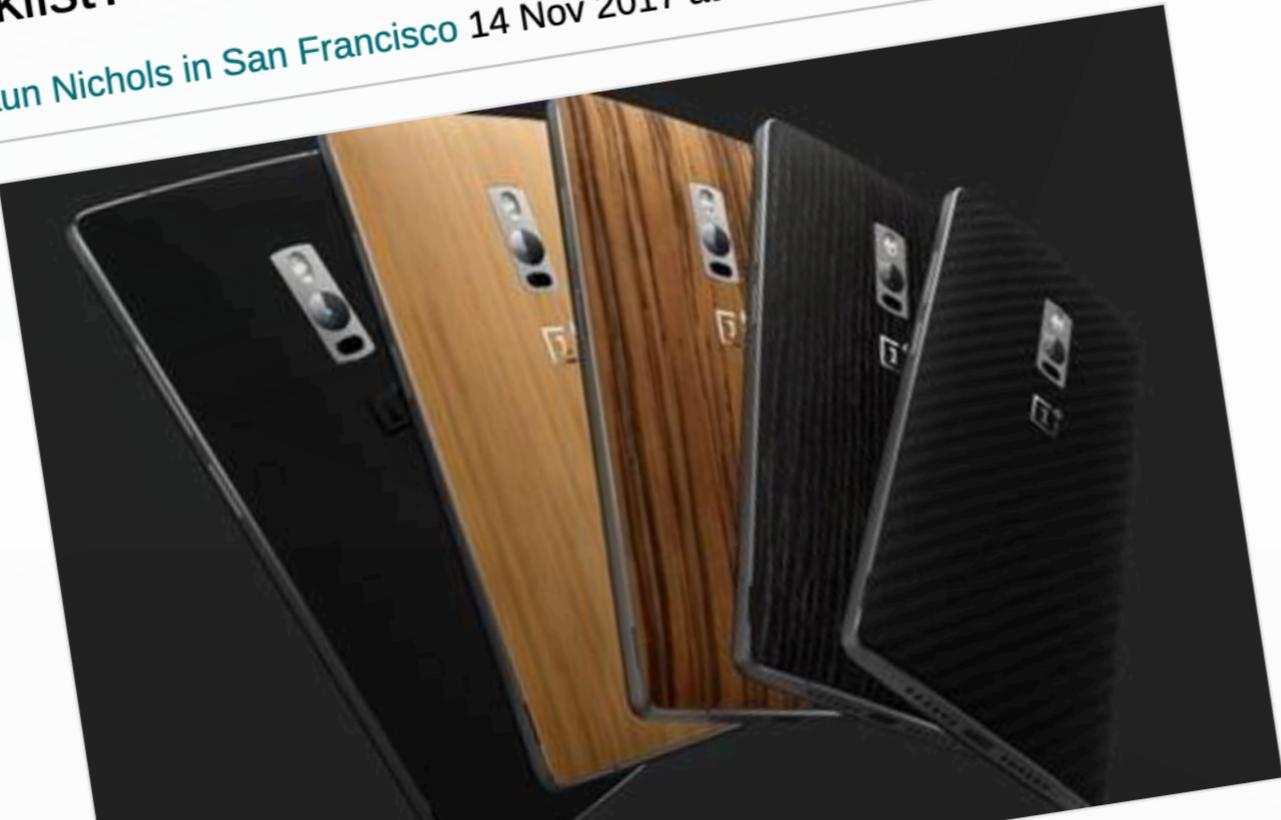
## Heads up: OnePlus phones have a secret root backdoor and the password is 'angela'

Who left 'wipe the engineering toolkit' off the factory checklist?

By [Shaun Nichols](#) in San Francisco 14 Nov 2017 at 21:32

12

SHARE ▾



### Most read



The NetCAT is out of the bag: Intel chipset exploited to sniff SSH passwords as they're typed over the network



Facebook: Remember how we promised we weren't tracking your location? Psych! Can't believe you fell for that



eBay eBabe enigma explained: Microsoft blamed after topless model slings e-souk's emails at stunned Brits



New lows at Bose as firmware update woes infuriate soundbar buyers



Geo-boffins drill into killing asteroid crater discover extinction involves bad smell

# Consequences

## Affected Build

- OxygenOS 4.5.14 and below
- Release date: 2017-10-31

## Angela Root Fixed

- OxygenOS 4.5.15
- Release date: 2017-12-05

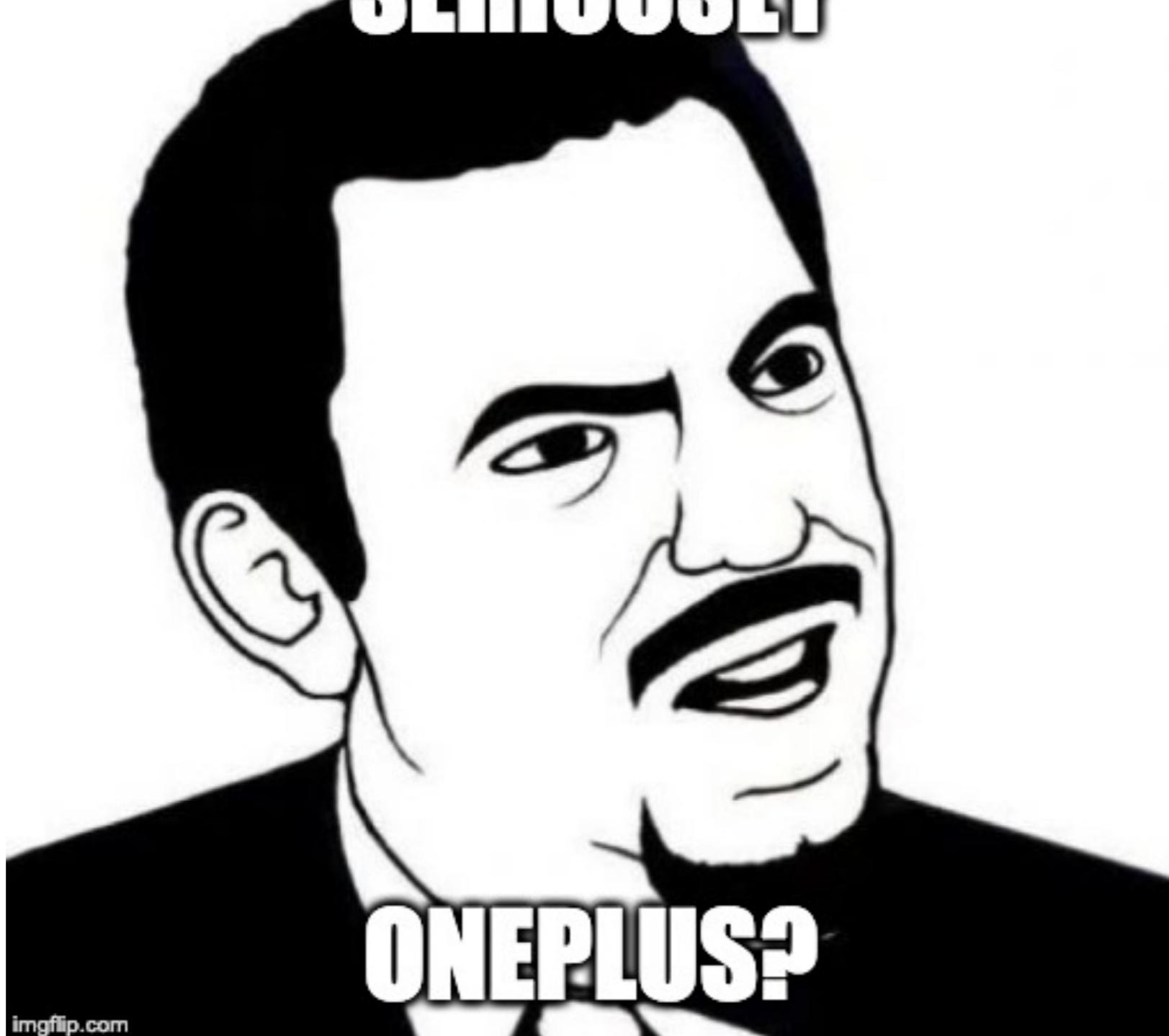


Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

# Conclusion

# SERIOUSLY



imgflip.com



Elliot Alderson  
@fs0c131y

➤ DevFest  
Toulouse 2019

Merci