Puis hacker votre application Android, please?

Meetup Paris Android/iOS Security - 24 Septembre 2019

Elliot Alderson

@fs0c131y

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

Intro to App Development

Android apps in a nutshell

- They are written in Java, Kotlin, C/C++
- Each app has unique identifier called **package name**
- Apps are built as a combination of « **components** »
  - Activity
  - Service
  - Broadcast Receiver
  - Content Provider

Activity

- It represents a single screen with a **user interface**
- You can have many: each of them defines a UI
- If the app allows it, an external app can start these activities at will

```
<manifest ... >
  <application ... >
      <activity android:name=".ExampleActivity" />
      ...
  </application ... >
  ...
</manifest >
```

Service

- Meant to perform an **action in the background** for some period of time, regardless of what the user is doing in foreground
- Example: a music player service
- They do not provide a user interface

```
<manifest ... >
  ...
  <application ... >
      <service android:name=".ExampleService" />
      ...
  </application>
</manifest>
```

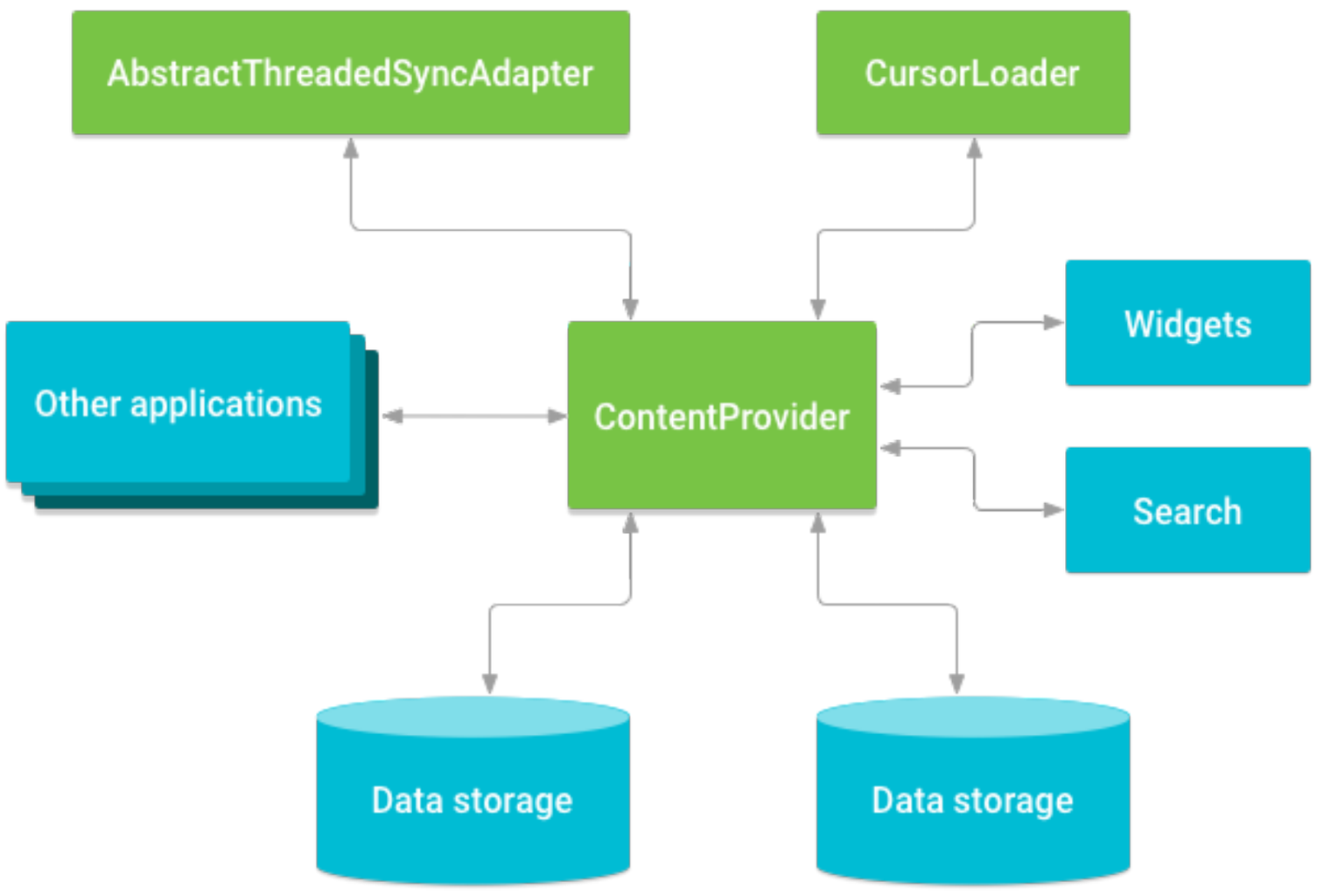
Broadcast Receiver

- They are meant to respond to **system-wide events**
- They have a well-defined entry point as well
- The system can deliver these events even to apps that are **currently not running**
- Example of events: battery charging, sms is received

```
<receiver android:name=".MyBroadcastReceiver"  android:exported="true">
    <intent-filter>
        <action android:name="android.intent.action.BOOT_COMPLETED"/>
        <action android:name="android.intent.action.INPUT_METHOD_CHANGED" />
    </intent-filter>
</receiver>
```

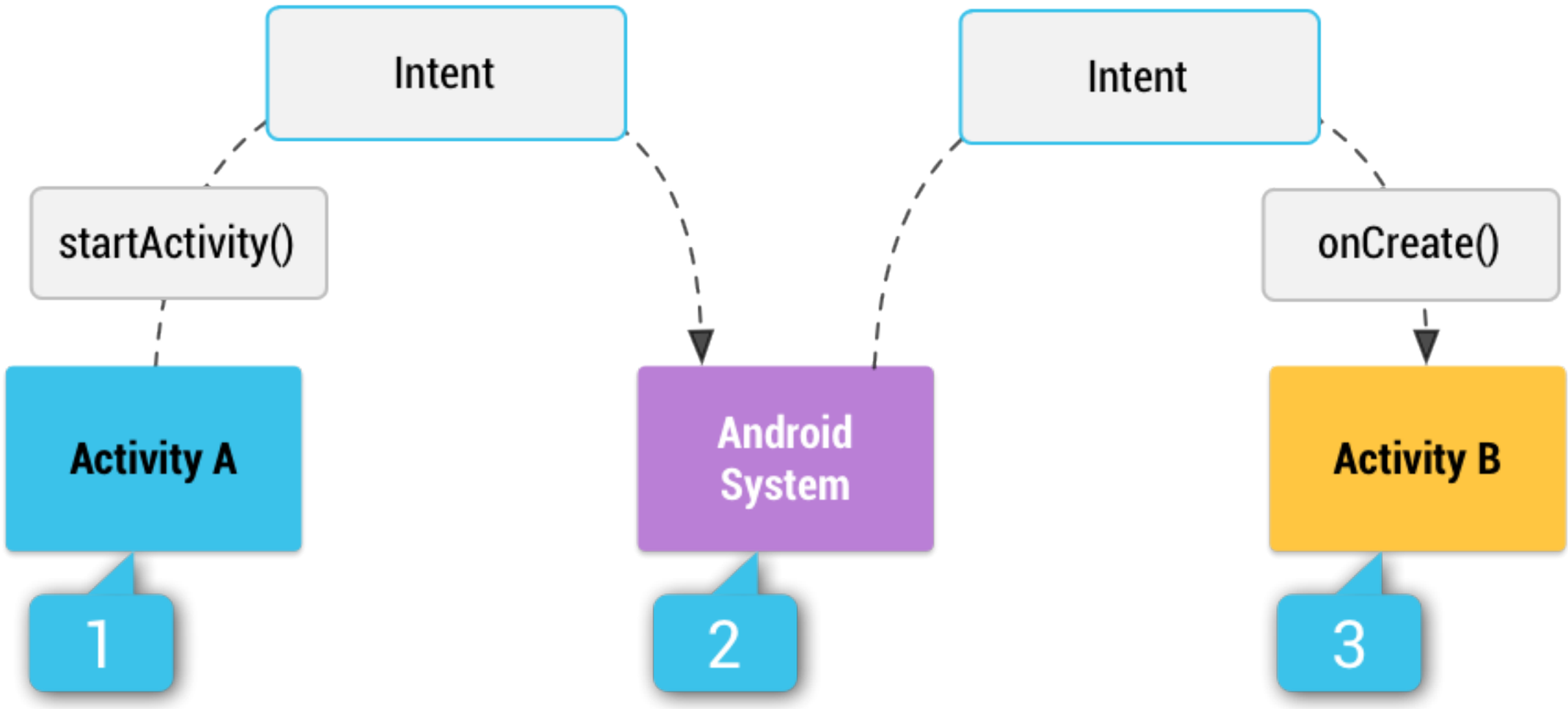
Content Provider

- High-level API to **access data** so that other apps and services can query / interact with it
- They abstract away the storing mechanism



Intents

- How can these components talk?
- Android-defined objects that encode an « intent »



Intro Android Security

App Isolation

- Android is based on Linux
- Each app has is own Linux user ID
- Each app lives in its own security sandbox
  - Standard Linux process isolation
  - Restricted file system permissions

App Isolation

- The android framework creates a new Linux user
- Each app is given private directory

```
fs0c131y@Elliots-MacBook-Pro:~$ adb shell
generic_x86_64:/ $ su
generic_x86_64:/ # cd /data/data/com.android.tencent.zdevs.bah
generic_x86_64:/data/data/com.android.tencent.zdevs.bah # ls -al
total 24
drwxr-x--x  5 u0_a67 u0_a67       4096 2019-03-11 01:18 .
drwxrwx--x 92 system system       4096 2019-03-11 01:18 ..
drwxrws--x  2 u0_a67 u0_a67_cache 4096 2019-03-11 01:18 cache
drwxrws--x  2 u0_a67 u0_a67_cache 4096 2019-03-11 01:18 code_cache
drwxrwx--x  2 u0_a67 u0_a67       4096 2019-03-11 01:18 shared_prefs
```

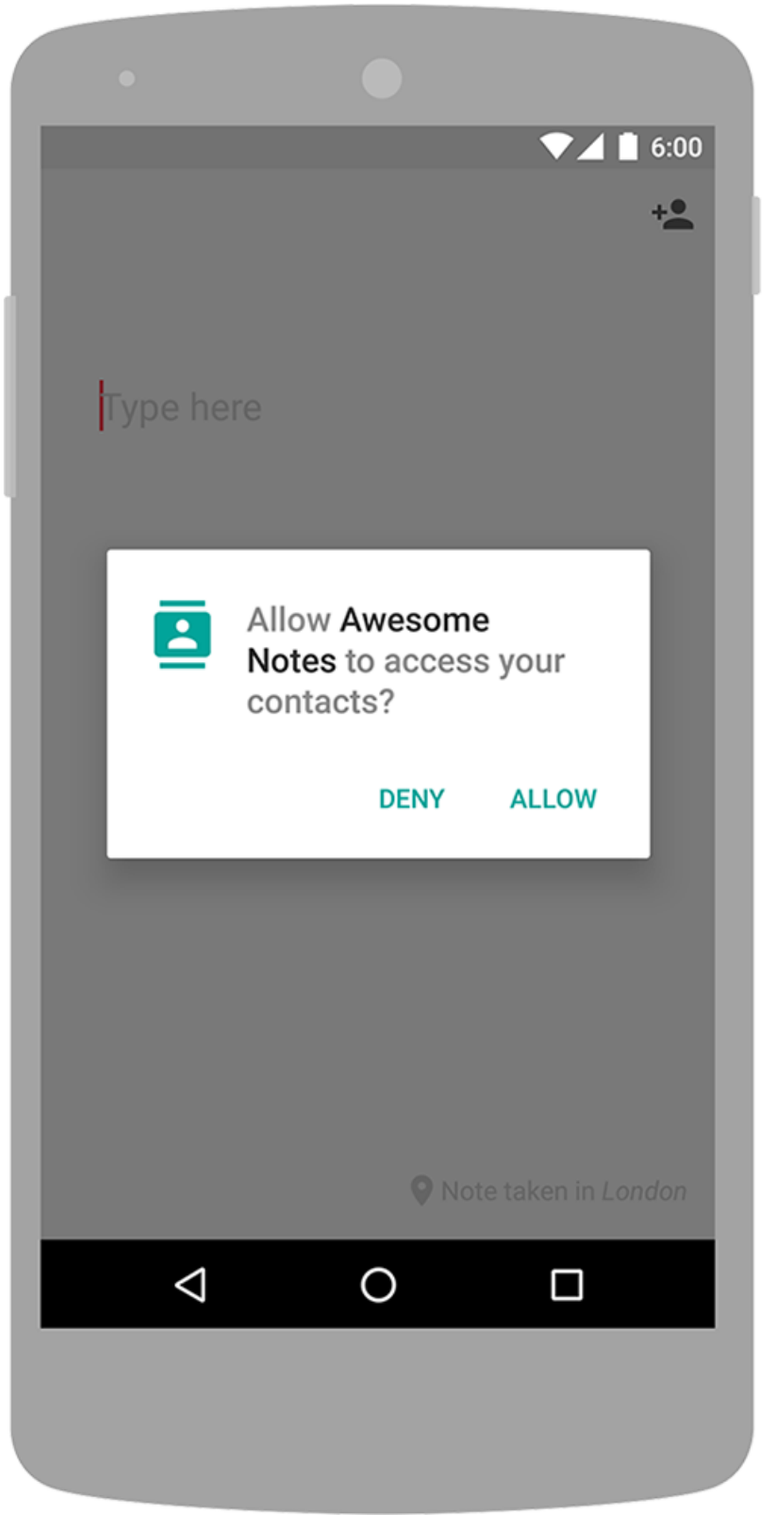
Permission System

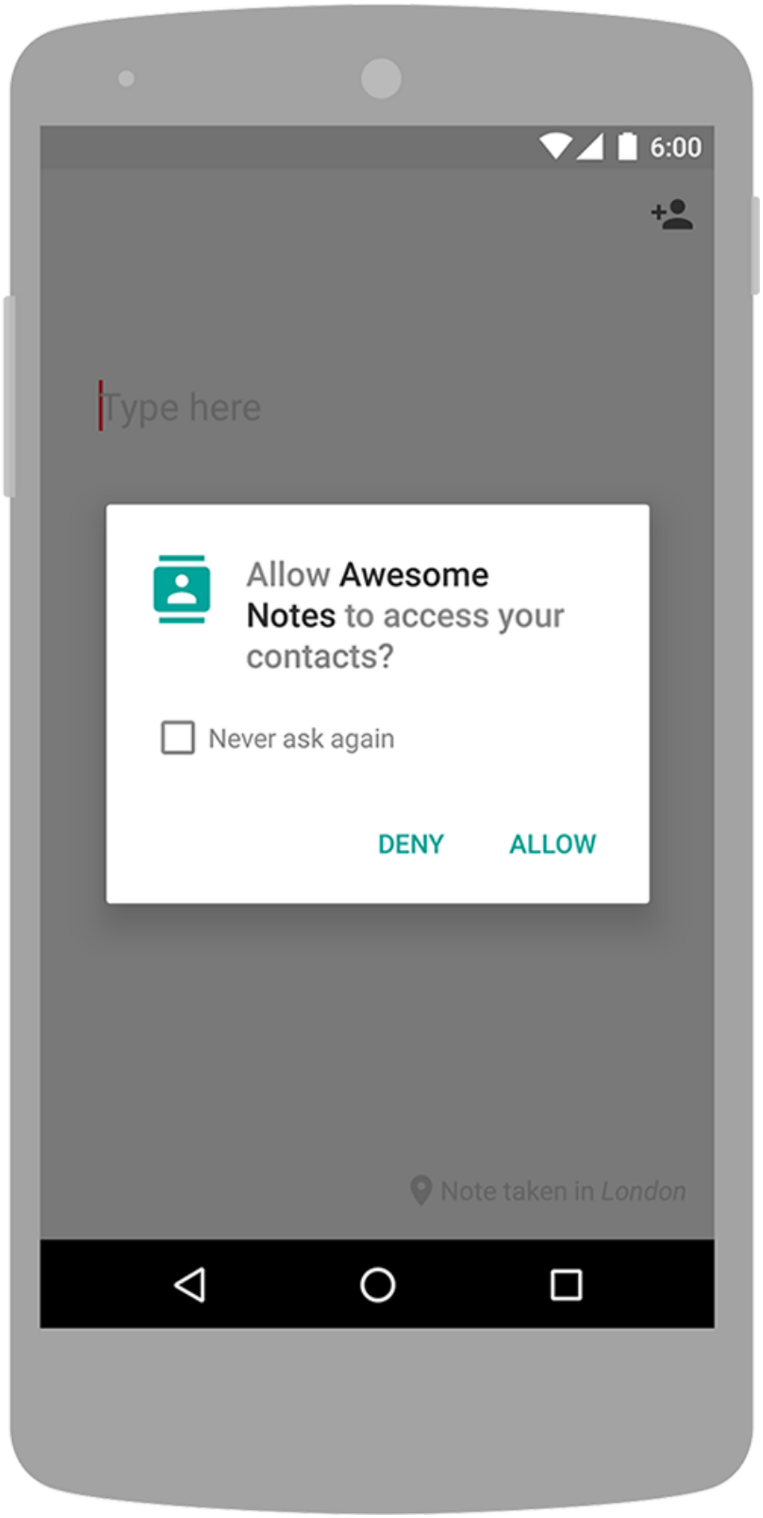
- The android framework defines a long list of permissions
- Each of these « protects » security sensitive capabilities
  - The ability to do **something sensitive**
  - The ability to **access sensitive information**

Permission Protection Levels

- **Normal**: The system automatically grants the app that permission at install time
- **Dangerous**: To use a dangerous permission, your app must prompt the user to grant permission at runtime.
- **Signature**: Granted at install time, but only when the app is signed by the same certificate as the app that defines the permission.
- **Special**: SYSTEM_ALERT_WINDOW and WRITE_SETTINGS are particularly sensitive, so most apps should not use them

Permission Protection Levels





Reverse Engineering

The art of reverse engineering

- The goal: understand ***what*** X does and ***how*** it does it
- Two approaches: Static analysis vs Dynamic Analysis
- **Static analysis**: inspect the app without running it
- **Dynamic analysis**: run the app and check what it does

Threat Model

- A "threat model" outlines what it is assumed an attacker **can do and cannot do**
- Keeping the threat model in mind is of critical importance when discussing attacks and defense systems
  - Attack X is possible under threat model T
  - Defense system Y is effective under threat model T

Tools

APKTOOL

Install Build Documentation Changes Contribute Current Version: **2.4.0**

Fork me on GitHub

A tool for reverse engineering Android apk files

```
$ apktool d test.apk
I: Using Apktool 2.4.0 on test.apk
I: Loading resource table...
I: Decoding AndroidManifest.xml with resources...
I: Loading resource table from file: 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...
$ apktool b test
I: Using Apktool 2.4.0 on test
I: Checking whether sources has changed...
I: Smaling smali folder into classes.dex...
I: Checking whether resources has changed...
I: Building resources...
I: Building apk file...
I: Copying unknown files/dir...
```

Apktool

A tool for reverse engineering 3rd party, closed, binary Android apps. It can decode resources to nearly original form and rebuild them after making some modifications. It also makes working with an app easier because of the project like file structure and automation of some repetitive tasks like building apk, etc.

It is **NOT** intended for piracy and other non-legal uses. It could be used for localizing, adding some features or support for custom platforms, analyzing

dex2jar

pxb1988 / dex2jar

Watch 397 | Star 5,763 | Fork 1,206

Code | Issues 202 | Pull requests 10 | Projects 0 | Wiki | Insights

Tools to work with android .dex and java .class files

557 commits | 2 branches | 32 releases | 6 contributors | Apache-2.0

Branch: 2.x | New pull request | Create new file | Upload files | Find File | Clone or download

pxb1988 support method handle of type INVOKE_CONSTRUCTOR/INVOKE_DIRECT/INVOKE... ... Latest commit 9cfc8ba on 5 Sep 2018

d2j-base-cmd	remove maven pom.xml	11 months ago
d2j-j6	support git/hg revision in meta-info	4 years ago
d2j-jasmin	fix build error	11 months ago
d2j-smali	support method handle of type INVOKE_CONSTRUCTOR/INVOKE_DIRECT/INVOKE...	6 months ago
dex-ir	insert a Nop between two LabelStmt if both have phis	9 months ago
dex-reader-api	support method handle of type INVOKE_CONSTRUCTOR/INVOKE_DIRECT/INVOKE...	6 months ago
dex-reader	support method handle of type INVOKE_CONSTRUCTOR/INVOKE_DIRECT/INVOKE...	6 months ago
dex-tools	[dex038] write class version 1.7 if dex version > DEX_037	9 months ago
dex-translator	support method handle of type INVOKE_CONSTRUCTOR/INVOKE_DIRECT/INVOKE...	6 months ago
dex-writer	remove maven pom.xml	11 months ago
gradle/wrapper	update gradle to 4.0	11 months ago
.hgignore	ignore build folder for gradle	5 years ago
.hgtags	clean exec mode from file	6 years ago
.travis.yml	use jdk8 for travis	11 months ago

JADX

skylot / jadx

Watch 647 | Star 17,791 | Fork 1,882

Code | Issues 129 | Pull requests 2 | Projects 1 | Insights

Dex to Java decompiler

android dex java decompiler

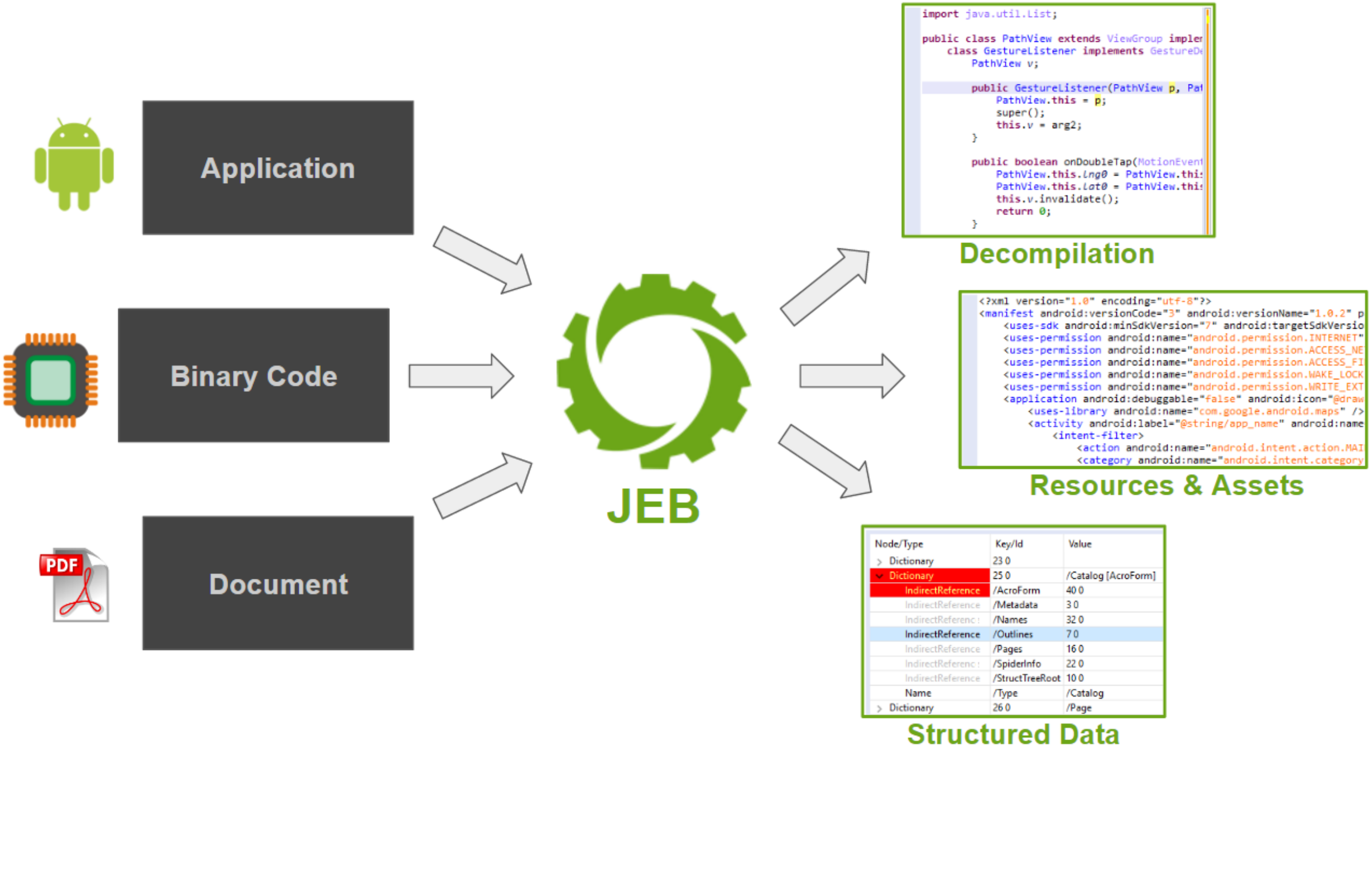
813 commits | 5 branches | 12 releases | 29 contributors | Apache-2.0

Branch: master | New pull request | Create new file | Upload files | Find File | Clone or download

jpstotz and skylot fix: restore support for AAR files (issue #95) (PR #464) — Latest commit 4353890 a day ago

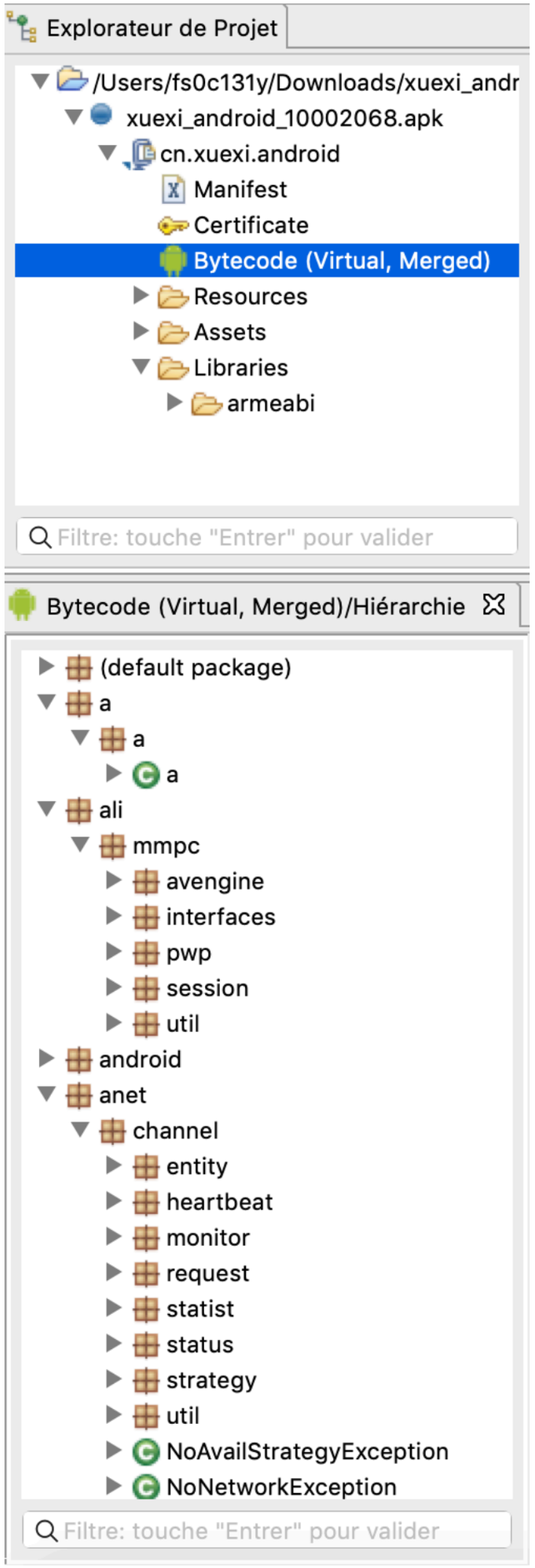
gradle/wrapper	chore: update dependencies and gradle	2 months ago
jadx-cli	fix: change not allowed access modifiers for methods (#387) (PR #439)	27 days ago
jadx-core	fix: restore support for AAR files (issue #95) (PR #464)	a day ago
jadx-gui	fix(gui): remove output directories from persistent settings (#447)	19 days ago
jadx-samples	build jadx-gui.exe	a year ago
jadx-test-app	build: remove sonar plugin from gradle config (fix #140)	2 years ago
scripts	feat(build): use semantic-release for automatic release publishing	7 months ago
.codecov.yml	build: disable codecov pull request report	a year ago
.gitignore	fix: force rename fields and methods with reserved names (#364)	5 months ago
.gitlab-ci.yml	build: fix gitlab config	6 months ago
.gitmodules	test: added module for check recompilation of test app	4 years ago
.releaserc.yml	chore: don't use labels for artifacts in github release	7 months ago
.travis.yml	build: add java 11 to build on travis	2 months ago
CONTRIBUTING.md	Create CONTRIBUTING.md	a year ago

JEB Decompiler



APK Structure

- Zip file
  - AndroidManifest.xml
  - classes.dex
  - res/*
  - lib/*
  - assets/*

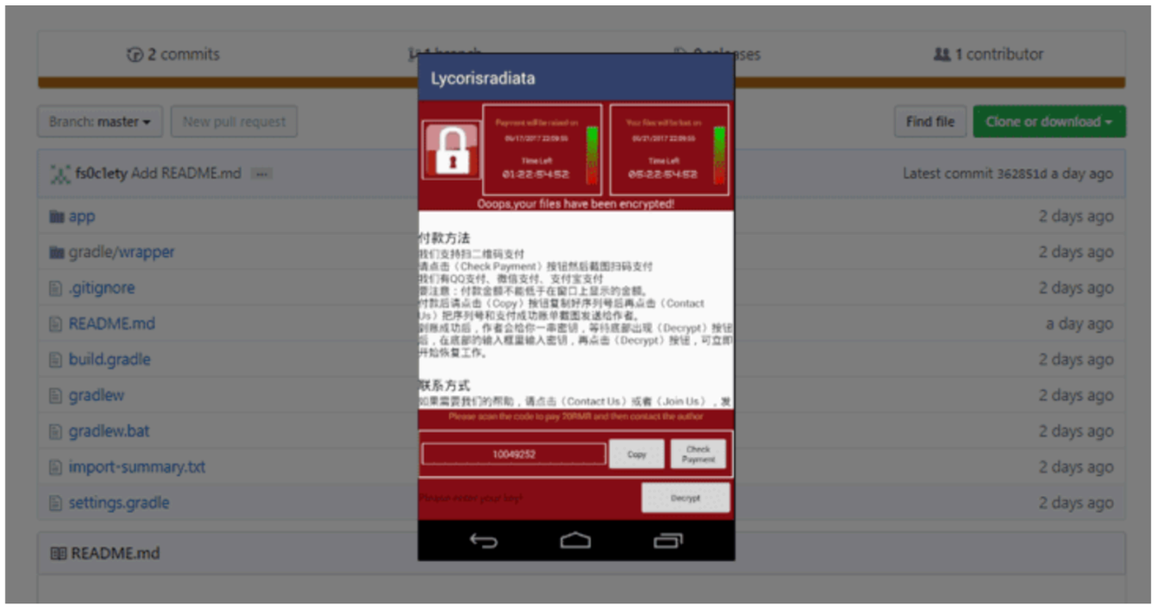


Exploitation

SLocker

Decompiled SLocker Android Ransomware Source Code Published Online

July 24, 2017 Swati Khandelwal



SLocker

SLocker Mobile Ransomware Starts Mimicking WannaCry

Posted on: July 5, 2017 at 7:00 am **Posted in:** Mobile, Ransomware
Author: Mobile Threat Response Team

by Ford Qin

Early last month, a new variant of mobile ransomware SLocker (detected by Trend Micro as ANDROIDOS_SLOCKER.OPST) was detected, copying the GUI of the now-infamous WannaCry. The SLocker family is one of the oldest mobile lock screen and file-encrypting ransomware and used to impersonate law enforcement agencies to convince victims to pay their ransom. After laying low for a few years, it had a sudden resurgence last May. This particular SLocker variant is notable for being an Android file-encrypting ransomware, and the first mobile ransomware to capitalize on the success of the previous WannaCry outbreak.

While this SLocker variant is notable for being able to encrypt files on mobile, it was quite short-lived. Shortly after details about the ransomware surfaced, decrypt tools were published. And before long, more variants were found. Five days after its initial detection, a suspect supposedly responsible for the ransomware was arrested by the Chinese police. Luckily, due to the limited transmission channels (it was spread mostly through forums like QQ groups and Bulletin Board

Featured Stories

systemd Vulnerability Leads to Denial of Service on Linux

qkG Filecoder: Self-Replicating, Document-Encrypting Ransomware

Mitigating CVE-2017-5689, an Intel Management Engine Vulnerability

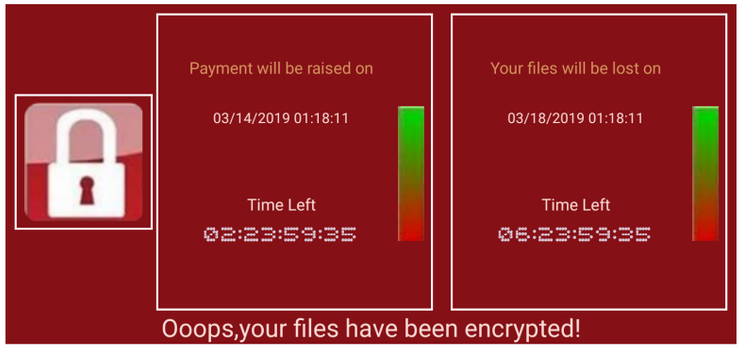
A Closer Look at North Korea's Internet

From Cybercrime to Cyberpropaganda

Security Predictions for 2019

Our security predictions for 2019 are based on our experts' analysis of the progress of current and emerging technologies, user behavior, and market trends, and their impact on the threat landscape. We have categorized them according to the main areas that are likely

SLocker



我的手机出了什么问题?

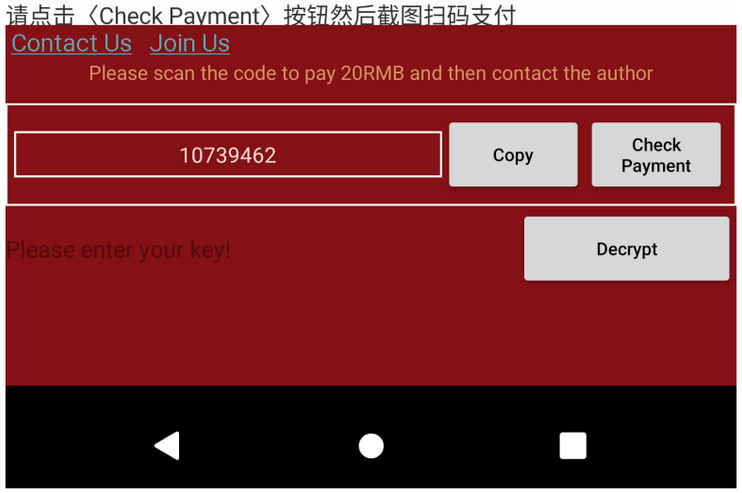
您的一些重要文件被我加密保存了。
照片、图片、文档、压缩包、音频、视频文件、txt文件等，几乎所有类型的文件都被加密了，因此不能正常打开。
这和一般文件损坏有本质上的区别。您大可在网上找找恢复文件的方法，我敢保证，没有我们的解密服务，就算老天爷来了也不能恢复这些文档。

有没有恢复这些文档的方法?

当然有可恢复的方法。只能通过我们的解密服务才能恢复。我以人格担保，能够提供安全有效的恢复服务。
但这是收费的，也不能无限期的推迟。请您放心，我是绝不会骗你的。
是否随时都可以固定金额付款，就会恢复的吗，当然不是，推迟付款时间越长对你不利。
最好3天之内付款费用，过了三天费用就会翻倍。
还有，一个礼拜之内未付款，将会永远恢复不了。
对了，忘了告诉你，对半年以上没钱付款的穷人，会有活动免费恢复，能否轮到你，就要看您的运气怎么样了。

付款方法

我们支持扫二维码支付
请点击〈Check Payment〉按钮然后截图扫码支付



Demo

OnePlus Angela Root

MOTHERBOARD

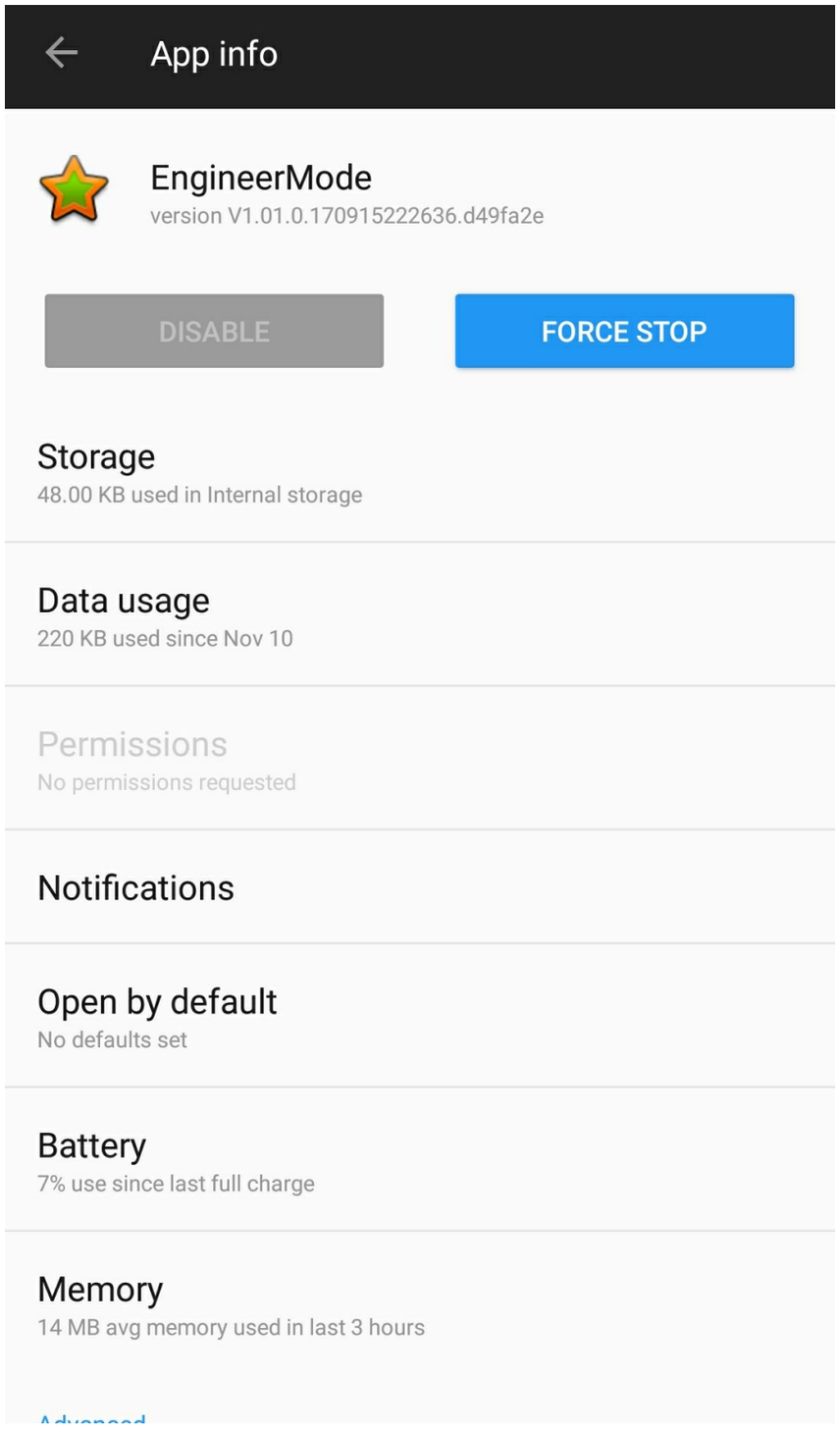
VICE

BUGS | By Louise Matsakis | Nov 14 2017, 7:07pm

OnePlus Phones Were Shipped With a Hidden Backdoor

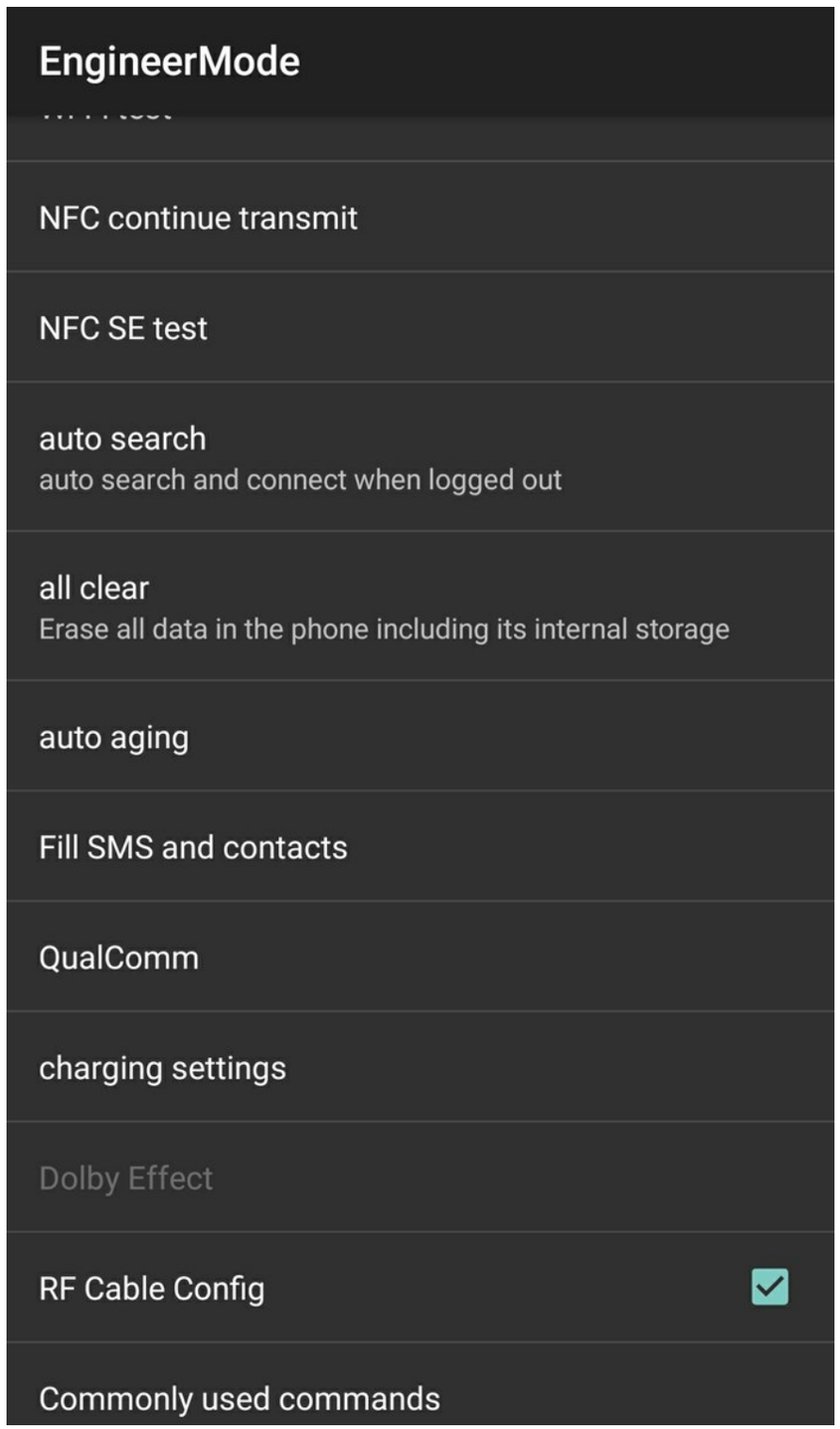
A pre-installed factory app called Engineer Mode can root devices.

OnePlus Angela Root



OnePlus Angela Root

adb shell am start com.android.engineeringmode/.EngineeringMode



Demo

mAadhaar

Flaw in mAadhaar app reportedly makes stealing data easier

Siladitya Ray

By Siladitya Ray (@SiladityaRay siladitya@medianama.com) January 12, 2018
Share This: Share via Email

DAILY NEWSLETTER

Enter your email address... →

HEADLINES

- Jio and Hathway blocking Reddit in some cities, working in others
- AIDSO protests the failure of biometric authentication system and student detention in colleges
- Gujarat bans PUBG in 5 districts citing violent behaviour among kids: reports
- CAIT wants to bundle cab and delivery cos into the commerce policy
- YouTube will label fact-checked videos in India; no partners specified
- Ecommerce policy: DPIIT extends the comments deadline to 29 March
- Key takeways from Election Commission's 2019 India's 2019 Elections announcement: On Fake News, Online Political Advertising and Model Code of Conduct
- Strange bedfellows: Facebook and Privacy
- BJP site offline 4 days after being hacked

mAadhaar

```
public static String generateDBPassword() {
    Random random = new Random();
    random.setSeed(123456789);
    String encodeBase64 = encodeBase64( str: "db_password_123" + random.nextInt( n: 10));
    Log.d(TAG,  msg: "Password: " + encodeBase64);
    return encodeBase64;
}
```

mAadhaar

Aadhaar Database Password

Database password = db_password_1235
Encoded database password = ZGJfcGFzc3dvcmRfMTIzNQ==

Donald Daters

CULTURE > NEWS

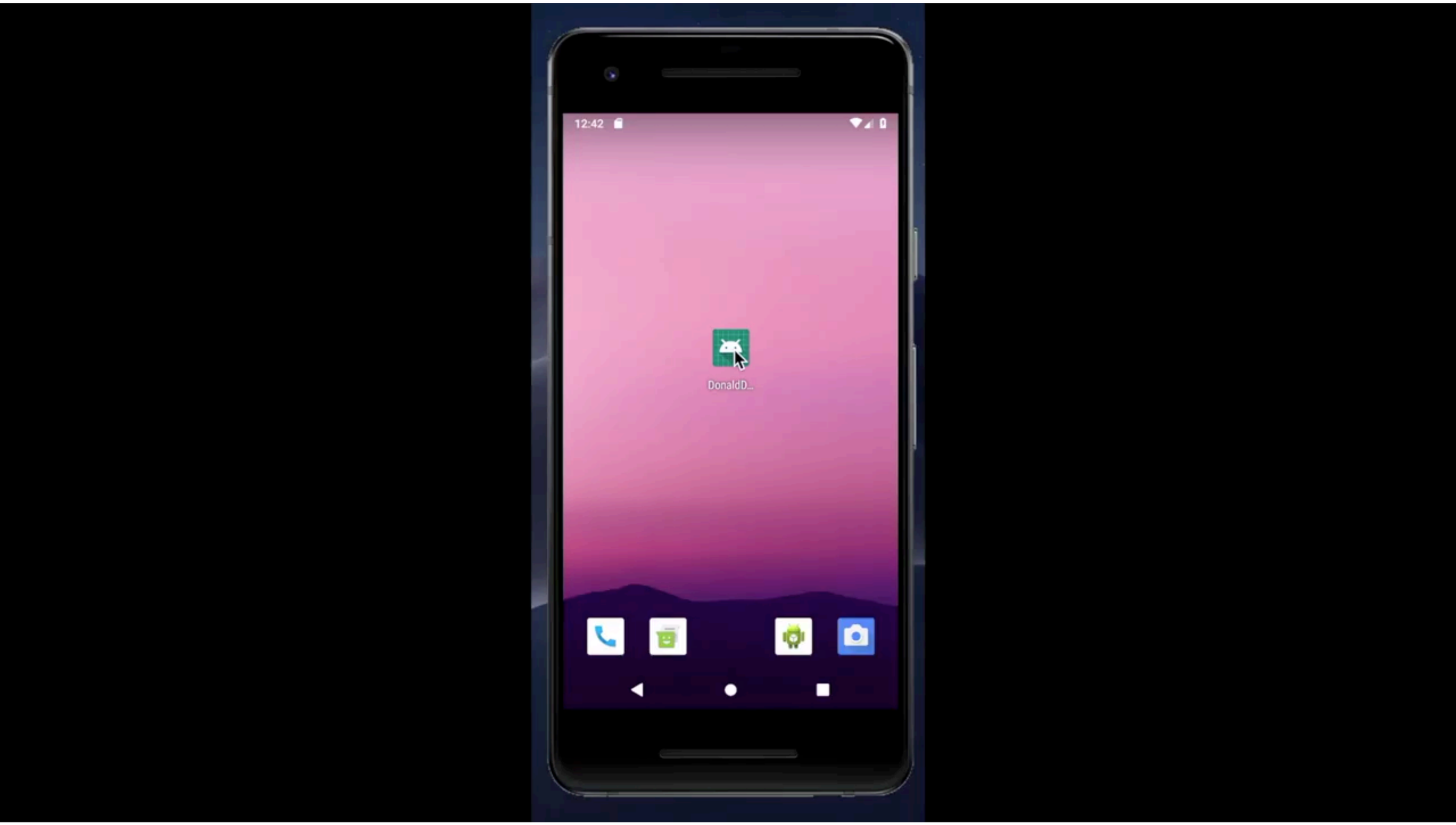
DonaldDaters, a Dating App for Trump Supporters, Launches—And Then Leaks Its Users' Data

OCTOBER 16, 2018 5:50 PM
by BRIDGET READ

Donald Daters

https://donalddaters2018.firebaseio.com/.json

Donald Daters



ES File Explorer

Login

Startups
Apps
Gadgets
Videos
Podcasts
Extra Crunch

—

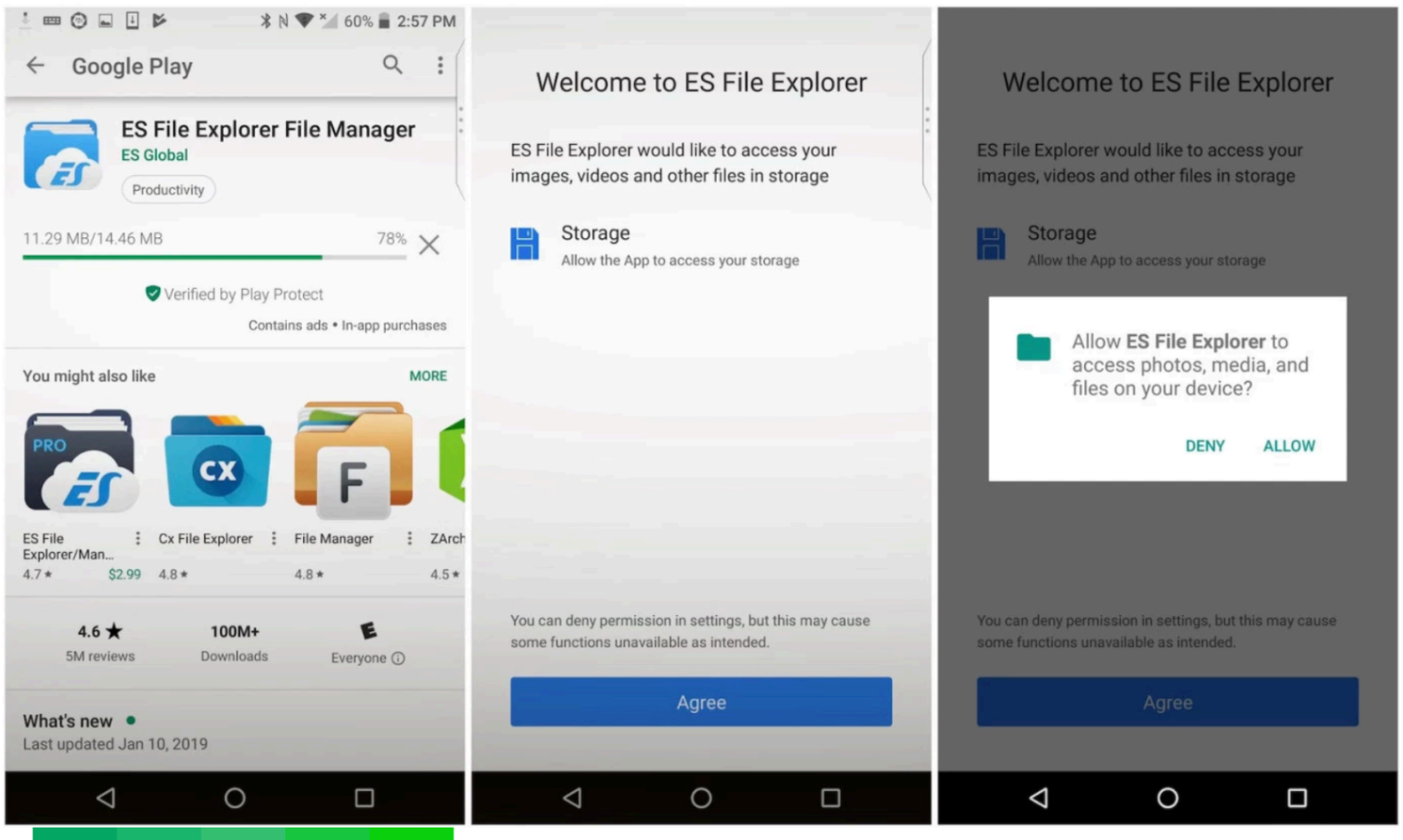
Events
Advertise
Crunchbase
More

Cybersecurity 101
Tesla
Fundings & Exits

Researcher shows how popular app ES File Explorer exposes Android device data

Zack Whittaker @zackwhittaker / 2 months ago

Comment



ost popular **Android** apps running a hidden web

ES File Explorer

ES File Explorer Open Port Vulnerability - CVE-2019-6447

As per their Google Play description:

> ES File Explorer (File Manager) is a full-featured file (Images, Music, Movies, Documents, app) manager for both local and networked use! With over 500 million users worldwide, ES File Explorer (File Manager) helps manage your android phone and files efficiently and effectively and share files without data cost.

Everytime a user is launching the app, a HTTP server is started. This server is opening locally the port 59777:

```
angler:/ # netstat -ap | grep com.estrongs
tcp6       0      0 :::59777                :::*                    LISTEN      5696/com.estrongs.android.pc
```

On this port, an attacker can send a JSON payload to the target

```
curl --header "Content-Type: application/json" --request POST --data '{"command":"[my_awesome_cmd]"}' http:/
```

These commands allow an attacker **connected on the same local network to the victim**, to obtain a lot of juicy information (device info, app installed, ...) about the victim's phone, **remotely get a file** from the victim's phone and **remotely launch an app** on the victim's phone.

Affected Versions

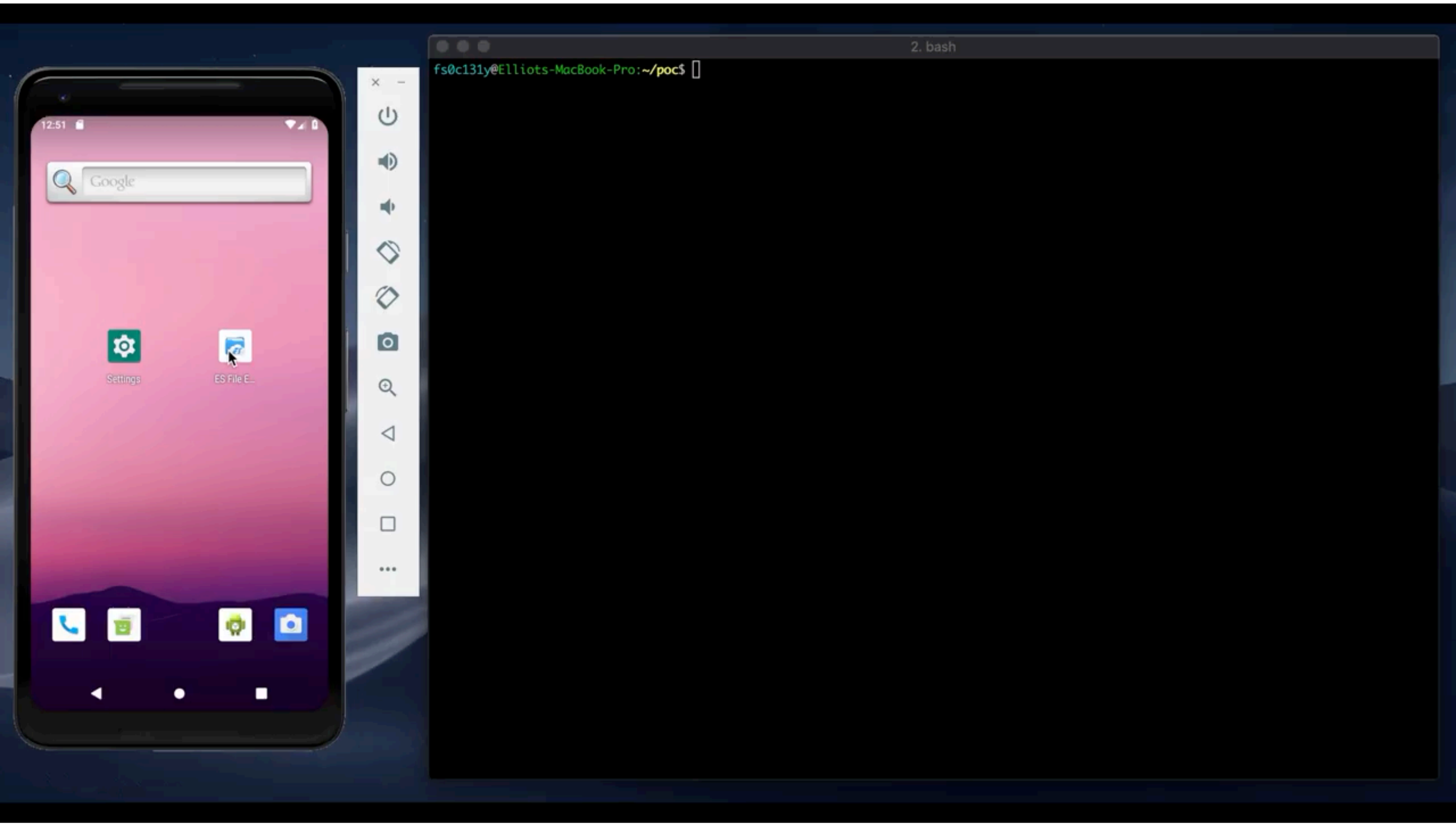
4.1.9.7.4 and below

POC Features

With the following Proof Of Concept (POC), you can:

- List all the files in the sdcard in the victim device
- List all the pictures in the victim device
- List all the videos in the victim device
- List all the audio files in the victim device
- List all the apps installed in the victim device
- List all the system apps installed in the victim device

ES File Explorer



63red

APPS · **Published** 1 day ago

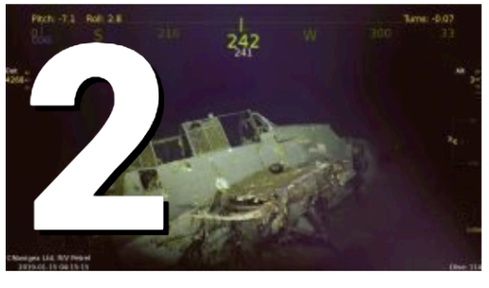
Pro-Trump app threatens expert for finding flaw in code

PCmag

Trending in Tech

1


Amazon warehouse conditions drove her to consider suicide, former employee tells Tucker Carlson

2

Wreck of WWII aircraft carrier USS Wasp discovered in the Coral Sea

Demo

Thank you