22-24 October 2013 - Luxembourg
9th edition of the infosec conference
"We're not computers, Sebastian, we're physical"
Roy Batty in Blade Runner

Phishing with permission

tom leclerc
tom.leclerc [-at-] telindus.lu

joany boutet
joany.boutet [-at-] telindus.lu

About the Speakers

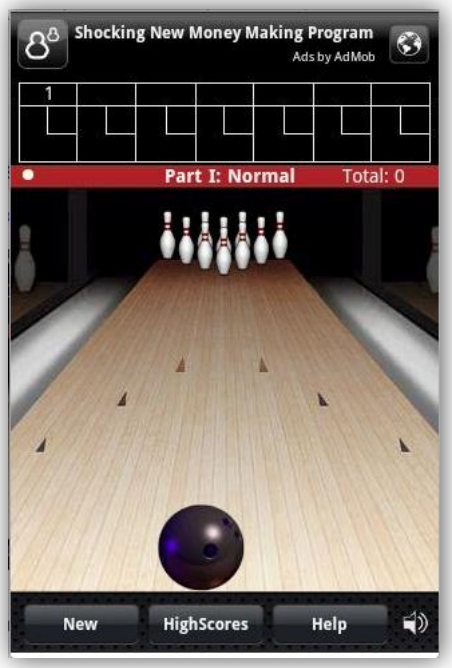
- Tom Leclerc and Joany Boutet are Security Consultants working for Security, Audit and Governance Services, a Telindus Luxembourg Security department.
- Tom
  - Ph.D. in computer science
    - Specialist in distributed systems and networks
    - Involved in several ESA projects
- Joany
  - Main focus on penetration testing
  - Has already written paper about Android security
    - Paper *Malicious Android Applications: Risks and Exploitation - "A Spyware story about Android Application and Reverse Engineering"* - (22/03/10) – Available in the SANS Reading Room

What we won't/will cover

- ~~New phishing technique~~
  - DEF CON 19 - Nicholas J. Percoco & Sean Schulte
    - "This is REALLY not the droid you're looking for…"
- ~~Distribution of free copy (virus-free) of GTA V~~ ☺
- Bypass the Android permission model
  - New Technique for hiding Android Malware

Evolution of Android Malware

- August 2010 - Application “Movie Player”
  - First Android SMS Trojan Found in the Wild
- December 2010 - Geinimi Trojan
  - First one that has botnet-like capabilities
  - Found in repackaged versions of legitimate applications
- March 2011 - DroidDream Malware
  - First one that uses an exploit to gain root permissions



Evolution of Android Malware

- April 2013 - “BadNews” malware family
  - Distributed as an ad framework for developers

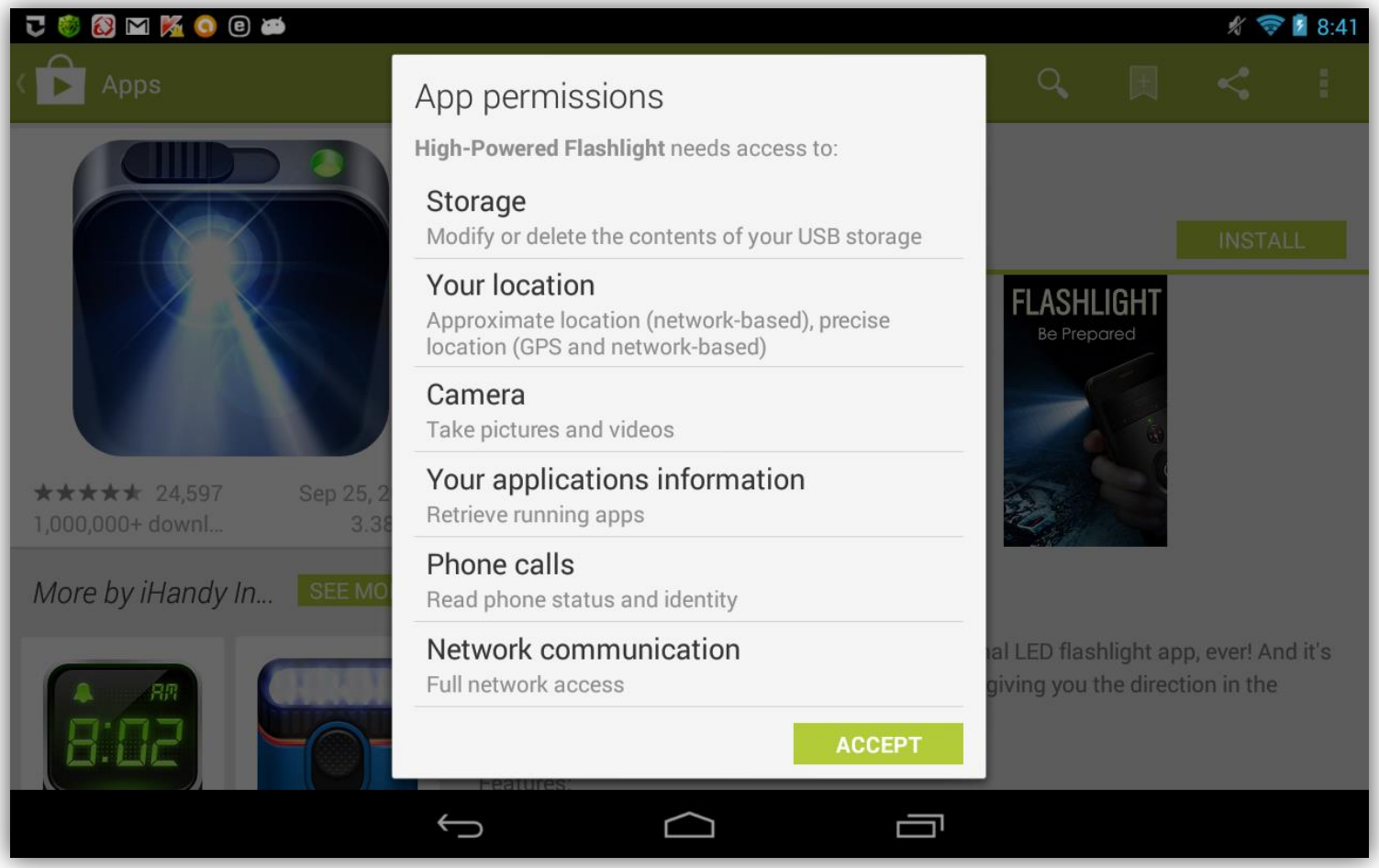
Icons of apps containing BadNews found by Lookout live in the Google Play Store.

- July 2013 - First Android Malware that uses the Master Key' Android Vulnerability
  - Allows attackers to inject malicious code into legitimate Android applications without invalidating the digital signature

- September 2013 - JollyBot - Malware as a Service

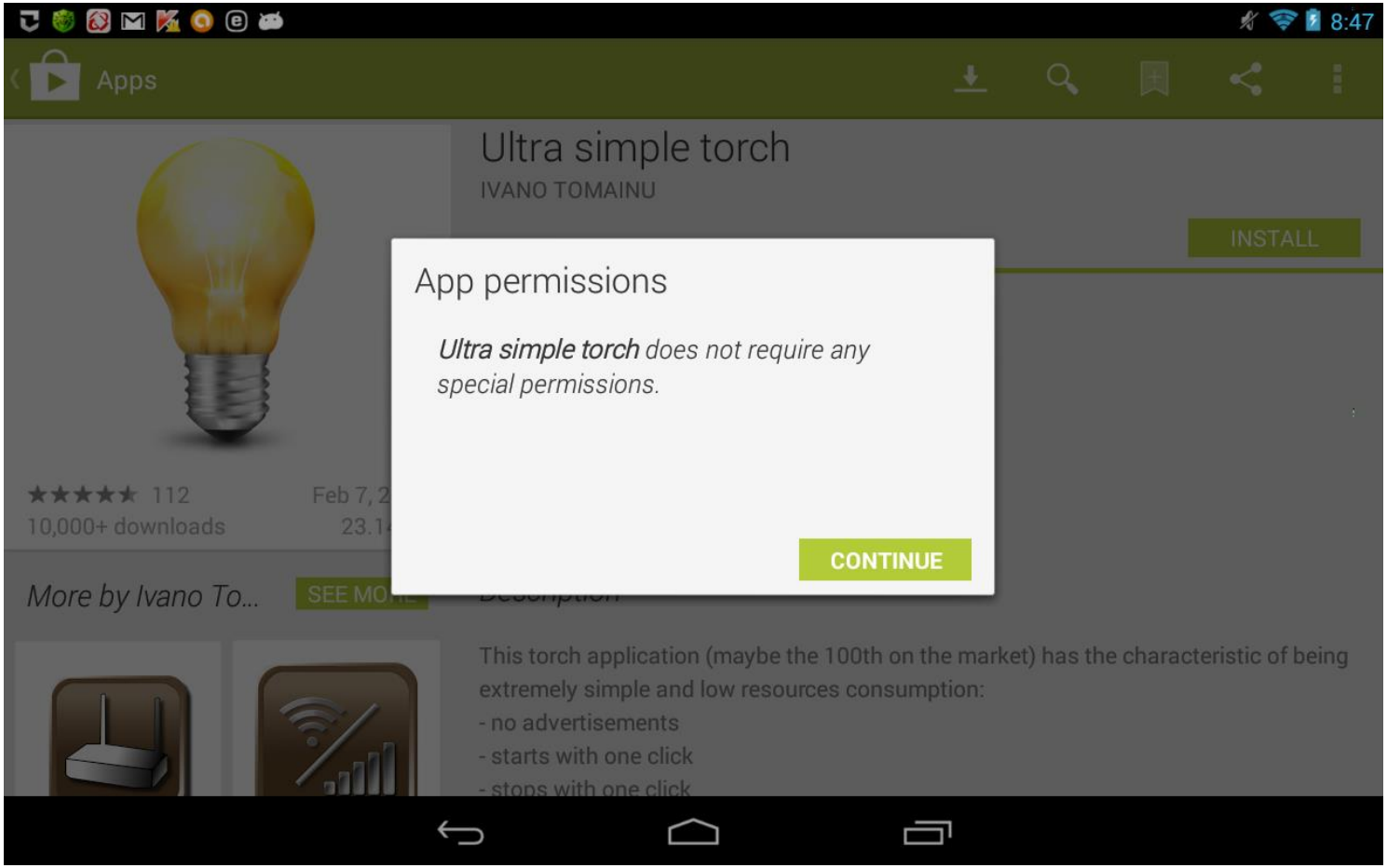
Audience Poll

- Would you install those applications ?



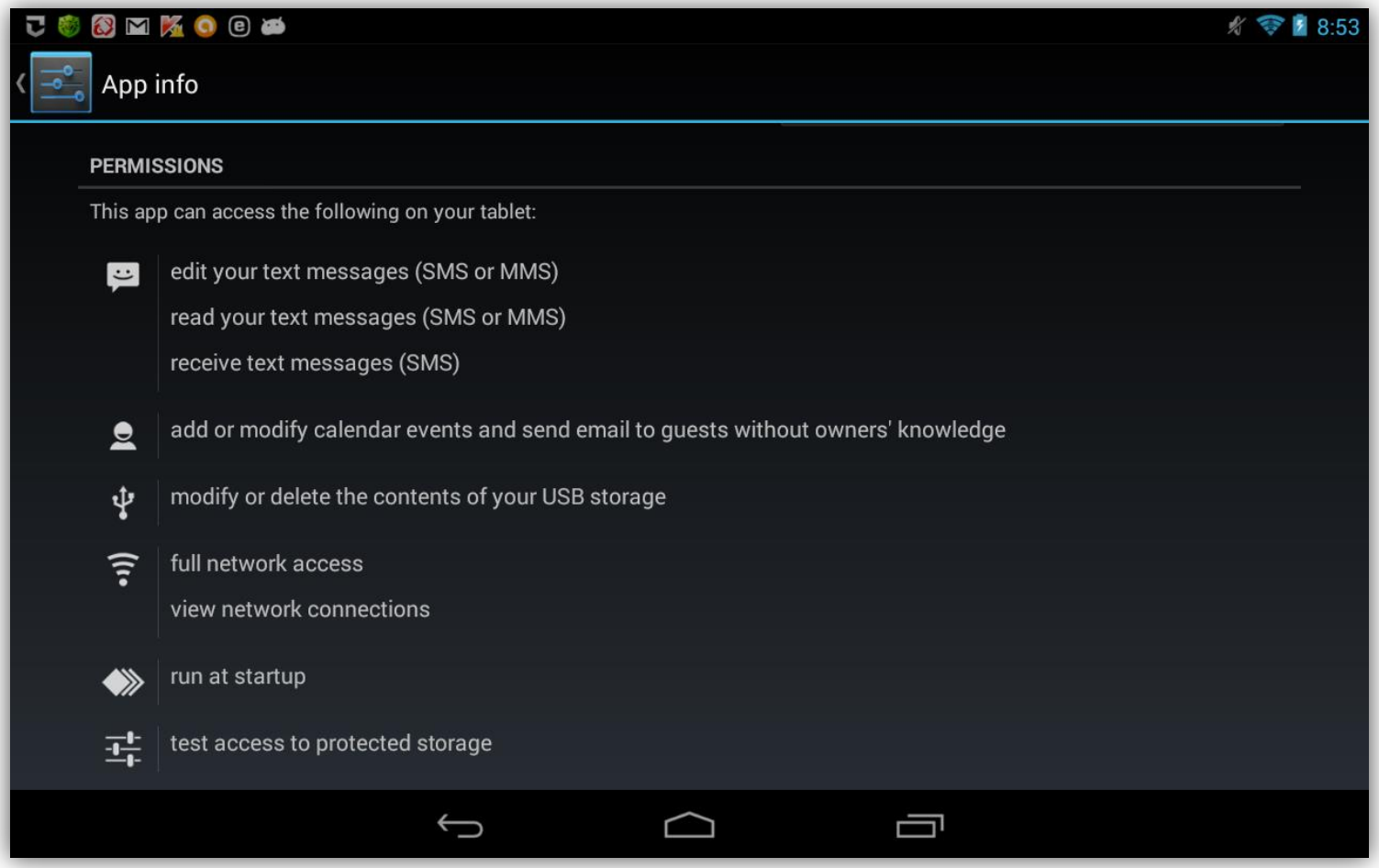
Audience Poll

- Would you install those applications ?



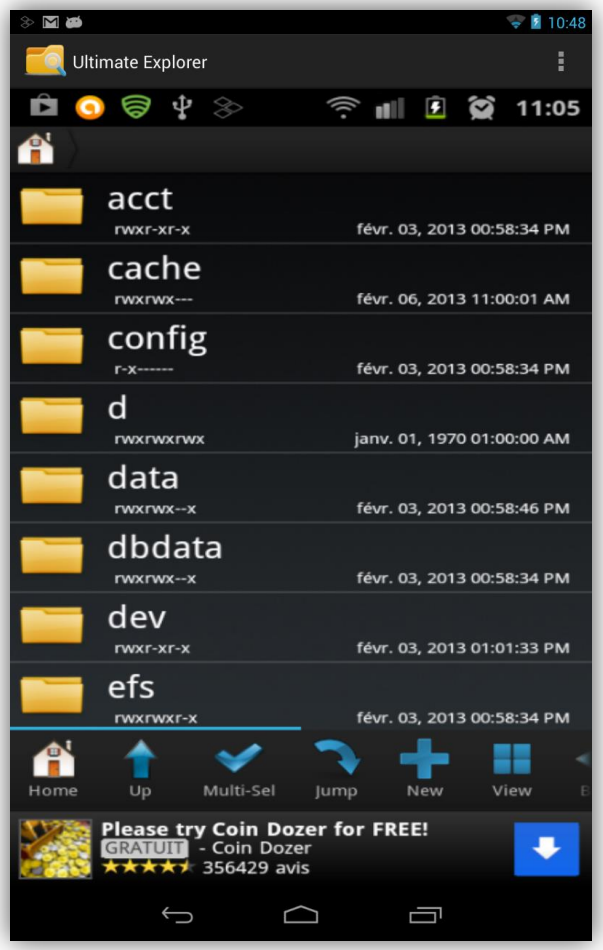
Audience Poll

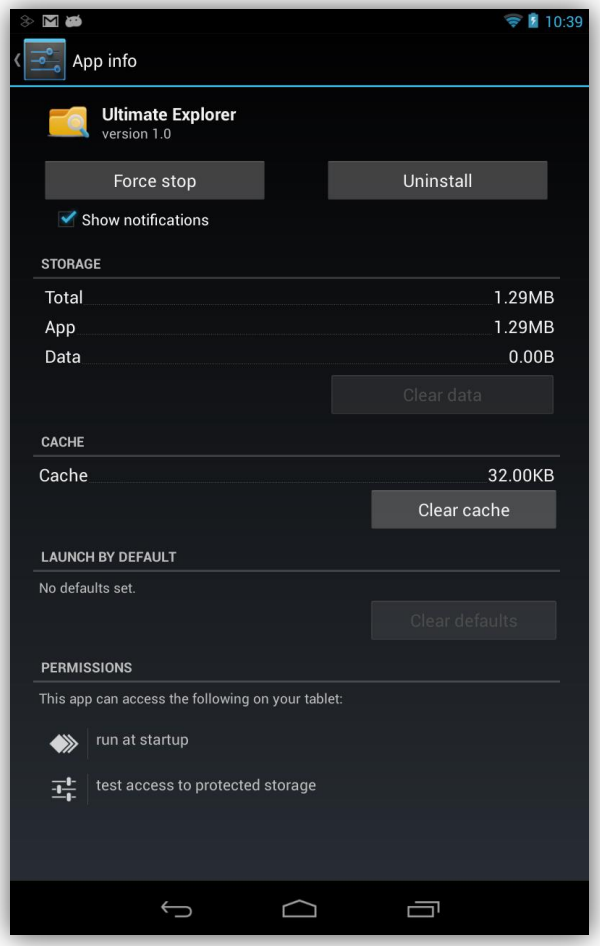
- Would you install those applications ?



Audience Poll

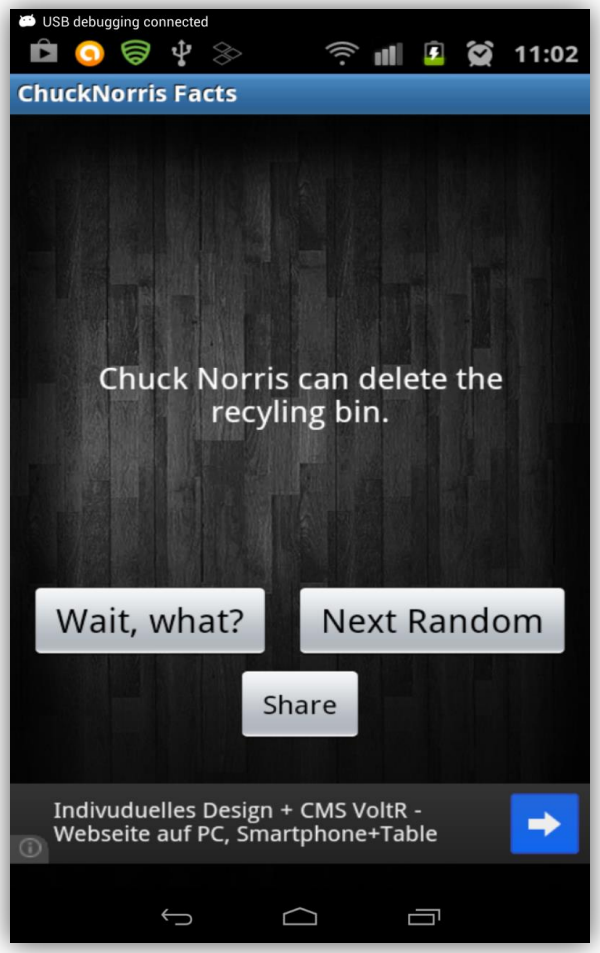
- Would you install those applications ?

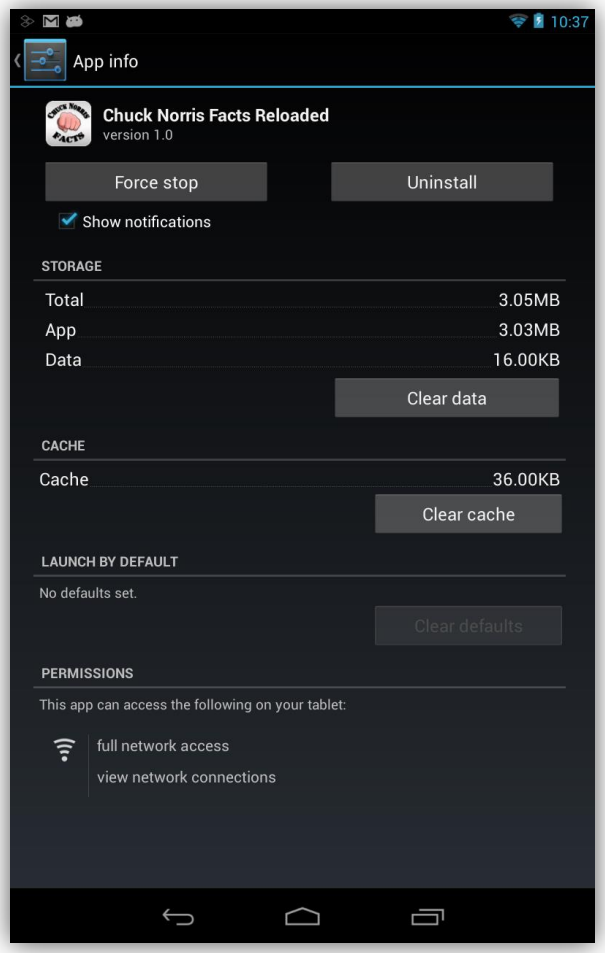




Audience Poll

- Would you install those applications ?





Have You Made the Right Choice ?

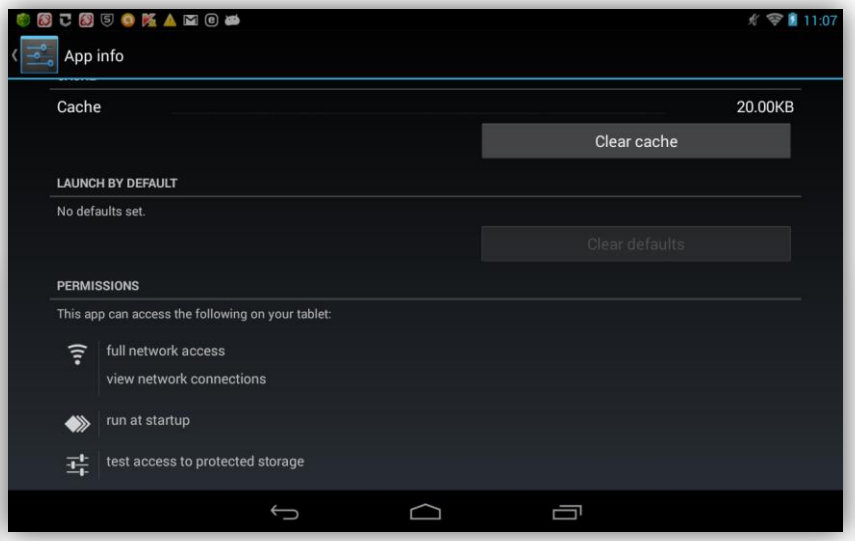
Video available on sagsblog.telinduslab.lu

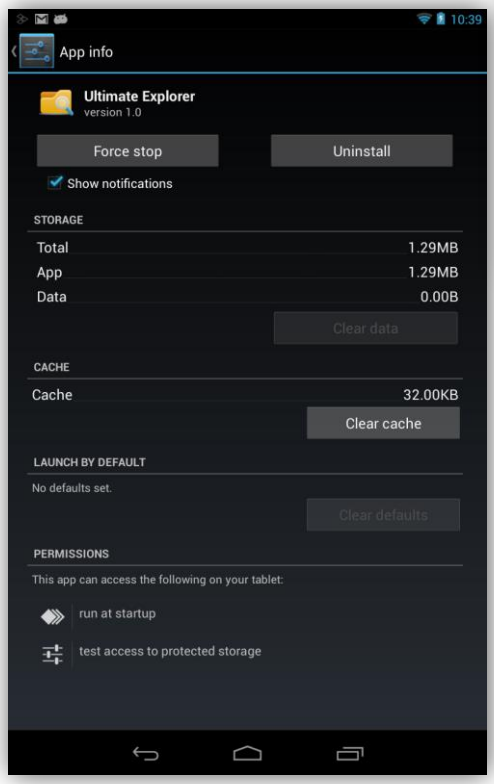
22-24 October 2013 - Luxembourg
9th edition of the infosec conference

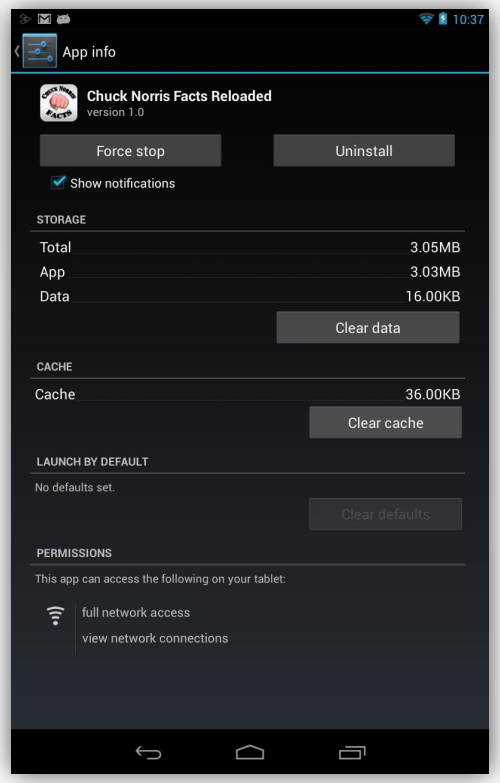
"We're not computers, Sebastian, we're physical"
Roy Batty in Blade Runner

Application Phishing via a Distributed Malware

Application Phishing via a Distributed Malware – *Phishing under the hood (1/5)*







Application Phishing via a Distributed Malware – *Phishing under the hood (2/5)*

Inter-Process Communication via Intents

Chuck Norris Facts Reloaded Application - AndroidManifest file

```
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
```

```
<activity
    android:name="lu.telindus.sags.chucknorrisfactsreloaded.AppMain"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />

        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
</activity>

<activity android:name="lu.telindus.sags.chucknorrisfactsreloaded.Login" android:configChanges="keyboardHidden|orientation" android:exported="true"/>

<service android:name=".CmdExec" android:exported="true">
    <intent-filter>
        <action android:name="isaca.telindus.get.dme.credentials"/>
        <action android:name="isaca.telindus.scan.shared.folders"/>
        <action android:name="isaca.telindus.download.network.file"/>
        <action android:name="isaca.telindus.upload.sdcard.content"/>
    </intent-filter>
</service>
```

Application Phishing via a Distributed Malware – *Phishing under the hood (3/5)*

- Security Weaknesses introduced by Intents

Proof of concept piece of malware - "Facebook" for Android 1.8.1

the *com.facebook.katana.LoginActivity* had a vulnerable intent which allowed the exfiltration of data

Facebook for Android - Information Diclosure Vulnerability

From: mbsdtest01 () gmail com
Date: Mon, 7 Jan 2013 13:58:14 GMT

```
Title:              Facebook for Android - Information Diclosure Vulnerability
Affected Software:  Facebook Application 1.8.1 for Android
                    (Confirmed on Android 2.2)
Credit:             Takeshi Terada
Issue Status:       v1.8.2 was released which fixes this vulnerability

Overview:
  The LoginActivity of Facebook app has improper intent handling flaw.
  The flaw enables malicious apps to steal Facebook app's private files.

Details:
  LoginActivity of Facebook app is "exported" to other apps. When
  the activity is called and the user is logged-in to Facebook, the
  activity pulls out an intent named "continuation_intent" from the
  extra data of the incoming intent. Then LoginActivity launches
  another activity by using continuation_intent.

  This behavior is dangerous because the actions described in the
  intent (continuation_intent) given by other apps is performed
  in the context (permission and identity) of Facebook app.

  This enables attacker's apps to call (and attack) Facebook app's
  private (not "exported") activities, by using LoginActivity as a
  stepping-stone.
```

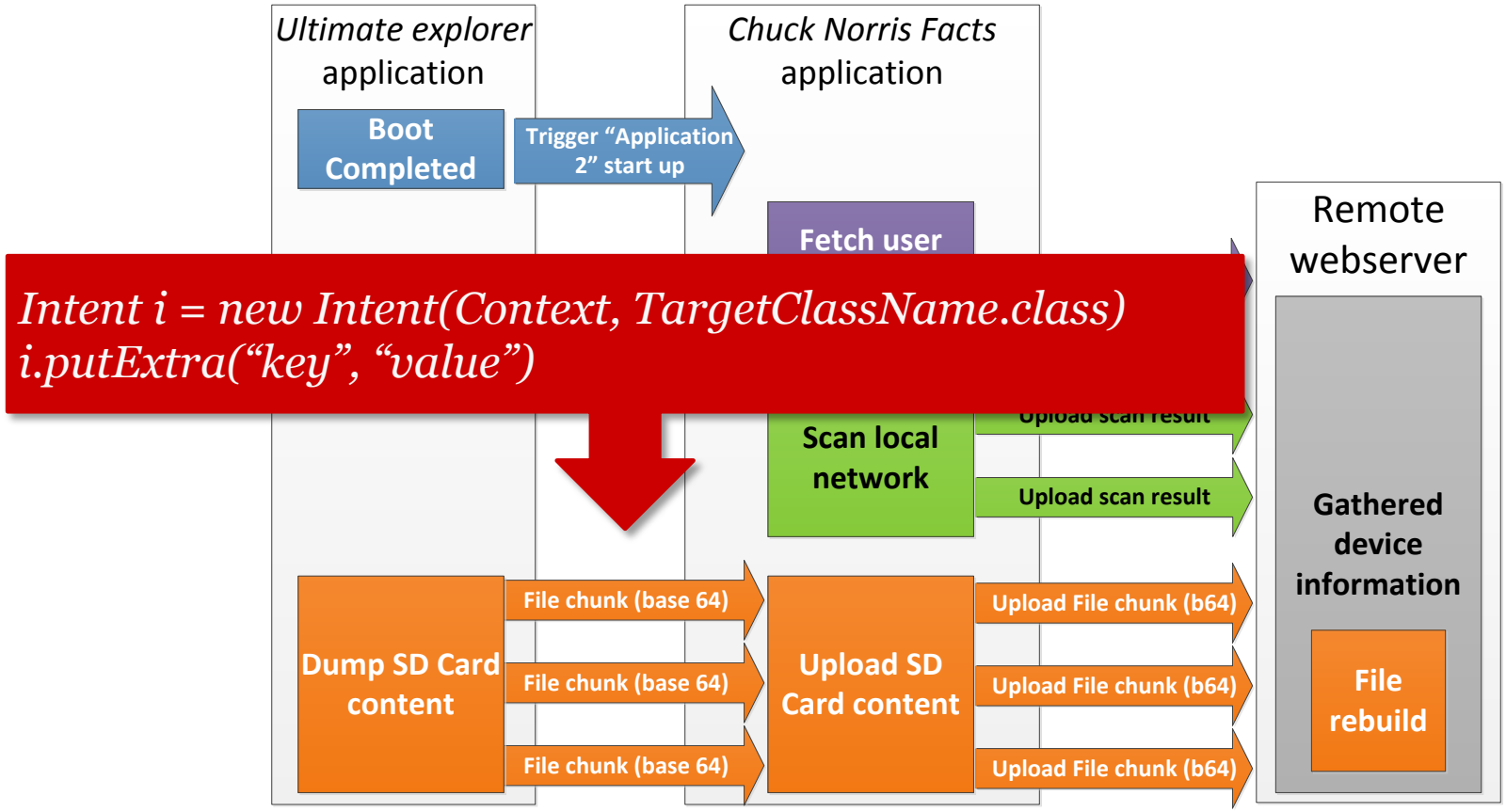
Application Phishing via a Distributed Malware – *Phishing under the hood (4/5)*

- Security Weaknesses introduced by Intents

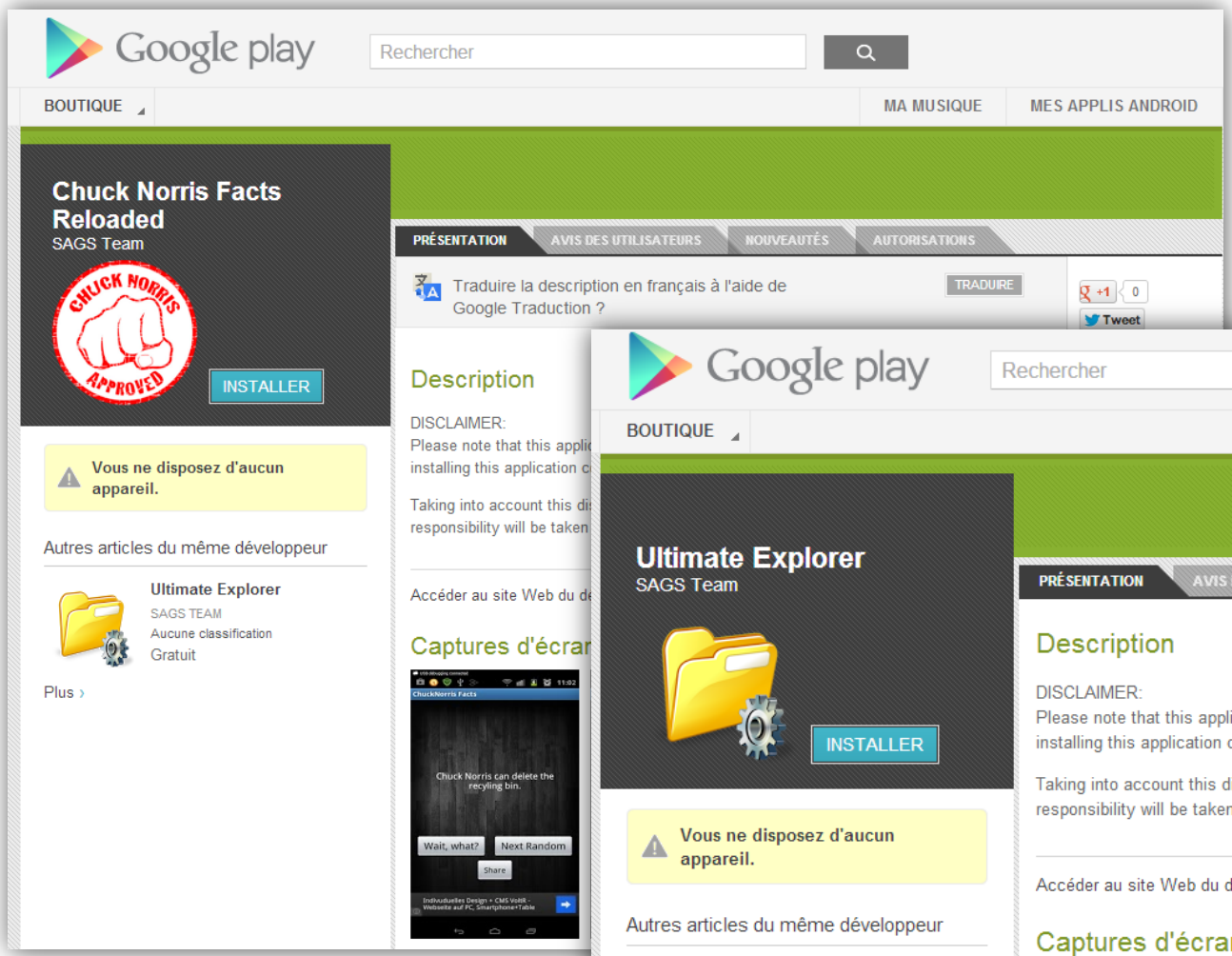
Android OEM's applications (in)security and backdoors without permission

- Presentation from *André Moulu* (quarkslab), SSTIC2013
- Around 10 Samsung OEM vulnerabilities related to misconfigured intents
- What about providing applications with (intentional) misconfigured intents ?

Application Phishing via a Distributed Malware – *Phishing under the hood (5/6)*

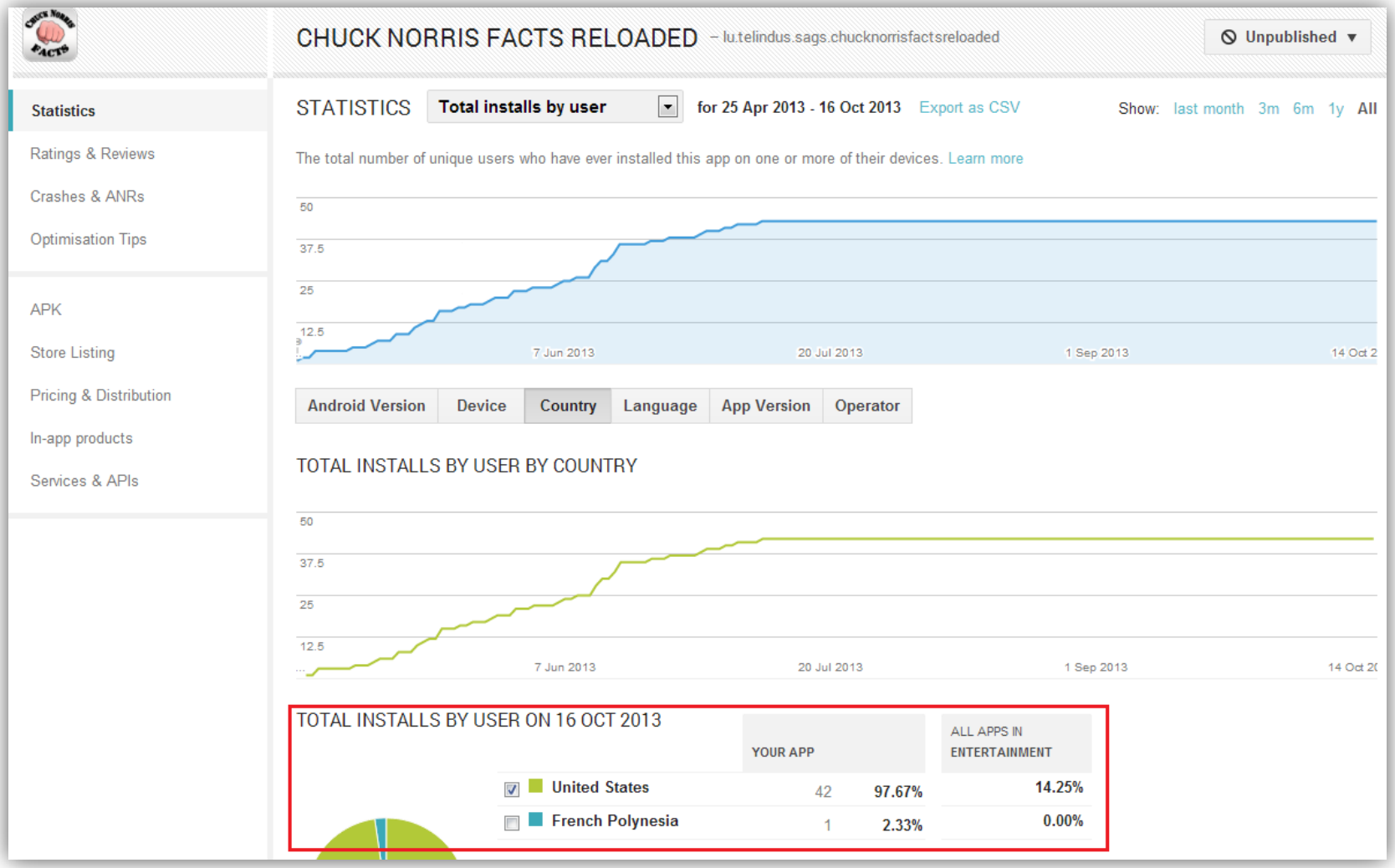


Application Phishing via a Distributed Malware – *Google Play Scenario (1/2)*

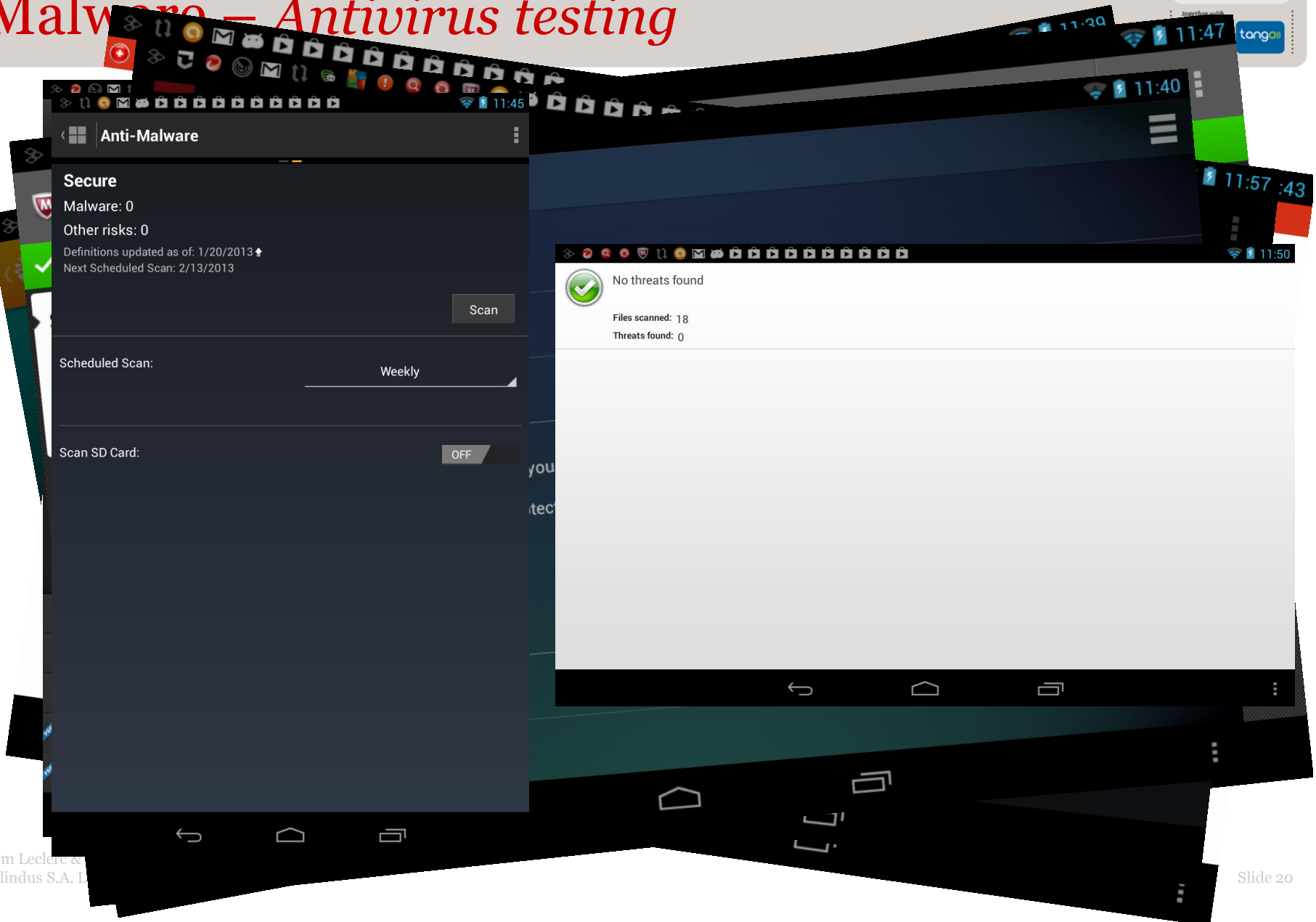


Tom Leclerc & Joany Boutet
Telindus S.A. Luxembourg

Application Phishing via a Distributed Malware – *Google Play Scenario (2/2)*



Application Phishing via a Distributed Malware – *Antivirus testing*



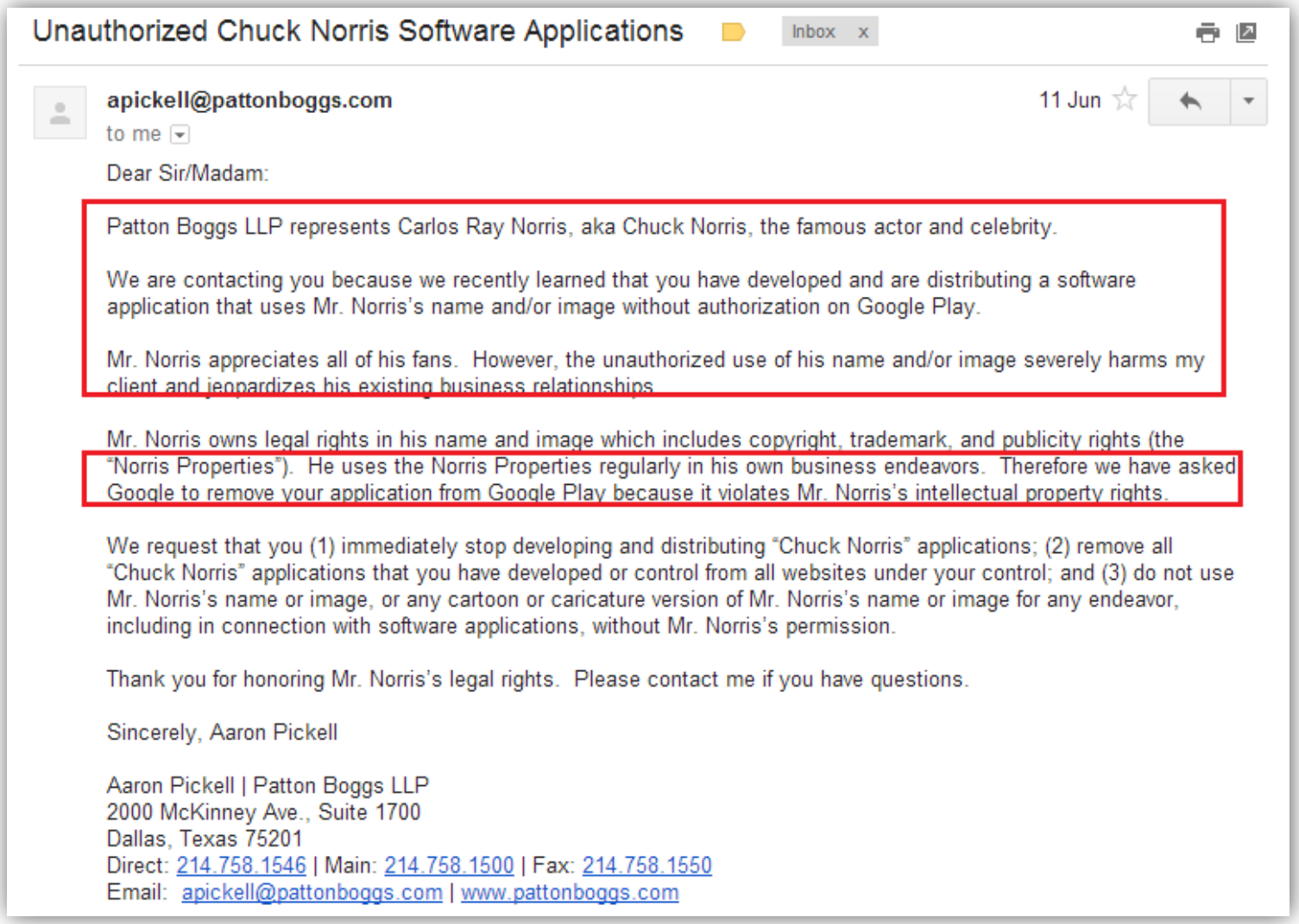
Application Phishing via a Distributed Malware – *Antivirus testing*

Tested with 10 antivirus programs:

- Avast! Mobile security
- Dr Web Light
- Ikarus mobile
- Lookout
- Mc Affee Security
- Zoner Antivirus
- AVG Antivirus
- Norton Mobile
- Eset Security
- Trend Micro Mobile Security

→0 detection!

Application Phishing via a Distributed Malware - *Here is the best antivirus ...*



Unauthorized Chuck Norris Software Applications

Inbox x

apickell@pattonboggs.com
to me

11 Jun

Dear Sir/Madam:

Patton Boggs LLP represents Carlos Ray Norris, aka Chuck Norris, the famous actor and celebrity.

We are contacting you because we recently learned that you have developed and are distributing a software application that uses Mr. Norris's name and/or image without authorization on Google Play.

Mr. Norris appreciates all of his fans. However, the unauthorized use of his name and/or image severely harms my client and jeopardizes his existing business relationships.

Mr. Norris owns legal rights in his name and image which includes copyright, trademark, and publicity rights (the "Norris Properties"). He uses the Norris Properties regularly in his own business endeavors. Therefore we have asked Google to remove your application from Google Play because it violates Mr. Norris's intellectual property rights.

We request that you (1) immediately stop developing and distributing "Chuck Norris" applications; (2) remove all "Chuck Norris" applications that you have developed or control from all websites under your control; and (3) do not use Mr. Norris's name or image, or any cartoon or caricature version of Mr. Norris's name or image for any endeavor, including in connection with software applications, without Mr. Norris's permission.

Thank you for honoring Mr. Norris's legal rights. Please contact me if you have questions.

Sincerely, Aaron Pickell

Aaron Pickell | Patton Boggs LLP
2000 McKinney Ave., Suite 1700
Dallas, Texas 75201
Direct: 214.758.1546 | Main: 214.758.1500 | Fax: 214.758.1550
Email: apickell@pattonboggs.com | www.pattonboggs.com

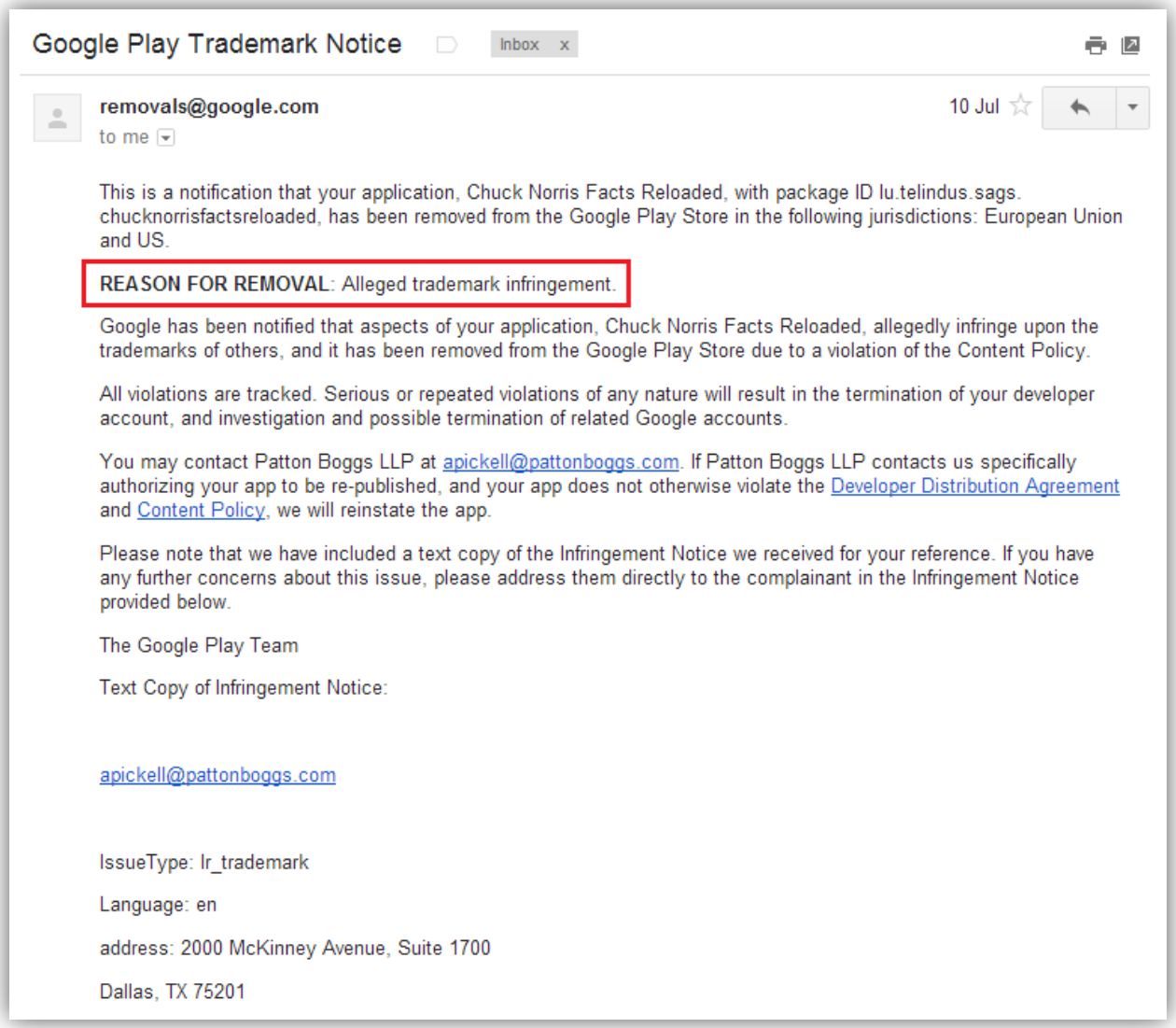
Application Phishing via a Distributed Malware - *Here is the best antivirus …*

Patton Boggs LLP ***represents Carlos Ray Norris, aka Chuck Norris****, the famous actor and celebrity.*

… we recently learned that you have developed and are distributing a software application that ***uses Mr. Norris's name and/or image without authorization on Google Play.***

Therefore we have asked Google to remove your application from Google Play because it violates Mr. Norris's intellectual property rights.

Application Phishing via a Distributed Malware - *Here is the best antivirus ...*



Google Play Trademark Notice

Inbox x

removals@google.com

10 Jul

to me

This is a notification that your application, Chuck Norris Facts Reloaded, with package ID lu.telindus.sags.chucknorrisfactsreloaded, has been removed from the Google Play Store in the following jurisdictions: European Union and US.

REASON FOR REMOVAL: Alleged trademark infringement.

Google has been notified that aspects of your application, Chuck Norris Facts Reloaded, allegedly infringe upon the trademarks of others, and it has been removed from the Google Play Store due to a violation of the Content Policy.

All violations are tracked. Serious or repeated violations of any nature will result in the termination of your developer account, and investigation and possible termination of related Google accounts.

You may contact Patton Boggs LLP at apickell@pattonboggs.com. If Patton Boggs LLP contacts us specifically authorizing your app to be re-published, and your app does not otherwise violate the Developer Distribution Agreement and Content Policy, we will reinstate the app.

Please note that we have included a text copy of the Infringement Notice we received for your reference. If you have any further concerns about this issue, please address them directly to the complainant in the Infringement Notice provided below.

The Google Play Team

Text Copy of Infringement Notice:

apickell@pattonboggs.com

IssueType: Ir_trademark

Language: en

address: 2000 McKinney Avenue, Suite 1700

Dallas, TX 75201

22-24 October 2013 - Luxembourg
9th edition of the infosec conference

"We're not computers, Sebastian, we're physical"
Roy Batty in Blade Runner

Distributed Malware via Repackaged Applications

Distributed Malware via Repackaged Applications – *Google Play Scenario (1/4)*

- Goal
  - Split permissions and malware components across several applications
  - Trick the user into installing all the required components
- Technical methods
  - Distribute malware content across repackaged applications
  - Communicate between applications using intents
- Social methods
  - Choose appealing applications
  - Advertise repackaged applications

Distributed Malware via Repackaged Applications – *Google Play Scenario (2/4)*

android.permission.ACCESS_WIFI_STATE

android.permission.INTERNET

App Type 2

Main Activity

Service

Generate 128-bit UUID
(androidId + macAddr)

Get Installed Apps

Bidirectional Communication

App Type 1

Main Activity

Service

Data Exfiltration

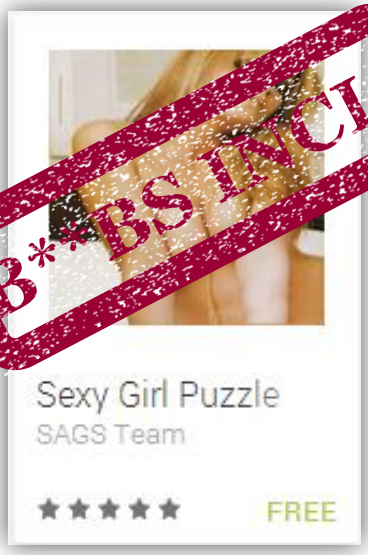
```
Intent i = new Intent()
i.setComponent(new ComponentName("app1_pkg", "app1_pkg.HackLuApp1Service"))
i.setAction("hack.lu.ex.data")
i.putExtra("device_unique_id", device_unique_id)
 i.putExtra("apps_installed", apps_installed)
```

Distributed Malware via Repackaged Applications – *Google Play Scenario (3/4)*

- Use the same technique using repackaged applications

Type 1: 4 Applications

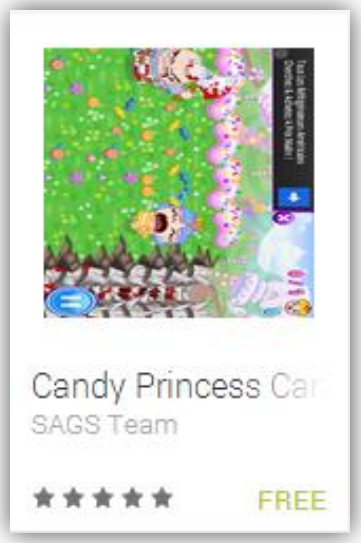
android.permission.INTERNET



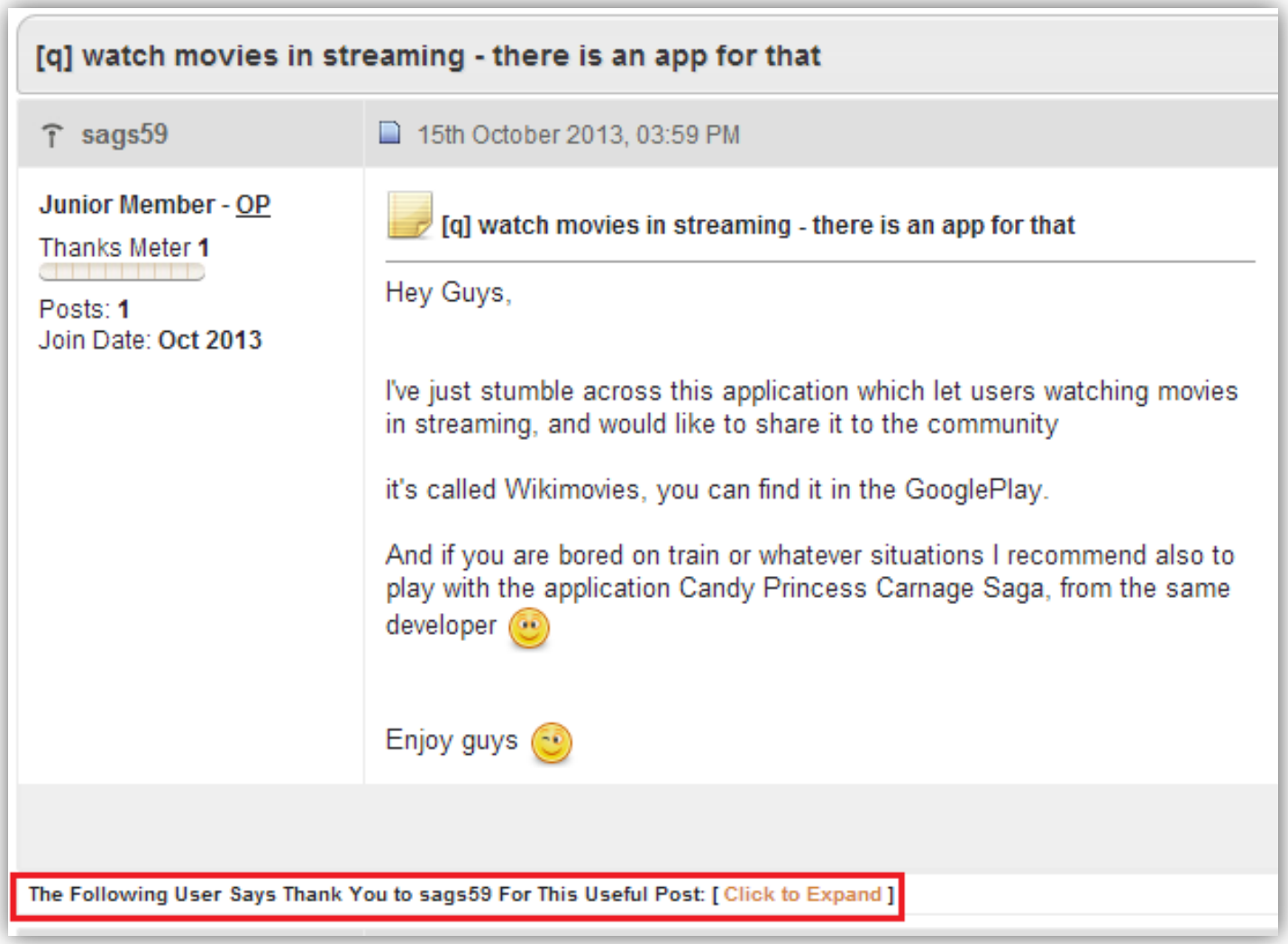
Distributed Malware via Repackaged Applications – *Google Play Scenario (4/4)*

Type 2: 4 Applications

android.permission.ACCESS_WIFI_STATE



Distributed Malware via Repackaged Applications – *Here is what can happen …*



[q] watch movies in streaming - there is an app for that

sags59 — 15th October 2013, 03:59 PM

Junior Member - OP
Thanks Meter 1
Posts: 1
Join Date: Oct 2013

[q] watch movies in streaming - there is an app for that

Hey Guys,

I've just stumble across this application which let users watching movies in streaming, and would like to share it to the community

it's called Wikimovies, you can find it in the GooglePlay.

And if you are bored on train or whatever situations I recommend also to play with the application Candy Princess Carnage Saga, from the same developer

Enjoy guys

The Following User Says Thank You to sags59 For This Useful Post: [Click to Expand]

Distributed Malware via Repackaged Applications – *Here is what can happen …*

fluck58
Junior Member

Posts: 2
Joined: Oct 2013
Reputation: 0

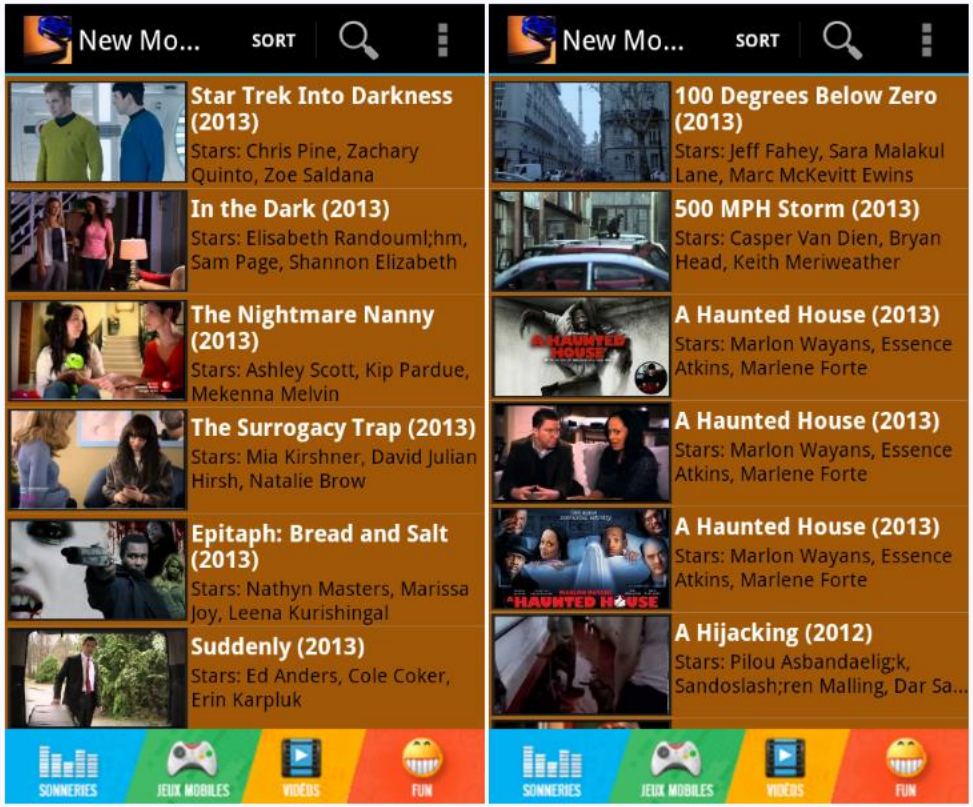
WATCH MOVIES IN STREAMING - THERE IS AN APP FOR THAT

Hey Guys,

WikiMovies App

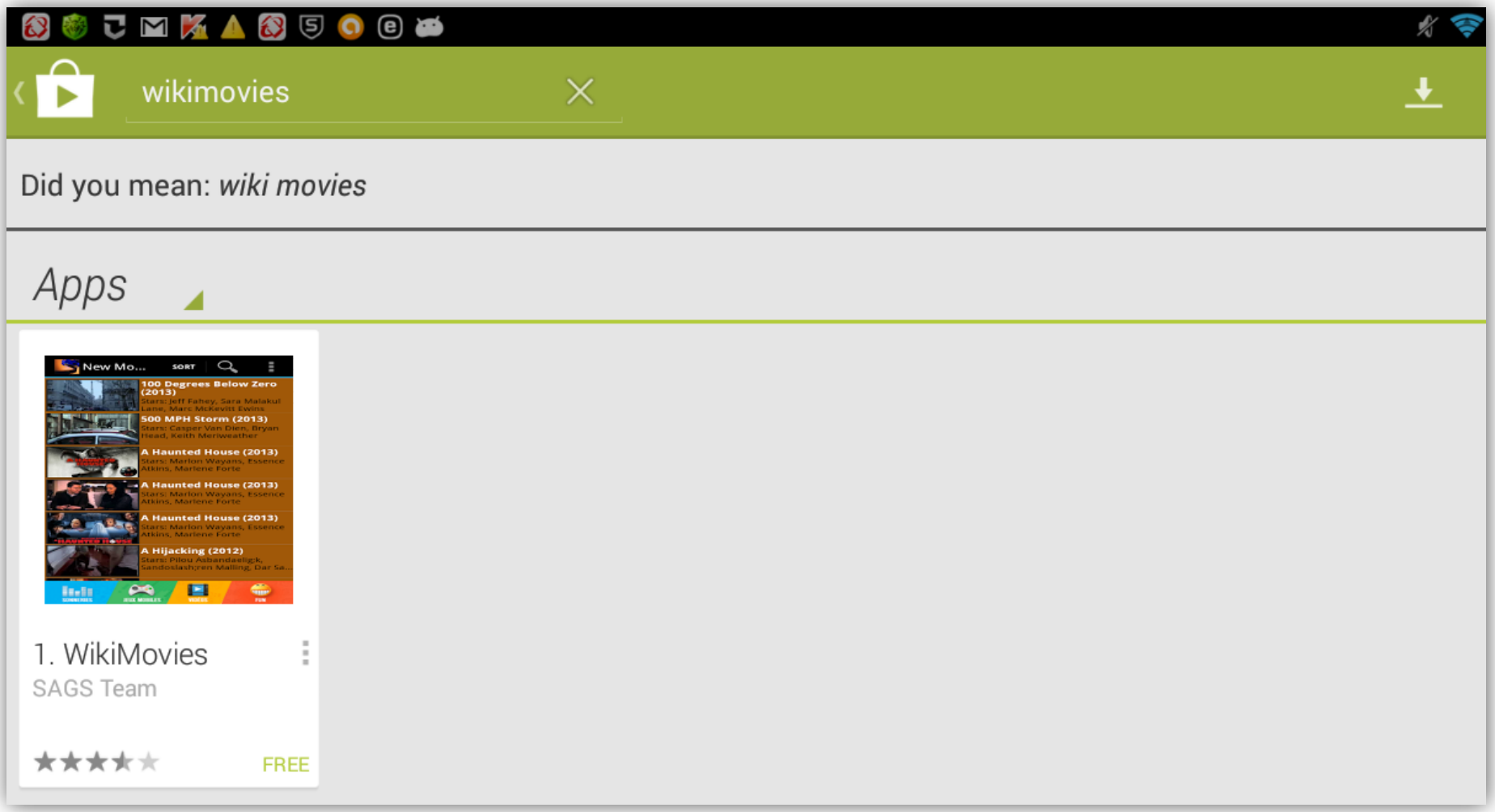
I've just stumble across this application which let users watching movies in streaming, and would like to share it to the community

Apps Screenshots

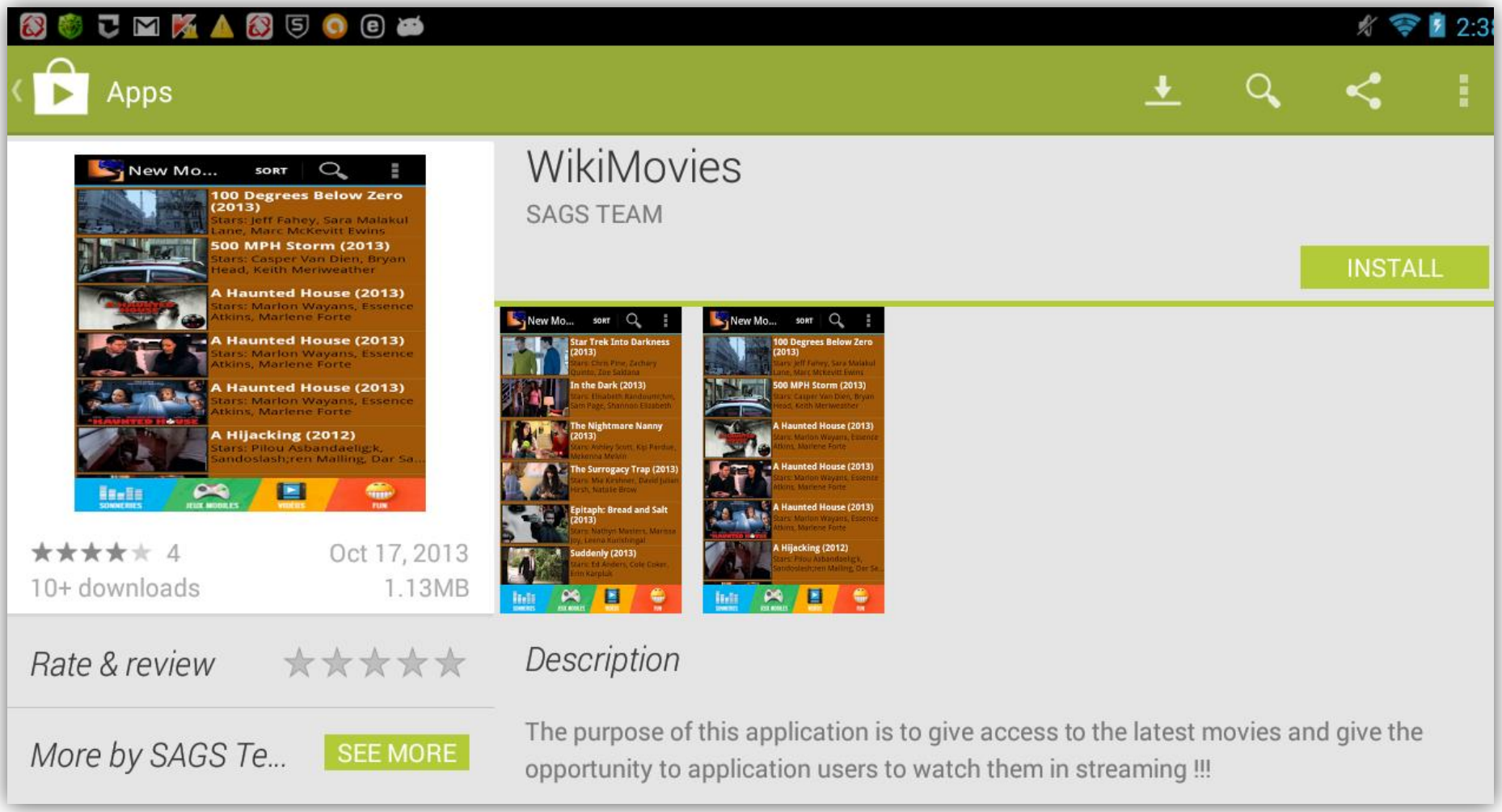


it's called Wikimovies, you can download it by one click on the following download button.

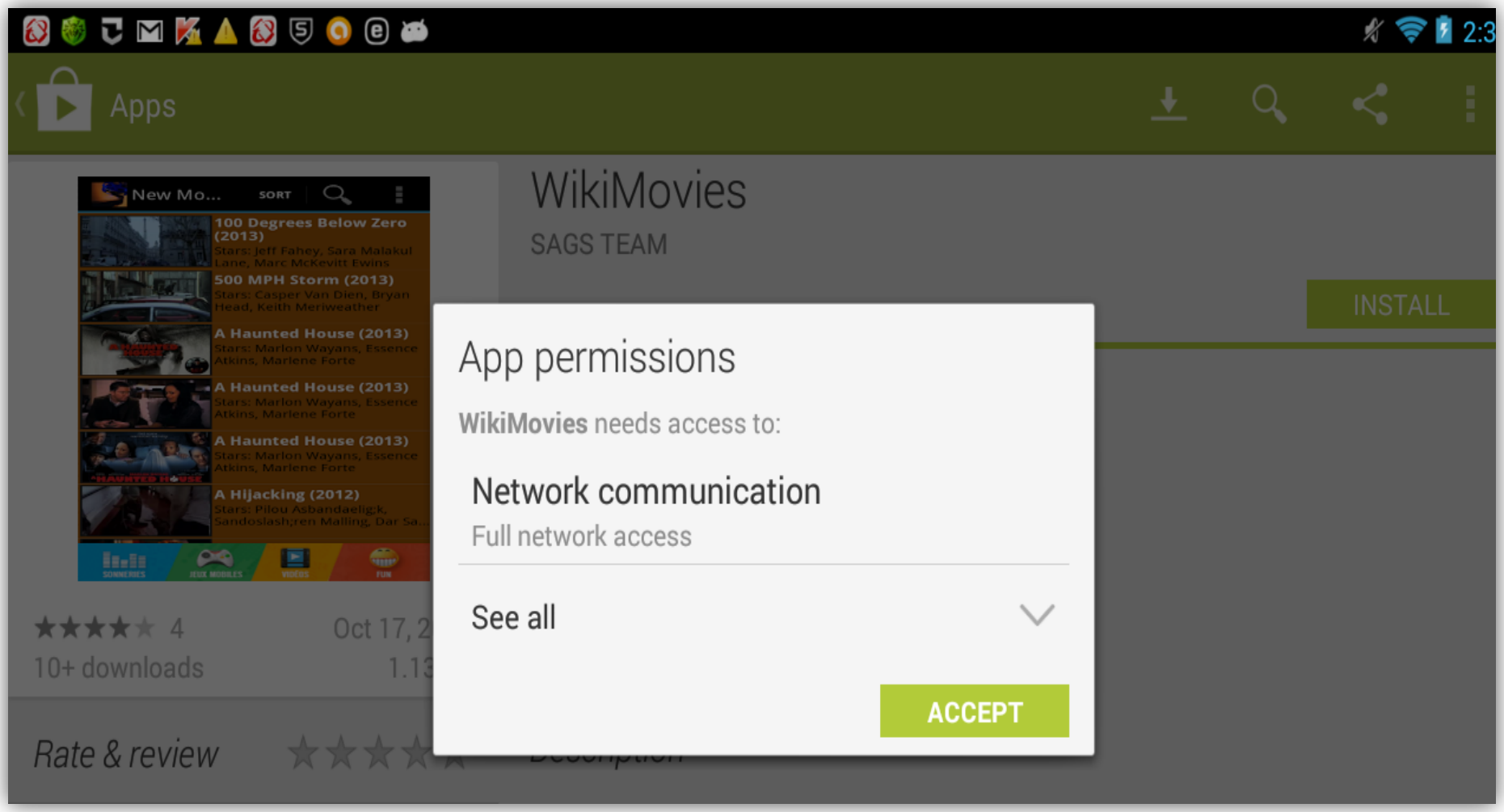
Distributed Malware via Repackaged Applications – *Here is what can happen …*



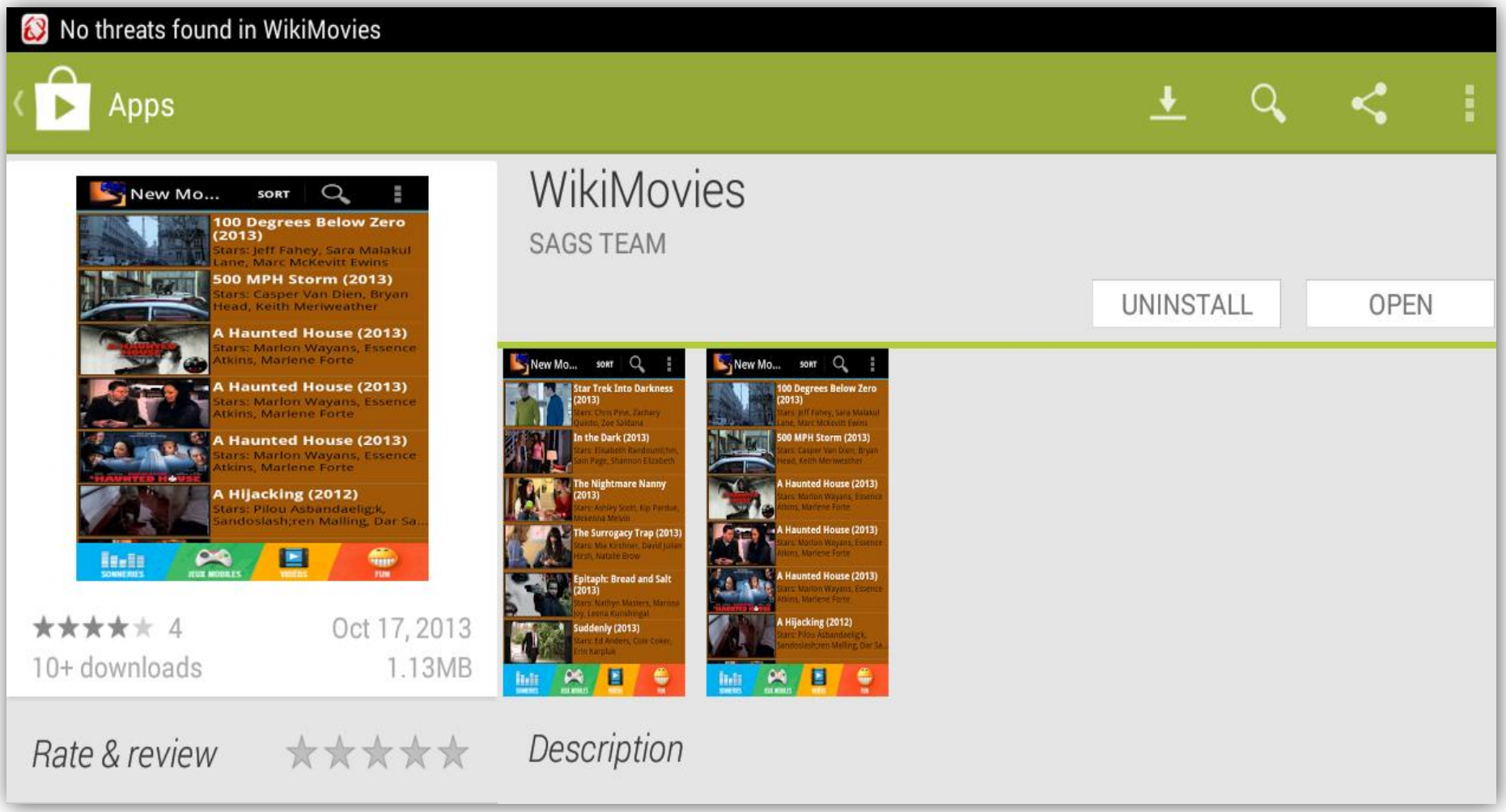
Distributed Malware via Repackaged Applications – *Here is what can happen …*



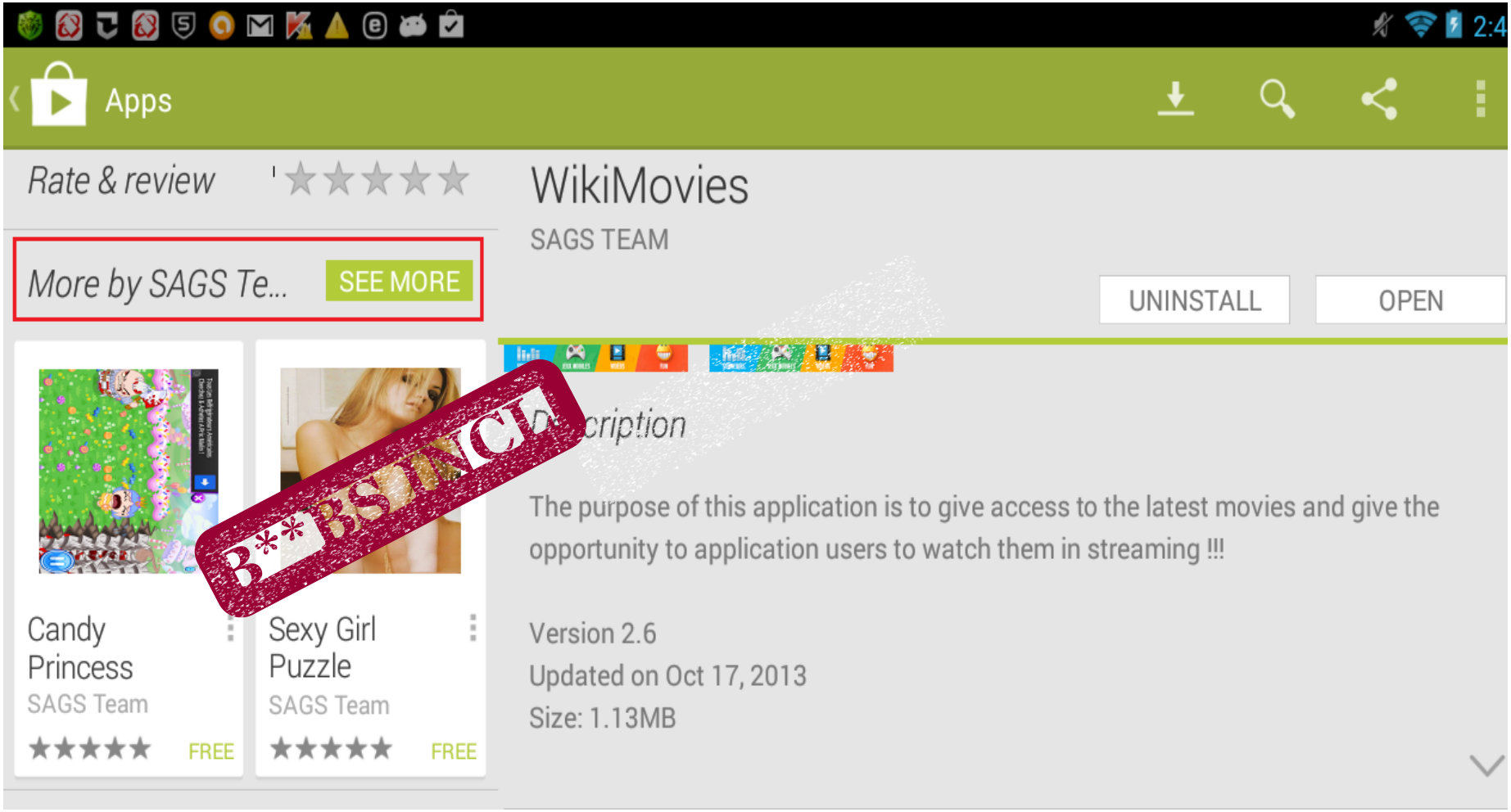
Distributed Malware via Repackaged Applications – *Here is what can happen …*



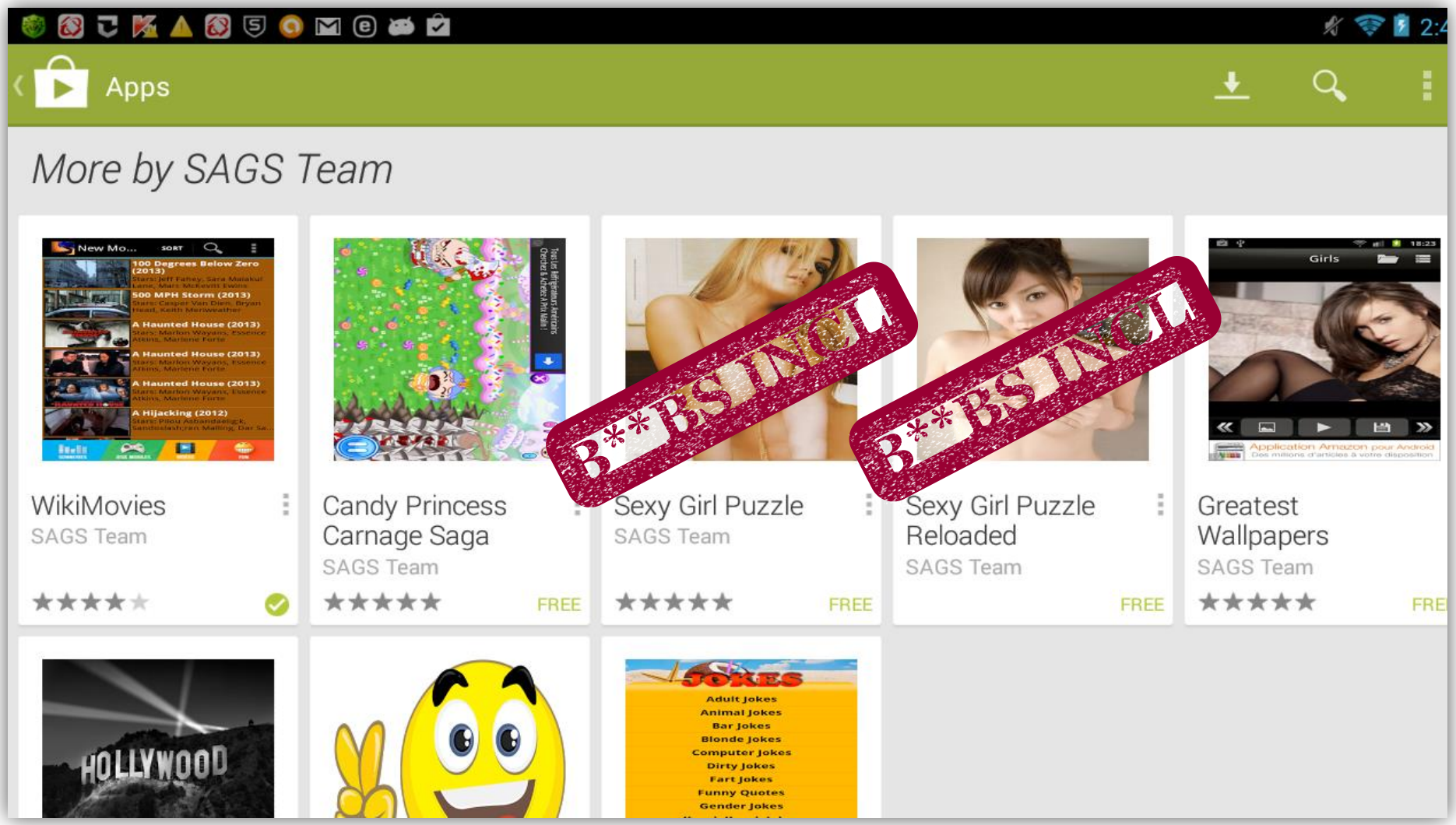
Distributed Malware via Repackaged Applications – *Here is what can happen …*



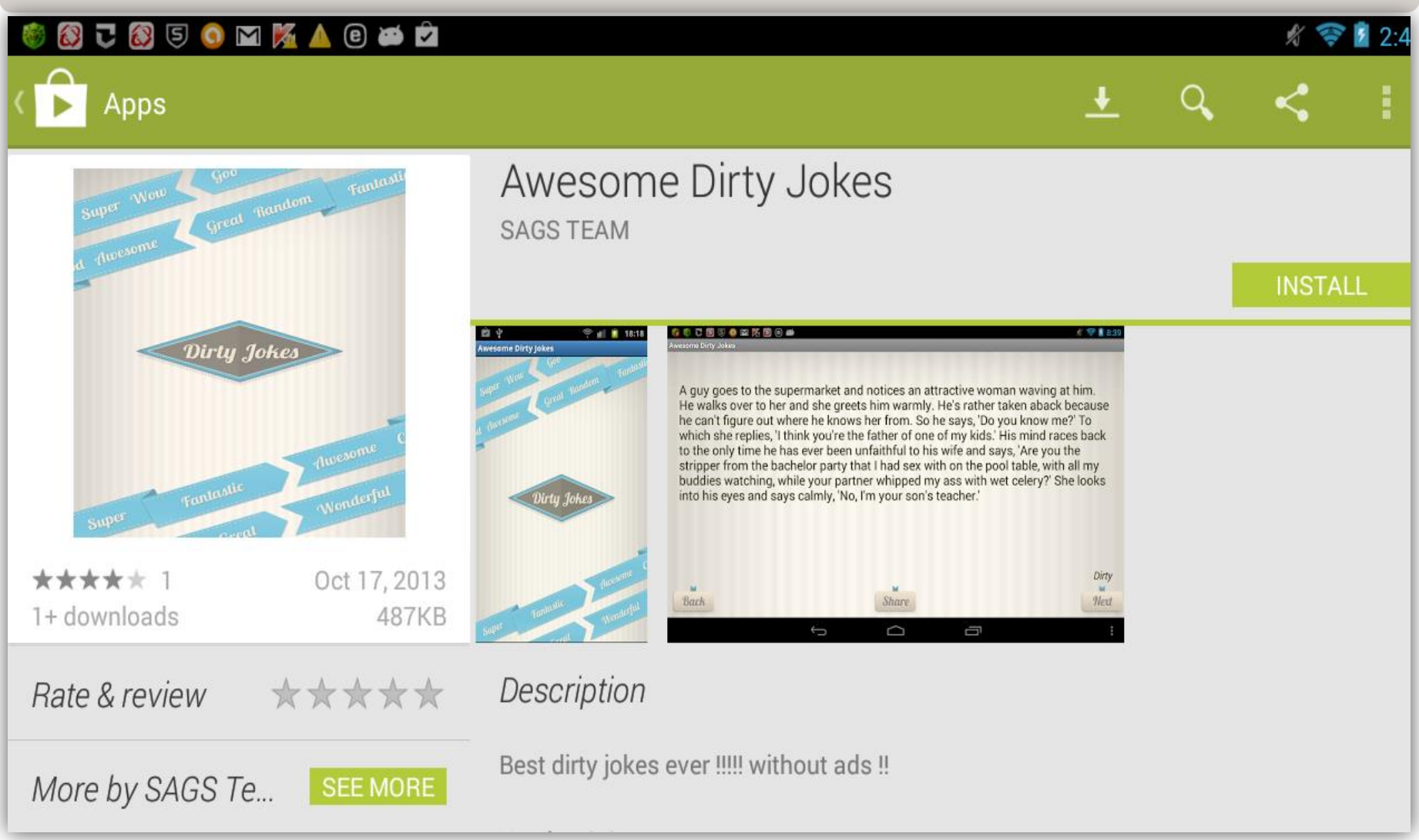
Distributed Malware via Repackaged Applications – *Here is what can happen …*



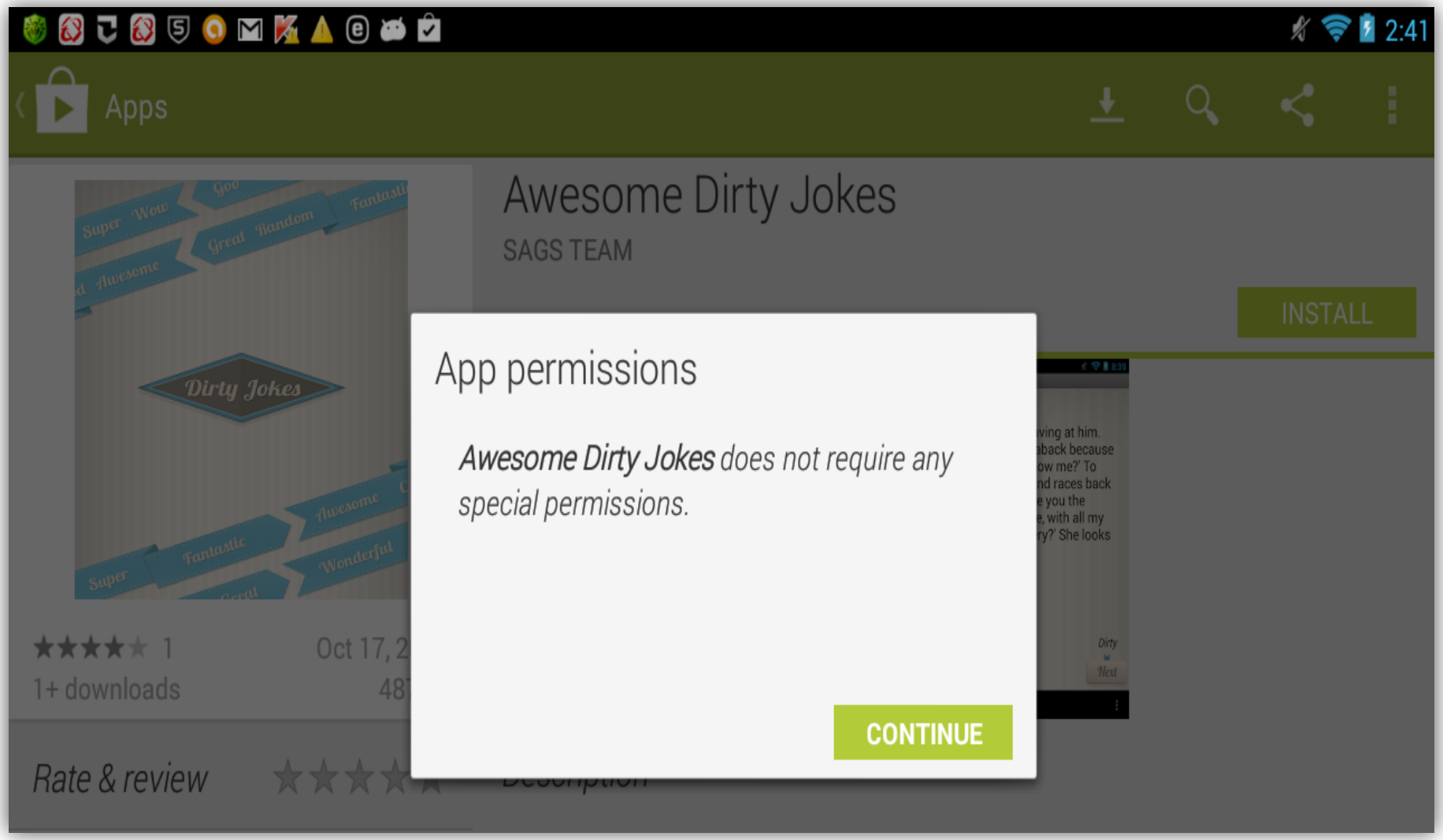
Distributed Malware via Repackaged Applications – *Here is what can happen …*



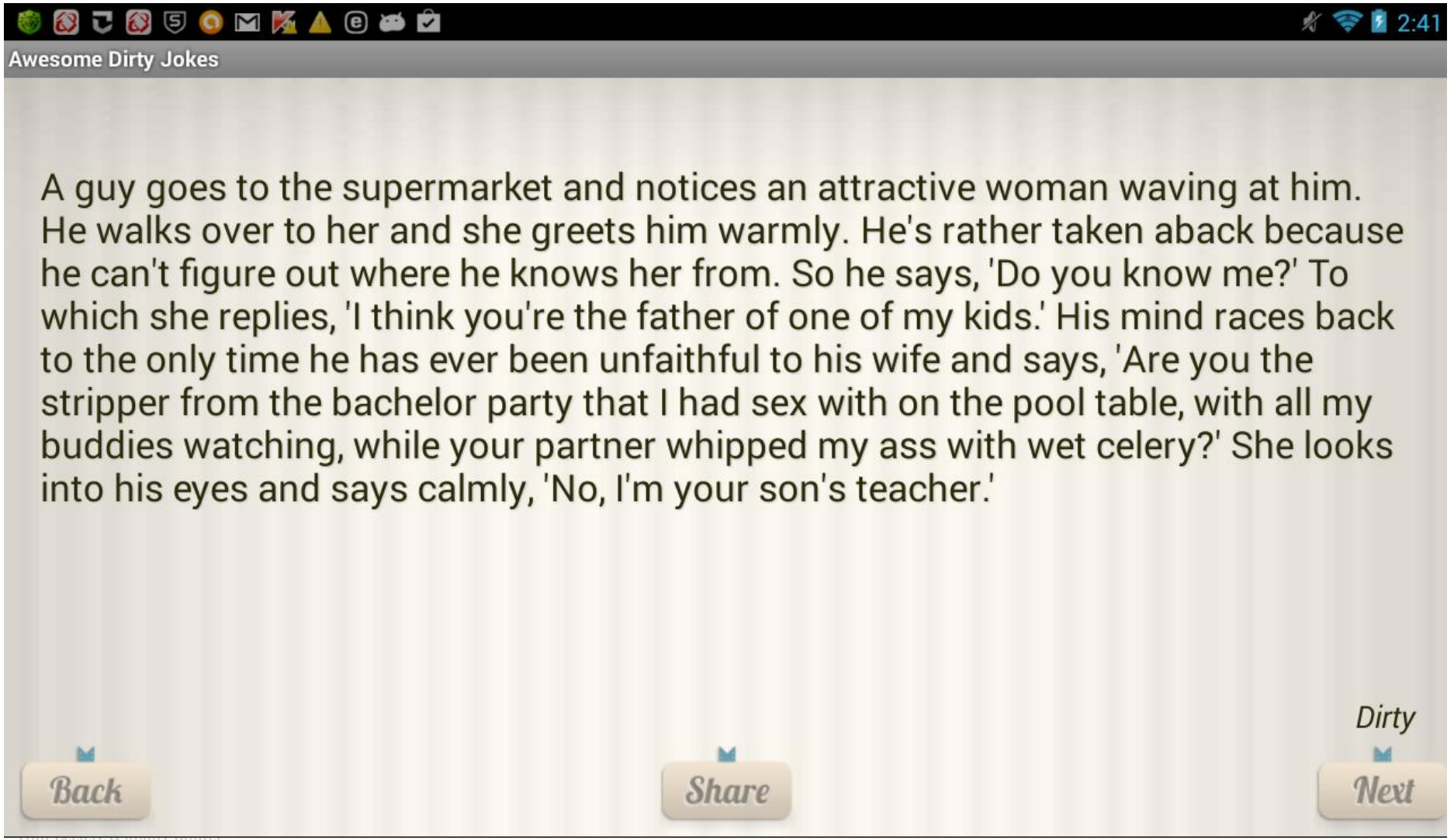
Distributed Malware via Repackaged Applications – *Here is what can happen …*



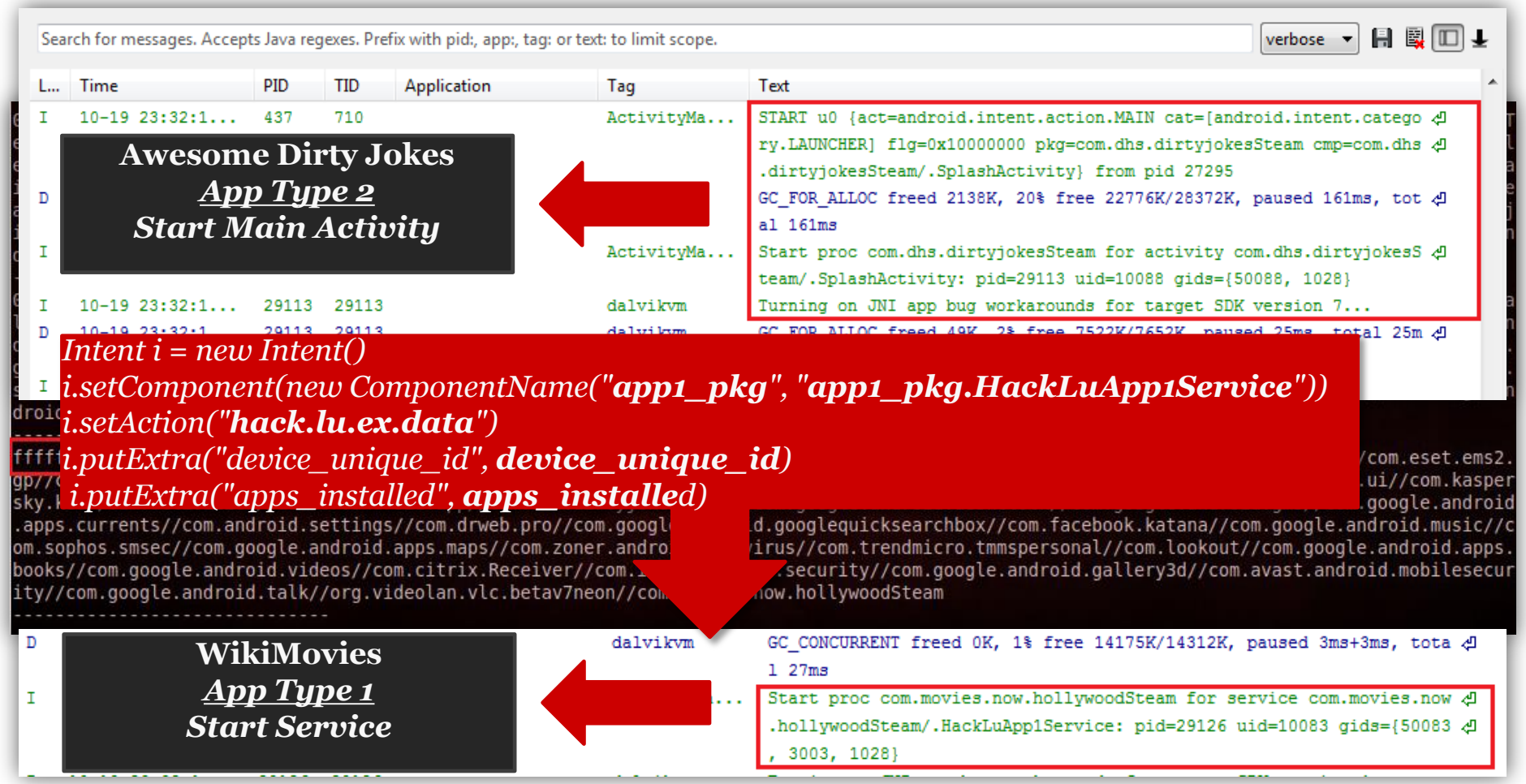
Distributed Malware via Repackaged Applications – *Here is what can happen …*



Distributed Malware via Repackaged Applications – *Here is what can happen …*



Distributed Malware via Repackaged Applications – *Here is what can happen …*



Distributed Malware via Repackaged Applications – *Technical Deep Dive (1/4)*

- Develop the malware using Eclipse

1. Type 1 service– used for data exfiltration

```
<uses-permission android:name="android.permission.INTERNET" />
```

2. Type 2 service – used for data fetching
   - Mobile device MAC address and installed apps

```
<uses-permission android:name="android.permission.ACCESS_WIFI_STATE" />
```

Distributed Malware via Repackaged Applications – *Technical Deep Dive (2/4)*

- Build the project and retrieve the APK file
- Reverse engineer this file to extract Dalvik bytecode

HackLuApp1Service.smali

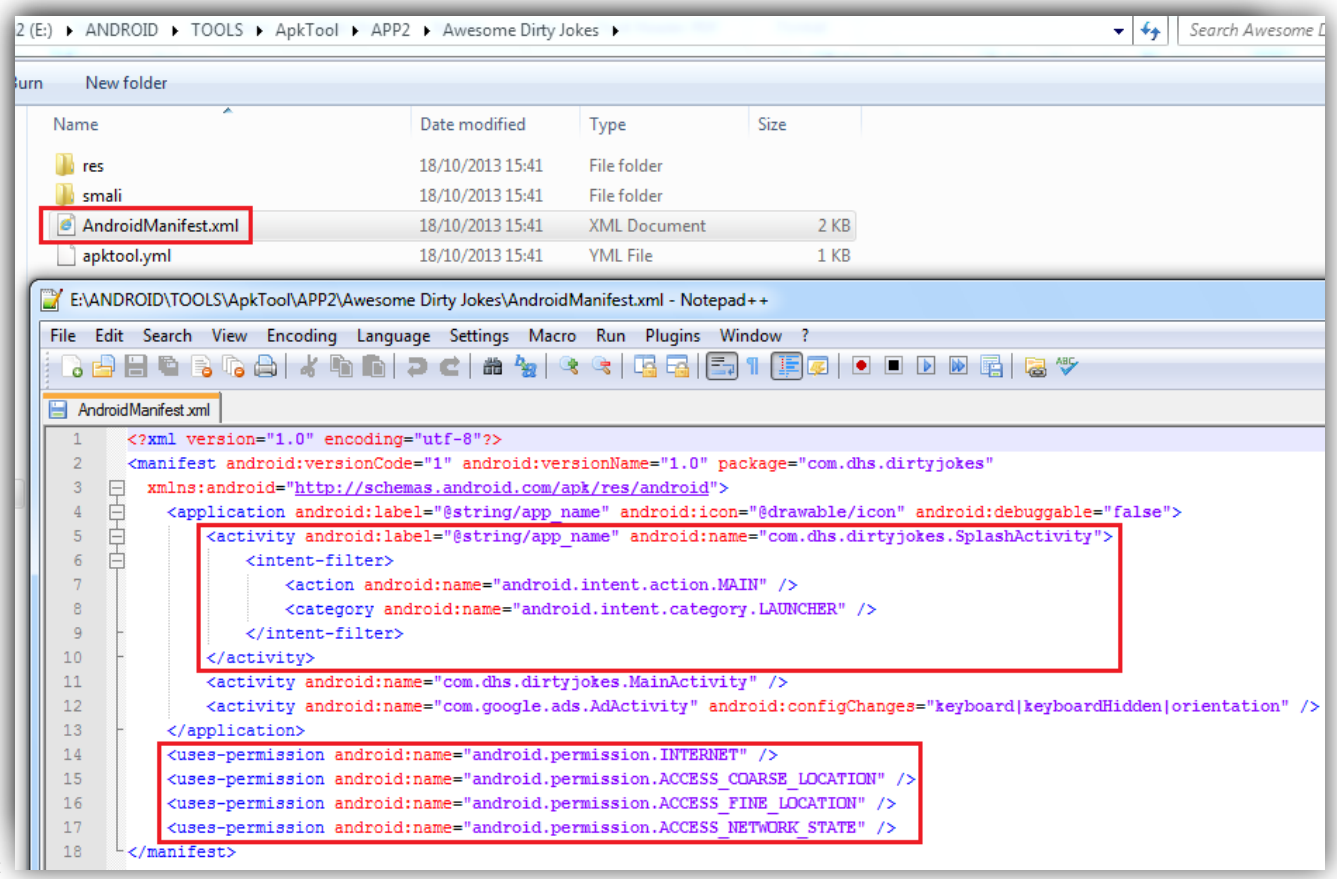
```
.class public Llu/telindus/hack/lu_app1/HackLuApp1Service;
.super Landroid/app/IntentService;
.source "HackLuApp1Service.java"


# instance fields
.field IP_Web_Server:Ljava/lang/String;

.field action:Ljava/lang/String;

.field app_fetch_data_package:Ljava/lang/String;

.field app_fetch_data_service:Ljava/lang/String;

.field app_name:Ljava/lang/String;

.field apps_installed:Ljava/lang/String;

.field device_unique_id:Ljava/lang/String;


# direct methods
.method public constructor <init>()V
    .locals 2

    .prologue
    const/4 v1, 0x0

    .line 42
    const-class v0, Llu/telindus/hack/lu_app1/HackLuApp1Service;

    invoke-virtual {v0}, Ljava/lang/Class;->getName()Ljava/lang/String;

    move-result-object v0

    invoke-direct {p0, v0}, Landroid/app/IntentService;-><init>(Ljava/lang/String;)V

    .line 25
    iput-object v1, p0, Llu/telindus/hack/lu_app1/HackLuApp1Service;->app_fetch_data_package:Ljava/lang/String;
```

HackLuApp2Service.smali

```
.class public Llu/telindus/hack/lu_app2/HackLuApp2Service;
.super Landroid/app/Service;
.source "HackLuApp2Service.java"


# instance fields
.field app_exfiltrate_data_package:Ljava/lang/String;

.field app_exfiltrate_data_service:Ljava/lang/String;

.field app_name:Ljava/lang/String;

.field apps_installed:Ljava/lang/String;

.field device_unique_id:Ljava/lang/String;


# direct methods
.method public constructor <init>()V
    .locals 1

    .prologue
    const/4 v0, 0x0

    .line 20
    invoke-direct {p0}, Landroid/app/Service;-><init>()V

    .line 22
    iput-object v0, p0, Llu/telindus/hack/lu_app2/HackLuApp2Service;->apps_installed:Ljava/lang/String;

    .line 24
    iput-object v0, p0, Llu/telindus/hack/lu_app2/HackLuApp2Service;->device_unique_id:Ljava/lang/String;
```

Distributed Malware via Repackaged Applications – *Technical Deep Dive (3/4)*

- Retrieve applications APK file on Google Play
- Reverse engineer those files to extract Dalvik bytecode



Distributed Malware via Repackaged Applications – *Technical Deep Dive (4/4)*

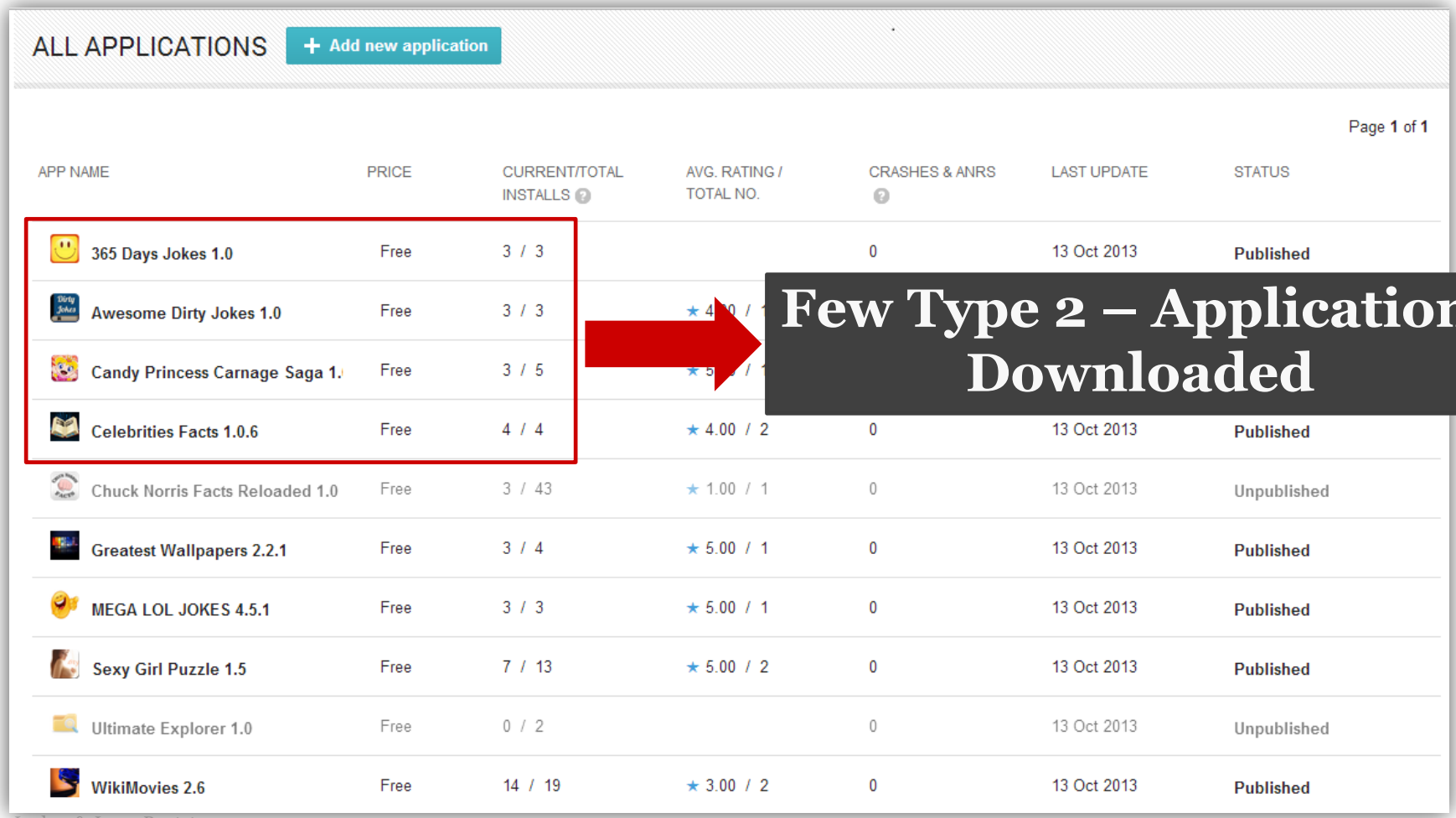
- Inject Services' Dalvik bytecode in reverse engineered apps
- Modify the AndroidManifest and Services files accordingly

SplashActivity.smali

```
15      .line 9
16      invoke-direct {p0}, Landroid/app/Activity;-><init>()V
17
18      return-void
19  .end method
20
21
22  # virtual methods
23  .method public onCreate(Landroid/os/Bundle;)V
24      .locals 4
25      .parameter "savedInstanceState"
26
27      .prologue
28      .line 16
29      invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V
30
31      .line 17
32      const v0, 0x7f030001
33
34      invoke-virtual {p0, v0}, Lcom/dhs/dirtyjokesSteam/SplashActivity;->setContentView(I)V
35
36
37      new-instance v0, Landroid/content/Intent;
38
39      const-class v1, Lcom/dhs/dirtyjokesSteam/HackLuApp2Service;
40
41      invoke-direct {v0, p0, v1}, Landroid/content/Intent;-><init>(Landroid/content/Context;Ljava/lang/Class;)V
42
43      invoke-virtual {p0, v0}, Lcom/dhs/dirtyjokesSteam/SplashActivity;->startService(Landroid/content/Intent;)Landroid/content/ComponentName;
44
45
```

Distributed Malware via Repackaged Applications – *Google Play Scenario Results*

- At first glance, not enough downloads … on October, 16th



ALL APPLICATIONS + Add new application

Page 1 of 1

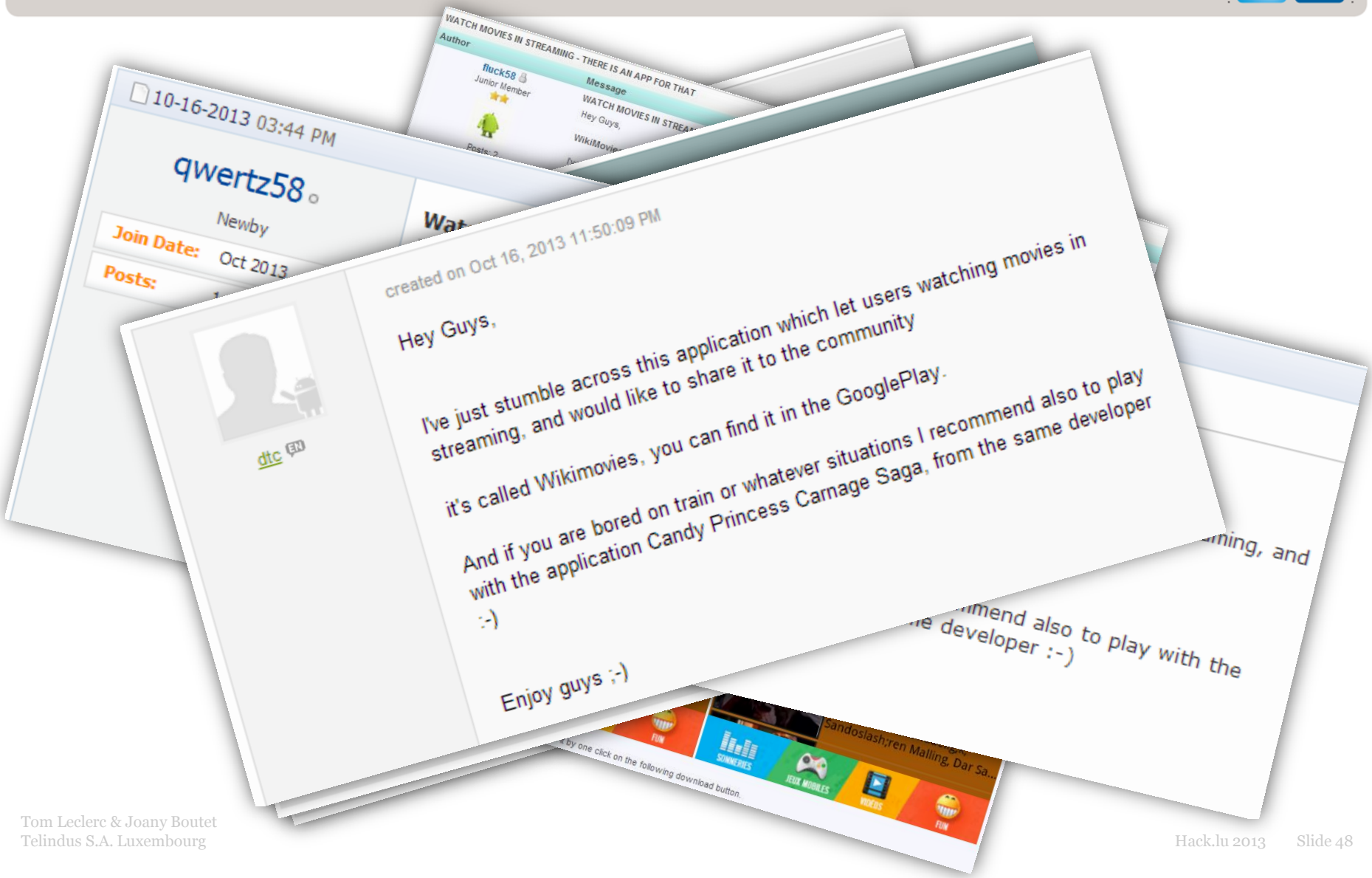
APP NAME	PRICE	CURRENT/TOTAL INSTALLS	AVG. RATING / TOTAL NO.	CRASHES & ANRS	LAST UPDATE	STATUS
365 Days Jokes 1.0	Free	3 / 3		0	13 Oct 2013	Published
Awesome Dirty Jokes 1.0	Free	3 / 3	★ 4.00 / 1			
Candy Princess Carnage Saga 1.	Free	3 / 5	★ 5.00 / 1			
Celebrities Facts 1.0.6	Free	4 / 4	★ 4.00 / 2	0	13 Oct 2013	Published
Chuck Norris Facts Reloaded 1.0	Free	3 / 43	★ 1.00 / 1	0	13 Oct 2013	Unpublished
Greatest Wallpapers 2.2.1	Free	3 / 4	★ 5.00 / 1	0	13 Oct 2013	Published
MEGA LOL JOKES 4.5.1	Free	3 / 3	★ 5.00 / 1	0	13 Oct 2013	Published
Sexy Girl Puzzle 1.5	Free	7 / 13	★ 5.00 / 2	0	13 Oct 2013	Published
Ultimate Explorer 1.0	Free	0 / 2		0	13 Oct 2013	Unpublished
WikiMovies 2.6	Free	14 / 19	★ 3.00 / 2	0	13 Oct 2013	Published

Few Type 2 – Applications Downloaded

Distributed Malware via Repackaged Applications – *Google Play Scenario Results*

- What about adding an additional Type 2 – Application and advertising our applications ? ☺

Distributed Malware via Repackaged Applications – *Google Play Scenario Results*



Distributed Malware via Repackaged Applications – *Google Play Scenario Results*

APP NAME	PRICE	CURRENT/TOTAL INSTALLS	AVG. RATING / TOTAL NO.	CRASHES & ANRS	LAST UPDATE	STATUS
365 Days Jokes 1.0	Free	1 / 3		0	13 Oct 2013	Published
Awesome Dirty Jokes 2.0	Free	3 / 5	★ 4.00 / 1	0	17 Oct 2013	Published
Candy Princess Carnage Saga 1.	Free	4 / 12	★ 5.00 / 1	0	13 Oct 2013	Published
Celebrities Facts 1.0.6	Free	2 / 5	★ 4.00 / 2	0	13 Oct 2013	Published
Chuck Norris Facts Reloaded 1.0	Free	3 / 43	★ 1.00 / 1	0	18 Oct 2013	Unpublished
Greatest Wallpapers 2.2.1	Free	7 / 18	★ 5.00 / 1	0	13 Oct 2013	Published
MEGA LOL JOKES 4.5.1	Free	1 / 3	★ 5.00 / 1	0	13 Oct 2013	Published
Sexy Girl Puzzle 1.5	Free	42 / 138	5.00			
Sexy Girl Puzzle Reloaded 1.0	Free	7 / 32		1	16 Oct 2013	Published
Ultimate Explorer 1.0	Free	0 / 2		0	21 Oct 2013	Unpublished
WikiMovies 2.6	Free	36 / 48	★ 3.75 / 4	0	17 Oct 2013	Published

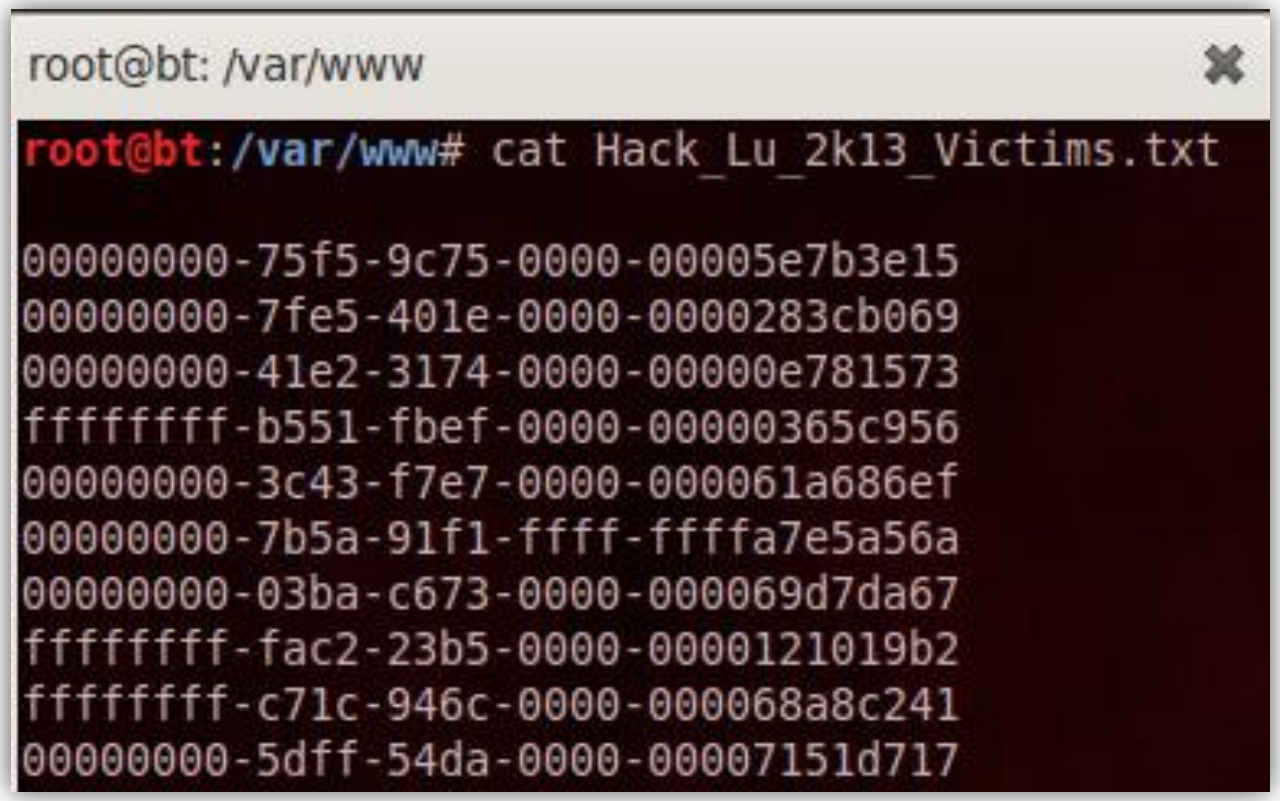
2 Type 1 Applications
android.permission.INTERNET

Distributed Malware via Repackaged Applications – *Google Play Scenario Results*

APP NAME	PRICE	CURRENT/TOTAL INSTALLS	AVG. RATING / TOTAL NO.	CRASHES & ANRS	LAST UPDATE	STATUS
365 Days Jokes 1.0	Free	1 / 3		0	13 Oct 2013	Published
Awesome Dirty Jokes 2.0	Free	3 / 5	★ 4.00 / 1	0	17 Oct 2013	Published
Candy Princess Carnage Saga 1.	Free	4 / 12	★ 5.00 / 1	0	13 Oct 2013	Published
Celebrities Facts 1.0.6	Free	2 / 5	★ 4.00 / 2	0	13 Oct 2013	Published
Chuck Norris Facts Reloaded 1.0	Free	3 / 43	★ 1.00 / 1	0	18 Oct 2013	Unpublished
Greatest Wallpapers 2.2.1	Free	7 / 18	★ 5.00 / 1	0	13 Oct 2013	Published
MEGA LOL JOKES 4.5.1	Free	1 / 3	★ 5.00 / 1	0	13 Oct 2013	Published
Sexy Girl Puzzle 1.5	Free	42 / 138	★ 5.00 / 2	0	13 Oct 2013	Published
Sexy Girl Puzzle Reloaded 1.0	Free	7 / 32				
Ultimate Explorer 1.0	Free	0 / 2		0	21 Oct 2013	Unpublished
WikiMovies 2.6	Free	36 / 48	★ 3.75 / 4	0	17 Oct 2013	Published

1 Type 2 Application
android.permission.ACCESS_WIFI_STATE

Distributed Malware via Repackaged Applications – *Google Play Scenario Results*



Recommendations

- (unintentional) misconfigured intents
  - Use of PendingIntent
  - Add permission on sensitive components
    - protectionLevel of “Signature”

- (intentional) misconfigured intents
  - Inform about intents that an application can send
    - Broadcast intents or specific intents
    - Modify permissions display accordingly

Conclusion

- Way to bypass Android permissions model
  →Hide permissions among several applications

- …Chuck Norris is one of the best Mobile AV ☺

Current/Future works

Ongoing whitepaper:

- Split well known malwares and test against antivirus programs
- Use broadcast intents as stealthier method

Future enhancements:

- Create a tool to automate the process of payload injection and split between several applications.

- Use techniques to hide malware code
  - Upcoming hack.lu talks of *Jurriaan Bremer* and *Axelle Apvrille*

→ Room for the spread of distributed Android malware

Questions ?

"As penetration testers, we need to figure out what our installed applications offer to perform on behalf of other apps, in an effort to better understand the security risk of the application and the overall device itself"

Chris Crowley, "Intentional Evil: A Pen Tester's Overview of Android Intents“
SANS Penetration Testing Blog, May 2013