Android Apps permissions model (in)security

Davide Danelon

Security Consultant @ Minded Security

OWASP EU Tour 2013 - Rome
Rome, 27th June 2013

The OWASP Foundation
http://www.owasp.org

About me

- Davide Danelon
  - MSc. in Computer Engineering
  - GWAPT, Comptia Security+, CCNA
  - Application Security Consultant @ Minded Security
  - OWASP Testing Guide Contributor

Agenda

- Android Growth
- Android Permissions Model
- Android Intents
- Permissions Re-delegation
- Permissions Avoidance
- Proof of Concept
- Conclusions

Android Growth

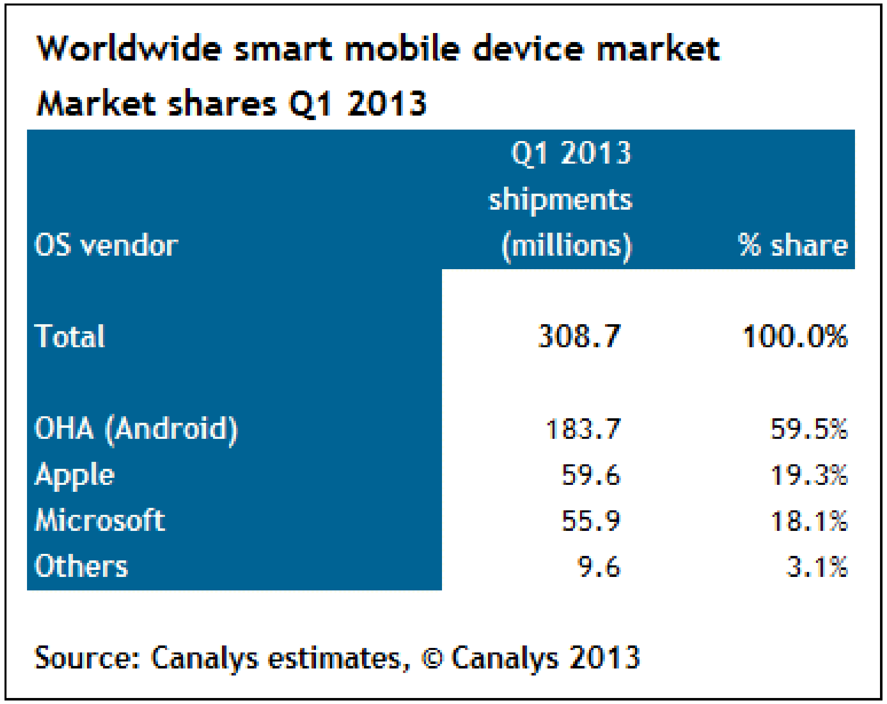
- Why Android security becomes more and more important?

Android Growth

- Android devices are growing exponentially
- Android Apps are growing too
- And accordingly… personal data treated!

Android Growth – Some Stats

- Every day more than 1 million new Android devices are activated worldwide
- 1.5 billion Apps downloads a month and growing.



Worldwide smart mobile device market

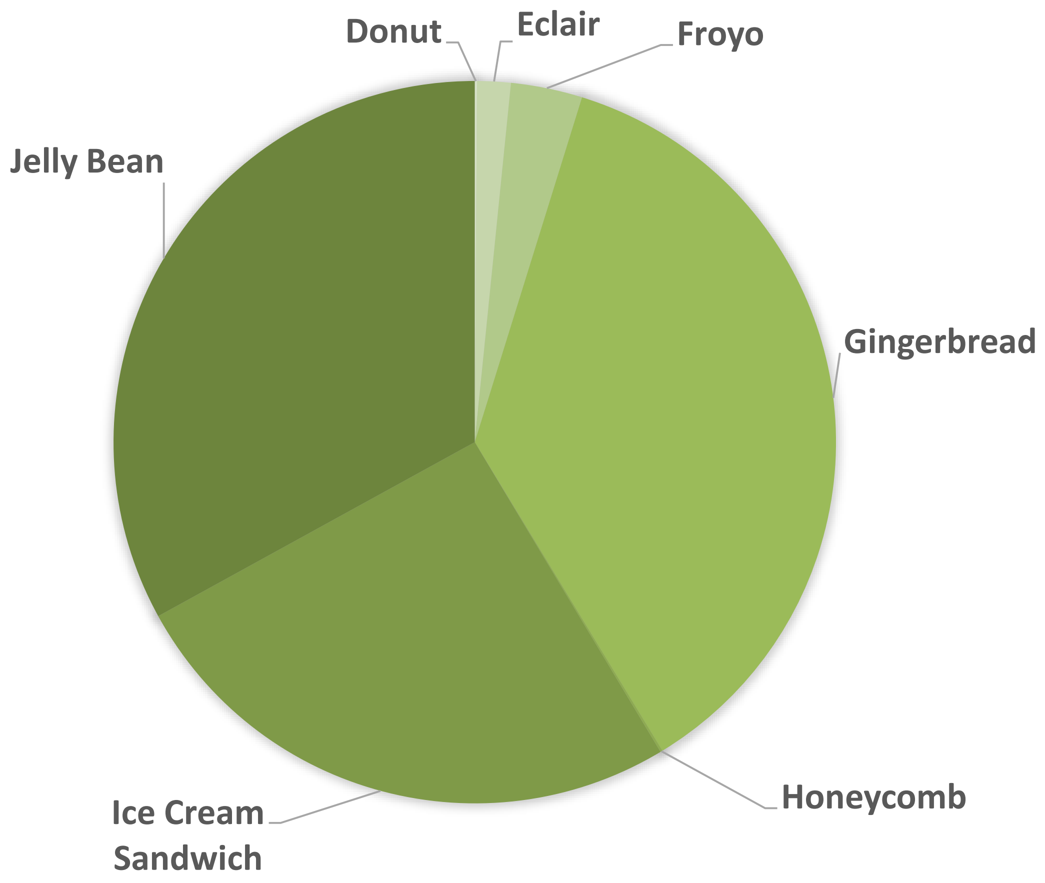
Market shares Q1 2013

OS vendor	Q1 2013 shipments (millions)	% share
Total	308.7	100.0%
OHA (Android)	183.7	59.5%
Apple	59.6	19.3%
Microsoft	55.9	18.1%
Others	9.6	3.1%

Source: Canalys estimates, © Canalys 2013

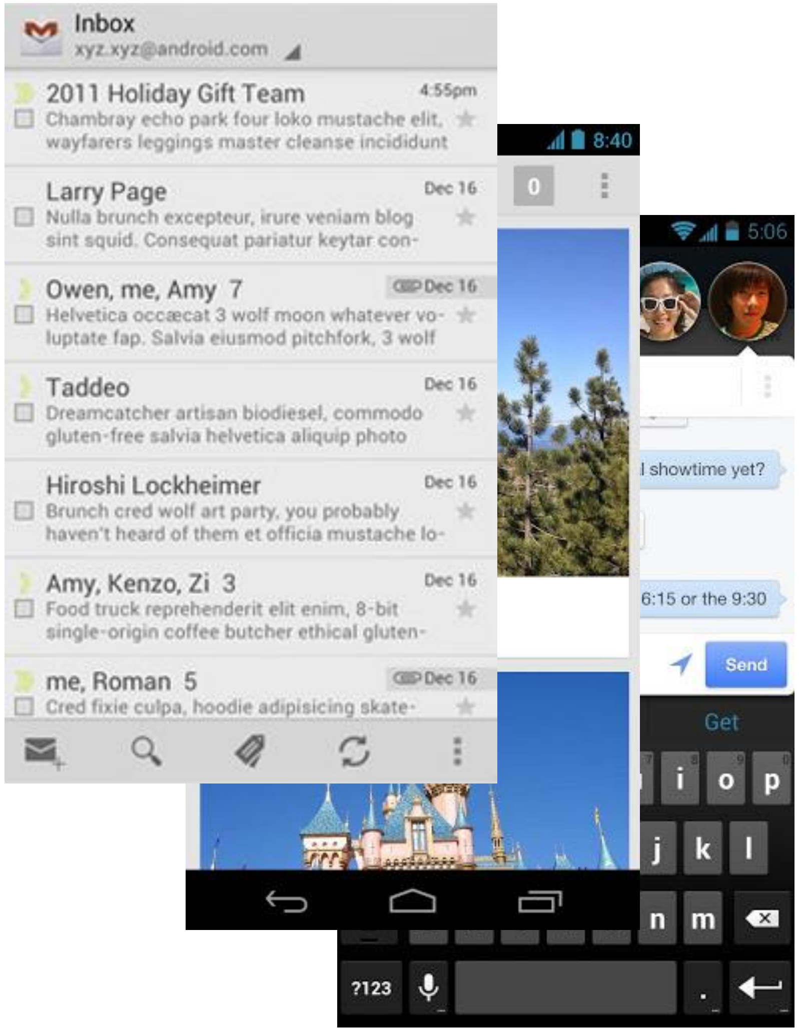
Android Growth – Some Stats

Version	Codename	API	Distribution
1.6	Donut	4	0.10%
2.1	Eclair	7	1.50%
2.2	Froyo	8	3.20%
2.3 - 2.3.2	Gingerbread	9	0.10%
2.3.3 - 2.3.7		10	36.40%
3.2	Honeycomb	13	0.10%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	25.60%
4.1.x	Jelly Bean	16	29.00%
4.2.x		17	4.00%



Android Permissions Model

- Mobile devices are full of data...
- How Android protects access to sensitive data and device capabilities?


Inbox
xyz.xyz@android.com

2011 Holiday Gift Team 4:55pm
Chambray echo park four loko mustache elit, wayfarers leggings master cleanse incididunt

Larry Page Dec 16
Nulla brunch excepteur, irure veniam blog sint squid. Consequat pariatur keytar con-

Owen, me, Amy 7 Dec 16
Helvetica occæcat 3 wolf moon whatever vo- luptate fap. Salvia eiusmod pitchfork, 3 wolf

Taddeo Dec 16
Dreamcatcher artisan biodiesel, commodo gluten-free salvia helvetica aliquip photo

Hiroshi Lockheimer Dec 16
Brunch cred wolf art party, you probably haven't heard of them et officia mustache lo-

Amy, Kenzo, Zi 3 Dec 16
Food truck reprehenderit elit enim, 8-bit single-origin coffee butcher ethical gluten-

me, Roman 5 Dec 16
Cred fixie culpa, hoodie adipisicing skate-

Android Permissions Model

- Applications are isolated from each other
  - Every app (data and resources included) runs under a different system identity (Linux user ID and group ID)
  - Only apps that are signed with the same digital certificates can weaken this isolation but requires explicit configuration

- Additional finer-grained security features are provided through a "permission" mechanism that enforces restrictions on the specific operations that a particular process can perform

Android Permissions Model

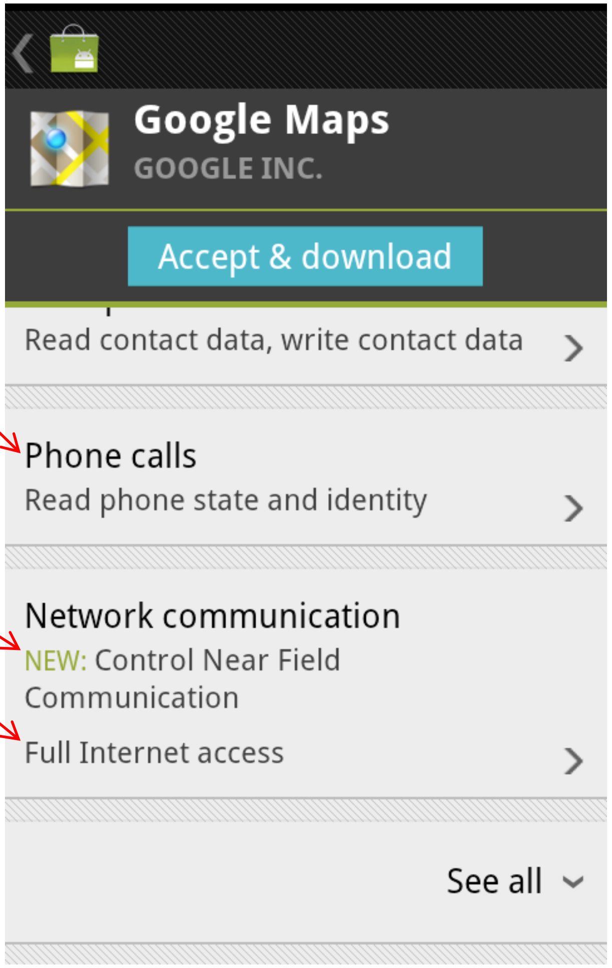
- Since Android sandboxes applications from each other, applications must explicitly share resources and data.
- They do this by declaring the permissions they need for additional capabilities not provided by the basic sandbox.
- Applications statically declare the permissions they require, and the Android system prompts the user for consent at the time the application is installed.

Android Permissions Model

- Some example of permissions provided by Android system
  - "android.permission.INTERNET"
  - "android.permission.READ_EXTERNAL_STORAGE"
  - "android.permission.SEND_SMS"
  - "android.permission.BLUETOOTH"
- But it is also possible to define custom permissions

Android Permissions Model

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



Google Maps
GOOGLE INC.
Accept & download

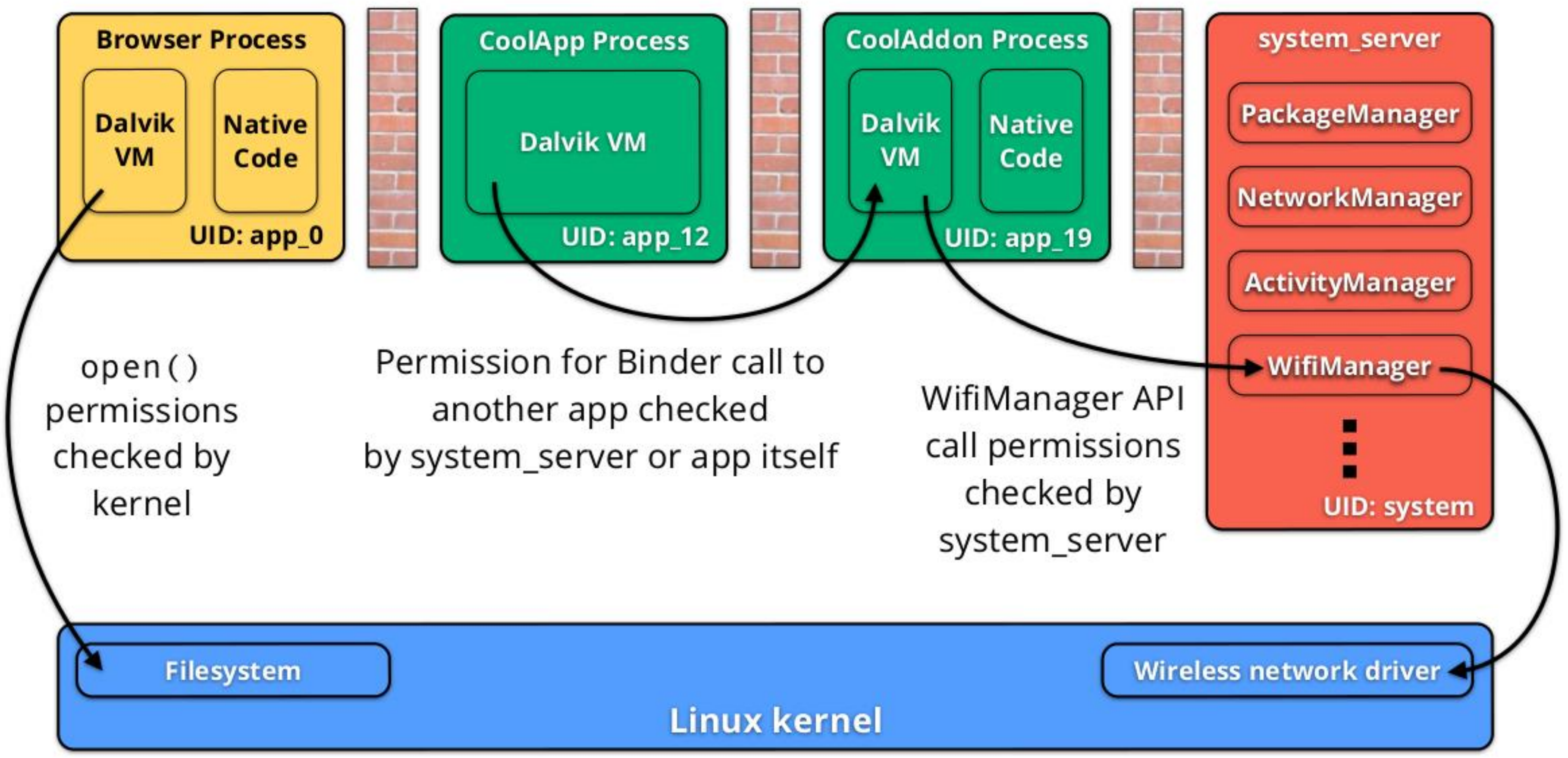
Read contact data, write contact data

Phone calls
Read phone state and identity

Network communication
NEW: Control Near Field Communication
Full Internet access

See all

Android Permissions Model



Android Intents

- Android supports some forms of Interprocess Communication (Intent, Binder, Messenger or traditional Linux techniques such as network sockets and shared files)
- Intents are the preferred mechanism for asynchronous IPC in Android (we focus on this method for this talk).

Android Intents

- Intent is a message used by components in the same or different applications to communicate with each other.
- An Intent is a bundle of information. It contains information of interest to the component that receives the intent (such as the action to be taken and the data to act on) plus information of interest to the Android system (such as the category of component that should handle the intent and instructions on how to launch a target activity).

Android Intents

- Intens can be:
  - *Explicit*: the message will be delivered only to a specific receiver.
  - *Implicit*: the message is created with a specific action and all the components that have registered to receive that action (using intent filters) will get the message.

Android Intents

- Why intents could became “dangerous”?
- Intents are used by almost all Android apps
  - All apps can send intents
  - Even malicious ones!
- Intents carry data
  - Received data can be malicious or fake!
  - Your app could leak confidential data!

Android Intents

- Developers have to handle malicious intents
- Senders of an intent can verify that the recipient has a permission specifying a non-Null permission with the method call.
- Moreover senders could use explicit intents to send the message to a single component (avoiding broadcasting).

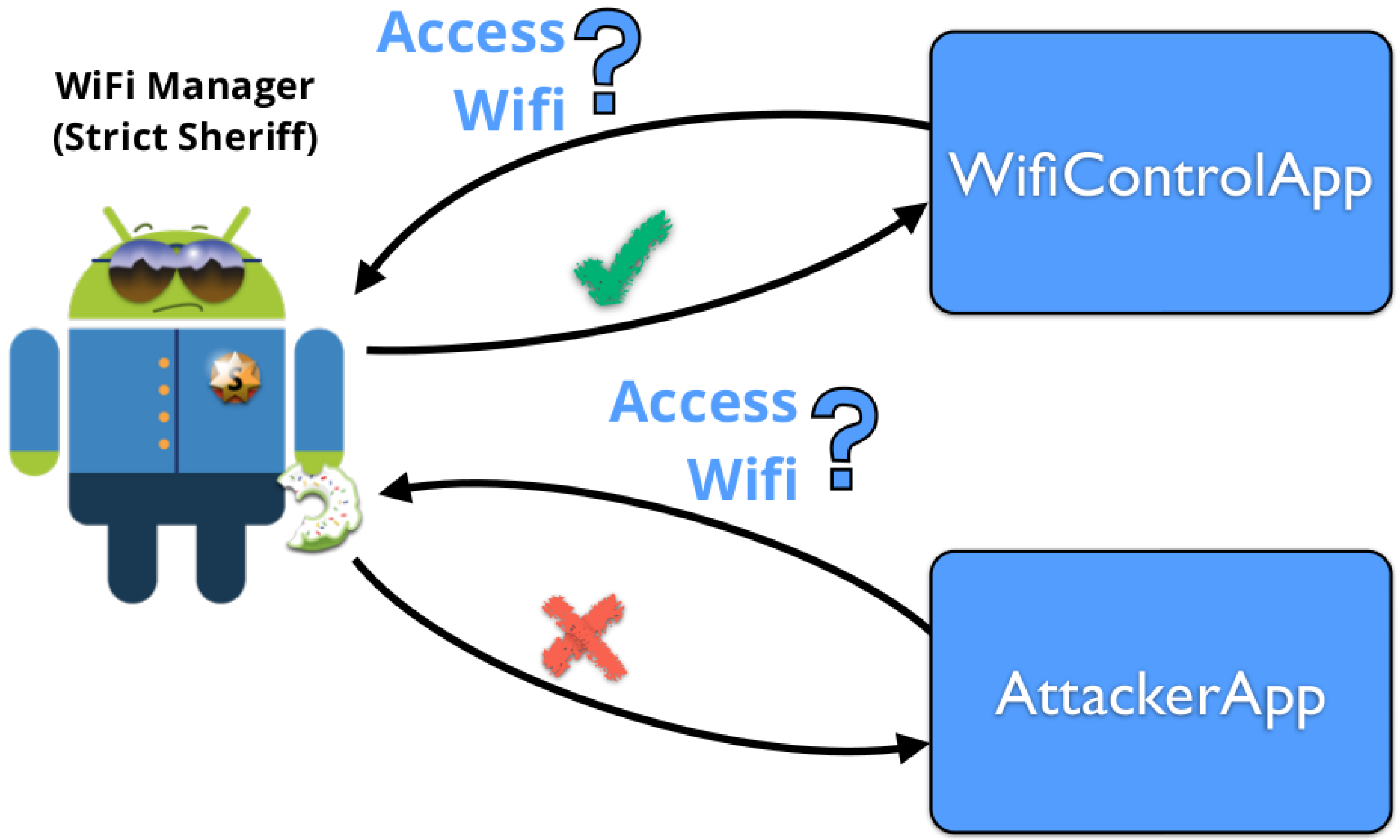
But…
Is this model really secure?

Permissions Re-delegation

- Permissions Re-delegation occurs when an application, without a permission, gains additional privileges through another application

- Is a special case of the confused deputy problem

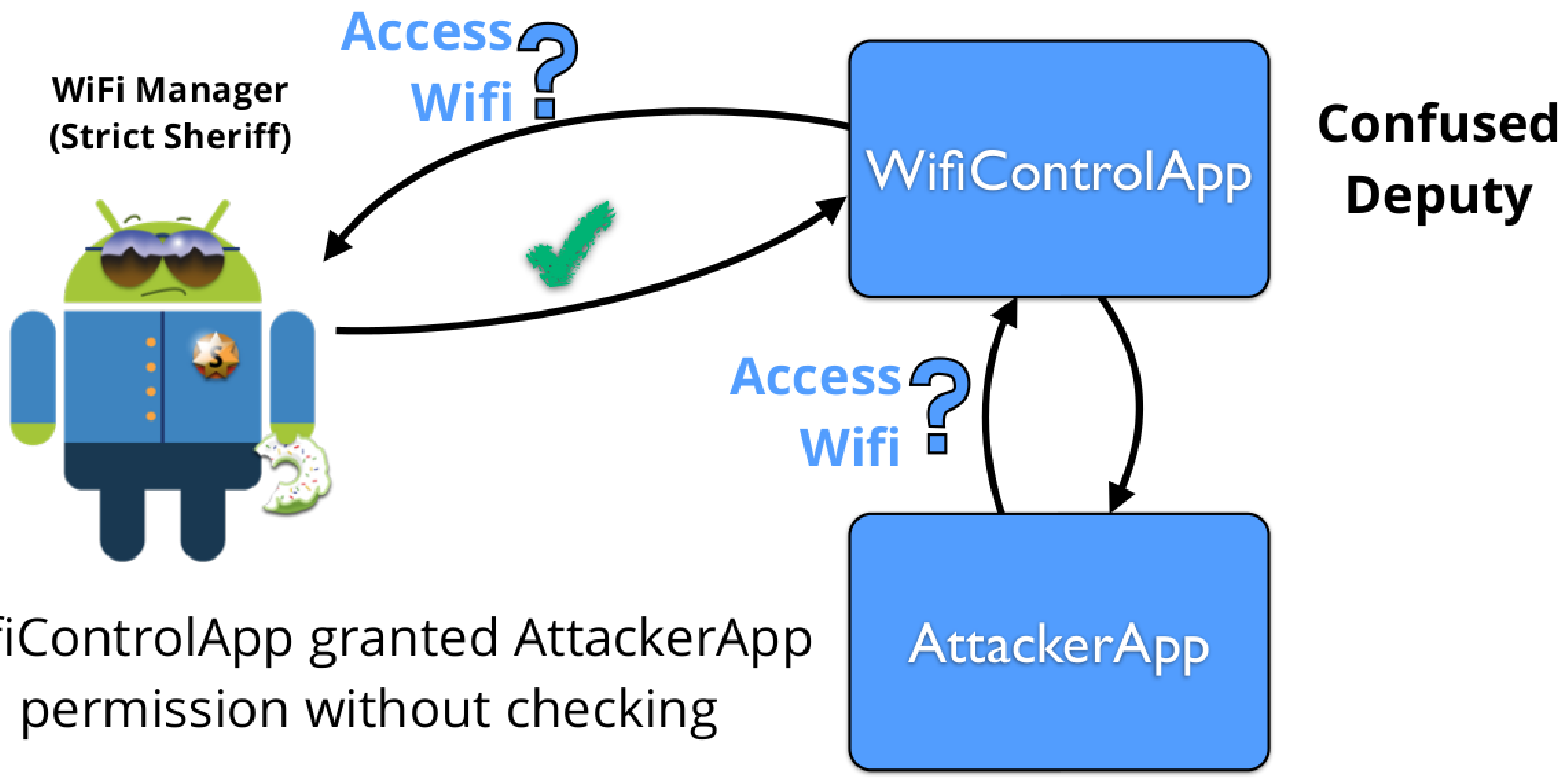
Permissions Re-delegation



Requested permission during install

No permissions during install

Permissions Re-delegation



WifiControlApp granted AttackerApp permission without checking

Permissions Re-delegation - Example

- This is possible since, if a public component (for example and Activity that do not have the "android:exported" property "false") doesn't explicitly have an access permission listed in its manifest definition, Android permits any application to access it.

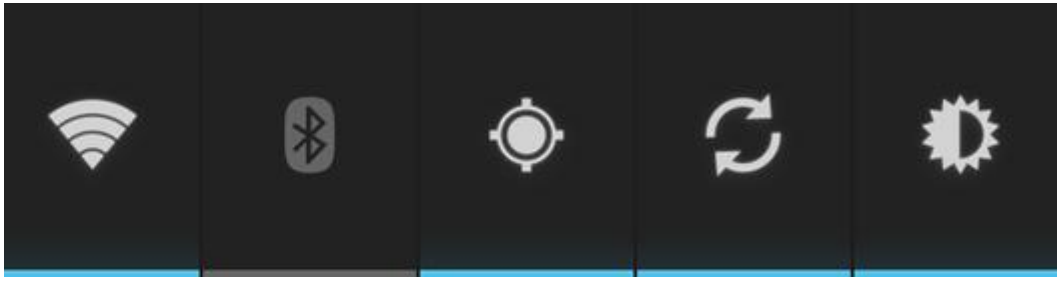
Permissions Re-delegation - Example

- The most famous example is the flaw found in the Power Control Widget that allows third party apps to change protected system settings without requesting permissions.

- Moreover this flaw is particularly dangerous because it permits to switch GPS and this action is not allowed to normal applications (at this moment Android do not provide any permission for toggling GPS).

Permissions Re-delegation - Example

- The “Power Control Widget” is one of the default widgets provided by Android (and therefore present on all devices)

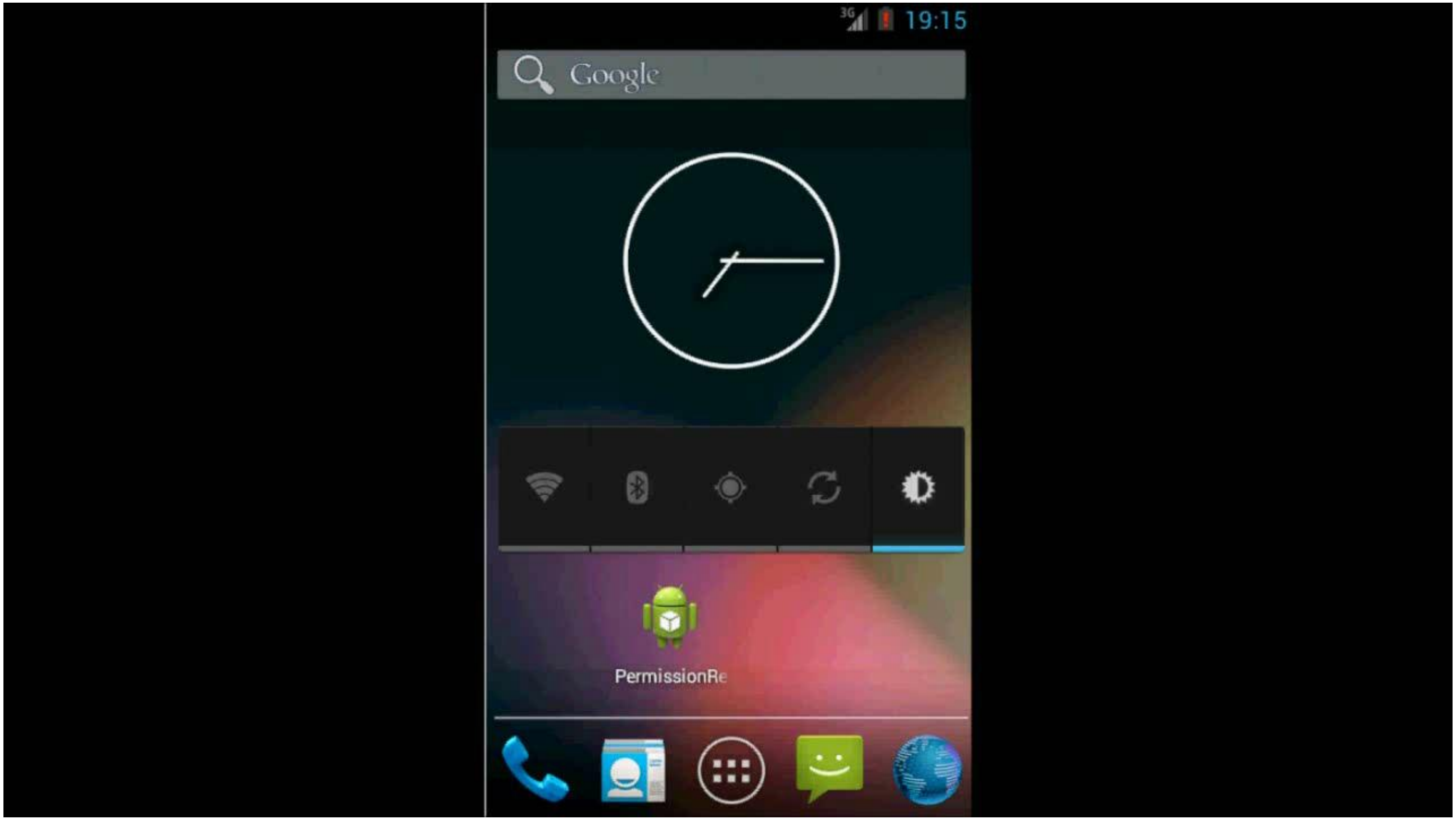


- It permits to toggle some settings (Wi-fi, BT, GPS, Data Sync, Screen Brightness) with only one click.

Permissions Re-delegation - Example

- Power Control Widget uses Intent to communicate the event of switching settings.
- A malicious app, without permissions, can send a fake Intent to the Power Control Widget, simulating the pressure of the button to switch settings.

Permissions Re-delegation - Example



Permissions Re-delegation - Example

- Vulnerable versions (red):

Version	Codename	API	Distribution
1.6	**Donut**	**4**	**0.10%**
2.1	**Eclair**	**7**	**1.50%**
2.2	**Froyo**	**8**	**3.20%**
2.3 - 2.3.2	Gingerbread	9	0.10%
2.3.3 - 2.3.7		10	36.40%
3.2	Honeycomb	13	0.10%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	25.60%
4.1.x	**Jelly Bean**	16	29.00%
4.2.x		**17**	**4.00%**

Permissions Re-delegation

- Good news
  - Principle of Least Privilege reduces the problem but is not the solution
  - Android provides developers some mechanisms to defend against this type of attack (custom permissions, manifest properties)

- Bad news
  - Currently final users have to rely on application developers for prevention since the protection of apps is not made at the system level

Permissions Avoidance

- Can we do more?

- Can an application, that do not require any permission, receives remote commands, access to sensitive data and send them to a remote handler?

Permissions Avoidance

- Currently, by default, all apps still have access to read from external storage without requiring permission.
  - So all apps could read data saved on external storage

- For example, in the external storage there is the DCIM dir, that include photos. But there are a lot of apps that store data in the external storage without protection.

Permissions Avoidance

- Moreover Android intents, allow opening some system applications without requiring permissions.
  For Example:
  - Open camera app
  - Open SMS app
  - Open contact list
  - And…Open Browser!

Permissions Avoidance

- Opening a browser, via intent, could be very dangerous since it permits to:
  - Transmit data
  - Receive remote commands
  - Download files

Permissions Avoidance

- Paul Brodeur of Leviathan Security creates an app that, with no permissions, can receive remote commands, fetch data from the external storage (and also some information about the phone and apps installed) and then send these data to a remote handler.

- The original “No Permissions” (v1.5) app was tested against Android 2.3.5 and Android 4.0.3

Permissions Avoidance

- In particular original “No Permissions” (v1.5) app can:
  - Receive commands from a remote handler
  - Scans the /sdcard directory and returns a list of all non-hidden files (including photos, backups, and any external configuration)
  - Fetch the /data/system/packages.list file to determine what apps are currently installed on the device (and so a list of any readable files stored by these apps)
  - Grab some identifiable information about the device itself (kernel version and possibly the name of the custom ROM installed, Android ID)

Permissions Avoidance

- And:
  - Exfiltrate any collected data via browser (using using a series of GET requests)
  - Fetch position by examining EXIF data of the photos
  - Donwload remote files
  - Execute java code
  - Start on boot
- All these action when the screen is locked/off…

Permissions Avoidance

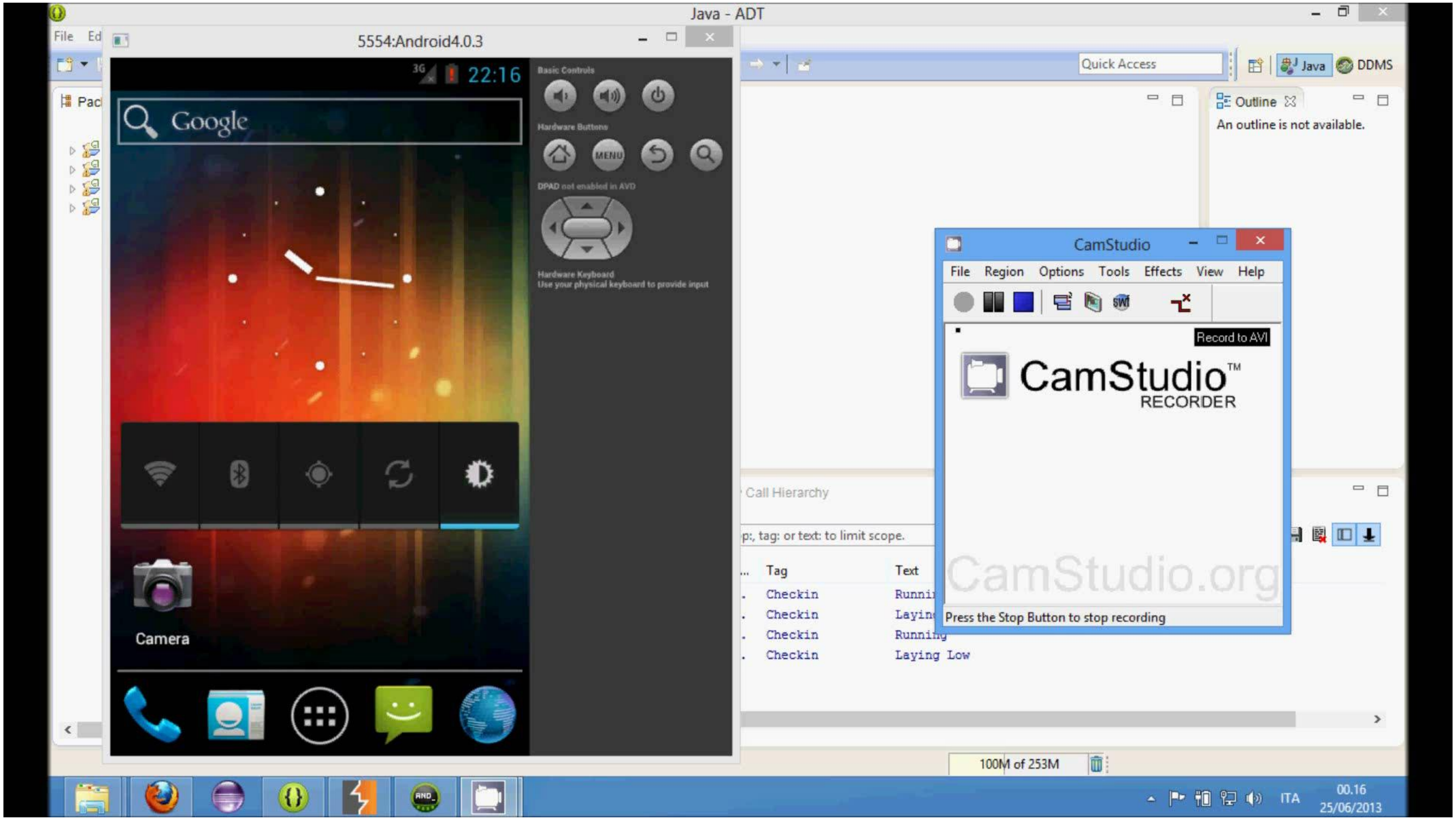
- I have done some changes to the original app, and then tested it with the latest Android versions available at the moment of writing (until Jelly Bean – 4.2.2).

- And with some “social engineering” and trickery it is possible to do almost all things also with Jelly Bean.

Permissions Avoidance

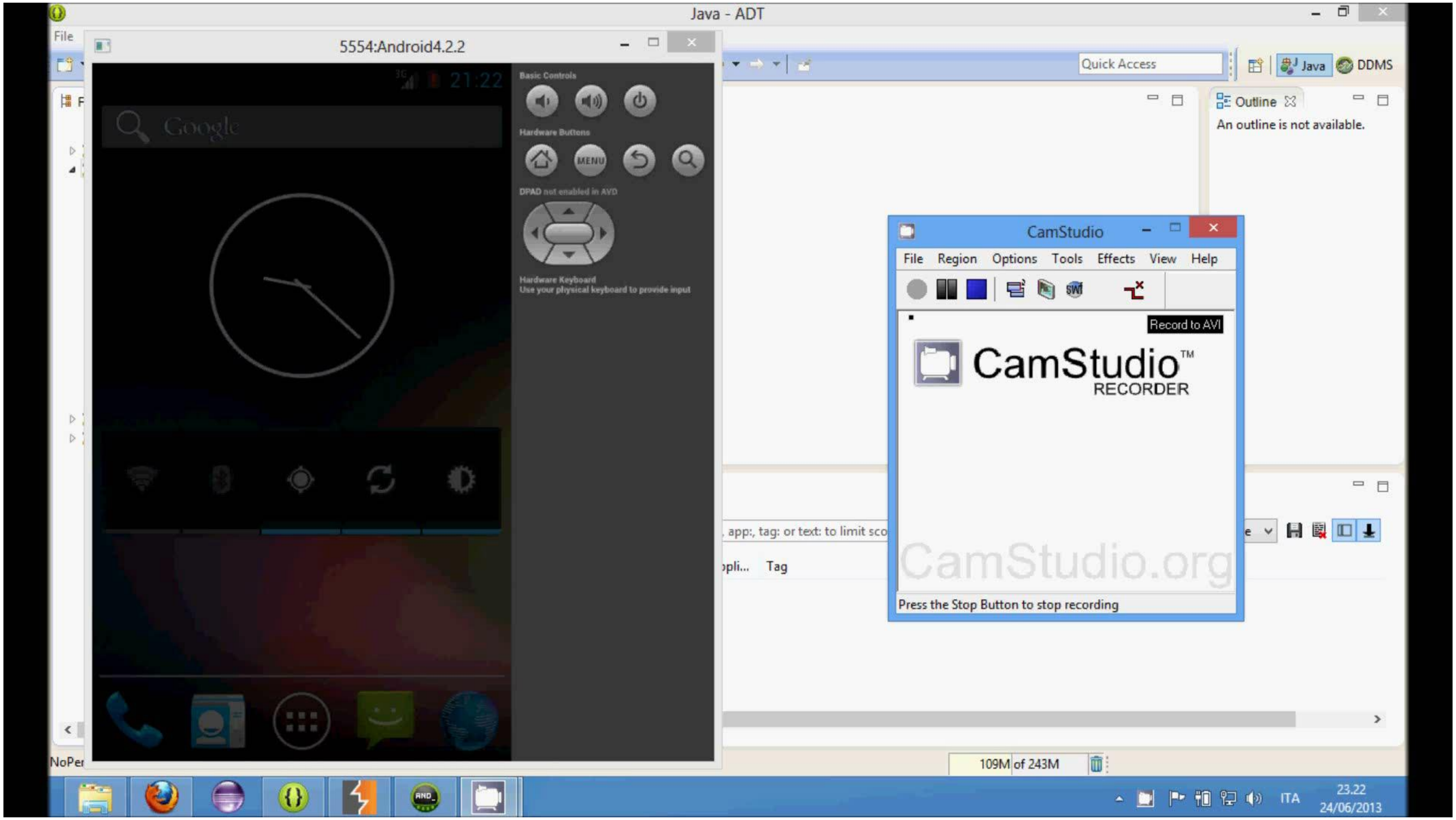
- With explicit intent is possible to target the stock web browser (in case the victim has more than one browser).
- With a trick it is possible to use a POST request instead GET making it easier to upload large amounts of data.
- Moreover, since with Jelly Bean is not possible to open web browser when the screen is off, it is possible to use some “social engineering”, for example simulating a game, to deceive the victim.

Proof of Concept - Ice Cream Sandwich



```
p:, tag: or text: to limit scope.
... Tag       Text
.   Checkin   Runni
.   Checkin   Layin
.   Checkin   Running
.   Checkin   Laying Low
```

Proof of Concept – Jelly Bean



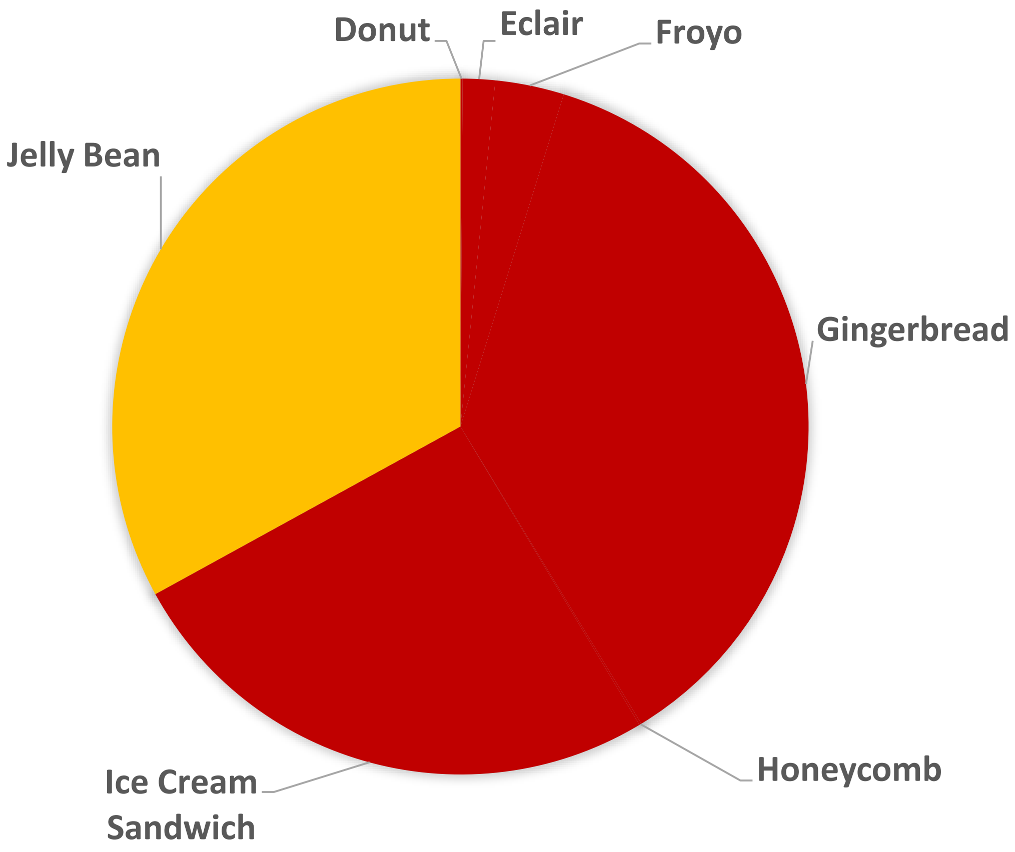
An outline is not available.

Press the Stop Button to stop recording

Proof of Concept - Summary

- Vulnerable (red), partially vulnerable (orange)

Version	Codename	API	Distribution
1.6	Donut	4	0.10%
2.1	Eclair	7	1.50%
2.2	Froyo	8	3.20%
2.3 - 2.3.2	Gingerbread	9	0.10%
2.3.3 - 2.3.7		10	36.40%
3.2	Honeycomb	13	0.10%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	25.60%
4.1.x	Jelly Bean	16	29.00%
4.2.x		17	4.00%



Proof of Concept - Summary

		DISTRIBUTIONS			
		Gingerbread	Honeycomb	Ice Cream Sandwich	Jelly Bean
ACTIONS	Fetch data on external storage	YES	YES	YES	PARTIAL
	Fetch installed apps	YES	YES	YES	YES
	Fetch identifiable information	YES	YES	YES	YES
	Send data via Browser	YES	YES	YES	YES
	Receive commands	YES	YES	YES	YES
	Fetch position by examining EXIF	YES	YES	YES	YES
	Donwload remote files	YES	YES	YES	YES
	Execute java code	YES	YES	YES	YES
	Act when the screen is off	YES	YES	YES	NO
	Start on boot	YES	YES	YES	N/A

Proof of Concept

- Good News
  - Users can enable “Protect USB storage” under Developer options in the Settings app on a device running Android 4.1 or higher.

- Bad News
  - Currently the protection of data is entrusted to the developer and is not made at the system level
  - Currently there is no way to protect from opening the browser and exfiltrate data

Conclusions

- Android has a lot of features that allow to keep apps and data safe but, currently, the protection from flaws seen previously rely on applications developers, that have to:
  - Follow the principle of Least Privilege
  - Protect components of the application from third party apps
  - Store confidential data correctly (if possible not on the external storage) and encrypted

Conclusions

- End users can partially defend themselves:
  - Install only trusted applications (is not enough to check the permissions required by applications)
  - Enable “Protect USB storage” under Developer options in the Settings app on a device running Android 4.1 or higher.
  - Do not keep sensitive data stored on external storage

References (1/2)

- *Android Dashboards*
  - http://developer.android.com/about/dashboards/index.html
- *Android Security Overview*
  - https://source.android.com/tech/security/
- *Zero-Permission Android Applications*
  - http://www.leviathansecurity.com/blog/zero-permission-android-applications/
  - http://www.leviathansecurity.com/blog/zero-permission-android-applications-part-2/
  - Paul Brodeur

References (2/2)

- *Security and Privacy in Android Apps*
  - https://developers.google.com/events/io/2012/sessions/gooio2012/107/
  - Jon Larimer , Kenny Root
- Permission Re-Delegation
  - http://www.cs.berkeley.edu/~afelt/felt_usenixsec2011.pdf
  - Adrienne Porter Felt, Helen J. Wang, Alexander Moshchuk, Steven Hanna, Erika Chin

Thanks for your attention!

- Contacts
  - Personal: davide.danelon@owasp.org
  - Work: davide.danelon@mindedsecurity.com
  - Site: http://www.mindedsecurity.com
  - Blog: http://blog.mindedsecurity.com
  - Twitter: http://www.twitter.com/mindedsecurity