Automatic Trust Based Segregation for Content Providers on Mobile Devices

Oren Poleg

Supervisor: Dr. David Movshovitz

Problem

- According to the Cloud Security Alliance Report, **2 out of the top 8 security threats to mobile are related to information leakage:**
  - Information-Stealing Malwares
  - Poorly written applications

Problem

- It is **easy to grab the entire phonebook** of an **android** device and transmit it to the net

```
Cursor phones = getContentResolver().query(
                        ContactsContract.CommonDataKinds.Phone.CONTENT_URI,
                        null,null,null, null);
while (phones.moveToNext())  {
            String name=phones.getString(phones.getColumnIndex(
                        ContactsContract.CommonDataKinds.Phone.DISPLAY_NAME));
            String phoneNumber =phones.getString(phones.getColumnIndex(
                        ContactsContract.CommonDataKinds.Phone.NUMBER));
             // Send over to your favourite site on the network
}
phones.close();
```

(+2 permissions in the manifest)

Problem

- A behavioral study shows that **only 17%** of participants **paid attention to permissions during installation**, and **only 3% actually understood the meaning** of the **various permissions**.

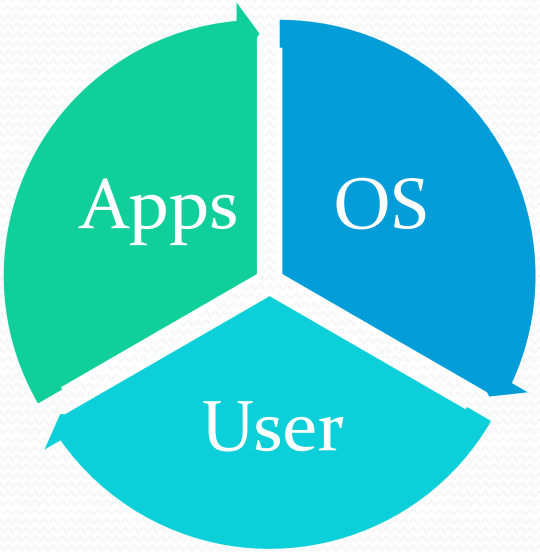
Adrienne Porter Felt, Elizabeth Hay, Serge Egelman, Ariel Haneyy, Erika Chin, David Wagner: Android Permissions: User Attention, Comprehension, and Behavior
http://www.cs.berkeley.edu/~afelt/felt-androidpermissions-soups.pdf

Goal

- **Prevent leakage of sensitive data** from mobile device (Android) by 3rd party applications
- **Take security decisions off the hands of the user**
- **Maintain usability**
  - Allow unified view of the data

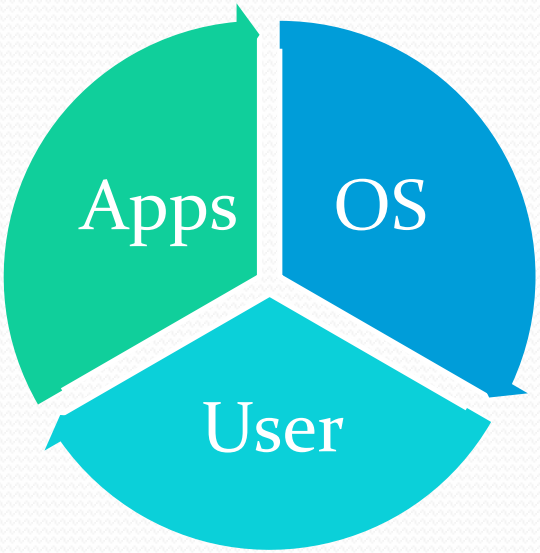
Threat Model

- **OS**
  - The underlying **OS** is **intact**, **trusted**
  - **System applications** (system contact manager / system calendar) are **benign** from privacy protection perspective and will not release private data of the device without authorizations.



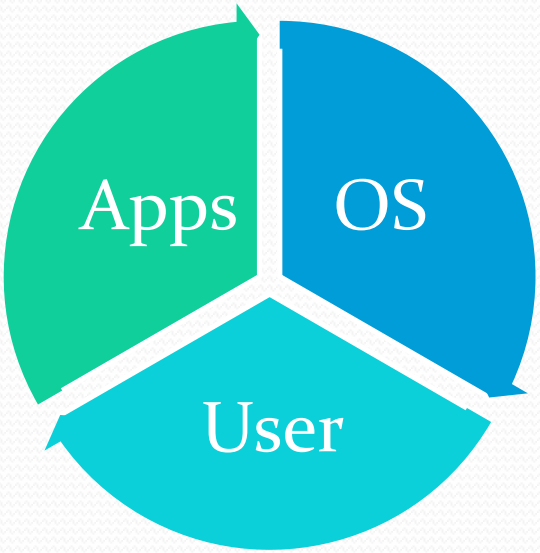
Threat Model

- **User**
  - User can be **trusted** with the data
  - User **cannot** be **trusted** to take the right **security decision**
  - May install any 3rd party application



Threat Model

- **3rd party applications**
  - **May** try to read and **leak sensitive** information
  - Have **no root** privileges



Current Solutions

Current Solutions

Secure Containers

- Airwatch
- FiberLink
- Zenprise (Cirtix)
- Good Technologies

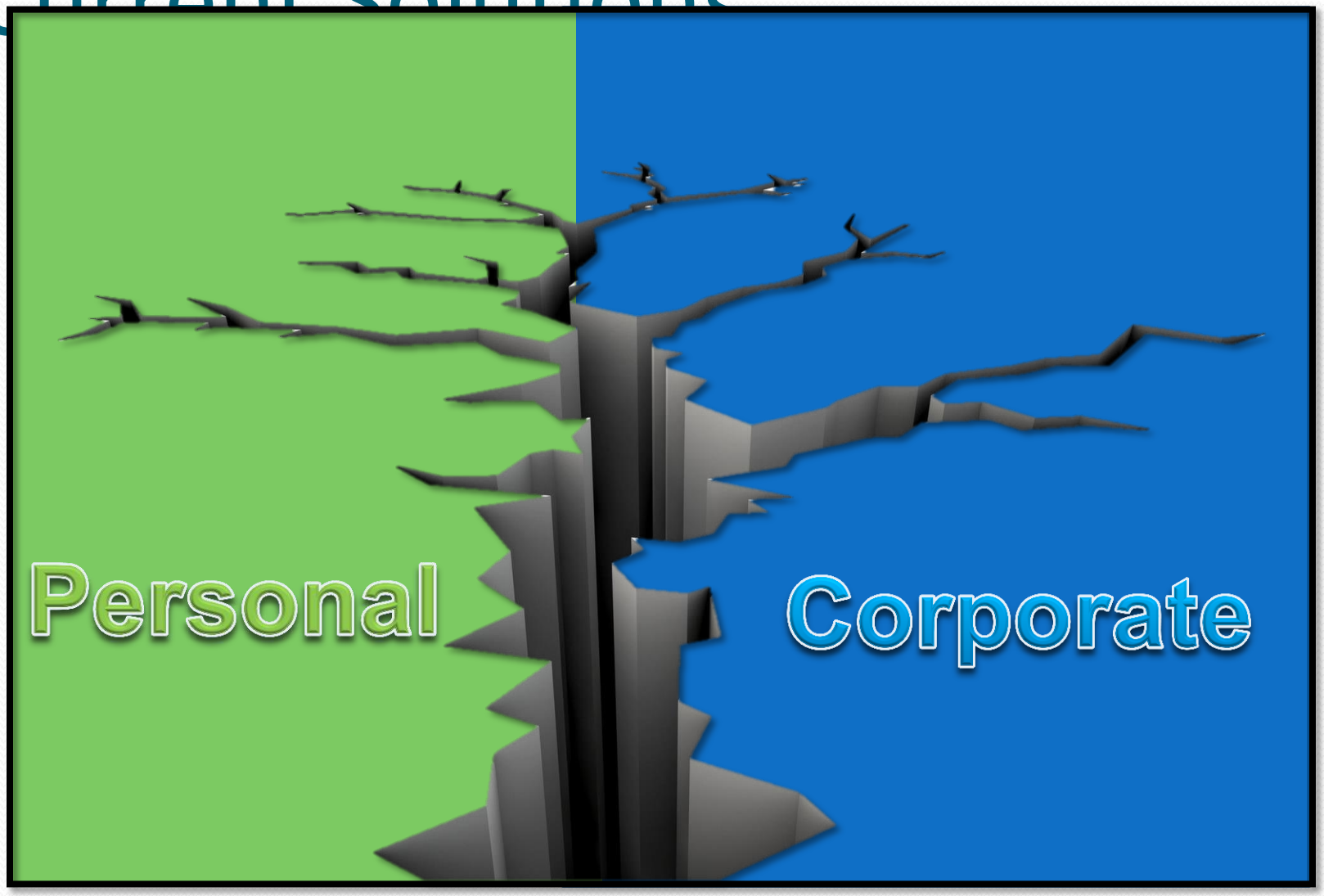
Remote Desktop

- VMware Horizon View 5.2
- Citrix Receiver

Virtualization

- CELLROX
- VMware Horizon Switch

Current Solutions



Current Solutions - Continued

Dynamic Tagging	• TaintDroid
Secure Package Manager	• MockDroid
Secure Content Handlers	• TISSA
Hardened OS	• SE Linux for Android

Current Solutions - Continued

Dynamic Tagging	oid
Secure Pack Manager	oid
Secure Content Handlers	A

- **User <u>Management</u> Required**
- **Application Centric Approaches**

Goal (revisited)

- **Prevent leakage of sensitive data** from mobile device (Android) by 3rd party applications
- **Take security decisions off the hands of the user**
- **Maintain usability**
  - Allow unified view of the data

Solution Principles

- We define **subsets of the user's data** (Contact list, Calendar, SMSs) that will only be available to the creating entity and to other entities that have the **creators trust**.

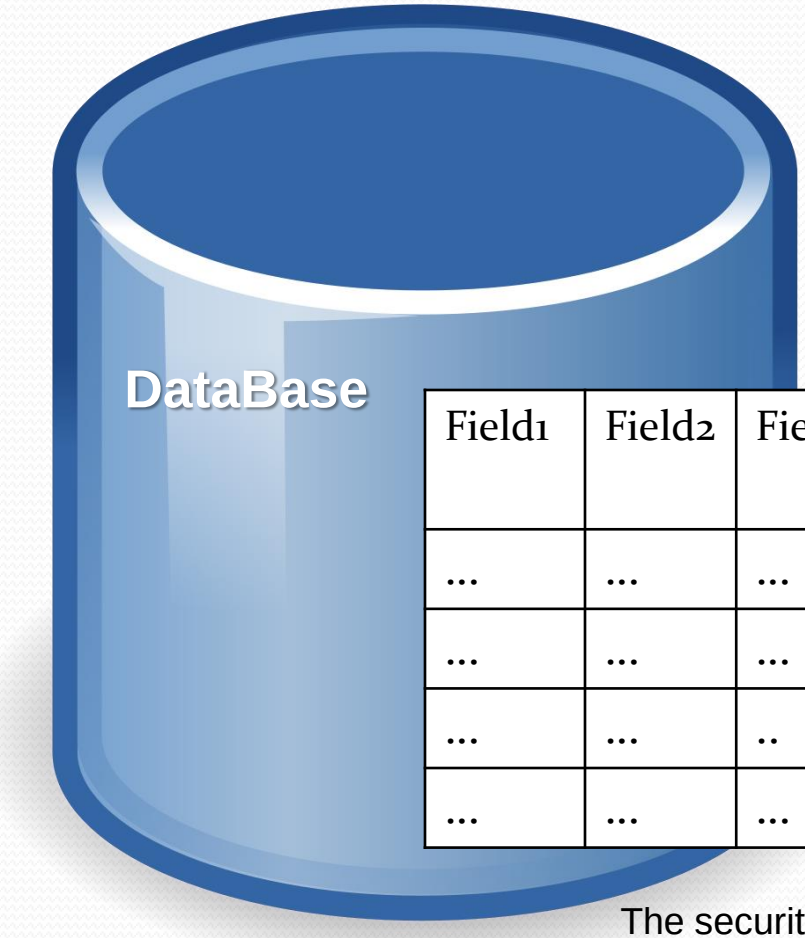
- **Untrusted 3rd party** applications will be **unaware** of the sensitive data nor could they access the data, thus will not be able to leak it.

Solution Principles

- **Data record** is the **basic element** to which the application may have access to.
- Each record in the content handler may have a **security tag** specifying an owner .

Data Centric

How it works



Field1	Field2	Field3	Security Tag
…	…	…	Owner1
…	…	…	Owner2
…	…	..	
…	…	…	Owner1

The security tag is derived from the public key of the Owner.

Solution Principles

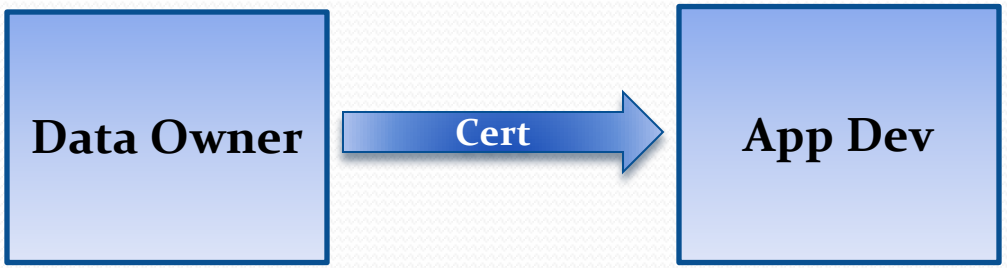
- An **application** is the equivalent of **user** in the current models
- The system maintains the **identity** (**public key**) of the **application developers** for each installed app.

App = User

Design

- An **owner** of a record is responsible for **granting** entitlements regarding the record.
- The system can directly verify if the owner is trying to access its own data by matching the caller's id with the tag of the record.
- When a different application tries to access the data, it will need to present a **certificate** to be validated by the system.

How it works



Certificate:

- Data **Owner** Public Key
- **Developer** Public Key
- **Entitlements**

- Signed with Data Owner Private Key

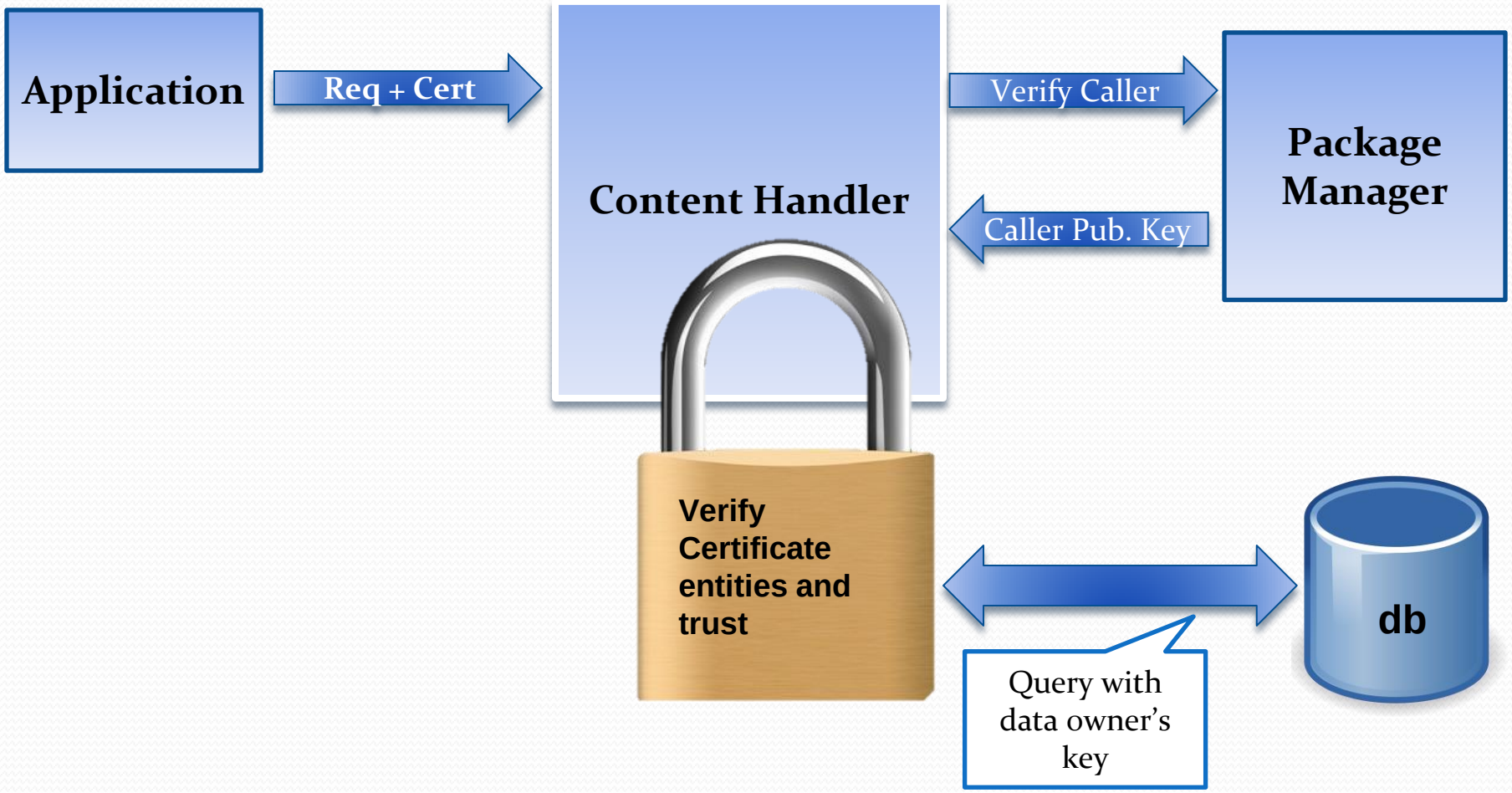
Design

- **Entitlements** will be implemented as **a list of keys and values**, to provide flexibility in implementation.
- **Basic entitlements** will map to the basic functionality provided by the content handler – **query**, **insert, delete, update**, controlling the functionality that the system will allow the user to perform on the data.

Design

- To maintain **compatibility** with **standard Android**, the following entitlements can be hard coded:
  - Owner have all entitlements
  - The **system applications** have **query/update/delete** entitlement.

How it works


Application
Req + Cert
Content Handler
Verify Caller
Caller Pub. Key
Package Manager
Verify Certificate entities and trust
db
Query with data owner's key

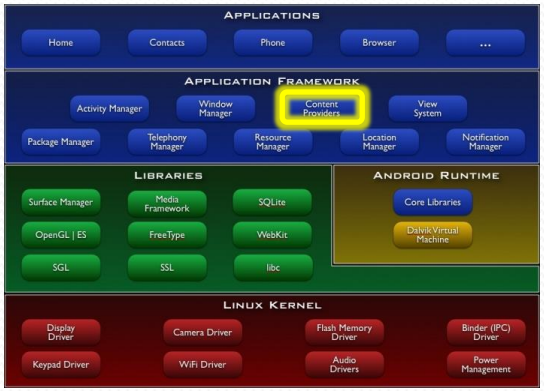
Design Considerations

- **Simple** implementation / Minimal changes
- Base segregation on **trust** between applications (Remove the user from the equation)
- Allow easy **integration** with **current Android** code

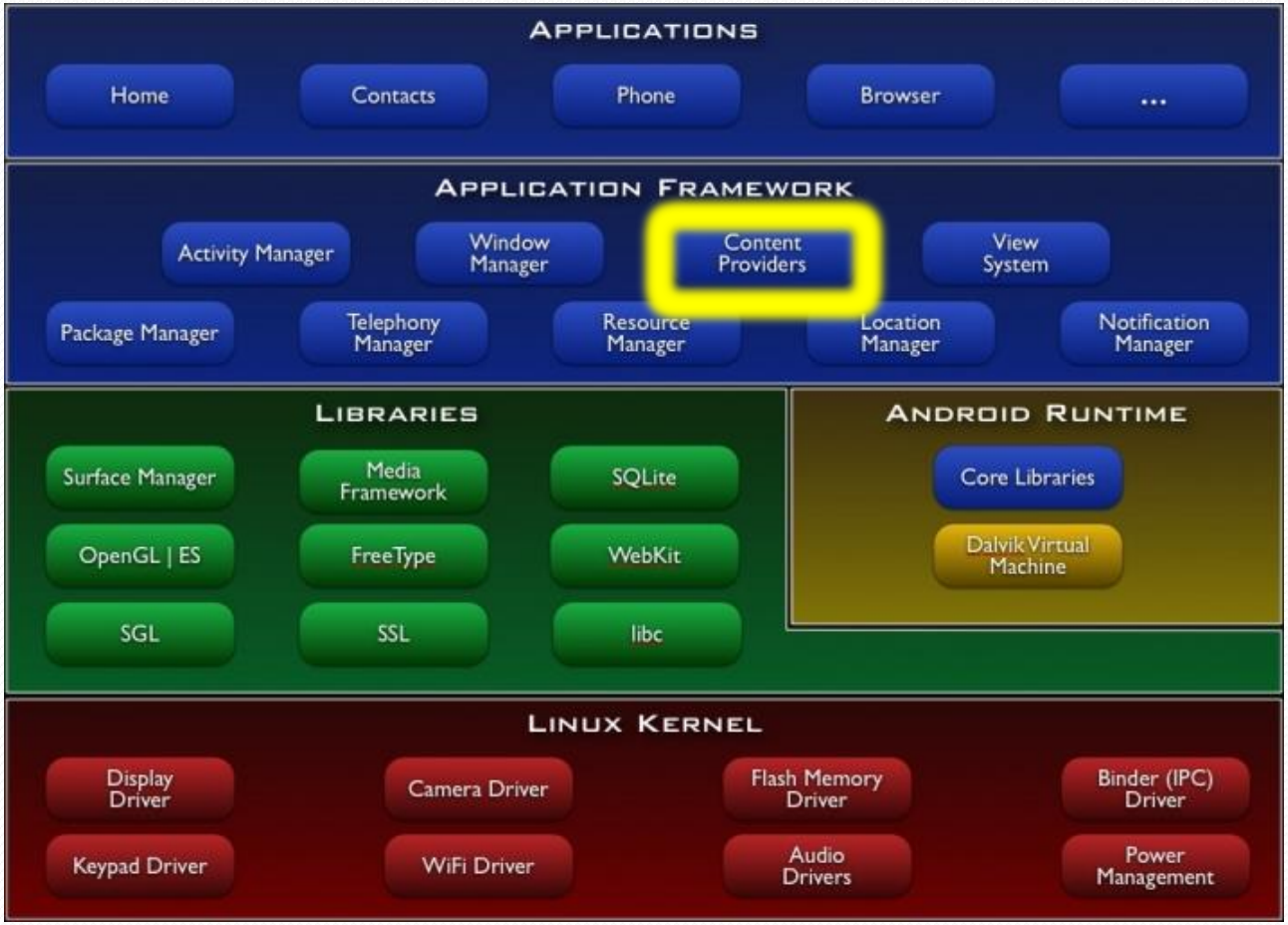
Implementation Overview

- In this work we have **designed** and **developed** a **security wrapper** around the **Android Content Providers** that allows us to **enforce access restrictions** on the **sensitive data** items they contain (such as contacts, calendar events)

- This approach is applicable to IOS



Implementation Overview

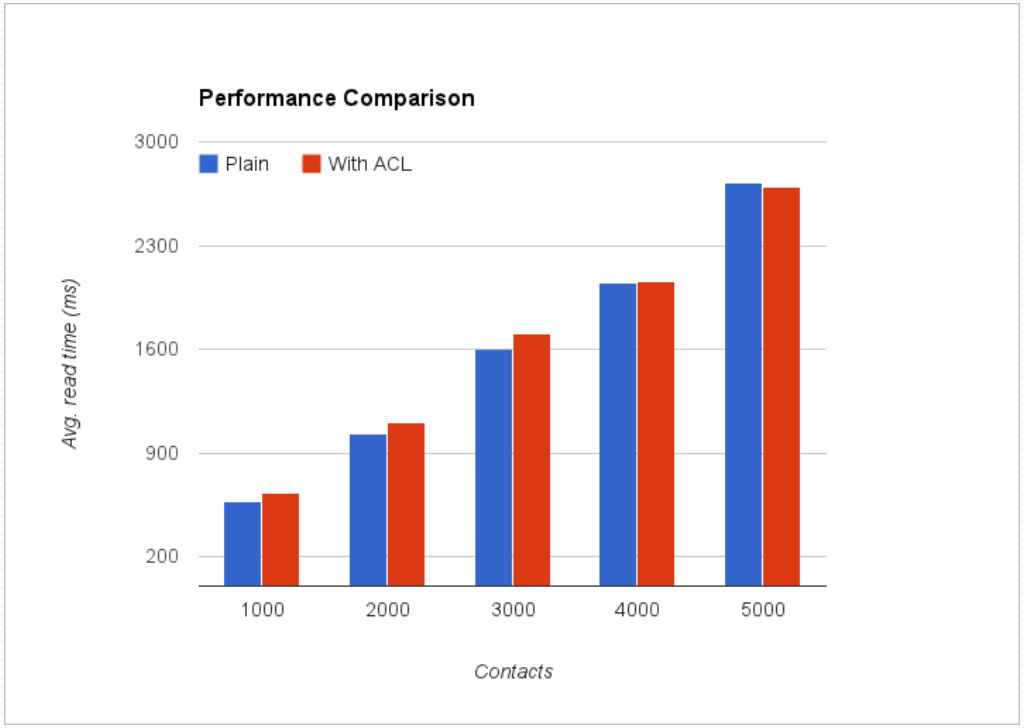


Proof of Concept

- We have implemented and tested the system on Android 4.0.3
  - Small modification to the Android System
    - Content Handler
    - Contact Database
  - Two test applications + System Application
    - “Corporate” contact list
    - “Evil” contact list
    - **System** Contact list
- Tested on Emulator and **Samsung S2**

Performance

- No measurable difference per transaction



Summary

- We have implemented a **data centric security wrapper** around the **Android content providers** that allows us to **enforce access restrictions** on the **sensitive data** .

☑ Isolates sensitive data
☑ Small footprint
☑ Can easily integrated with current Android
☑ No user interaction
☑ Can provide a unified view of data to the user

Open Issues

- Handling the data when the creating application is removed
- Reading data from multiple owners

Q&A

Oren Poleg
owasp.2013@poleg.org
+972-544-537405