An Access Control Model for Applications on Mobile Devices using Common Criteria Certifications

Helmut Kurth, Trang Huynh

atsec information security corp.

Overview

The Problem

- **Access control on smartphones:**
  - Why do we need it?
  - Who are the players?
    - What are their expectations?
  - What are suitable “objects”?
  - Who are the “subjects”?
  - What are the access control rules?
  - How are access rights initialized and managed?
  - How is access control enforced?
  - How can this be integrated in current systems?
    - Using Android as the example

Overview

From the problem to a possible solution

- **Elements of the access control policy**
  - Explaining the roots of the model: The Caernarvon access control model
  - Enhancements made to adopt it to smartphones
  - The role of digital certificates
  - Binding access control information to apps
    - Both access rights and access authorizations
  - Evaluating access control information apps and the OS
  - Managing access control information
  - The role of Common Criteria certified apps

Why not a “traditional” OS Access Control System?

What does exist

- **Traditional server operating systems:**
  - They manage different users (as subjects)
  - Usually files as objects
- **Client operating systems:**
  - They manage different roles (partly different users)
- **Both models are not suited for smartphones!**
  - Clearly just a single “user”
  - Files are not suitable “objects”
  - Do you really want to manage “roles” on your smartphone? *(I don’t!)*

Traditional OS access control is not suited for smartphones!

Expectations on Access Control

The Device Owner

- Access control should be “secure by default”
- “No app should be able to misuse my phone by stealing personal information and/or using phone services undercover”
- “Don’t bother me with complex access control management!”
- “Give me assurance that my phone and my personal data are protected!”
  (whatever “protected” means)

Expectations on Access Control

The Service Provider

- Access control should protect against misuse of phone services – as far as my liability is involved!
- Please no bad press that my phones “got hacked”!
- “As long as it generates revenue and I am protected – I am fine”
- I don’t care too much about apps – as long as I can control which phone services they are allowed to access
- Don’t bother me with interaction between apps
  - That’s not my business

Expectations on Access Control

Application Provider

- For sensitive applications we *(the application provider)* want to control which other apps use our services – and how they use them
  - Example: Payment application offered by a bank
    - Should not be misused for covert payments
    - Should only be usable by applications somehow “approved” by the bank
- We do not want anybody to weaken those controls!

Which “Objects” Need Protection

The Assets – Part 1

- **Personal information**
  - Contact details
  - Calendar
  - E-mail
  - ...
- **Phone resources**
  - Phone service
  - SMS service
  - GPS
  - Wi-Fi (Internet)
  - Bluetooth
  - ...

Which “Objects” Need Protection

The Assets – Part 2

- **Software resources provided by apps**
  - Service requests from other apps
  - Application defined resources (Objects)
  - Application defined “access types”
  - ...
- **All those lists are open ended**
  - New personal files may be created
  - New phones may provide new services
  - New apps may provide new software services requiring access restrictions
  - ...

What are the “Subjects”?

The Stakeholders on a Smartphone

- **Stakeholders are:**
  - The smartphone owner (as the single user)
  - The primary service provider
  - Application providers

The Basics of the Access Control Model

Basic Ideas

- **The model we propose is a combination of ideas from:**
  - Capability based systems *(anybody remember those?)*
  - The access control model of IBM Research's smart card operating system, Caernarvon
  - Controlling access using a central "access manager"
  - Defining and managing access rights using digital certificates
  - Authorizing specific access for "trusted" (evaluated) apps only, not for "untrusted" apps

The Basics of the Access Control Model

Objects and Access Rights

- The OS and apps can “register” new objects (resources)
- With each object it registers a “default access”. Any app not explicitly authorized for a “higher level” of access gets this default access
  - Semantics of the access rights are defined by the app
- Higher levels of access require explicit authorization
  - Either by the “object owner” (the service provider for OS objects or the developer of the app that has registered the object)
  - Or by a rule that allows CC evaluated apps to get higher levels of access
    - Details are explained later

The Basics of the Access Control Model

What about the Owner of the Phone?

- He may always define more restricted “default access rights”
- If allowed by the OS or the app, he may define less restrictive default access rights
- When the default are set “securely”, it should not be necessary for him to do anything!
- For personal files like contacts, etc.:
  - Usually managed by one specific application
  - This app has a default policy for access of other apps
  - User may overwrite this default policy (if app allows)

The Basics of the Access Control Model

Certificates

- **Each application is digitally signed (potentially by multiple entities) and comes with the following:**
  - A set of phone services, phone OS interfaces, and app services it requires or optionally wants to use
  - A set of certificates for a kind of “pre-approved” access rights
    - One certificate from the service provider allowing access to phone services and OS interfaces (optional)
    - Potentially multiple certificates from app developers allowing access to services provided by their apps (optional)
    - One certificate from a CC certification body (optional)
  - For access requests not “pre-approved” the default access rights apply

The Basics of the Access Control Model

Drawbacks (without CC based Certificates)

- Every app would potentially need to get certificates from the service provider and many app developers!
  - That's not very practical
- What is needed is assurance that the apps satisfy defined objectives
  - That's what the Common Criteria schemes are able to provide
- How can this be included in the access control model?

The Role of CC Certification

Digital Certificates for CC Certifications

- **Let us assume:**
  - Service providers and app providers publish a set of security objectives plus a minimum set of (CC) assurance requirements an application needs to satisfy to access specific services in a mode higher than the default access
  - App provider then may get their apps CC evaluated against those objectives and the required assurance components
  - The phone would then need to validate that the app either
    - Comes with a certificate from the “owner” of the resource he is requesting, allowing access, or
    - Comes with a certificate from a CC certification body demonstrating compliance with the security objectives and the minimum assurance requirements

What Would this Require?

Subjects for “Standardization”

- **Security objectives would need to be identifiable**
  - Requires some kind of common coding of security objectives (and assurance components)
  - Requires the code of the security objective to be part of the certificate issued by the CC certification body
  - Requires the certificate to be bound to the app (as for all other certificates coming with the app)
  - Requires a PKI for CC certification bodies

How to Encode Security Objectives

A Suggestion

- An app provider obtains a “provider ID” (PIn) from a CC certification body (needs to be unique, nothing else)
- The app provider registers (with a CC certification body) a list of security objectives and numbers them (SO1 to SOn)
- Now those security objectives can be encoded in a digital certificate issued by a CC certification body encoded as “PIn.SOm”
- In his app he issues as a set of requirements like:
  - PI5.SO7, PI5.SO15, PI5, SO21, EAL4, AVA_VAN.4

Example 1

Payment Application – Part 1

- Provided by application provider PA7
- Application requires:
  - Any app that wants to use a specific service (O) of the payment application has to be:
    - Either signed by the payment service provider, or
    - Evaluated with security objectives PA7.SO5, PA7.SO8 and being evaluated at EAL4 augmented with AVA_VAN.5
    - Default access is “none”
  - When installed, the payment app registers “object” O with those access conditions with the access manager

Example 2

Payment Application – Part 2

- User installs an app X that wants to use the payment application P
- During installation, app X provides its certificates to the access manager who validates them
- Access manager stores the access rights defined by the certificates for the application
- App X connects to app P and requests type A access to service O
  - App P connects to the access manager and asks if app X has access type A to “object” O
  - The access manager checks his access database and tells the payment app if access can be granted or not

Example 2

Database of Contacts – Part 1

- Database of contacts is usually managed by one app
  - Let's name it CM (for Contact Manager)
- This app registers several "objects"
  - Object 1: contacts marked by the owner as "public"
  - Object 2: name and phone number of non-public contacts
  - Object 3: name and address of non-public contacts
  - Object 4: all fields of non-public contacts

It is up to the CM app to define those "objects" with their semantics

Example 2

Database of Contacts – Part 2

- Access authorizations registered with access manager
  - Object 1:
    - Default access is READ, access UPDATE and CREATE requires certificate from the CM app provider or EAL2 with security objectives CM.1, CM.2, and CM.3
  - Object 2:
    - Default access is NONE, access UPDATE and CREATE requires certificate from CM app provider or EAL3 with security objectives CM.1, CM.2, CM.3, and CM.4
  - .... And so on

Example 2

Database of Contacts – Part 3

- App X wants to obtain the name and address of a contact
  - Sends request to App CM
  - App CM identifies that this requires READ access to its Object 3
  - App CM sends a request to the access manager asking:
    - Does App X have READ access to my Object 3?
  - Access Manager evaluates the information in his access database and answers with ‘yes’ or ‘no’

Managing Access Rights

Granting and Revoking Access

- Access authorizations are bound to apps (and the OS)
- Change requires an “update” of the app or OS that defines the access authorizations
  - Update process would automatically register the modified access authorizations with the access manager, invalidating the previous access authorizations
- An app that wants to have a higher level of access to a resource would also need to be updated (potentially just with additional certificates)
  - Update process would automatically register the modified access rights with the access manager, invalidating the previous access rights

Possible Integration into Android

Overview

- **What Android does today**
  - The current access control model of Android
- **How Android could use the new model**
  - Implementation of a “policy server” (as the “access manager”)
    - An old idea for microkernel architectures developed in the late 80’s
    - All policy requests are directed to a specific trusted application, which makes the access decision
      - This allows for different access control models

Current Android Access Control Model

Android Policy

- "All-or-nothing" policy enforcement model
- Apps declare permissions they require to use the phone's resources (fits with our model)
- Each app runs on their own process with a unique Linux user ID assigned at install time (fits with our model)
- App's own resources (components) are assigned with access permission labels (somehow fits)
- Apps are signed with developer's self-signed digital certificate (no use for this in our model)

Current Android Access Control Model

"All-or-nothing" permission model

- For a successful installation of an app, user must accept all permissions requested at install time of the app (needs to be done by the access manager in our model)
- Permissions are set at installation time and cannot be modified until reinstall (fits with our model)

Permission levels (need to change!)

- Normal – granted by system without explicit user approval
- Dangerous – granted at install time after user approval
- Signature – granted only if the requesting and granting apps both have the same certificates
- SignatureSystem – granted to packages in the Android system image or that are signed with the same certificates

Current Android Access Control Model

Applications define and enforce their own permissions

- Apps statically declare permissions they require, i.e., app developer defines the app's security policy *(fits with our access authorizations)*
- Each app has its own AndroidManifest.xml file where the app declares:
  - Permissions it requires to interact with other apps *(fits)*
  - Permissions it requires to access protected parts of the OS API *(fits)*
  - Permissions that other apps require to interact with the app's resources (components) *(fits)*

Current Android Access Control Model

Application signing

- Apps must be digitally signed with their developer/provider certificate (needs to change unless it is for integrity verification only)
- Signing with self-signed certificates are allowed (not for access authorizations)
- Apps signed with the same signature may share the same user ID thus allowing sharing of code and data (could be modeled in the access manager rules)

How Android Could Use the New Model

Implementation of a “policy server”

- An old idea for microkernel architecture developed in the late 80’s
- Taken up by the definition of “access managers”
- All policy requests are directed to a specific trusted application, which makes the decision
  - This allows for flexible access control policies
    - Dynamic definition of new subjects and objects
    - Flexible access control rules
    - Integration of access control policies using CC evaluations as one element
    - ...

What is a Policy Server?

Policy Server

- Acts as an “access manager”
- Centralized access control for the OS and apps where services for the phone are registered
- One policy server per phone
- Comes with a set of pre-defined meta rules for the evaluation of access rights
  - If an app does not meet any of the default rules, explicit authorization is required
  - Explicit authorizations can be defined in several ways. CC based certificates is one of them

The Policy Server

How does it work

- At app install time:
  - OS retrieves all the certificates that come with the app and submits them to the policy server for validation
  - Policy server validates the certificates (i.e., ensures that they have been issued by a trusted entity)
  - Policy server validates the correct binding to the app (i.e., the hash value of the app is part of the data that is signed)
  - Policy server stores the resulting access rights in its “access database”
  - Policy server uses those access rights to make access decisions later

Standard Access Control Policy

What's it for?

- For apps that do not have CC certifications
- Verifies (via the policy server) that the app requesting access to phone services is either authorized by the default access right or by a digital certificate of the "resource owner"

What does it require?

- Identification of app requesting access
  - Needs to be provided by the OS
- For access other than the default access: valid/trusted certificate from object owner
- ...

CC-Related Access Control Policy

What's it for?

- For apps requesting services that do not possess required digital certificates from the object owner
- Checks if the object owner has defined access rights bound to CC certifications

What does such a CC based certificate contain?

- (Encoded) Security objectives included as part of the evaluation
- Assurance components/assurance level
- Application Developer identifier
- Hash value of the application
- Additional restrictions/conditions (e.g., OS version required, access manager version required, not allowed when roaming)
- Signature from a CC Certification Body

Conclusion

Benefits

- The model allows for flexible access control on smartphones (and other mobile or embedded systems)
- The model can be integrated into existing smartphone operating systems
- The model allows for dynamic definition of new subjects, new objects, and new access authorizations and access rights
- The model allows for defining access authorizations bound to a CC evaluation of the requesting app
  - Not just for an assurance level, but also for clearly defined security objectives

Contact Information

atsec information security corp.

Helmut Kurth – helmut@atsec.com

Trang Huynh – trang@atsec.com

Thank you