Flexible and Fine-Grained Mandatory Access Control on Android for Diverse Security and Privacy Policies

Sven Bugiel
Joint work with Stephan Heuser and Ahmad-Reza Sadeghi

Cryptography and Information Security Group
Saarland University

Cyber-physical Systems Security
Fraunhofer SIT, Darmstadt

System Security Lab
Technische Universität Darmstadt
This talk is about:

**FlaskDroid**: Generic security architecture for Android

- Motivation for building a generic security architecture
- Design of FlaskDroid
- Policy language and example use-cases
Android’s Security Architecture Shown To Be Insufficient
Example Attacks on Android

- Root Exploits
- Confused Deputy Attacks
- Privacy Violations
- Collusion Attacks
Academic Security Extensions

http://1.androidauthority.com/wp-content/uploads/2012/03/security-notice-your-app-is-being-monitored.jpg
Key Observations
OBSERVATION #1: MOST SECURITY EXTENSIONS ARE MANDATORY ACCESS CONTROL SOLUTIONS TAILORED TO A SPECIFIC PROBLEM
Runtime policy:
(expose|access) (source app, type, action) (destination app, component) conditions
XManDroid / TrustDroid
[Bugiel et al., 2012/2011]

VALID policy language with Android-specific extensions
SE Android
[Smalley and Craig, 2013]

SELinux policy language (kernel) and MMAC extensions (middleware)
Nice to have: Policy-driven instantiations of use-cases

Existing work (e.g. Saint, TrustDroid,...) → Generic security architecture → New use-cases (e.g. Phone Booth Mode)
OBSERVATION #2:
ACCESS CONTROL REQUIRED AT BOTH USER-SPACE AND KERNEL LEVEL
Observation #2:
Access Control required at user-space and kernel level
FlaskDroid
Main Contributions

• System-wide security framework operating on both middleware and kernel layer
• Policy language specifically designed for the rich semantics at middleware layer
• Policy-driven instantiations of use-cases and related work
Design

- Dynamic, system-state aware policies
- Support multiple stakeholder
- Preserve security invariants (e.g., no root)
- Low-level enforcement in alignment with middleware

```
AccessControlCheck(Subject, Object, Operation, System State)  Policy provisioning/sync
```
Implementation
Policy Language

type android_t;
type contacts_email_v2_t;

class contentProvider_c
{
    query insert update delete
    readAccess writeAccess
};

class contactsProvider_c
inherits contentProvider_c;

Type definitions for Type Enforcement

New classes for middleware-specific objects
Policy Language (cont.)

```plaintext
bool phoneBooth_b = false;
kbool app_network;

if(!phoneBooth_b)
{
    allow app_telephony_t any: contactsProvider_c {query};
};

context phoneBooth_con;
switchBoolean
{
    context=phoneBooth_con;
    auto_reverse=true;
    phoneBooth_b=true;
};
```

- Boolean definitions for middleware and kernel
- Conditional policies
- Context definitions and mapping to boolean values
Policy Language (cont.)

```markdown
appType app_contacts_t
{
    Package:package_name=com.android.contacts;
};

intentType intentLaunchHome_t
{
    Action:hasAction=android.intent.action.MAIN;
    Categories:hasCategory=android.intent.category.HOME;
};
```

Metrics for dynamically assigning application and Intent types
## Base Policy

<table>
<thead>
<tr>
<th>Policy</th>
<th>#Types</th>
<th>#Classes</th>
<th>#Rules</th>
</tr>
</thead>
<tbody>
<tr>
<td>FlaskDroid Middleware MAC</td>
<td>111</td>
<td>18</td>
<td>109</td>
</tr>
<tr>
<td>(base policy from 12/04/2012)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SE Android 4.0.4</td>
<td>232</td>
<td>84</td>
<td>1359</td>
</tr>
<tr>
<td>(master branch, 12/04/2012)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SELinux Fedora 17</td>
<td>3900</td>
<td>83</td>
<td>103235</td>
</tr>
<tr>
<td>(targeted, policy.27 from 12/04/2012)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Use-case: “Phone booth mode”
Use-case: “Phone booth mode”

User activates phone booth mode (Context)

Query

Calllog Provider

Contacts Provider

“Contacts”

“Calllog”

“Home”
Use-case: Fine-grained Access to Contacts Data

Do these apps *really* need all my contacts data?
Or are just the telephone numbers or email addresses enough?
Use-case: Fine-grained Access to Contacts Data
Use-case: Fine-grained Access to Contacts Data
Further use-cases

• **App developer policies**
  (Saint)

• **Secure integration of higher privileged 3\textsuperscript{rd} party apps**
  (Firewall and Anti-Virus apps, no root required)

• **Multi-level security**
  (private vs. business domain)

• **Context-aware policies**
  (prevent reading sensor data while keyboard in foreground)
Quo Vadis?

• Port to SE Android 4.3
  – Integration with SE Android MMAC

• Towards *completeness*
  – Static analysis of API for hook placement
  – Formal analysis of policy subspaces

• More fine-grained types
  – Currently assigned to application sandboxes
Thank You!
Questions?

www.flaskdroid.org
References


