QuantDroid: Quantitative Approach towards Mitigating Privilege Escalation on Android

Tobias Markmann 1  Dennis Gessner 2  Dirk Westhoff 3

1 HAW Hamburg, Germany

2 NEC Laboratories Europe, Heidelberg, Germany

3 HFU Furtwangen, Germany

IEEE ICC 2013 - Communications and Informations Systems Security Symposium
Motivation

- Android popularity → increasing
- Privacy under attack! → Soundcomber (NDSS, 2011), PlaceRaider (NDSS, 2013), ...
- Permission model → confusing & inflexible

Source: PlaceRaider [2]
Android Security & Communication

System Security

- Common Linux security
- High-level permissions
- Sandbox for apps

↓

High-level IPC

Source: Programming Android [3]
Android Security & Communication

System Security
- Common Linux security
- High-level permissions
- Sandbox for apps

↓
High-level IPC

Communication
- High-level Middleware
- Unicast, Broadcast & RPC
- Poorly secured

Activity
- Intent
- Result

Service
- Remote method (AIDL)

Content Provider
- Queries/cursor

Broadcast Receiver
- Broadcast intent

Source: Programming Android [3]
Objective

- Identifying privilege escalation
- Detecting illegal information flow
  - Dishonest/Colluding apps
  - Abused apps
→ Prevent mobile privacy invasion
→ Using information flow analysis
Related Work

XManDroid (NDSS, 2012)

- Graph based
- App permissions
- Direct & indirect communication

Source: XManDroid [4]
Related Work

**XManDroid (NDSS, 2012)**
- Graph based
- App permissions
- Direct & indirect communication

**IPC Inspection (USENIX Sec., 2011)**
- Focus on permission redelegation
- Adjust IPC callee permissions
- Only reduced, never extended

Merely message independent interface-level permission control.

Source: XManDroid [4]
IPC Monitoring with FlowGraphService

IPC Monitoring

- At IPC boundary
- High-level communication methods
- Forwarding data collection
IPC Monitoring with FlowGraphService

IPC Monitoring
- At IPC boundary
- High-level communication methods
- Forwarding data collection

Monitoring Characteristics
- Sender (PID, UID)
- Receiver (PID, UID)
- Size
- Taint Tag (⋯)
IPC Monitoring with FlowGraphService

**IPC Monitoring**
- At IPC boundary
- High-level communication methods
- Forwarding data collection

**FlowGraphService**
- Real-time collection
- Communication graph
  - Containing all running apps
  - Quantitative data flow

**Monitoring Characteristics**
- Sender (PID, UID)
- Receiver (PID, UID)
- Size
- Taint Tag (,...)
IPC Monitoring with FlowGraphService

IPC Monitoring
- At IPC boundary
- High-level communication methods
- Forwarding data collection

FlowGraphService
- Real-time collection
- Communication graph
  ▶ Containing all running apps
  ▶ Quantitative data flow

Monitoring Characteristics
- Sender (PID, UID)
- Receiver (PID, UID)
- Size
- Taint Tag (薰, 薫, 釈,  nắm, …)

Limit enforcement
- Enforce data flow limits
- Based on taint tags
- Countermeasures
  ▶ Kill app
  ▶ Block IPC message
Utilising Dynamic Taint Tagging

TaintDroid (OSDI, 2010)

- Dynamic taint tagging
- Tag = data source
- Dalvik VM only, no native code
- Across IPC →
Utilising Dynamic Taint Tagging

TaintDroid (OSDI, 2010)
- Dynamic taint tagging
- Tag = data source
- Dalvik VM only, no native code
- Across IPC →

Source: TaintDroid [6]
Visualisation

- Current graph via custom `fgdump-tool`
- Graphviz for rendering

Example Snapshot

```
UID 10012
android.process.media
UID 10014
com.example.servicecomreceiver
UID 10008
Tag: IMEI
Throughput: 1664 Bytes/min
Tag: CONTACTS
Throughput: 1664 Bytes/min
Tag: CONTACTS
Throughput: 3968 Bytes/min
com.example.servicecomsender
```
Evaluation

Criteria

- Privilege escalation $\rightarrow$ sensitive data propagates across apps
- Works with standard Android SDK APIs

Test Scenarios

i) Conspiring apps

ii) Confused-deputy
**Scenario: Conspiring apps**

**Setup**

**Attack scenario: conspiring apps**

**Objective**

Innocent looking apps siphoning off contact data to send it off-site.
Scenario: Conspiring apps

Execution

UID 10042
com.example.snr_a.custommapping

UID 10041
Tag: CONTACTS
Throughput: 828 Bytes/min

UID 10040
com.example.snr_a.weatherreporter

UID 10040
com.example.snr_a.weatherwidget
Scenario: Conspiring apps

Execution

$T_2$

UID 10042

com.example.snr_a.custommapping

UID 10041

com.example.snr_a.weatherreporter

UID 10040

Tag: CONTACTS

Throughput: 216 Bytes/min

com.example.snr_a.weatherwidget

216 bytes
≈ 1 contact
Scenario: Conspiring apps

Execution

**T₃**

- **UID 10042**
  - com.example.snr_a.custommapping
    - Tag: CONTACTS
    - Throughput: 828 Bytes/min

- **UID 10041**
  - com.example.snr_a.weatherreporter
    - Tag: CONTACTS
    - Throughput: 216 Bytes/min

- **UID 10040**
  - com.example.snr_a.weatherwidget
Scenario: Conspiring apps

Execution

$T_3$ to $T_4$

UID 10042

Tag: CONTACTS
Throughput: 828 Bytes/min

com.example.snr_a.custommapping

UID 10041

Tag: CONTACTS
Throughput: 216 Bytes/min

com.example.snr_a.weatherreporter

UID 10040

Tag: CONTACTS
Throughput: 216 Bytes/min

com.example.snr_a.weatherwidget
Scenario: Conspiring apps

Execution

UID 10042
Tag: CONTACTS
Throughput: 828 Bytes/min

com.example.snr_a.custommapping

UID 10041

com.example.snr_a.weatherreporter

com.example.snr_a.custommapping

UID 10041

Tag: CONTACTS
Throughput: 828 Bytes/min

com.example.snr_a.weatherreporter
Scenario: Confused-deputy

Objective

SMS theft due to insecure/open API.

Execution

See our paper.
Conclusion

- Mitigate privilege escalation
- Quantitative IPC monitoring
- Limitation: Not monitoring IP-/UNIX-sockets

Outlook

- Analyse apps from PlayStore
- Investigating dataflow threshold heuristics
Conclusion & Outlook

Conclusion
- Mitigate privilege escalation
- Quantitative IPC monitoring
- Limitation: Not monitoring IP-/UNIX-sockets

Outlook
- Analyse apps from Play Store
- Investigating data flow threshold heuristics
Questions?

Tobias Markmann
Department of Computer Science
HAW Hamburg
tobias.markmann@haw-hamburg.de


