Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

NEC

HOCHSCHULE FURTWANGEN UNIVERSITY

QuantDroid: Quantitative Approach towards Mitigating Privilege Escalation on Android

Tobias Markmann [1] Dennis Gessner [2] Dirk Westhoff [3]

[1]HAW Hamburg, Germany

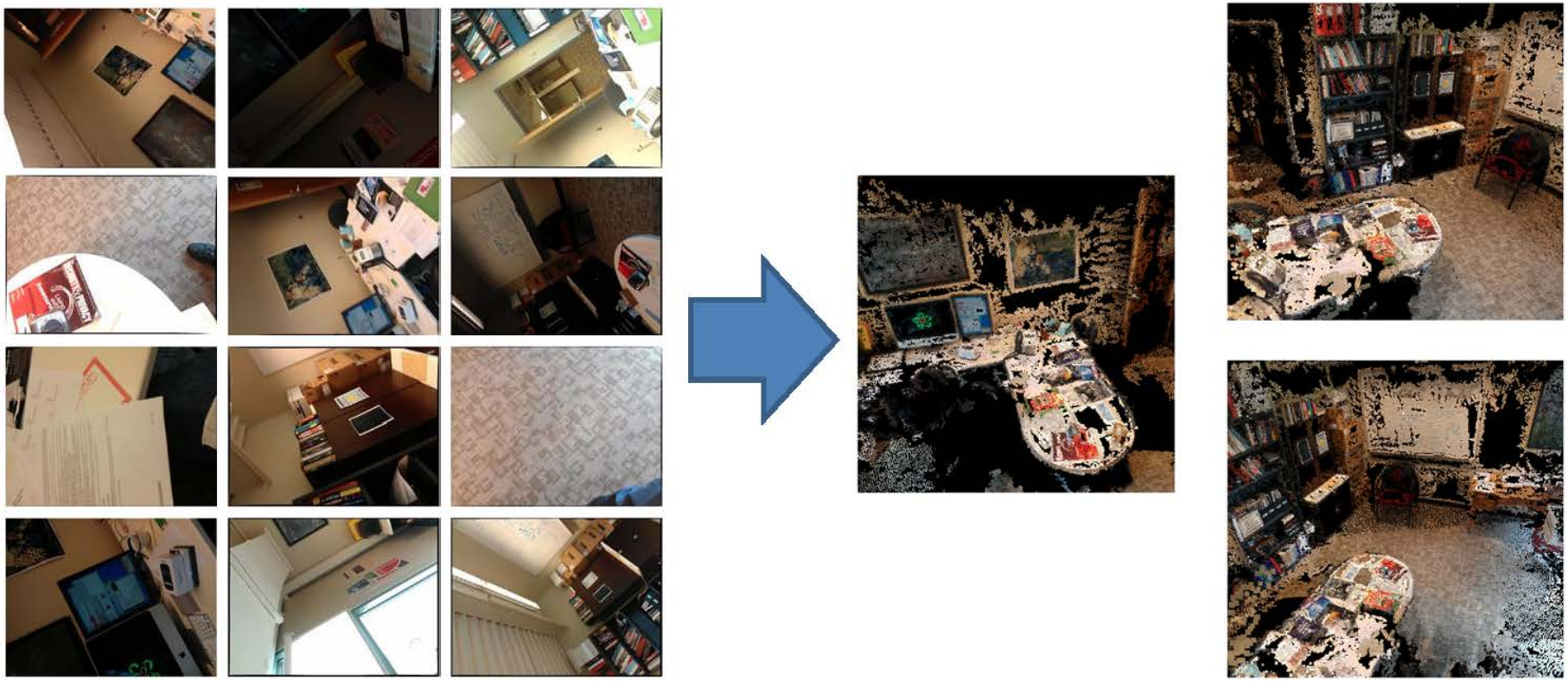
[2]NEC Laboratories Europe, Heidelberg, Germany

[3]HFU Furtwangen, Germany

IEEE ICC 2013 - Communications and Informations Systems Security Symposium

Motivation

- Android popularity $\rightarrow$ increasing
- Privacy under attack! $\rightarrow$ Soundcomber (NDSS, 2011), PlaceRaider (NDSS, 2013), ...
- Permission model $\rightarrow$ 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

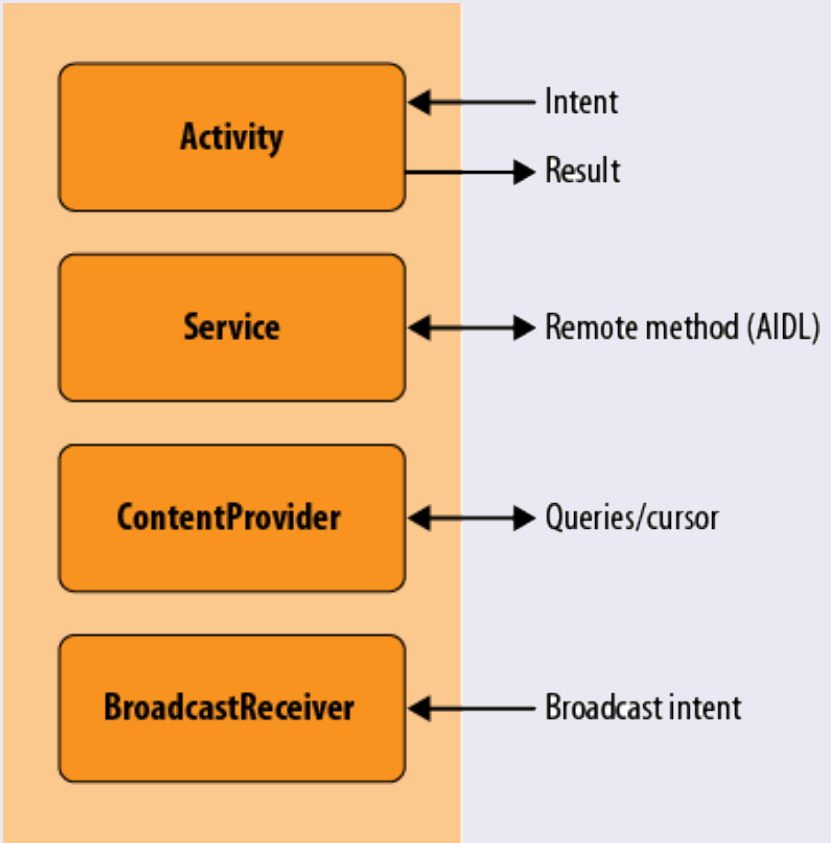
- Common Linux security
- High-level permissions
- Sandbox for apps

↓

High-level IPC

Communication

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



Source: Programming Android [3]

Objective

- Identifying privilege escalation
- Detecting illegal information flow
  - Dishonest/Colluding apps
  - Abused apps

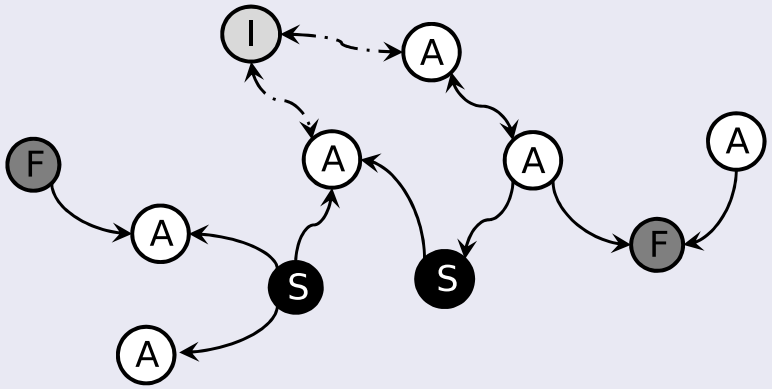
$\rightarrow$ Prevent mobile privacy invasion

$\rightarrow$ Using information flow analysis

Related Work

XManDroid (NDSS, 2012)

- Graph based



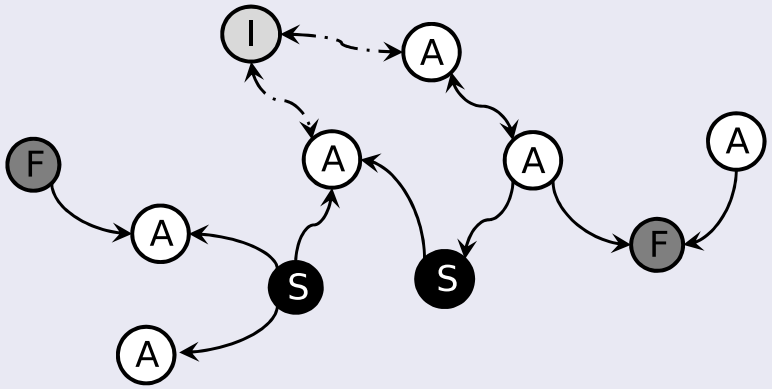
- App permissions
- Direct & indirect communication

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 ( ,  ,  ,  , …)

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 $\longrightarrow$

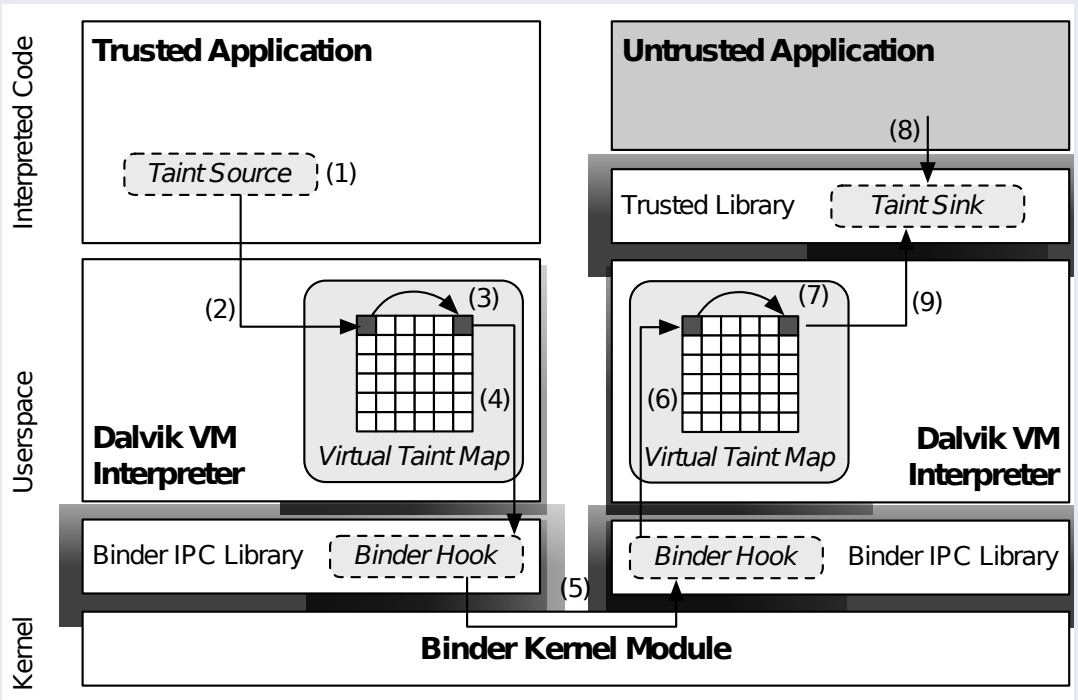
Source: TaintDroid [6]

Utilising Dynamic Taint Tagging

TaintDroid (OSDI, 2010)

- Dynamic taint tagging
- Tag = data source
- Dalvik VM only, no native code
- Across IPC $\longrightarrow$

Taint Tagged IPC

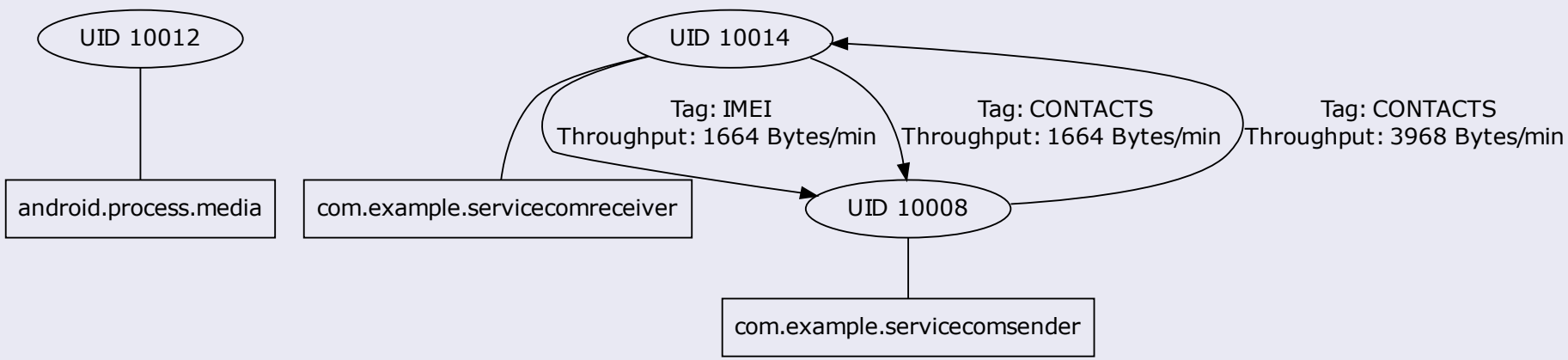


Source: TaintDroid [6]

Visualisation

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

Example Snapshot



Evaluation

Cirteria

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

Test Scenarios

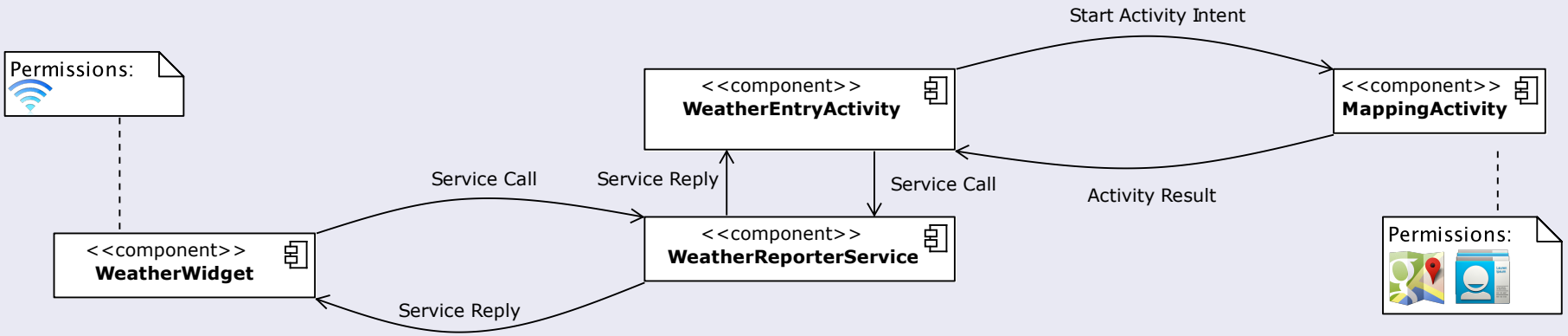
i) Conspiring apps

ii) Confused-deputy

Scenario: Conspiring apps

Setup

Attack scenario: conspiring apps



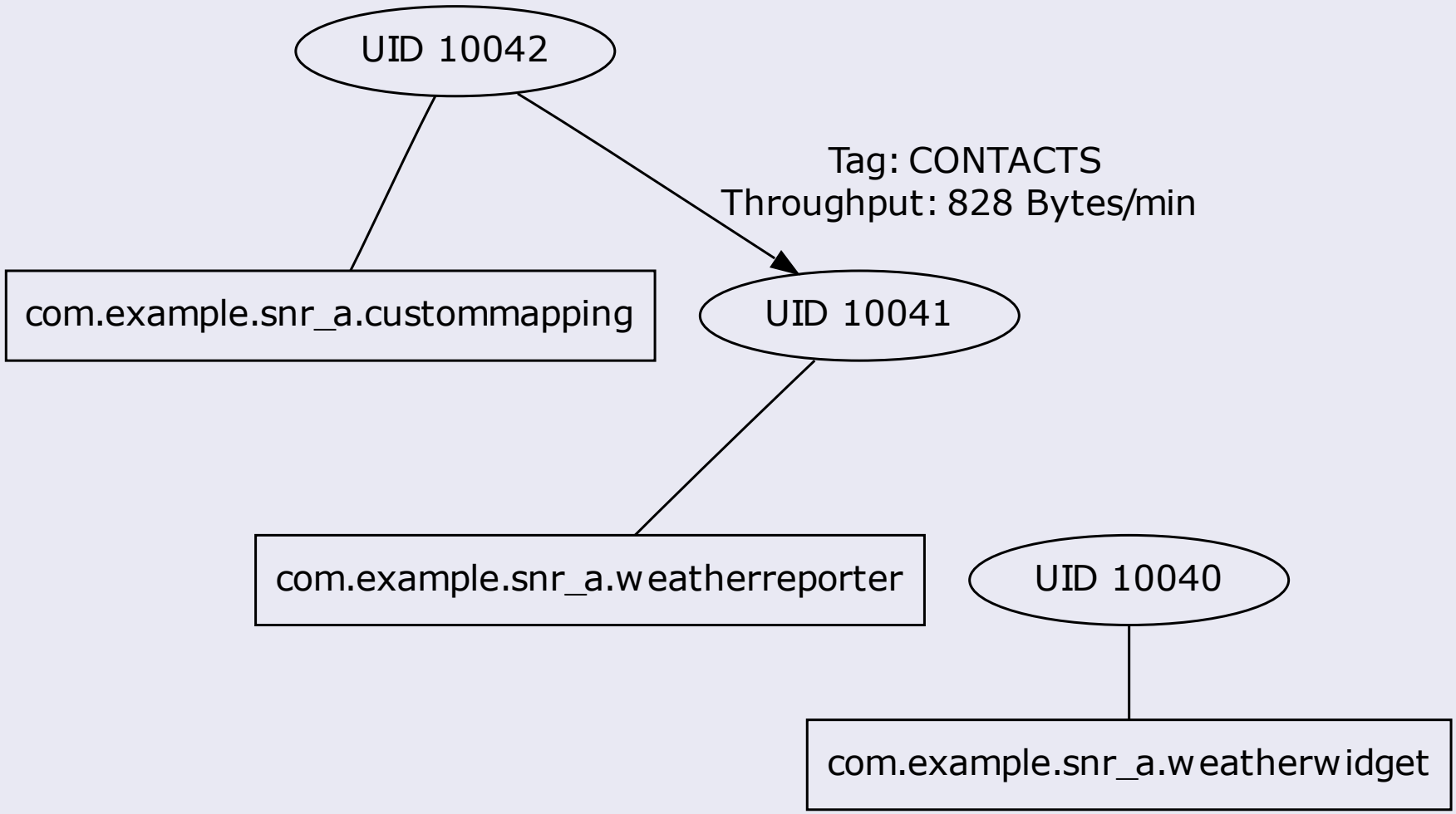
Objective

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

Scenario: Conspiring apps

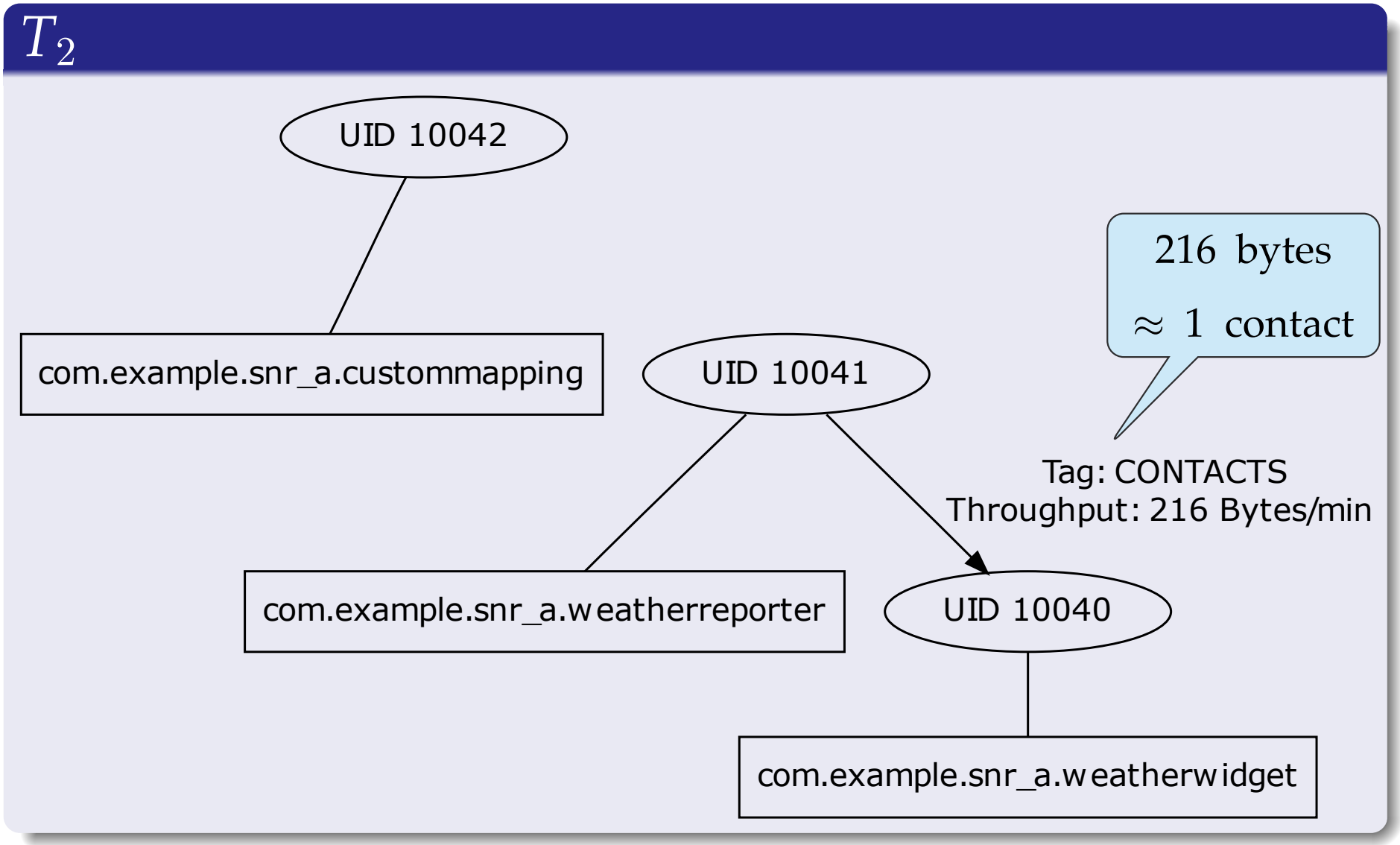
Execution

T_1



Scenario: Conspiring apps

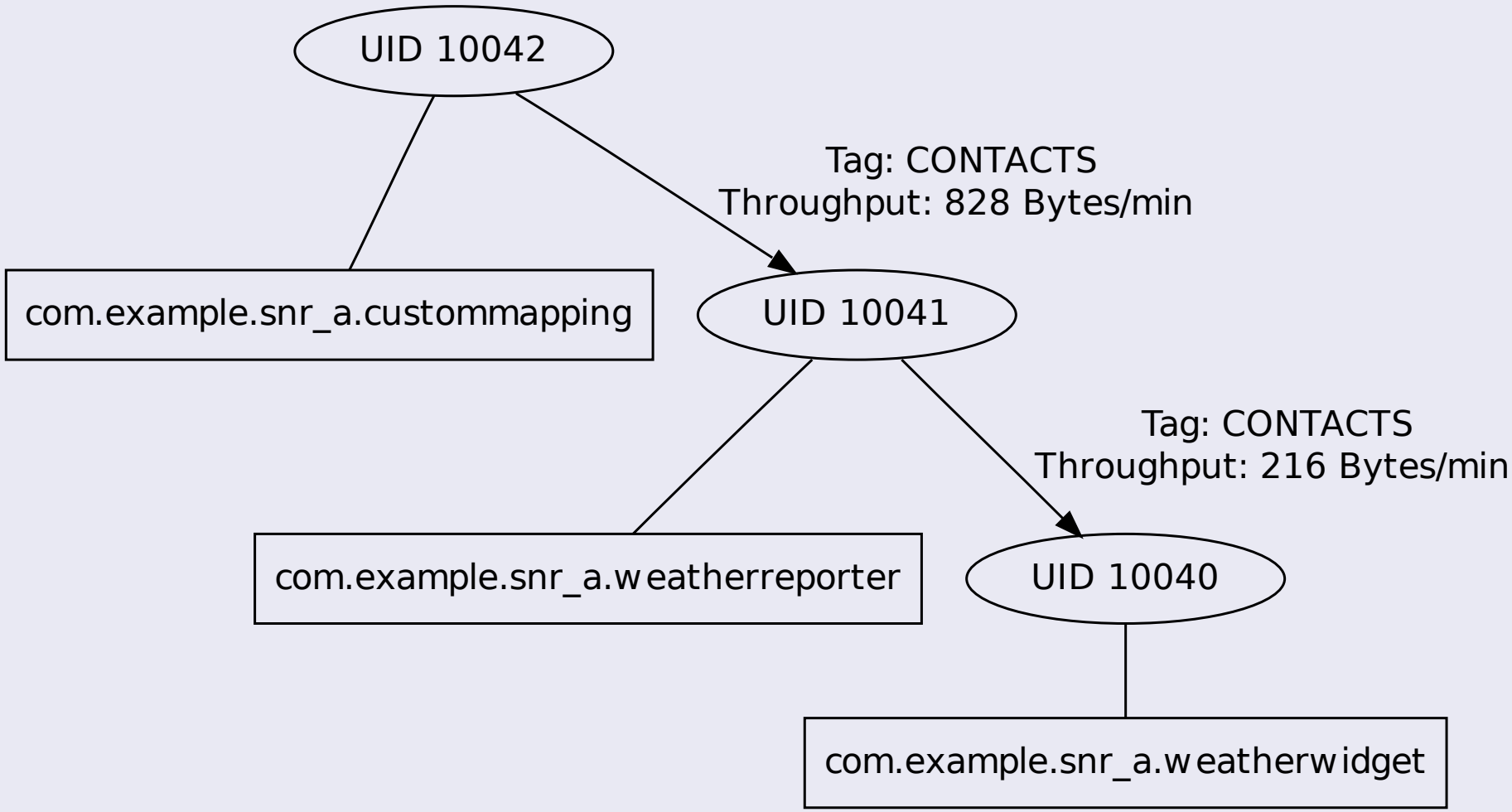
Execution



Scenario: Conspiring apps

Execution

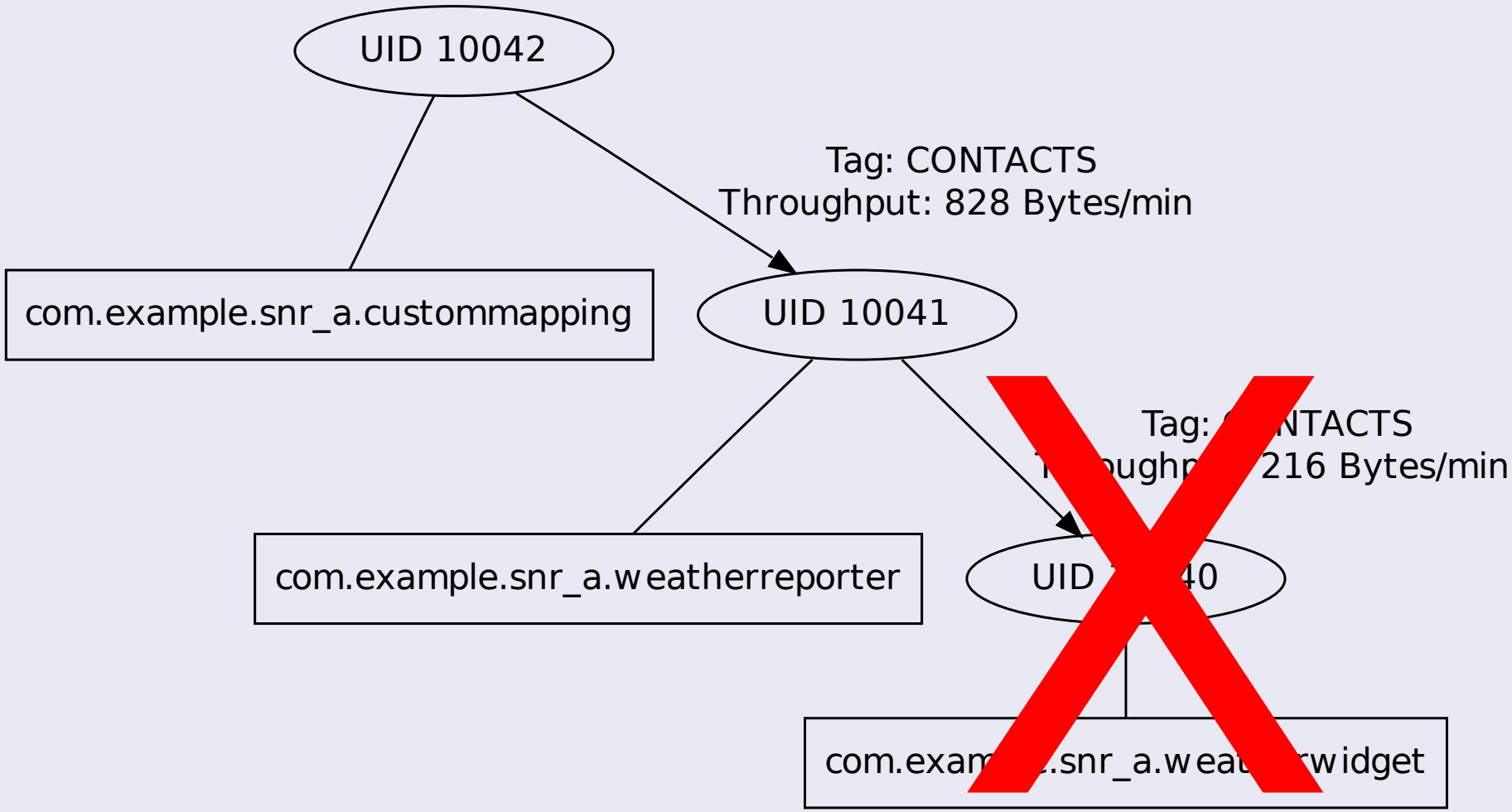
T_3



Scenario: Conspiring apps

Execution

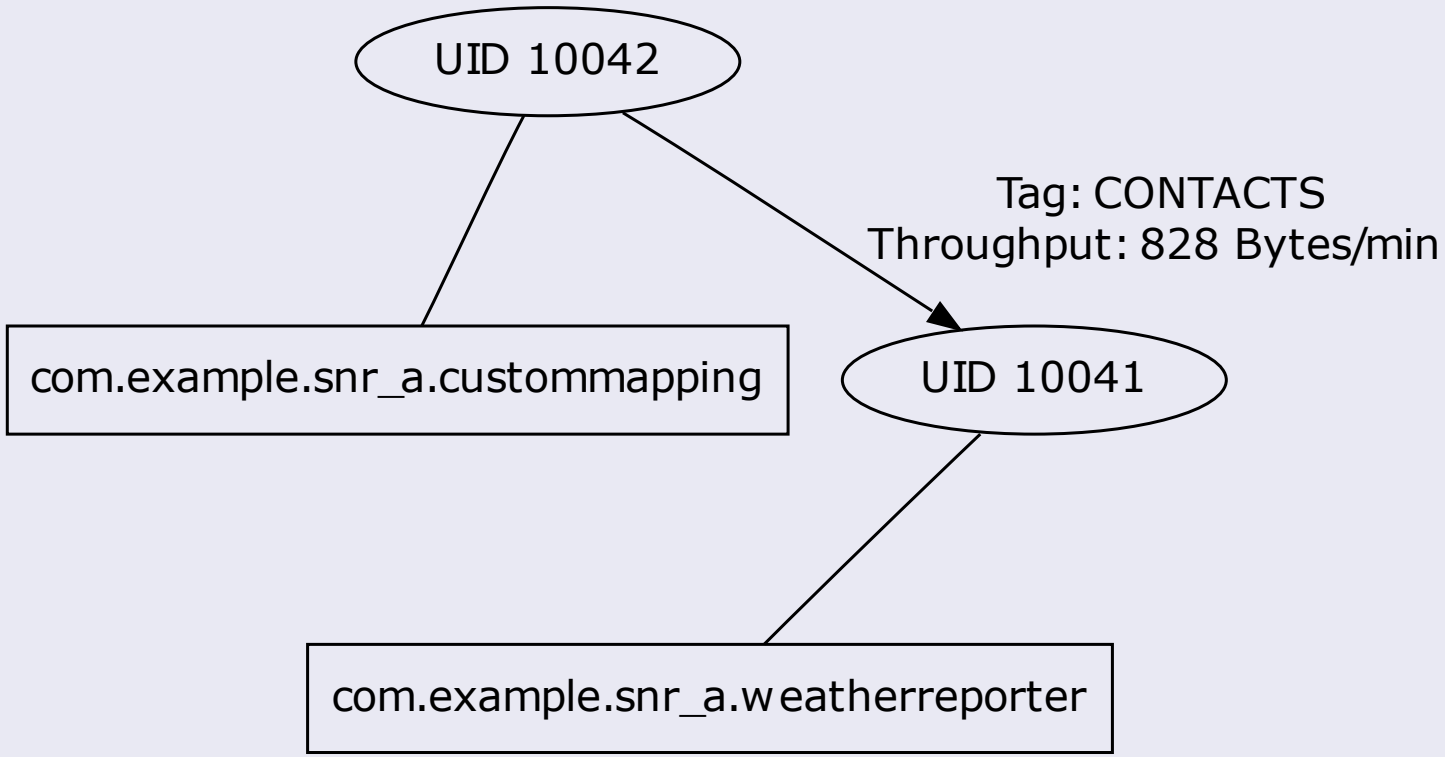
T_3 to T_4



Scenario: Conspiring apps

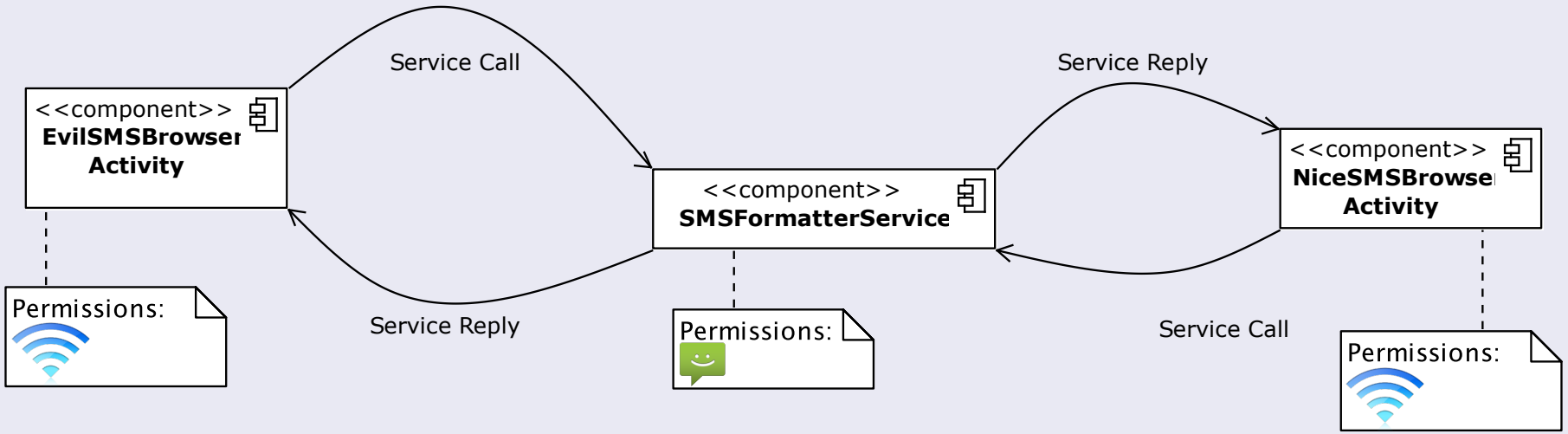
Execution

T_4



Scenario: Confused-deputy

Attack scenario: *confused-deputy*



Objective

SMS theft due to insecure/open API.

Execution

See our paper.

Conclusion & Outlook

Conclusion

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

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

[1] R. Schlegel, K. Zhang, X. yong Zhou, M. Intwala, A. Kapadia, and X. Wang, "Soundcomber: A Stealthy and Context-Aware Sound Trojan for Smartphones," in *NDSS*, The Internet Society, 2011.

[2] R. Templeman, Z. Rahman, D. J. Crandall, and A. Kapadia, "PlaceRaider: Virtual Theft in Physical Spaces with Smartphones," *CoRR*, vol. abs/1209.5982, 2012.

[3] Z. R. Mednieks *et al.*, *Programming Android: Java programming for the new generation of mobile devices*. O'Reilly & Associates, Inc., second ed., 2012.

[4] S. Bugiel, L. Davi, A. Dmitrienko, T. Fischer, and A.-R. Sadeghi, "XManDroid: A New Android Evolution to Mitigate Privilege Escalation Attacks," Technical Report TR-2011-04, Technische Universität Darmstadt, Apr. 2011.

[5] A. P. Felt, H. J. Wang, A. Moshchuk, S. Hanna, and E. Chin, "Permission re-delegation: attacks and defenses," in *Proceedings of the 20th USENIX conference on Security*, SEC'11, (Berkeley, CA, USA), pp. 22–22, USENIX Association, 2011.

[6] W. Enck, P. Gilbert, B. gon Chun, L. P. Cox, J. Jung, P. McDaniel, and A. Sheth, "TaintDroid: An Information-Flow Tracking System for Realtime Privacy Monitoring on Smartphones," in *OSDI* (R. H. Arpaci-Dusseau and B. Chen, eds.), pp. 393–407, USENIX Association, 2010.