What is Android Colluded Applications Attack and How to Detect It?

Igor Khokhlov, Leon Reznik
ixk8996@rit.edu, lr@cs.rit.edu
Rochester Institute of Technology
Rochester, NY

This research is partially based upon work supported by the NSF under Award # ACI-1547301 and NSA under Award # H98230-17-1-0200
Content

- Data Quality and Security in real life
- Android security mechanisms
- Overt communication channel
  - Overview
  - Attack scenario
  - Attack analysis
- Covert communication channel
  - Overview
  - Attack scenario
  - Attack analysis
- Colluded application attack detection
- Conclusion
Data Quality

Old data collection model

From a scientist to a scientist

Quality Data

Modern data collection model

Citizen science

Internet of Things

What is data quality?
Data Quality

How do we do it?

Our Solution:

A Cyclic Distributed Hierarchical Framework for Data Quality Evaluation and Assurance
Data Quality

- Data Quality
- Data Trustworthiness
- User’s Privacy
- Communication security
- Device Security
- Colluded Applications
- Channel type
- Accuracy
- Freshness
- Noise
What is application collusion?

Colluded applications – are collaborating applications that can bypass permission restrictions through communicating with each other.

Applications can communicate with each other either through overt communication channel or covert communication channel.
Hypothesis

Colluded applications may create distinctive patterns in the memory consumption and CPU usage signals.
Typical Android Architecture

- Application layer
- Android Runtime (ART) executes Java code
- (HAL) provides standard interfaces of hardware components.
- Native C/C++ Libraries layer contains high performance libraries.
- Linux kernel is the basic layer that communicates with platform hardware and sensors.

Colluded applications: violation of major security mechanisms

Sandboxing

Application UID 1

Application UID 2

Permissions

In order to use device's resources, an application should ask for a permission.
Colluded applications: violation of major security mechanisms

Sandboxing

Permissions

In order to use device’s resources, an application should ask for a permission

Application UID 1
Application UID 2
Colluded applications: violation of major security mechanisms

Sandboxing

Permissions

In order to use device’s resources, an application should ask for a permission
Overt communication channel

Overt communication is used for explicit data transmission between installed applications.
Overt communication channel: 
Explicit Intent

Application A

Explicit intent

Launch service of the application B

Data to transmit

Intent

Process received intent and data

Application B
Overt communication channel: Implicit Intent

Application A

Explicit intent

Perform some action

Data to transmit

Intent

Process received intent and data
Application B

Process received intent
Application C
Overt communication channel: Implicit Intent
Overt communication channel: Attack scenario

Explicit Intent

Implicit Intent
Colluded Applications Definition

\[(A, B \in S) \land (P_{DA}, P_{DB} \subseteq DP) \land P_{DA} \neq P_{DB} \land (p_{D} \in P_{DA}) \land (p_{D} \notin P_{DB}) \land (p_{L} \in P_{DB}) \land (p_{L} \notin P_{DA}) \land t_{A}(B, D_{p_{D}}, background) \rightarrow A \text{ and } B \text{ are colluded}\]
Initial Experiment Description

- Device: Google Nexus 4
- Android version 5.1
- Colluded applications do not follow up normal procedures for retrieving user’s data, which commonly have to request permission for data acquisition
- Colluded application transfer data using Android OS services
- Transmitted data: 300 MB of user’s data
- Chrome web-browser runs at the background
Overt communication channel: Attack analysis – no attacks
Overt communication channel: Attack analysis – 1 attack at a time
Overt communication channel: Attack analysis – 3 attacks simultaneously
Overt communication channel: Attack analysis - comparison
Covert communication channel

Covert inter-application communication creates a capability to transfer data between applications that are not supposed to be allowed to communicate.
Covert communication channel: Time based
Covert communication channel: Time based – attack analysis

- Minimal time interval between requests is 1ms;
- 125 bytes per second – expected to be used for small amount of data
- A device cannot go into a sleep mode
- We have not detected patterns in the CPU usage
Covert communication channel: Storage based
Conclusion

- Colluded applications can bypass permissions and cause leak of a private information
- Time-based covert channel is not expected to be used for communicating big amounts of data
- Transferring big amounts of data through Intents creates distinguishing patterns in memory consumption and CPU usage
- These patterns can be used for application collusion detection in a real-time
More information?

- Download our apps from Google Play
- https://play.google.com/store/apps/details?id=dataqualitylab.rit.ver_app_finder and more are coming
- Watch our webinar: https://youtu.be/nkp0kvJvTWw
- Take a look at our publications (next slide)
- And yes, we are developing the project website
- Contact us

Leon Reznik, Igor Khokhlov
Department of Computer Science
Rochester Institute of Technology
email: lr@cs.rit.edu, ixk8996@rit.edu
Publications


What is Android Colluded Applications Attack and How to Detect It?

Igor Khokhlov, Leon Reznik
ixk8996@rit.edu, lr@cs.rit.edu
Rochester Institute of Technology
Rochester, NY

This research is partially based upon work supported by the NSF under Award # ACI-1547301 and NSA under Award # H98230-17-1-0200