Preserving User Privacy from Third-party Applications in Online Social Networks

Yuan Cheng, Jaehong Park and Ravi Sandhu
Institute for Cyber Security
University of Texas at San Antonio

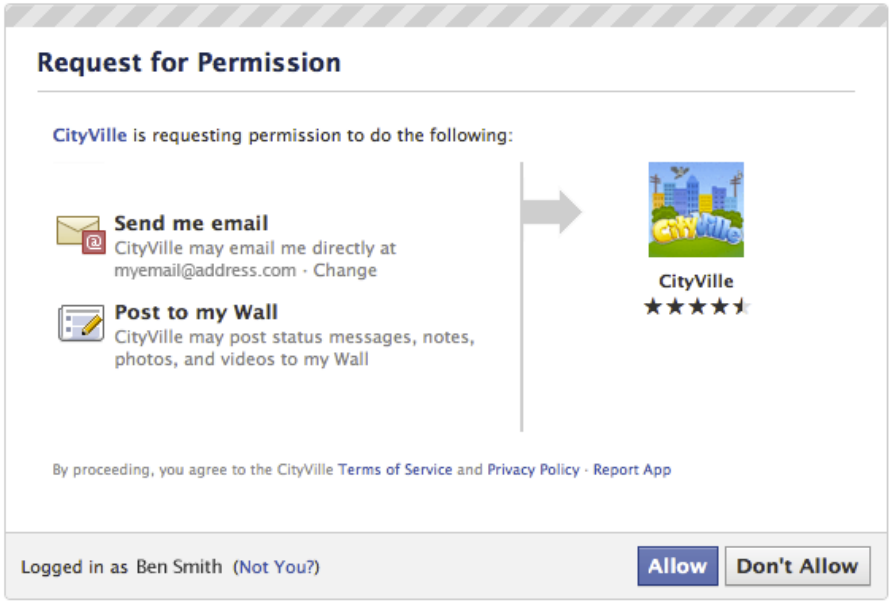
Presentation at PSOSM13, Rio de Janeiro, Brazil
May 14, 2013

Agenda

- Privacy Issues of 3rd-party Apps
- Countermeasures
- Access Control Framework
- Policy Model
- Conclusions

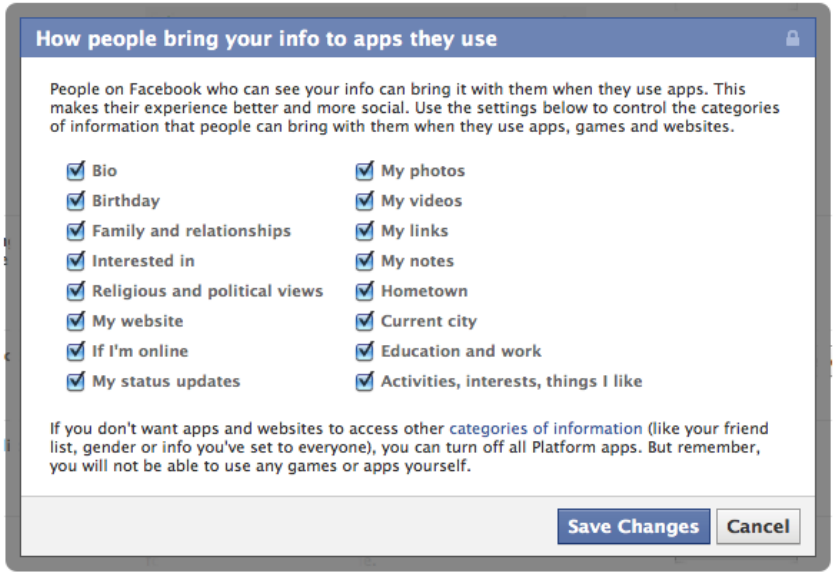
Privacy Issues

- An all-or-nothing policy for application-to-user interactions
  - User has to grant the app *full* access, even if the app only needs partial data
- Users are not aware of the application's real needs



Privacy Issues (cont.)

- Coarse-grained opt-in/out privacy control does not let user specify policies for each piece of data
- Some permissions are given by user's friend who installed the app, without user's knowledge



Countermeasures

	Summary	Pros	Cons
Data Generalization	Convert private data to a privacy-nonsensitive form	Have been widely accepted in recent solutions	
User-specified Privacy Preference	Allow user to express their preference more flexibly		
Communication Interceptor	Intercept requests, exert user preferences, and return sanitized or dummy data		Lose functionality and integrity
Information Flow Control	Confine app execution and mediate information flow	Enable post-authorization	Need substantial modification to current architecture
User-to-application Policy Model	Provide a complete policy model for users to define, use and manage their own policies		

Goal

- Protect inappropriate exposure of users' private information to untrusted 3rd party apps

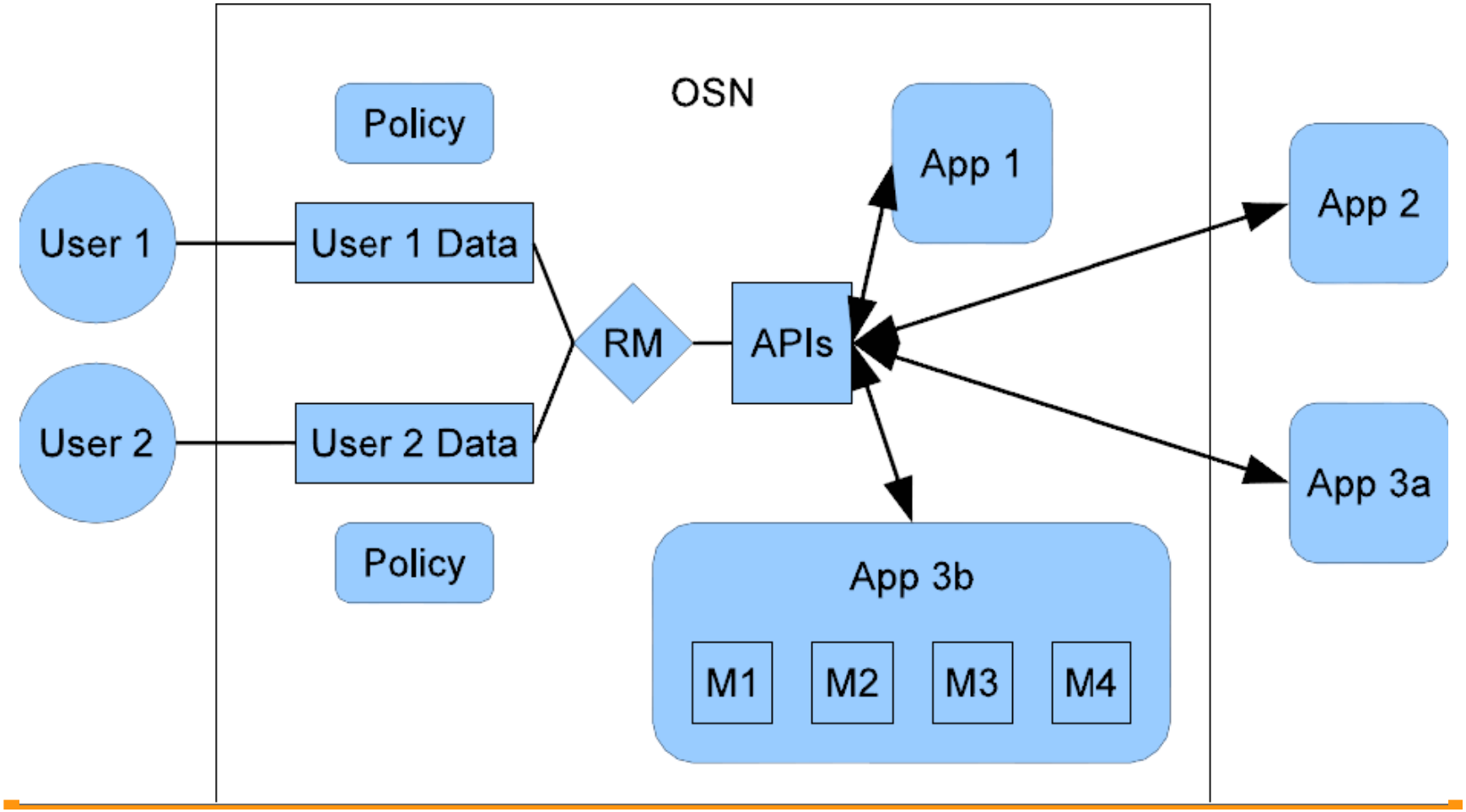
- Propose an policy model for controlling application-to-user activities
  - More flexible
    - further utilize the relationships and the social graph in OSN
  - Finer grained
    - e.g., per resource vs. per resource type, distinction of different types of access

Framework Overview

- Prevent applications from learning user's private information while still maintaining the functionality
- Leave private information within OSN system and allow external servers of applications to retrieve non-private data

Data Classification	Strategy
unnecessary & private	do not permit
unnecessary & non-sensitive	user's choice
essential & non-sensitive	transmittable outside of OSN
essential & private	processable within OSN

Proposed Architecture



Application Components

- Internal component
  - High trustworthy; can handle private data
  - Can be provided by OSN and 3rd-party entities
- External component
  - Provided by 3rd-party entities
  - Low trustworthy; cannot consume private data

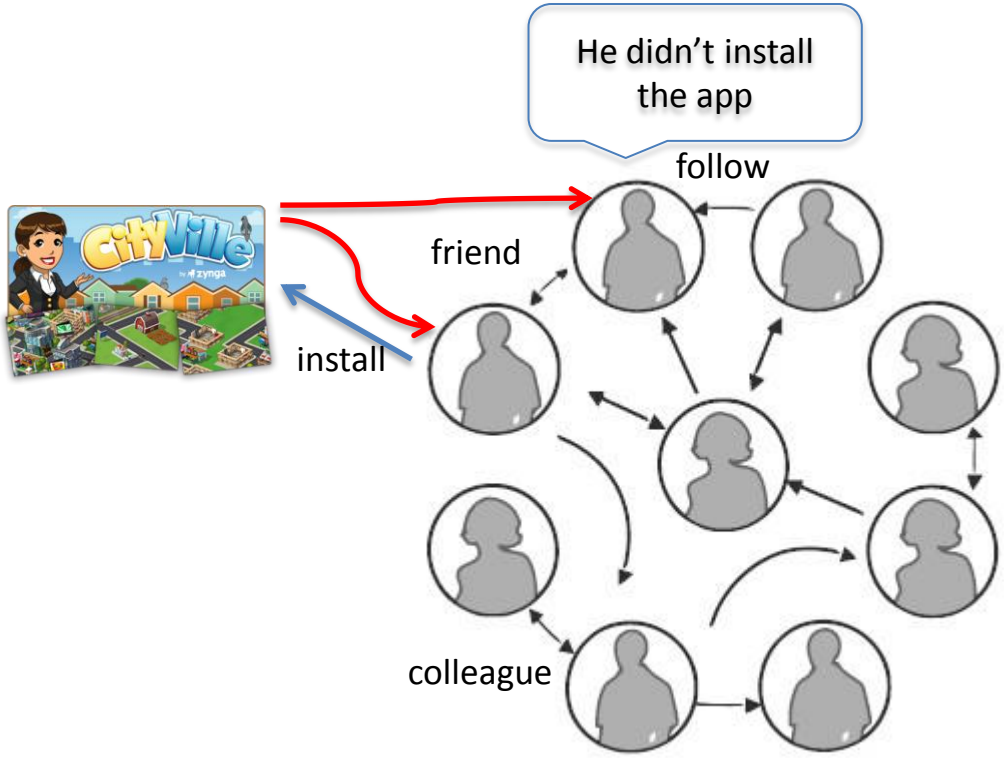
Communications

	OSN provided	3rd-party provided
Communication w/ system calls	M1	M2
Communication w/ non-private data	M3	M4

Communication between components only through OSN-specified APIs

- Communication w/ system calls
- Communication w/ non-private data
- Communication w/ private data (not allowed)

Relationship-based Access Control w/ Apps



Policy Specifications

- *<action, target, (start, path rule), $2^{ModuleType}$>*
  - *action* specifies the type of access
  - *target* indicates the resource to be accessed
  - *start* is the position where access evaluation begins, which can be either *owner* or *requester*
  - *path rule* represents the required pattern of relationship between the involved parties

e.g., “install”, “friend·install”

Policy Specifications

- *<action, target, (start, path rule), $2^{ModuleType}$>*
  - *action* specifies the type of access
  - *target* indicates the resource to be accessed
  - *start* is the position where access evaluation begins, which can be either *owner* or *requester*
  - *path rule* represents the required pattern of relationship between the involved parties
  - *ModuleType* = {*M1, M2, M3, M4, external*}, $2^{ModuleType}$ indicates the set of app module types allowed to access

Example: App Request Notification

- <app request, _, (target user, install), {M1, M2, M3, M4, external}>
  - For apps she installed; Protect her data
- <app request, _, (requester, install·friend), {M1, M2}>
  - For apps she installed ; Protect her friends' data
- <app request, _, (target user, friend·install), {M1, M2}>
  - For apps her friends installed; Protect her data

Example: Accessing User's Profile

- <access, dateofbirth, (owner, install), {M1, M2}>
  - DOB is private
- <access, keystroke, (owner, install), {external}>
  - Keystroke is non-private
  - Keystroke information is crucial for fulfilling functionality
- <access, emailaddress, (owner, friend·install), {M1, M2, M3, M4}>
  - Protect his friends' data

Conclusions

- Presented an access control framework
  - Split applications into different components with different privileges
  - Keep private data away from external components
- Provided a policy model for application-to-user policies
  - Specify different policies for different components of the same application

Q&A

Questions?

ycheng@cs.utsa.edu

http://my.cs.utsa.edu/~ycheng

Twitter: @nbycheng