Science of Human Circumvention of Security

PIs: Tao Xie (Illinois), Jim Blythe (USC),
Ross Koppel (U Penn), Sean Smith (Dartmouth)

User Expectations in Mobile App Security

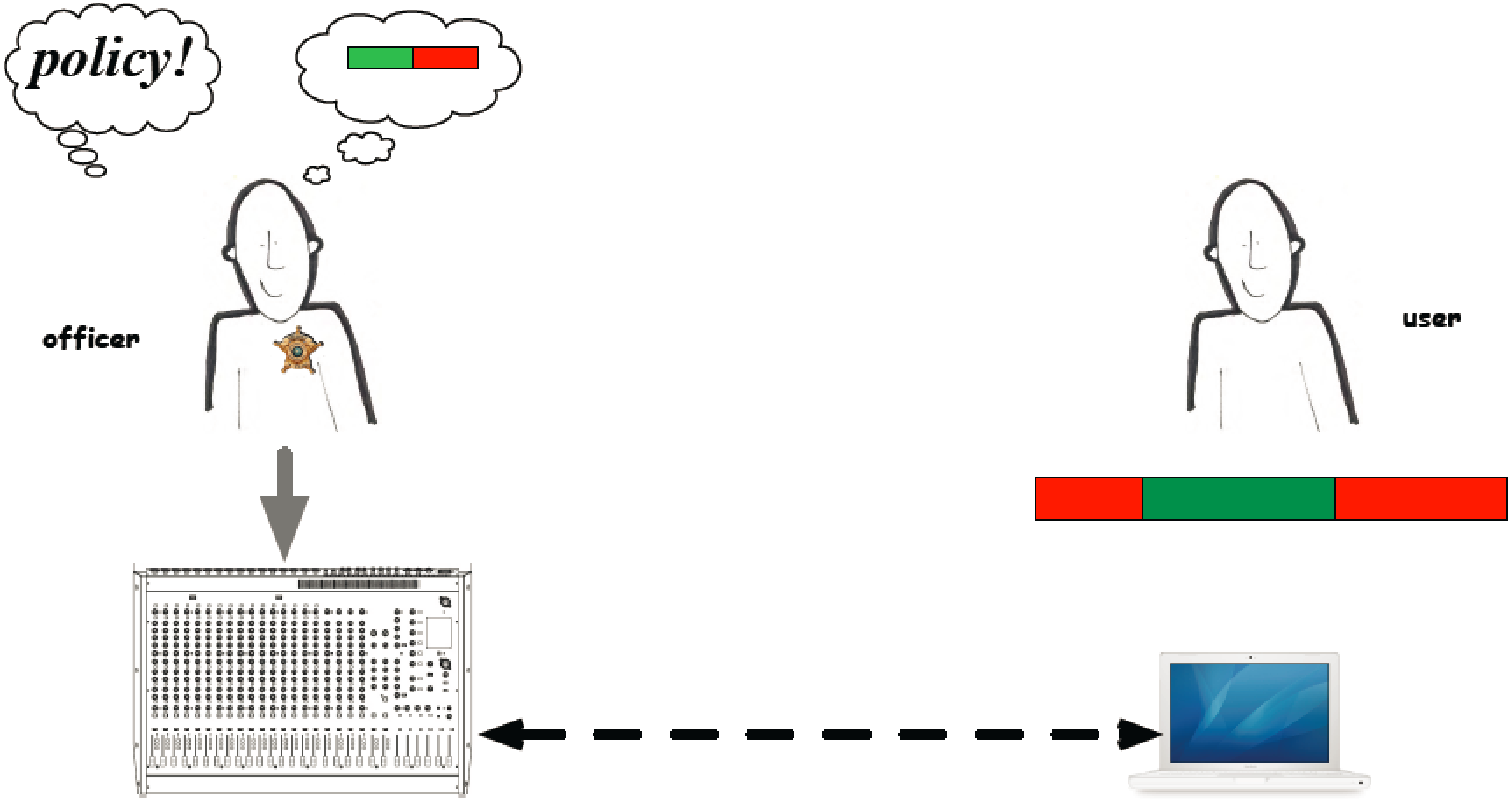
Tao Xie

Our View of Science of Security: When Human and Machine (Security Control) Meet

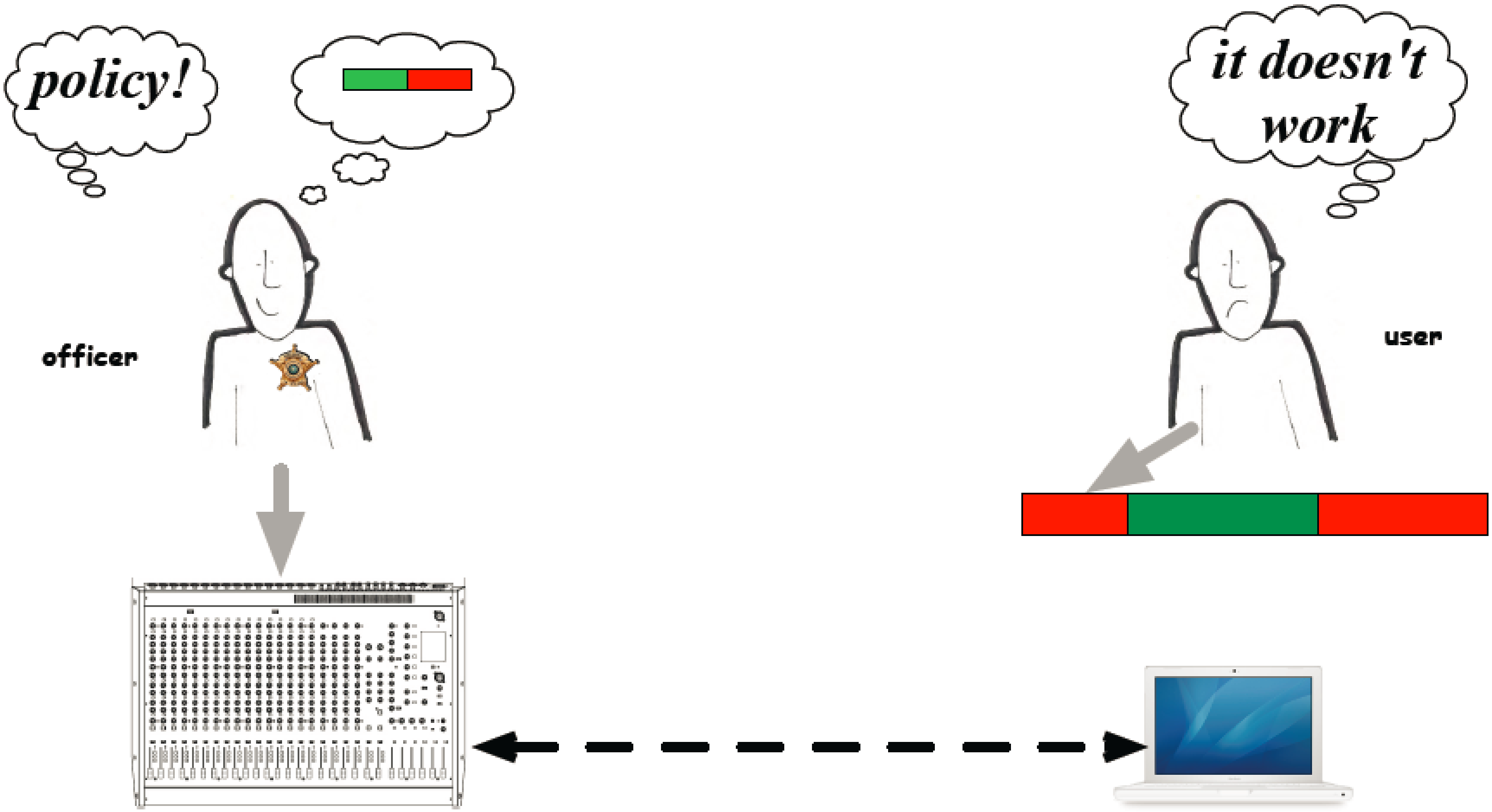
- **Assumption**: human decision on security control is perfect
- **Reality**: well-intentioned human users continually circumvent security controls or make uninformed security decision
- **Consequence**: ubiquitousness of this circumvention or uninformed decision undermines the effectiveness of security designs

- To develop metrics and mechanisms to enable stakeholders to make meaningful, quantifiable **comparisons**, **decisions**, and **evaluations** of proposed security controls *in light of what really happens when these controls are deployed*

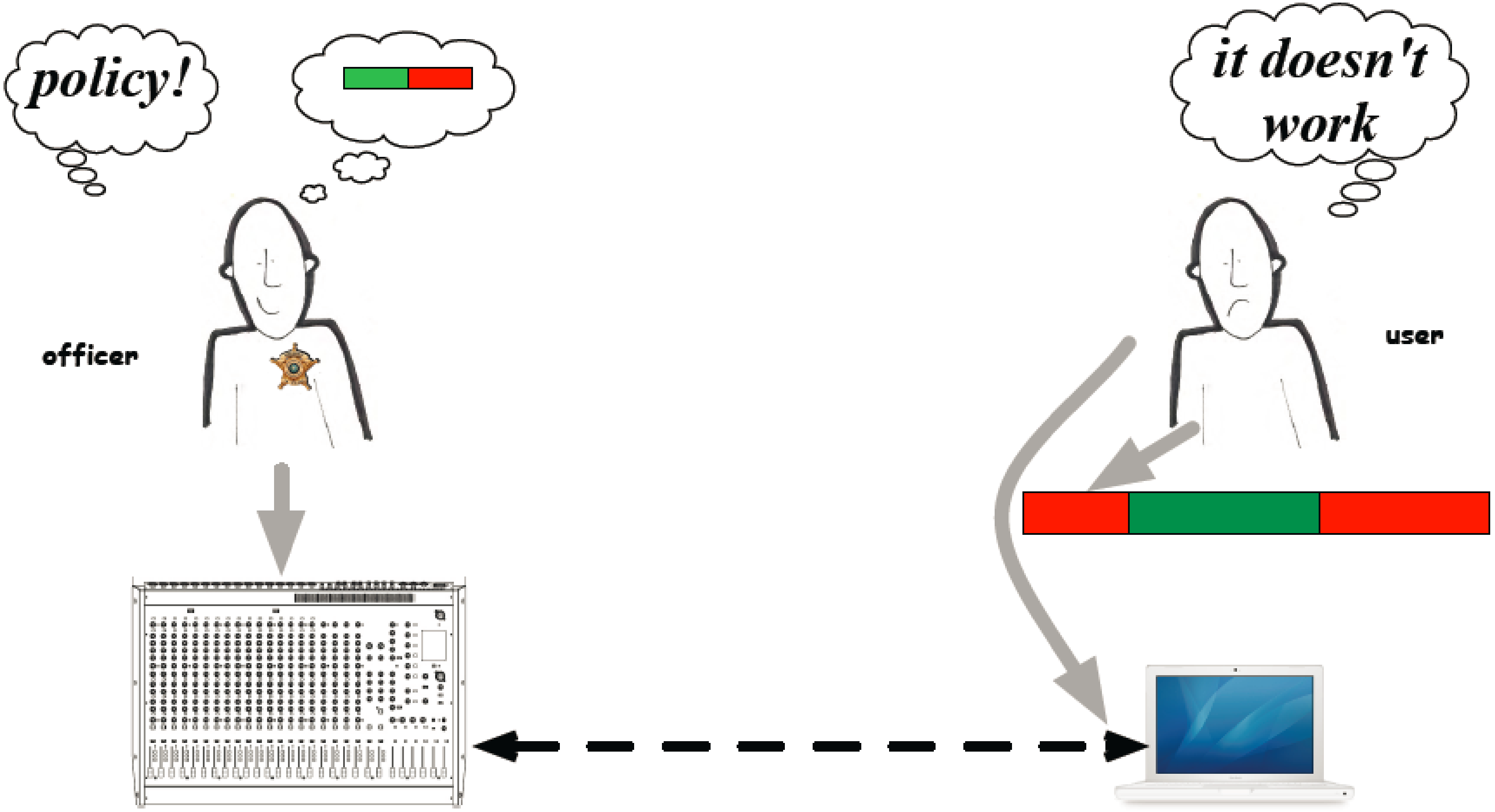
Manageability – Access Control Example



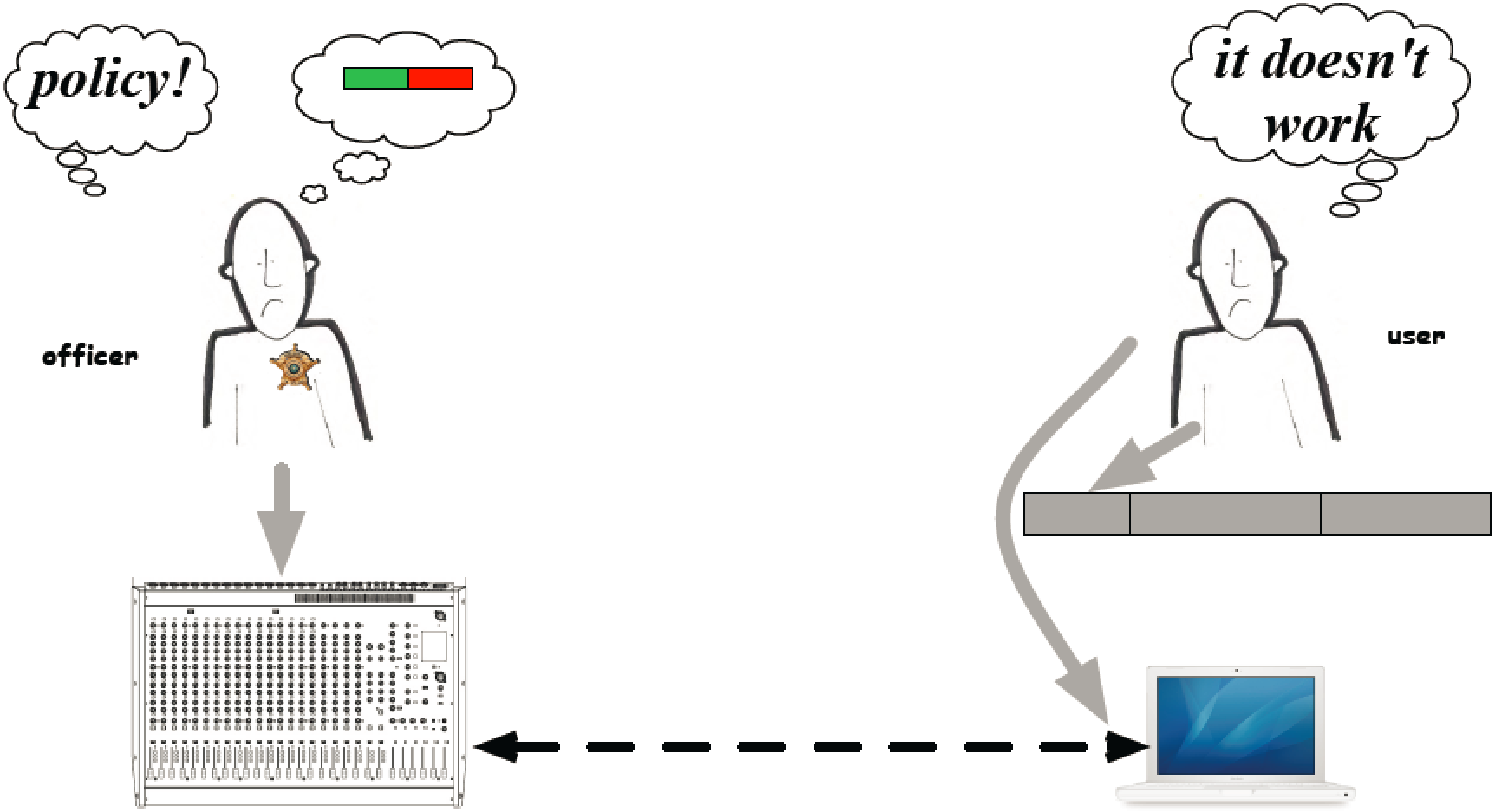
Manageability – Access Control Example



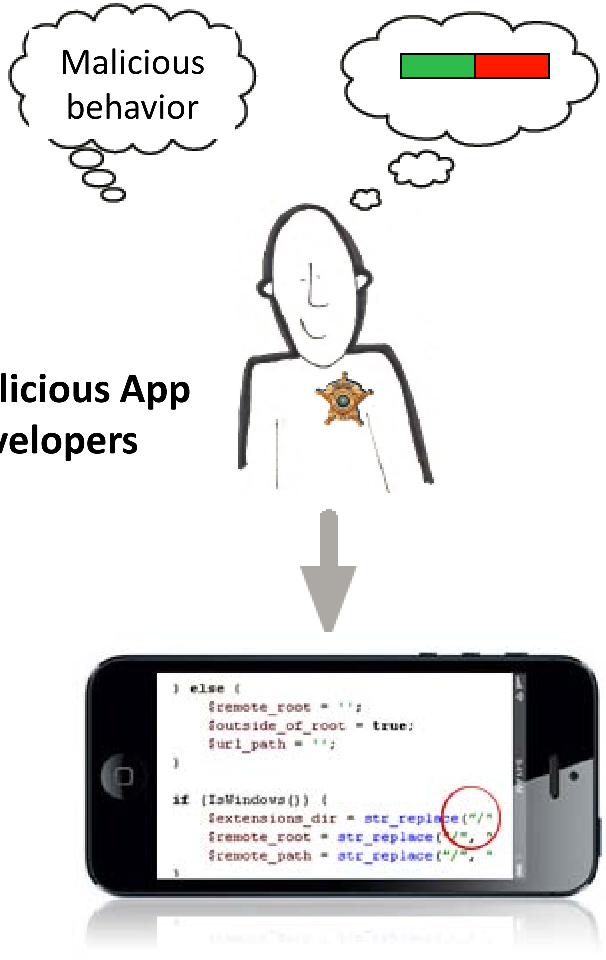
Manageability – Access Control Example



Manageability – Access Control Example



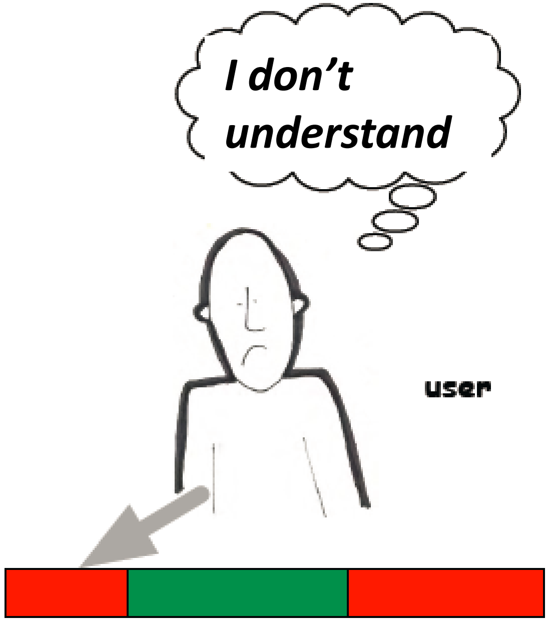
Manageability – Mobile App Permission Example

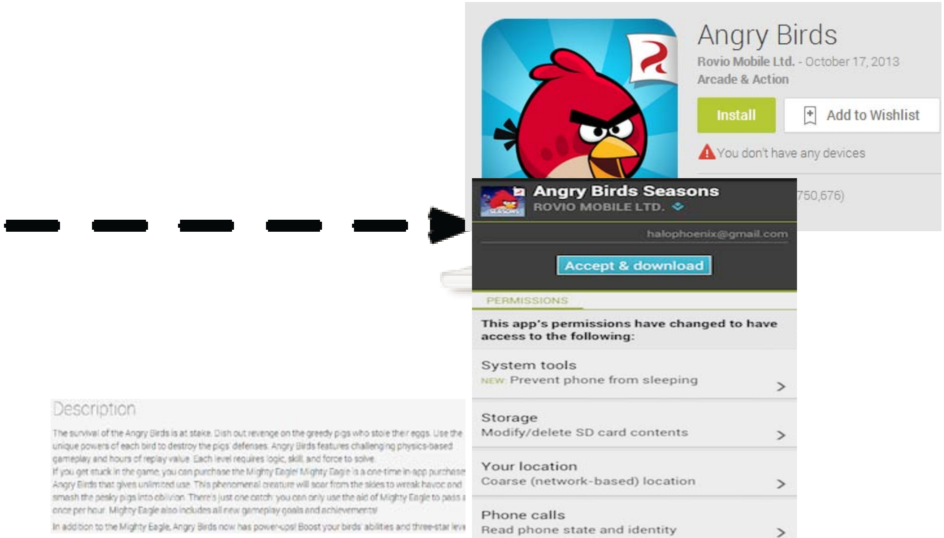


Malicious App Developers

```
} else {
    $remote_root = '';
    $outside_of_root = true;
    $url_path = '';
}

if (IsWindows()) {
    $extensions_dir = str_replace("/"
    $remote_root = str_replace("/", "
    $remote_path = str_replace("/", "
}
```




Angry Birds

Rovio Mobile Ltd. - October 17, 2013

Arcade & Action

You don't have any devices

750,676)

Angry Birds Seasons

ROVIO MOBILE LTD.

halophoenix@gmail.com

This app's permissions have changed to have access to the following:

System tools
NEW: Prevent phone from sleeping >

Storage
Modify/delete SD card contents >

Your location
Coarse (network-based) location >

Phone calls
Read phone state and identity >

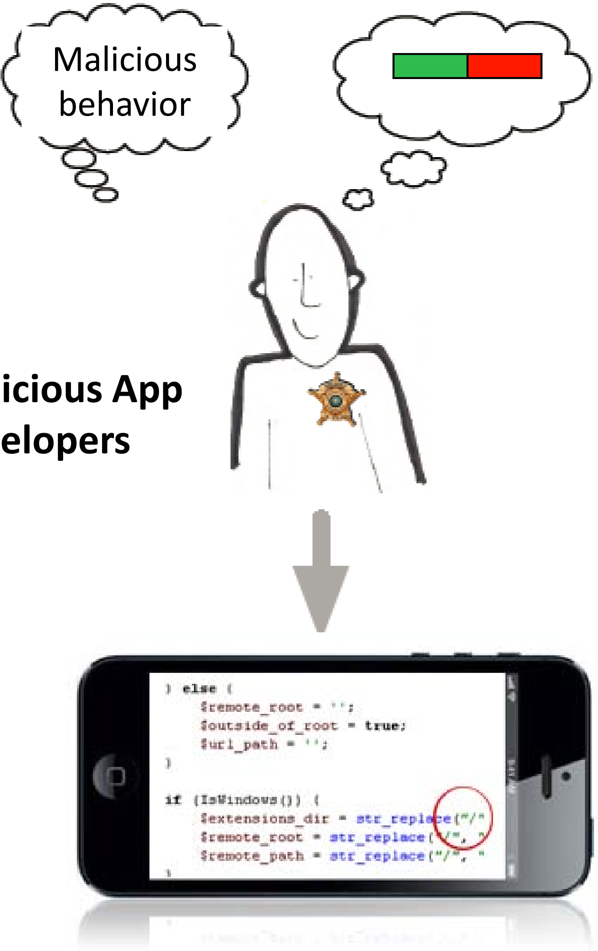
Description

The survival of the Angry Birds is at stake. Dish out revenge on the greedy pigs who stole their eggs. Use the unique powers of each bird to destroy the pigs' defenses. Angry Birds features challenging physics-based gameplay and hours of replay value. Each level requires logic, skill, and force to solve.

If you get stuck in the game, you can purchase the Mighty Eagle! Mighty Eagle is a one-time in-app purchase Angry Birds that gives unlimited use. This phenomenal creature will soar from the skies to wreak havoc and smash the pesky pigs into oblivion. There's just one catch: you can only use the aid of Mighty Eagle to pass a once per hour. Mighty Eagle also includes all new gameplay goals and achievements!

In addition to the Mighty Eagle, Angry Birds now has power-ups! Boost your birds' abilities and three-star leve

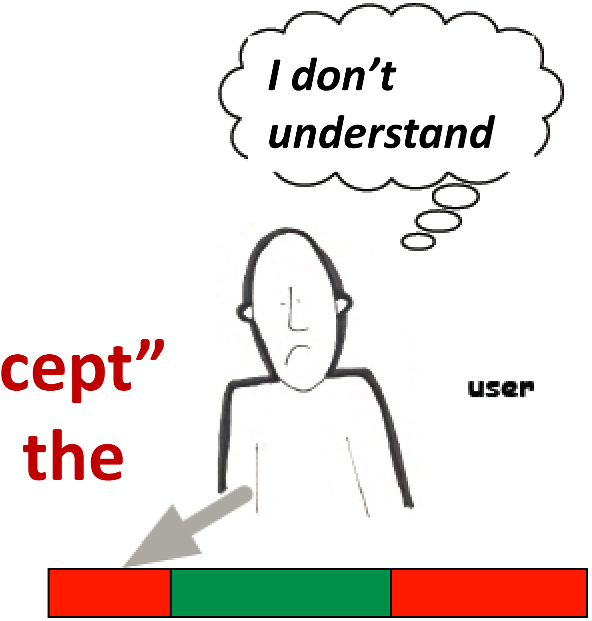
Manageability – Mobile App Permission Example



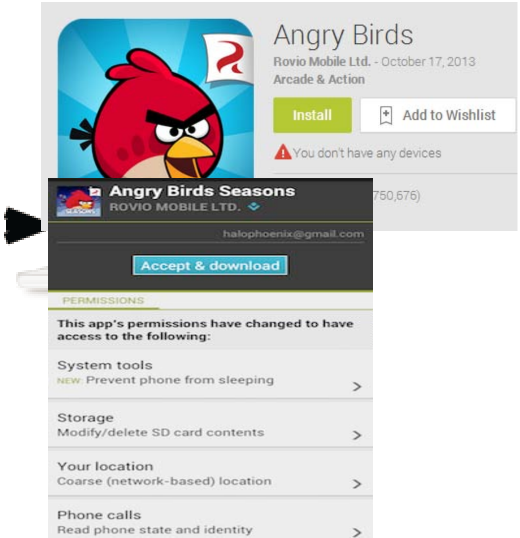
Malicious App Developers

```
} else {
    $remote_root = '';
    $outside_of_root = true;
    $url_path = '';
}

if (IsWindows()) {
    $extensions_dir = str_replace("/"
    $remote_root = str_replace("/", "
    $remote_path = str_replace("/", "
}
```



Click "Accept" to Install the App



Angry Birds
Rovio Mobile Ltd. - October 17, 2013
Arcade & Action
Install | Add to Wishlist
You don't have any devices
750,676)

Angry Birds Seasons
ROVIO MOBILE LTD.
halophoenix@gmail.com
Accept & download

PERMISSIONS

This app's permissions have changed to have access to the following:

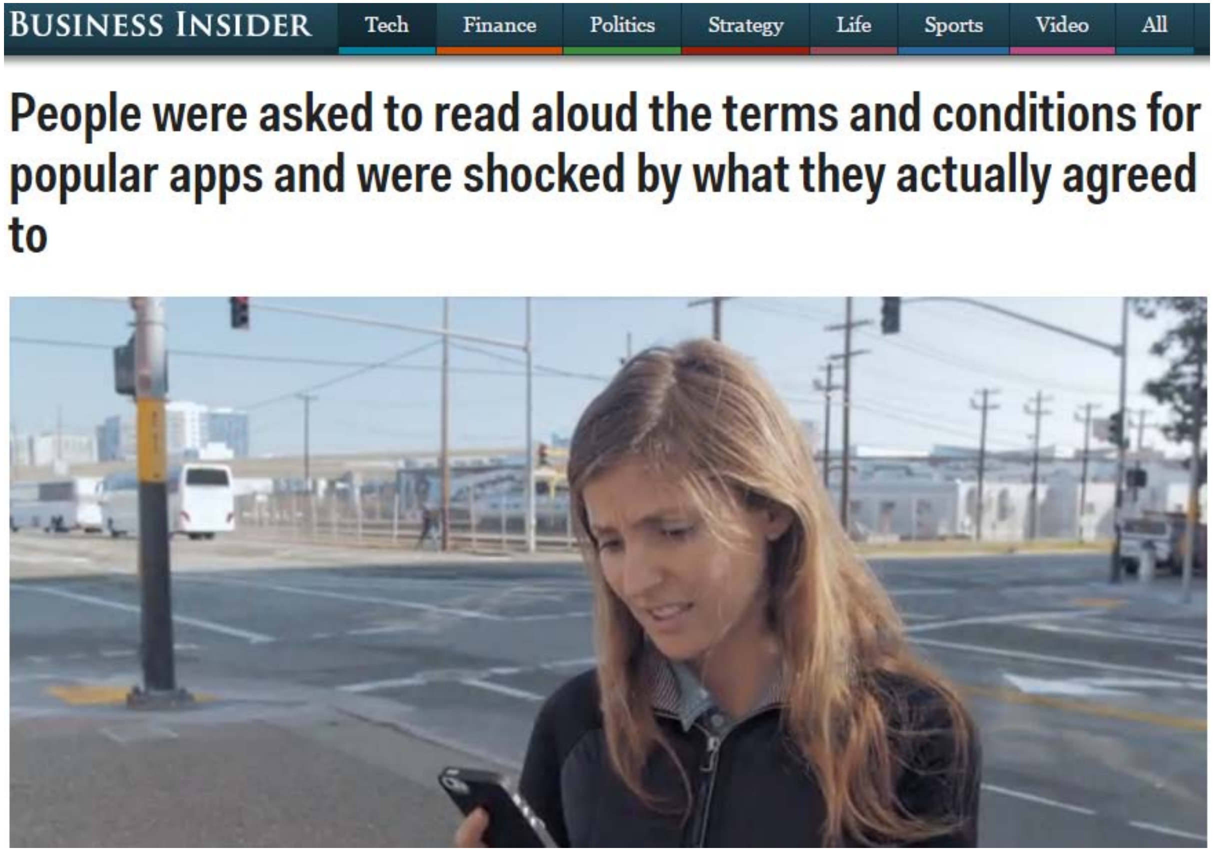
System tools
NEW: Prevent phone from sleeping >

Storage
Modify/delete SD card contents >

Your location
Coarse (network-based) location >

Phone calls
Read phone state and identity >

It is NOT that People Don’t Care

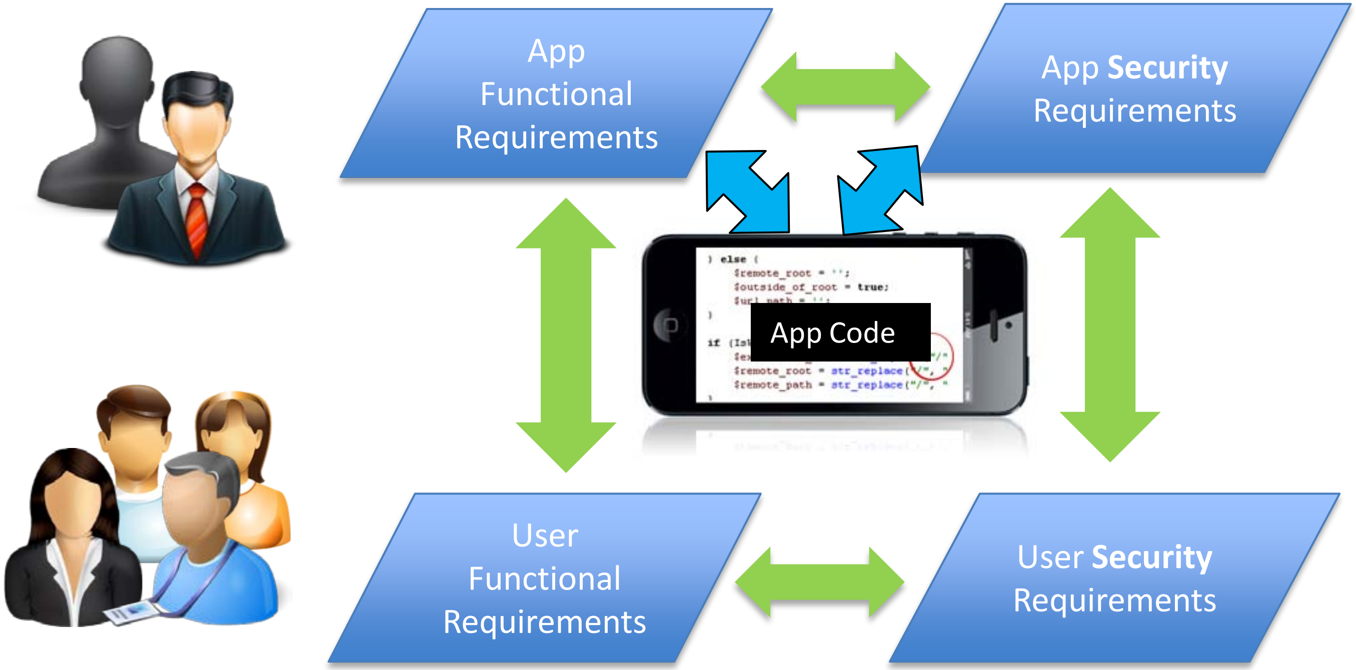


BUSINESS INSIDER Tech Finance Politics Strategy Life Sports Video All

People were asked to read aloud the terms and conditions for popular apps and were shocked by what they actually agreed to

http://www.businessinsider.com/app-permission-agreements-privacy-video-2015-2

“Conceptual” Model

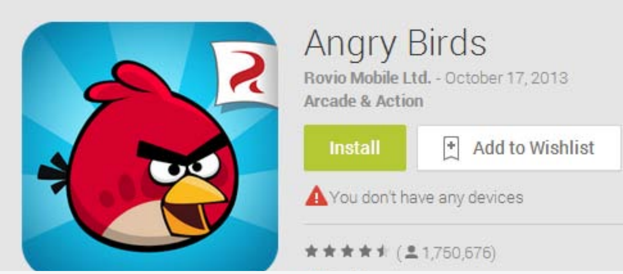

App Functional Requirements
App **Security** Requirements
App Code
User Functional Requirements
User **Security** Requirements

```
} else {
    $remote_root = '';
    $outside_of_root = true;
    $url_path = '';
}

if (Is
    $ex                          "/"
    $remote_root = str_replace("/", "
    $remote_path = str_replace("/", "
}
```

User Expectation: User Perception + User Judgment

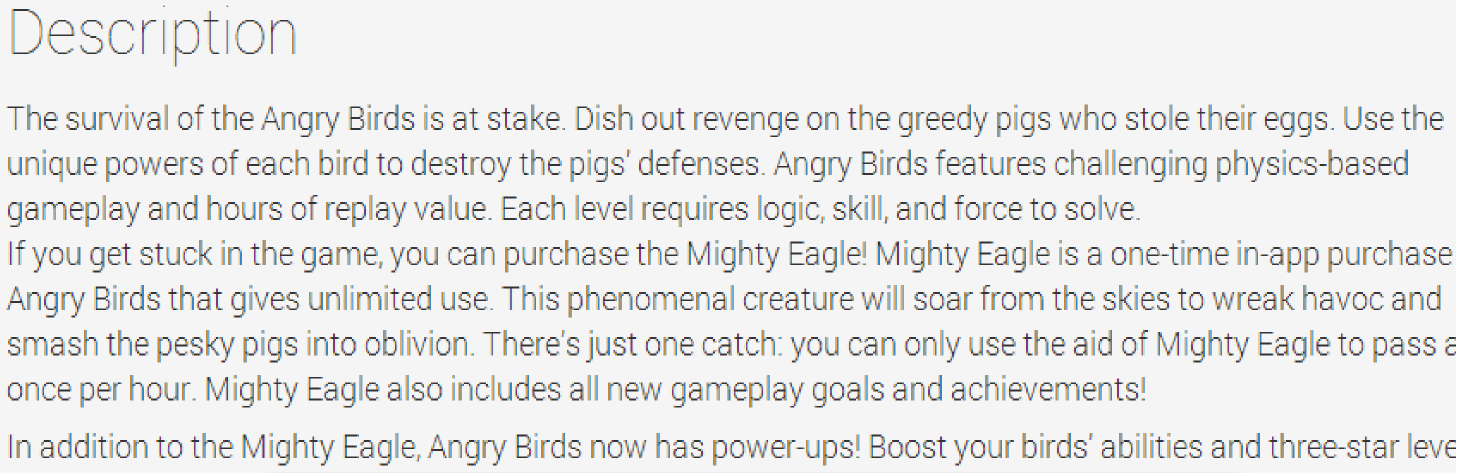
Informal App Functional Requirements: App Description



Angry Birds

Rovio Mobile Ltd. - October 17, 2013

Arcade & Action



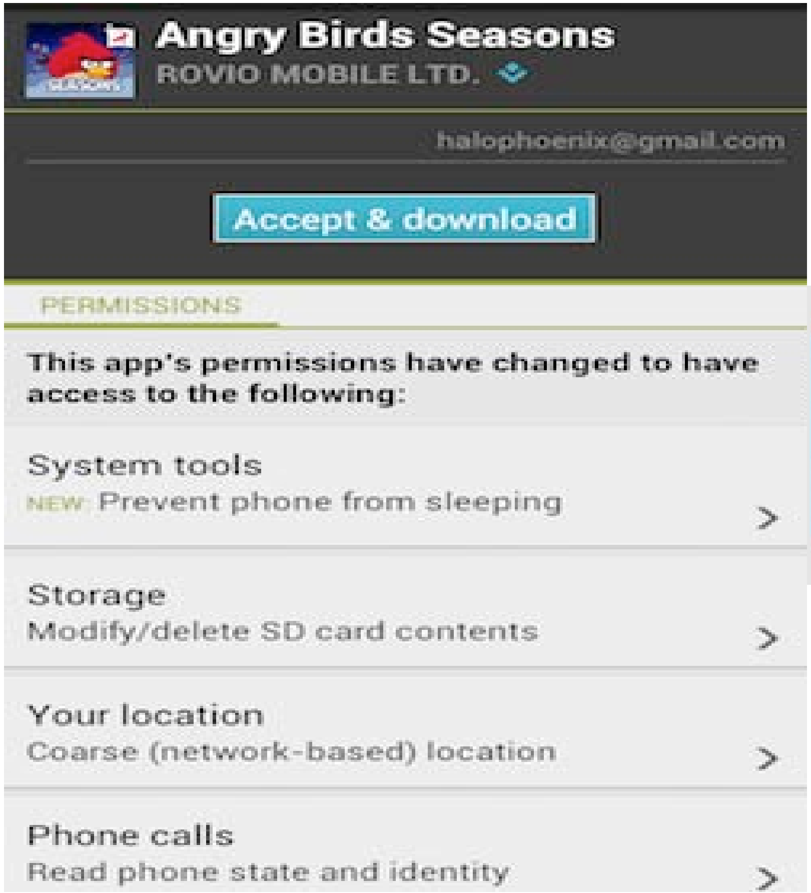
Description

The survival of the Angry Birds is at stake. Dish out revenge on the greedy pigs who stole their eggs. Use the unique powers of each bird to destroy the pigs' defenses. Angry Birds features challenging physics-based gameplay and hours of replay value. Each level requires logic, skill, and force to solve.

If you get stuck in the game, you can purchase the Mighty Eagle! Mighty Eagle is a one-time in-app purchase Angry Birds that gives unlimited use. This phenomenal creature will soar from the skies to wreak havoc and smash the pesky pigs into oblivion. There's just one catch: you can only use the aid of Mighty Eagle to pass a once per hour. Mighty Eagle also includes all new gameplay goals and achievements!

In addition to the Mighty Eagle, Angry Birds now has power-ups! Boost your birds' abilities and three-star leve

App Security Requirements: Permission List



Angry Birds Seasons

ROVIO MOBILE LTD.

halophoenix@gmail.com

PERMISSIONS

This app's permissions have changed to have access to the following:

System tools

NEW Prevent phone from sleeping

Storage

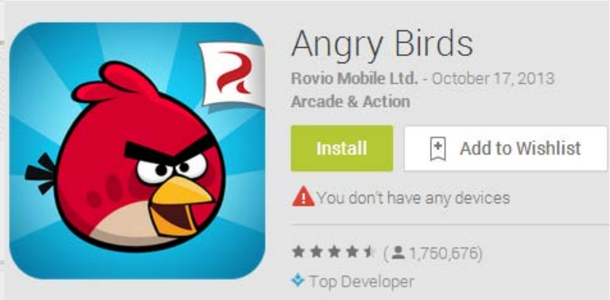
Modify/delete SD card contents

Your location

Coarse (network-based) location

Phone calls

Read phone state and identity



Angry Birds

Rovio Mobile Ltd. - October 17, 2013

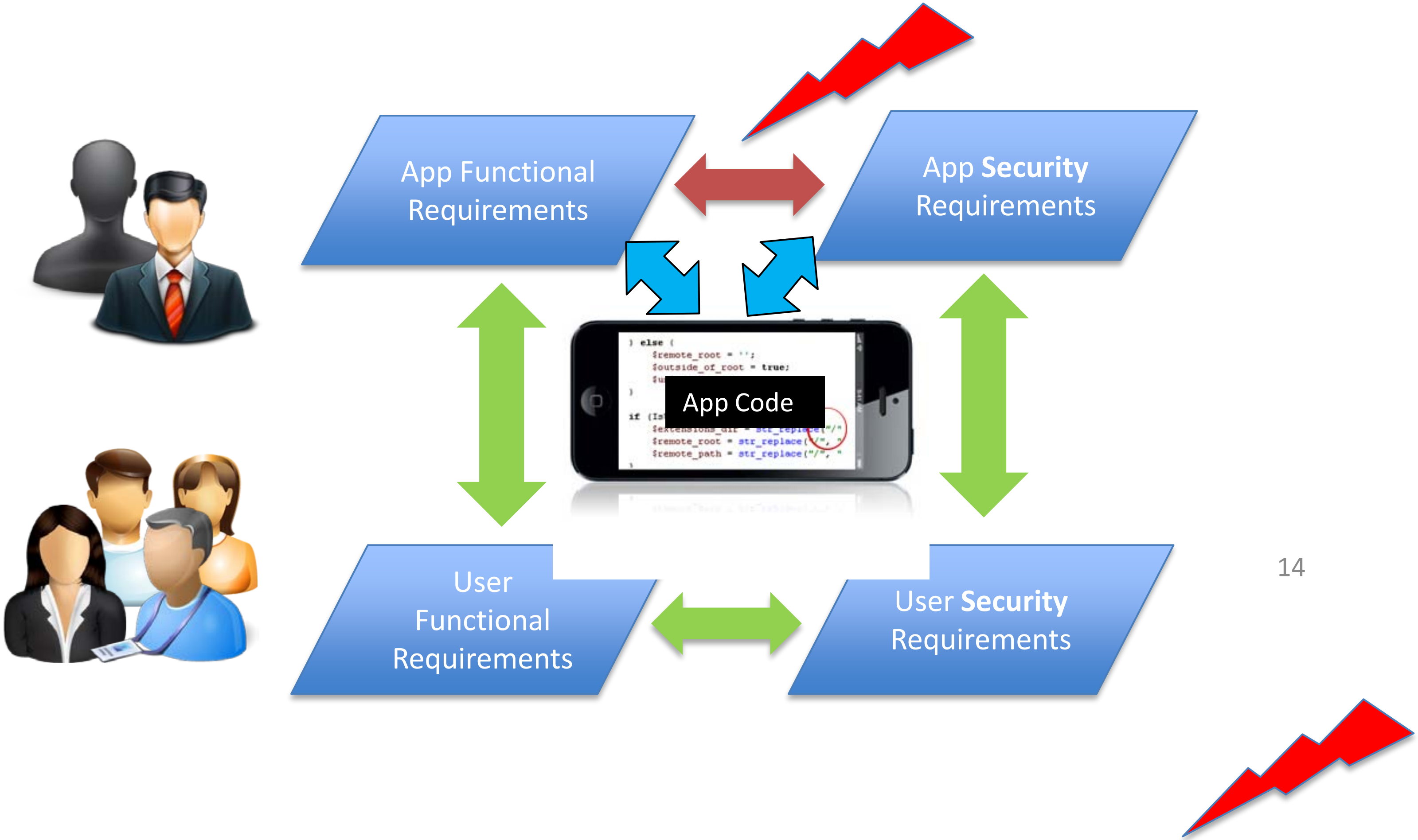
Arcade & Action

You don't have any devices

★★★★½ (1,750,676)

Top Developer

“Conceptual” Model

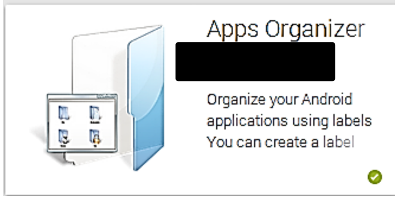

App Functional Requirements
App **Security** Requirements
App Code
User Functional Requirements
User **Security** Requirements

User Expectation: User Perception + **User Judgment**

WHYPER: Text Analytics for Mobile Security

- Focus on permission ←→ app descriptions
  - permissions (protecting user understandable resources) should be discussed
- ***What does the users expect (w.r.t. app functionalities)?***
  - **GPS Tracker:** record and send location
  - **Phone-Call Recorder:** record audio during phone call

App Description Sentence



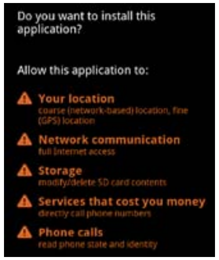
Apps Organizer

[redacted]

Organize your Android applications using labels You can create a label

Linkage

Permission



Do you want to install this application?

Allow this application to:

⚠ Your location
coarse (network-based) location, fine (GPS) location

⚠ Network communication
full Internet access

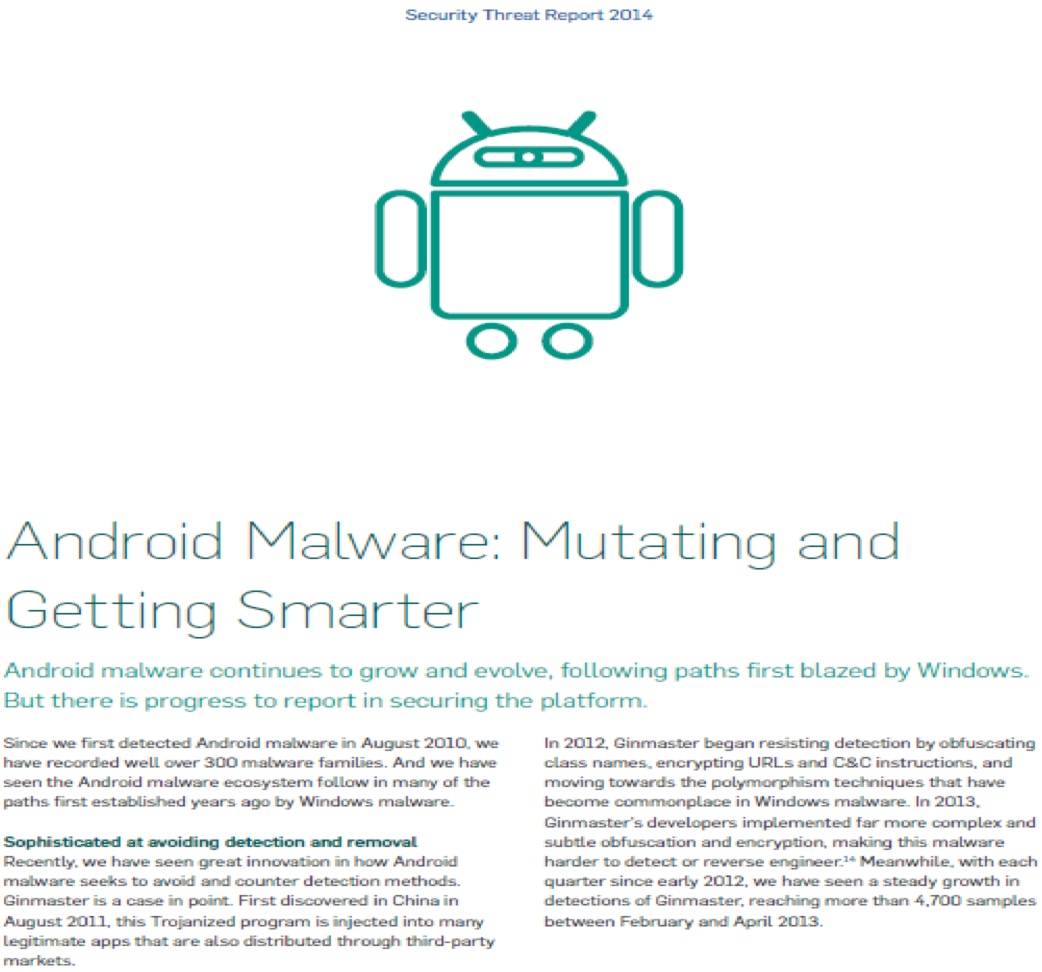
⚠ Storage
modify/delete SD card contents

⚠ Services that cost you money
directly call phone numbers

⚠ Phone calls
read phone state and identity

Pandita et al. WHYPER: Towards Automating Risk Assessment of Mobile Applications. *USENIX Security 2013*
http://taoxie.cs.illinois.edu/publications/usenixsec13-whyper.pdf

Not All Malware Developers Are "Dumb" or "Lazy"



Security Threat Report 2014

Android Malware: Mutating and Getting Smarter

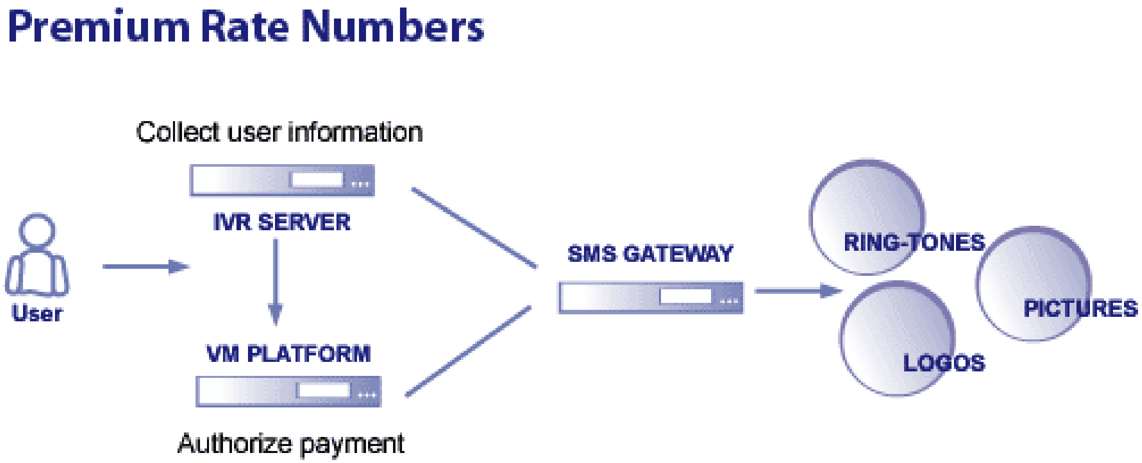
Android malware continues to grow and evolve, following paths first blazed by Windows. But there is progress to report in securing the platform.

Since we first detected Android malware in August 2010, we have recorded well over 300 malware families. And we have seen the Android malware ecosystem follow in many of the paths first established years ago by Windows malware.

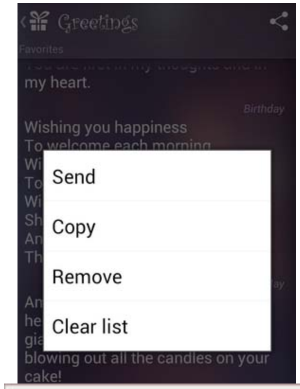
Sophisticated at avoiding detection and removal
Recently, we have seen great innovation in how Android malware seeks to avoid and counter detection methods. Ginmaster is a case in point. First discovered in China in August 2011, this Trojanized program is injected into many legitimate apps that are also distributed through third-party markets.

In 2012, Ginmaster began resisting detection by obfuscating class names, encrypting URLs and C&C instructions, and moving towards the polymorphism techniques that have become commonplace in Windows malware. In 2013, Ginmaster's developers implemented far more complex and subtle obfuscation and encryption, making this malware harder to detect or reverse engineer.[14] Meanwhile, with each quarter since early 2012, we have seen a steady growth in detections of Ginmaster, reaching more than 4,700 samples between February and April 2013.

Example Malicious App


Premium Rate Numbers

Benign? Malicious?



Greetings

Favorites

my heart.

Birthday

Wishing you happiness
To welcome each morning

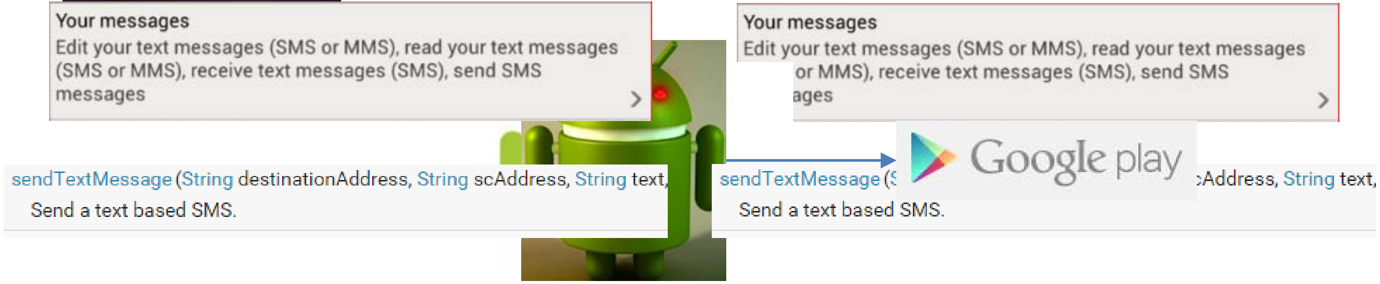
Send

Copy

Remove

Clear list

blowing out all the candles on your cake!



Your messages
Edit your text messages (SMS or MMS), read your text messages (SMS or MMS), receive text messages (SMS), send SMS messages

sendTextMessage (String destinationAddress, String scAddress, String text,
Send a text based SMS.

Your messages
Edit your text messages (SMS or MMS), read your text messages
or MMS), receive text messages (SMS), send SMS
ages

sendTextMessage (S :Address, String text,
Send a text based SMS.

Our Insight

Different goals of benign apps vs. malware.

- Benign apps
  - Meet requirements from users (as delivering utility)

- Malware
  - Trigger malicious behaviors frequently (as maximizing profits)
  - Evade detection (as prolonging lifetime)

User Expectation: User Perception + User Judgment

Differentiating Characteristics

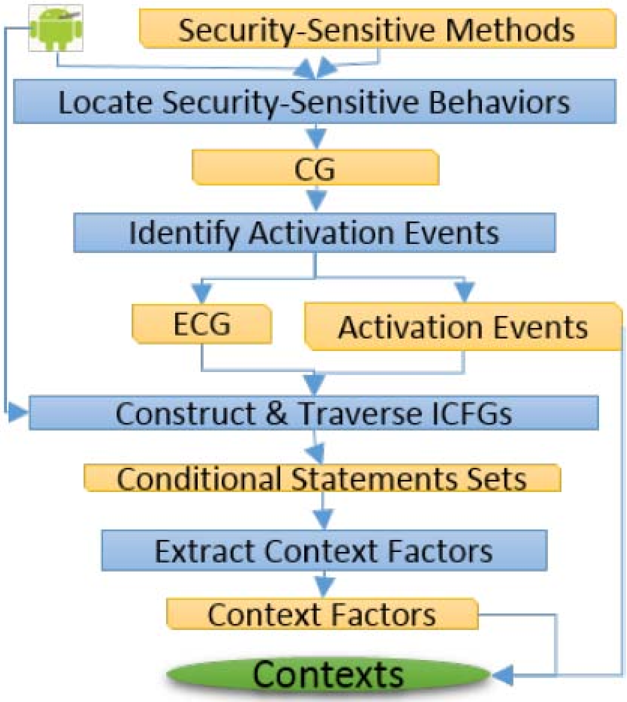
Mobile malware (vs. benign apps)

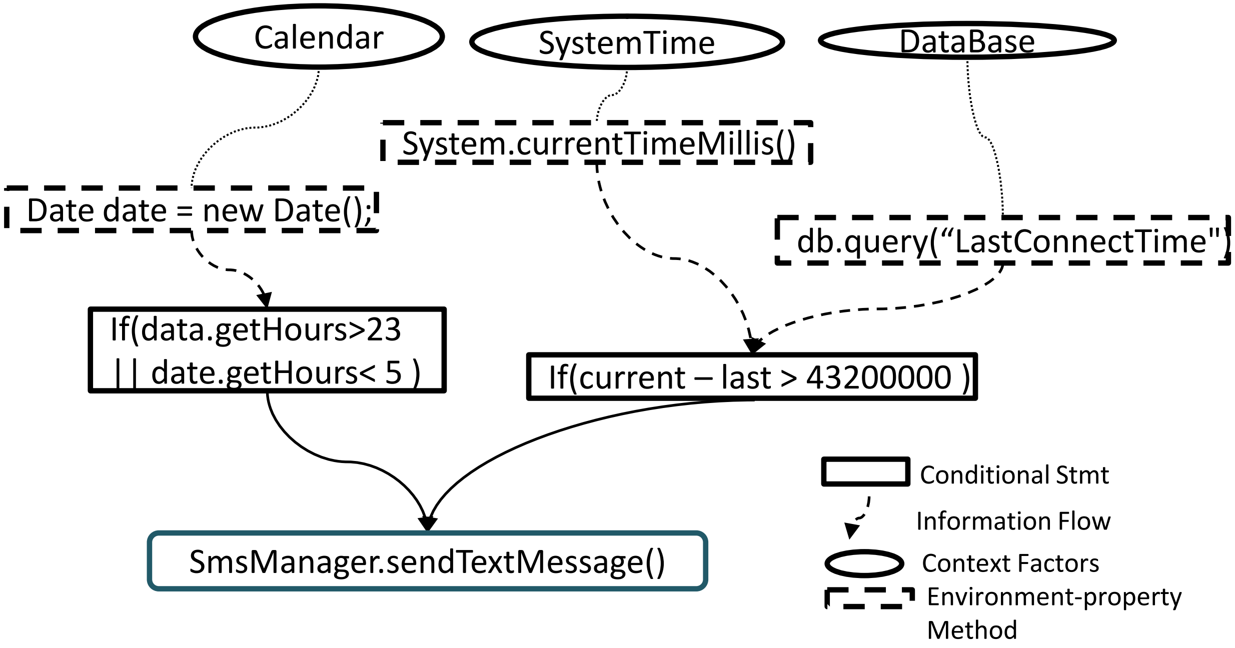
- **Frequently enough** to meet the need: **frequent** occurrences of **imperceptible** system events;
    - E.g., many malware families trigger malicious behaviors via background events

Balance!!!

- **Not too frequently** for users to notice anomaly: **indicative** states of external environments
    - E.g., Send premium SMS every 12 hours

Our AppContext Approach


Security-Sensitive Methods
Locate Security-Sensitive Behaviors
CG
Identify Activation Events
ECG
Activation Events
Construct & Traverse ICFGs
Conditional Statements Sets
Extract Context Factors
Context Factors
Contexts


Calendar
SystemTime
DataBase
System.currentTimeMillis()
Date date = new Date();
db.query("LastConnectTime")
If(data.getHours>23 || date.getHours< 5)
If(current – last > 43200000)
SmsManager.sendTextMessage()
Conditional Stmt
Information Flow
Context Factors
Environment-property Method

Context1: (Event: Signal strength changes), (Factor: Calendar)
Context2: (Event: Entering app), (Factor: Database, SystemTime)
Context3: (Event: Clicking a button)

Context factors: environmental attributes for affecting security-sensitive behavior's invocation (or not)

Yang et al. AppContext: Differentiating Malicious and Benign Mobile App Behavior Under Contexts. ICSE 2015.
http://taoxie.cs.illinois.edu/publications/icse15-appcontext.pdf

Context-based Security-Behavior Classification

Step 1. Transform contexts for each app's security behavior as features

Step 2. Label each behavior in training set as malware or benign

Step 3. Learn a predictive model via ML technique, e.g., support vector machine (SVM)

Step 4. Classify an unlabeled behavior as malware or benign via the model

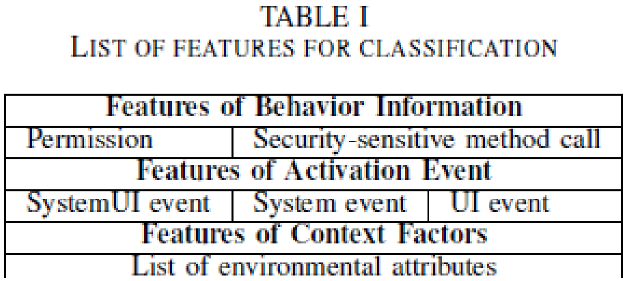
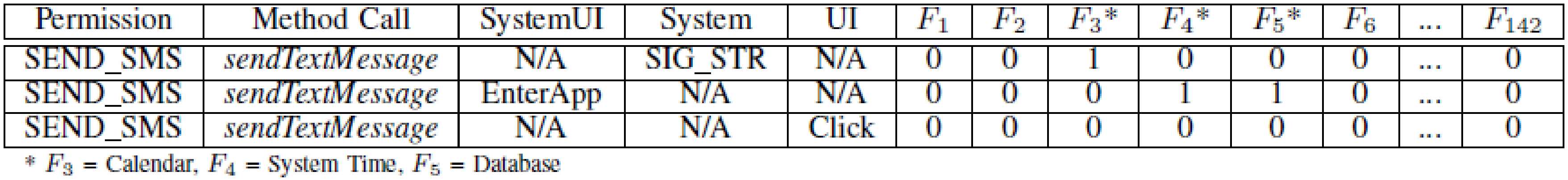


TABLE I
LIST OF FEATURES FOR CLASSIFICATION

Features of Behavior Information		
Permission	Security-sensitive method call	
Features of Activation Event		
SystemUI event	System event	UI event
Features of Context Factors		
List of environmental attributes		



Permission	Method Call	SystemUI	System	UI	F_1	F_2	F_3*	F_4*	F_5*	F_6	...	F_{142}
SEND_SMS	*sendTextMessage*	N/A	SIG_STR	N/A	0	0	1	0	0	0	...	0
SEND_SMS	*sendTextMessage*	EnterApp	N/A	N/A	0	0	0	1	1	0	...	0
SEND_SMS	*sendTextMessage*	N/A	N/A	Click	0	0	0	0	0	0	...	0

* F_3 = Calendar, F_4 = System Time, F_5 = Database

Summary: AppContext

- Capture differentiating characteristics with contexts of security-sensitive behavior.
- Leverage contexts in machine learning (classification) to differentiate malware and benign apps.

User Expectation: User Perception + User Judgment

Yang et al. AppContext: Differentiating Malicious and Benign Mobile App Behavior Under Contexts. ICSE 2015.
http://taoxie.cs.illinois.edu/publications/icse15-appcontext.pdf

(Mobile) Privacy vs. Utility: A Balancing Act in User Expectation

- A likely scenario for a professor
  - **Student A**: "May I record our 1-on-1 meeting so that I don't miss anything?"
  - **Professor**: "Hmmhh… OK… but please don't post it on public domain or redistribute it…"
  - Hopefully….
- Mobile utility apps: app store management, IME (input method editor), …
  - even non-mobile ones: search engines, ….
- Assurance case for privacy policy compliance by app or service providers [Sen et al. Oakland'13]

Sen et al. Bootstrapping Privacy Compliance in Big Data Systems, Oakland 2013. http://research.microsoft.com/apps/pubs/default.aspx?id=208626

PIs: Tao Xie (Illinois), Jim Blythe (USC),
Ross Koppel (U Penn), Sean Smith (Dartmouth)

User Expectations in Mobile App Security

Tao Xie

Questions??

Science of Human Circumvention of Security

To better understand and *to model* computer access *workarounds*—their:

- Reasons, norms, and justifications
- Tasks, urgency, and environments
- Role in others rule-following behaviors
- Methods of discovery
- Sensible (responsible & used) controls

via

- Fieldwork
- Modeling individuals and systems
- Validation
- Application to hard problems in the real world

Computer-Access Workarounds in Healthcare

- Workarounds to computer access in healthcare are common but often go unnoticed (clinicians focus on patient care, not cybersecurity)
- Need to do analyses of computer rules, and interviews & observations w/ clinicians
- Conducted Interviews and observations with hundreds of medical workers and with 19 cybersecurity experts, CIOs, CMIOs (chief medical informatics officer), CTO, and IT workers
- Shadowed clinicians as they worked
- **Findings**: dozens of ways workers ingeniously circumvent security rules

Computer Security Perils of Reuse

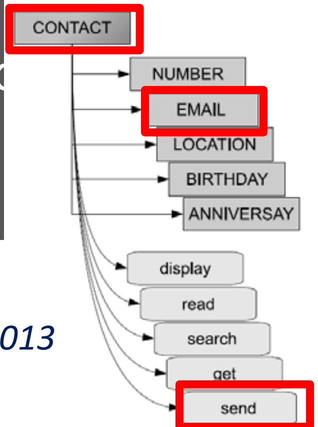
- System designers routinely reuse existing policies, technologies, and architectures—frequently with little or no changes
- Reuse is good software engineering practice
- **Findings**: Careless reuse in a different or even similar domain can introduce failures and new challenges that subvert security goals and impede organizational objectives

J. Blythe, R. Koppel, V. Kothari, S. Smith. "The Computer Security Perils of Reuse." March 2015.

Natural Language Processing on App Description

- *"Also you can share the yoga exercise to your friends via Email and SMS."*
  - Implication of using the contact permission
  - Permission sentences
- **Confounding effects:**
  - Certain keywords such as "**contact**" have a confounding meaning
  - E.g., "… displays user contacts, …" vs "… contact me at abc@xyz.com".
- **Semantic inference:**
  - Sentences describe a sensitive action w/o referring to keyword
  - E.g., "share yoga exercises with your friends via Email and SMS"

*NLP + **Semantic Graphs/Ontologies** Derived from Android API Documents*



Pandita et al. WHYPER: Towards Automating Risk Assessment of Mobile Applications. *USENIX Security 2013*
http://taoxie.cs.illinois.edu/publications/usenixsec13-whyper.pdf

Challenges

- *Ex non-permission sentence: "You can now turn recordings into ringtones."*
  - *functionality that allows users to create ringtones from previously recorded sounds but NOT requiring permission to record audio*
  - *false positive due to using synonym: (turn, start)*
- *Ex. permission sentence: "blow into the mic to extinguish the flame like a real candle"*
  - *false negative due to failing to associate "blow into" with "record"*
- Automatic mining from user comments and forums

Pandita et al. WHYPER: Towards Automating Risk Assessment of Mobile Applications. *USENIX Security 2013*
http://taoxie.cs.illinois.edu/publications/usenixsec13-whyper.pdf