Improving Mobile Application Security via Bridging User Expectations and Application Behaviors

Wei Yang  
University of Illinois  
at Urbana-Champaign  
weiyang3@illinois.edu

Xusheng Xiao  
North Carolina State University  
(xxiao2,rpandit,whenck)@ncsu.edu

Rahul Pandita  
William Enck  
Tao Xie  
University of Illinois at Urbana-Champaign

Objective

• To help classify a mobile application as malicious, privacy infringing, or benign via incorporating user expectations (reflected via user perception in combination with user judgment) in application analysis to determine whether application behavior is within user expectations

Motivation

• Existing analyses do not distinguish between user-expected application behaviors and unexpected application behaviors
• Existing analyses may potentially report all security/privacy-sensitive operations as malicious

Vision

• Bridge the gap between user expectation and application behaviors
• Enforce application-description discussion of permissions protecting user-understandable resources

WHYPER: Why Permissions?

• A framework using Natural Language Processing (NLP) techniques to identify sentences that describe the need for a given permission in an application’s description

Use Cases

• Enhance user experience while installing Apps
• Enforce functionality disclosure on developers
• Complement program analysis to ensure more appropriate justifications

WHYPER achieved an average precision of 82.8%, and an average recall of 81.5% for three permissions (address book, calendar, and record audio)

WHYPER analyzed 581 application descriptions, containing 9,953 sentences (parsed by WHYPER), and identified 751 sentences as permission sentences
• 622 sentences correctly identified as permission sentences
• 129 sentences incorrectly identified as permission sentences
• 141 sentences incorrectly identified as not-permission sentences

Automating Risk Assessment

• Automated risk assessment framework (WHYPER) – Identification of sentences that indicate the need for a given permission in an application’s description
• User-aware privacy control – Application validation and access granting by computing information flows and identifying user-imperceptible information

Information-flow Visualization

• Notify users of potential information leak via presenting information flows to show what private data type flows to what output channels
• Allow users to inspect outgoing information at runtime

Flexible Choices – Anonymized Data

• Users can experience the application without compromising their privacy
• Users can change accesses after experiencing the application

User-Imperceptible Information Flows

Discover two types of information flows that may escape/camouflage themselves from users’ attention:
• Escaping flows - information may flow to output channels (e.g., network sockets) where users cannot perform runtime inspection
• Tampering flows - information may be tampered before it is presented to users for inspection

Tampered data has low integrity (unsafe) e.g., picture -> location encoded -> shared

Reference: