Validating Application Behavior against User Expectations

Wei Yang and Wing Lam
QInF 2016 Final Presentation
Context Matters: Malware vs Benign Apps
Related Work

• Signature based
• Learning based

Limitation:
Existing approaches focuses on limited data sources
Yin-Yang View on Mobile App Security

Goal of our system:
• To comprehensively characterize expectation contexts and security behaviors,
• And check their consistency.

Contextual data: user usage data, user reviews, UI screen (labels, hints, screen, buttons, sequence of screen), app descriptions, privacy policy, pictures/videos, tags (app category)

Behavioral data: API invocations, network incoming and outgoing traffic, keyboard logs, app execution trace, bug/crash reports, static analysis

Expectation Context Check Consistency

Security Behavior Characterize
Traceability-Centric Security Vetting System

• **Key Insights:**
  – Malicious behaviors can be detected as inconsistencies across expectation contexts and security behaviors
  – Leveraging heterogeneous data sources can provide comprehensive characterization

• **Challenges:**
  1. Traceability recovery across expectation context and security behaviors (consistency checking)
  2. Traceability recovery among heterogeneous data sources (characterization)
Challenge 1: Traceability Recovery for Consistency Checking

**Goal:** Develop a set of automated analyses to check inconsistencies and warn users about potential risks.
Roadmap (~12 months)

• **Permission Refinement (~2 months)**
  – Refine permission used by detecting which capability is actually leveraged

• **Context Recovery (~3 months)**
  – Combine *context factors* with *activation events* to generate a context tuple

• **Checking Unexpected Behaviors (~3 months)**
  – Report inconsistencies between synthesized descriptions and app descriptions

• **Removing Unwanted Behaviors (~4 months)**
  – Develop a suite of repair strategies to repair the apps at four levels of granularity (“where”, “when”, “what”, and “how”)

~2 months
~3 months
~3 months
~4 months
~12 months
Challenge 2: Traceability Recovery for Characterization

- Categories
- Descriptions
- UI Texts
- User Reviews
- Privacy Policies
- API Methods
- Permissions
- Private Info

Traceability Recovery

Linkage

Security Behaviors
Strength of the Team

Wei Yang
- Experienced in Android app security and testing, and natural language processing
- Published in *USENIX Security ‘13* and in *FASE ‘13, ICSE ‘15, NDSS ‘16*

Wing Lam
- Experienced in empirical studies, software testing, program analysis, and Android development
- Published in *ISSTA ‘14*

**WHYPER** (Published in Usenix Security)
Uses NLP to analyze app descriptions and permissions
Results attracted Google’s strong interest and attention

**Pluto** (Published in NDSS)
Examines in-app information available to libraries at runtime

**AppContext** (Published in ICSE)
Leverage context information to check malicious behaviors
Conclusion

• **Key Insights:**
  – Malicious behaviors can be detected as inconsistencies across expectation contexts and security behaviors
  – Leveraging heterogeneous data sources can provide comprehensive characterization

• **Proposed Techniques:**
  1. Traceability recovery across expectation context and security behaviors (consistency checking)
  2. Traceability recovery among heterogeneous data sources (characterization)