WILLIAM & MARY

CHARTERED 1693

CSCI 445:
Mobile Application Security

Lecture 24

Prof. Adwait Nadkarni

Announcements

- HW5 released → extra credit assignment, relies entirely on slides.
- Project presentations and final review split across 2 lectures
  - **Thursday (04/23)**: 2 talks, 10 min each
    - We end class early for the inauguration ceremony
  - **Tuesday (04/28):** 2 talks
    - Plus, final review (study up, and ask questions!)
- ***Ask for an extension for Milestone 4 if you need one***

Detecting Mobile Malware!

Traditional detection systems?

- Can we use antivirus software built for desktops?
  - Android/iOS malware is increasingly *mobile-specific*
- Important to understand the attacker's goals and abilities
  - Stealing private information
  - Costing money (e.g., premium messages)
  - Remote control
  - ...
- In many cases, *no exploits*, simple permission misuse
- *We need* ***(1) techniques that detect traditional malware*** *(e.g., rootkits), and also* ***(2) tailored techniques for smartphones.***

Attack Vectors

- Comprehensive characterization: Zhou and Jiang [1]
- *How does malware get on our devices?*
  - Mostly using *social engineering*
- Malware relies on the user to initiate installation
  1. Repackaging
     a. All at once
     b. Runtime download of payload
  2. Drive-by download

[1] Zhou, Yajin, and Xuxian Jiang. "Dissecting android malware: Characterization and evolution." In *Security and Privacy (SP), 2012 IEEE Symposium on*, pp. 95-109. IEEE, 2012.

Repackaged Malware

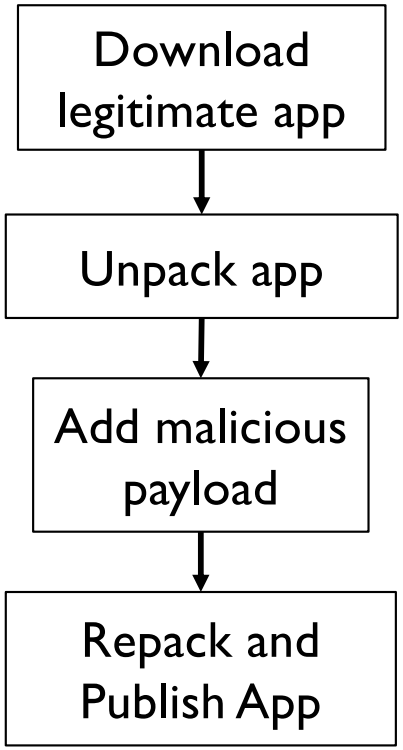
- Where is it published?
  - Third-party stores (generally)
  - Official Stores (e.g., Google Play)

I. To find repackaged apps in ***third-party stores***

i. Look for an app with the same package name as an *official app*

ii. Static/dynamic analysis: Is the difference benign?

General Approach

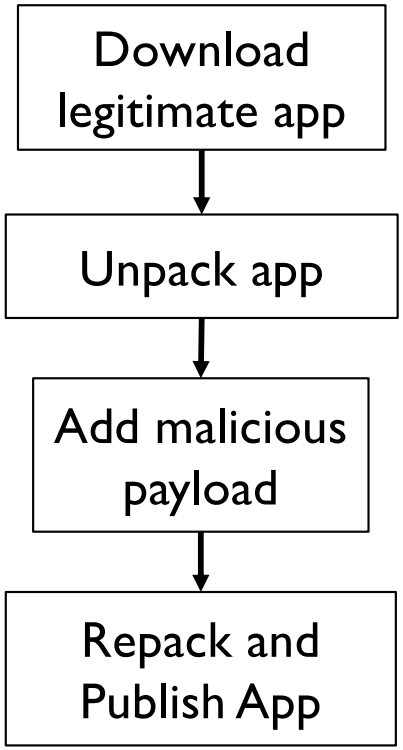


Repackaged Malware

- Where is it published?
  - Third-party stores (generally)
  - Official Stores (e.g., Google Play)

2. To find repackaged apps on ***Google Play***
   i. Can't use package names: package names are *unique!*
   ii. Detect similarity using *metadata*

General Approach



Automated similarity analysis using metadata

- Text analytics (e.g., bag of words)
  - Titles, descriptions, developer names (in that order)
- *Fuzzy* image matching:
  - icons
- Effective for detecting *grayware*
  - May also help detect retargeted malware (e.g., free repackaged versions of paid apps)

Original App		*Impostor App*
The Coupons App	**Title**	The Coupons App
Most Popular Download	**Developer Name**	*Most Popular Downloads*
thecouponsapp.coupon	**Package Name**	*thecouponsapp.dailydeals*
	**Icon**	
10 - 50 million	**Downloads**	0.5 - 1 million

Andow, Benjamin, Adwait Nadkarni, Blake Bassett, William Enck, and Tao Xie. "A study of grayware on google play." In *Security and Privacy Workshops (SPW), 2016 IEEE*, pp. 224-233. IEEE, 2016.

Malware Detection in Practice

- **Target mobile-specific objectives:**
  - Privilege Escalation: Generally, gain *root* privilege
    - Execute one or more root exploits
    - Many exploits are publicly available! (e.g., *towelroot*)
  - Remote Control: Botnets!
  - Charging users: Premium messages, phone calls
  - Stealing private data
  - ...

Malware Detection in Practice

- **Know the limitations of analysis**
  - Malware often hides behavior to evade static analysis
    - Code obfuscation
    - Encrypting code/ root exploits
      - Storing it as an asset
    - Dynamically updating the malicious app
    - JNI
- Problem: *Some of these behaviors are also exhibited by benign apps!*

Malware Detection in Practice

- Boils down to a **classification problem**
- Typical approach:
  1. Select interesting features/feature-types
  2. Train with known malware/benign apps: Use lightweight static analysis to extract features
  3. Use machine learning on feature vectors to classify as benign or malicious
  4. *Test on unlabeled samples*
- VirusTotal: Aggregates results from over 70 virus scanners (most of these are signature based)

Feature Selection

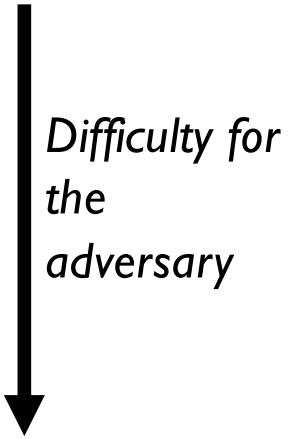
1. **AndroidManifest.xml:**
   - Requested permissions: Sensors, sensitive/private API
   - App components
   - Intent Filters
2. **Disassembled code:** Sensitive API calls
   - APIs for which permissions *have not* been requested. Why?
     - *Sign of potential privilege escalation*
   - Permissions actually *used*
   - Suspicious APIs: get IMEI, dynamic code loading
   - URLs/host names for network communication. Why?
     - *Attributing/ connecting malware samples*

Advantages

- Automated
- Explainable (sometimes; e.g., DREBIN)
- It *generalizes:* Why is this important?
  - Need to detect *variations* of malware
  - More robust against typical evasive maneuvers (e.g., dynamic code loading, obfuscation, etc.)
    - Relies on *a diverse array of features*

Limitations

- Craft **adversarial examples:** Make changes that evade detection, but without changing behavior
  1. Adversary has your model
  2. Adversary does not have model, *but,* can query for *malicious/benign*, and *confidence score*
  3. Adversary can query for *malicious/benign*



- How to get labeled data?: *like everyone does*
  - Query VirusTotal! (or specific scanners you want to evade)
- **Is #3 feasible?:** Train a neural network on this labeled dataset.
  - *Key property: Transferability: If an adversarial sample evades my model, it will also evade other similar models*

Research Methods 2

Parts of a paper

- Parts of paper (vast generalization)
- These points apply to your analysis reports as well (again, *in general)*

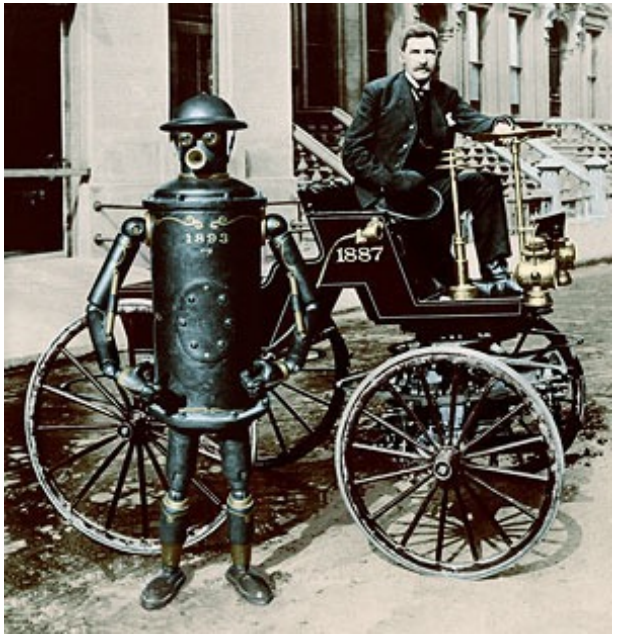
1. Abstract
2. Introduction
3. Related Work/Background
4. Solution/Problem
5. Evaluation/Analysis/Experiment
6. Discussion (often, but not always)
7. Conclusions

Abstract

- One sentence each for:
  - Area
    - Topic of work
  - Problem
    - What's the issue?
  - Solution
    - How do you propose to address the problem?
  - Methodology
    - What's the experiment?
  - Results
    - What did you find?
  - Take Away: Lesson

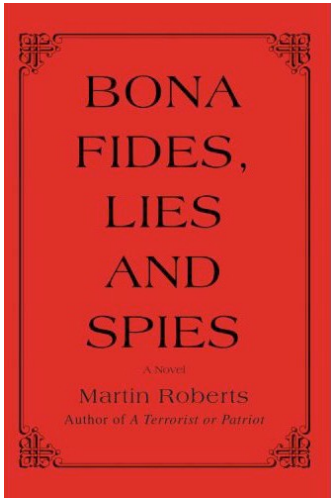
Introduction

- One paragraph each on:
- Area
  - More elaborate
- Problem
  - Scenario
- Why is problem not solved
  - Brief of related work or the challenge
- Proposed insight ("In this paper, ...")
  - What is the experiment?
- Contributions -- What will the reader learn?
- Boilerplate outline (?)



Related work

- This is a statement of the work that led to this one.
  - who this work relies on
  - who has done work in the area
  - areas that inspired this work (not just technology)
  - Not a laundry list
- There are several reasons for related work section:
  - Motivate the current work
  - Differentiate from past work
  - Establish "bona fides"



Motivation, Background

- **Motivation**
  - Why is this a problem?
  - Motivating Example: Alice…
  - Why isn't the problem solved?
    - Forward/backward reference to the related work.
- **Problem, assumptions**: Problem statement, threat model, TCB.
- **Background:** What all does the reader need to know to understand your approach?
    - Already known material related to the solution
    - Tip: You can always move text from the design to the background, to focus on the *novel contributions in the design*.

System Architecture and Design

- How do you solve the problem?
- General Architecture / Overview
- What are the
  - Design Goals?
  - Challenges?
  - Contributions of your design (i.e., the design decisions) that help overcome the design challenges, hence achieving the design goals?

Experiment

- Research questions: i.e., questions the research has set out to answer.
- Experiment
  - Means of showing truth
  - Big Insight -- Hypothesis -- Claim
    - Show why it is interesting
  - Expected Results
    - Informal proof/argument that is true
- Experiment types
  - *Empirical* - measure some aspect of the solution
  - *Analytical* - prove something about solution
  - *Observational* - show something about solution

Results and Findings

- Results
  - Summarize -- what do the results mean?
  - Specific experiments
    - We did X, saw Y
  - What do the experiments prove
  - What other experiments would you want to do based on these results?
- Key Findings
  - *What do the results mean?*
  - What are the lessons?
  - *Lead to the takeaway.*

Conclusion

- Like the abstract in past tense
- Problem
  - What was the problem?
- Solution
  - What was the insight and why was it expected to work?
- Method and Results
  - What did you find?
- Take away: Lesson
- Future work

Hint

- Intro: tell them what you are going to tell them
- Body: tell them
- Conclusion: tell them what you told them.