# BenchPress: Analyzing Android App Vulnerability Benchmark Suites

**Joydeep Mitra** Venkatesh-Prasad Ranganath Aditya Narkar

Department of Computer Science
Kansas State University, USA

International Workshop on Advances in Mobile App Analysis (A-Mobile) 2019
San Diego, USA
November 11, 2019

# Context

Lots of interest in tools and techniques for Android security analysis

- AmanDroid, FlowDroid, COVERT, MobSF, Qark, etc.

Tool evaluation is based on benchmarks or real-world apps

- Choice of benchmarks are not informed by their characteristics or relevance
- Real-world apps do not contain ground truths

To trust the evaluations, are these benchmarks representative?

- How about measuring the representativeness of benchmark suites?
- This effort is inspired by a similar evaluation of Ghera benchmarks (ASE'19 JF)

# Objectives/Research Questions

1. Are Android app vulnerability benchmark suites representative of real-world apps in terms of API usage?
    a) What is the representativeness of a benchmark?
    b) How can it be measured?
2. Do real-world apps use security-related APIs not used by any benchmark suite?
    a) Is there an opportunity to extend the benchmarks?

# Representativeness of a Vulnerability Benchmark

*The manifestation of a vulnerability in the benchmark should be similar to its manifestation in real-world apps*

Measuring representativeness == Compare benchmarks with real-world vulnerable apps

- Consider producers and consumers of data
- Consider APIs involved in handling and processing data
- Consider data/control flow paths
- We used API usage as a metric for representativeness

We used API usage as a metric for representativeness

- Extent to which APIs used in a benchmark are used by real-world apps

# Experiment

1. Select benchmark suites used for tool evaluations
   - DroidBench, Ghera, ICCBench, UBCBench
2. Collect real-world apps from AndroZoo
3. Identify the (security-related) APIs used in an app
4. Measure the representativeness of a benchmark suite based on API usage
5. Identify the APIs used in real-world apps but not in any benchmark

# Observations & Open Questions about Benchmark Suites

1. All considered benchmark suites contain benchmarks in the form of apks and source code
2. Except Ghera, every considered benchmark suite provide APKs containing unnecessary code and resources
3. Most benchmarks run on currently supported versions of Android, even if they were not designed to run on them
4. 35 benchmarks across DroidBench and UBCBench crashed when run currently supported versions of Android

| Benchmark Suite | # Benchmarks | # Successfully built | # Successfully Executed |
|---|---|---|---|
| DroidBench | 211 | 201 | 169 |
| Ghera | 60 | 60 | 60 |
| ICCBench | 24 | 24 | 24 |
| UBCBench | 16 | 16 | 13 |

# Experiment

1. Select benchmark suites used for tool evaluations
   - DroidBench, Ghera, ICCBench, UBCBench
2. Collect real-world apps from AndroZoo
   - 226K apps with target API level 23-27
3. Identify the (security-related) APIs used in an app
4. Measure the representativeness of a benchmark suite based on API usage
5. Identify the APIs used in real-world apps but not in any benchmark

# Experiment

1. Select benchmark suites used for tool evaluations
   - DroidBench, Ghera, ICCBench, UBCBench
2. Collect real-world apps from AndroZoo
3. Identify the (security-related) APIs used in an app
4. Measure the representativeness of a benchmark suite based on API usage
5. Identify the APIs used in real-world apps but not in any benchmark

# Identifying APIs in Benchmark Suites

1. Collect API profile for each app/benchmark
   - Elements and attributes in an app's manifest
   - Callback methods
   - APIs used but not defined
2. Use developer discussions on Stack Overflow
   - Collect relevant tags related to *Android* and *security of Android* (apps)
   - Consider all posts with *Android* tag
   - Consider all posts with *Android security-related* tag
   - If an API used in a benchmark occurred in a post with *Android* tag, then deem it as *relevant*
   - If an API used in a benchmark occurred in a post with *Android security-related* tag, then deem it as *security-related*

# Identifying APIs in Benchmark Suites

| Benchmark Suite | # Total APIs | # Relevant APIs | # Security-related APIs |
|---|---|---|---|
| DroidBench | 2188 | 769 | 744 |
| Ghera | 1906 | 504 | 494 |
| ICCBench | 185 | 70 | 70 |
| UBCBench | 751 | 98 | 96 |

# Experiment

1. Select benchmark suites used for tool evaluations
   - DroidBench, Ghera, ICCBench, UBCBench
2. Collect real-world apps from AndroZoo
3. Identify the (security-related) APIs used in an app
4. Measure the representativeness of a benchmark suite based on API usage
   - Percentage of real-world apps that use a relevant (security-related) API
5. Identify the APIs used in real-world apps but not in any benchmark

# Observations about Representativeness of Benchmark Suites

## Relevant (security-related) APIs used by all benchmarks suites are also used by real-world apps

- 73%, 67%, 78%, and 81% relevant APIs in DroidBench, Ghera, ICCBench, and UBCBench, respectively, are used in more than 60% of real-world apps

## DroidBench and Ghera use more than four times the number of APIs used by ICCBench and UBCBench

# Open Questions about Using Stack Overflow

All APIs discussed by more than 15% of Stack-overflow posts are used in more than 60% of real-world apps

- Is there less clarity about such APIs amongst app developers?

For all benchmark suites the number of security-related APIs were almost the same as relevant APIs

- Is community knowledge of security-related APIs more effective than expert based knowledge?

# Experiment

1. Select benchmark suites used for tool evaluations
   - DroidBench, Ghera, ICCBench, UBCBench
2. Collect real-world apps from AndroZoo
3. Identify the (security-related) APIs used in an app
4. Measure the representativeness of a benchmark suite based on API usage
5. Identify the APIs used in real-world apps but not in any benchmark

# Observations about Unused Real World APIs

26K APIs are used in real-world apps but not in any benchmark

Approx. 70% (18K out of 26K) of the APIs not used in the benchmarks are not related to security

Approx. 90% (7K out of 8K) of the APIs not used in benchmarks but related to security belong to parts of the Android framework that are covered by the benchmarks

# Suggestions to Extend Benchmark Suites

## Analyze the 8K unused APIs related to security

- At least 17 APIs can be used to create new benchmarks
- We used 2 APIs to create 2 new benchmarks in Ghera

## Explore APIs in parts of the Android framework that have been covered by the benchmarks

- E.g., android.app, android.media, android.content, android.net, android.os

## Explore parts of the Android framework not covered by the benchmarks

- E.g., android.preference, android.renderscript, android.nfc, android.service, android.speech

# Open Questions about Using Stack Overflow

## Is Stack-overflow a good source for community-wide information about Android app security?

- APIs that do not occur in Stack-overflow posts related to security might still be related to security

## Why are security-related APIs not being discussed in Stack-overflow?

- Are developers discussing in other forums?
- Are developers not interested in these APIs?
- Do developers understand such APIs enough to not discuss them?
- Are developers not directly using these APIs for app development?

# Threats to Validity

1. API usage is a weak measure of representativeness
2. The occurrence of an API in a Stack Oveflow post might not always imply discussion of the API

# Takeaways

## Call for Action

Based on API usage, DroidBench is the most representative benchmark suite followed by Ghera, ICCBench, and UBCBench

- Evaluate the representativeness of benchmarks

Benchmark suites are not comprehensive and can be extended with security-related APIs used in real-world apps

- Explore the APIs used in real-world apps but not in any benchmark

Stack-overflow posts might not be the best markers for identifying security-related APIs used in Android app development

- Identify markers that can be accurately and subjectively used to identify security-related APIs