Sustainable Defenses against Evolving Malware in the Android Ecosystem: A Manifesto

Haipeng Cai
Washington State University
Sustainable defense manifesto -- the Android case

- Context
- Problem
- Sustainable Defense
- Current Results
- Looking ahead
The dominance of Android

2019 Q3

Operating systems on Smartphones shipped during 2019 Q3[1]

Mobile operating systems distribution of page visits from StatCounter [2]

The Android ecosystem

End user, app developer

Android apps

User applications

OS/Framework/SDK API, etc.

Android platform

Context – Problem – Sustainable Defense Manifesto – Current Results – Looking Ahead
...is evolving constantly & rapidly

- The problem:
  - Security defense techniques deteriorate
  - Compatibility issues aggravate
  - ......
It is a problem…

Numerous and continuously more defense solutions available

Malware in Android still growing [1]

Compatibility issues also thriving [2]

99% of mobile malware run on Android

Context – Problem – Sustainable Defense Manifesto – Current Results – Looking Ahead
Sustainability is essential

- A **manifesto** on sustainable defense against malware

A defense solution against malware (e.g., malware detection technique) should keep (sustain) its desirable capabilities and performance over time without frequent updating/maintenance

- Why **no** frequent updating/maintenance?
  - Overhead
  - Sample availability
    - Zero-days
Evolution is the cause: embrace it!

- **Gain** an understanding of the evolution of the holistic ecosystem
  - Through **systematic characterization**
- **Sustain** the evolutionary understanding
  - Through **continuous mining**
Continuous and evolutionary ecosystem mining and characterization

Context – Problem – **Sustainable Defense Manifesto** – Current Results – Looking Ahead
Ecosystem data mining

- Efficient data harvesting and storage
- Data on holistic ecosystem
  - Platform
  - Apps
  - User community

Continuous Ecosystem Data Mining

- Search & Crawling
- Preprocessing & Linking
- Storage & Indexing

Source

Android repository

App stores

Android markets

Database (HBase)

Raw data & characteristics of the ecosystem
Systematic ecosystem characterization

- App characterization
  - Code structure
  - Component communication
  - Sensitive access
- Platform characterization
  - Permission mechanism
  - API
  - SDK
- User community characterization
  - End-user dynamics
  - Developer dynamics

Context – Problem – **Sustainable Defense Manifesto** – Current Results – Looking Ahead
Practical applications/tools

- Proactive app quality (via app health profiling)
  - Quantitative metrics on compatibility, vulnerabilities, usability, and predicted user rating
- Sustainable app security classification
  - Identifying evolution-resilient differentiating features
Put it into action

- Evolutionary characterization
  - Run-time behavioral evolution: malware vs. benign
- Assessing/quantifying “sustainability”
  - Introduce a new performance metric
- Improving sustainability: the malware detection case
  - Achieve a more sustainable solution
A need for understanding

- How Android apps evolve in run-time behaviors
- What are the differences in code-level execution structure
- How benign apps differ from malware in the evolution

Informing the actual, rather than estimated, behaviors of apps

Offering deeper insights into explaining external behaviors

Providing immediate references for developing better malware defense techniques
Our study

• **Dataset**
  - 15,451 benign apps
  - 15,183 malware

• **Profiling**
  - With random inputs: 10 mins, line coverage 60~100% (mean 74.85, stdev 11.97)
  - On emulator: SDK 6.0

• **Characterization**
  - From 30,634 traces (i.e., purely dynamic study)
  - Dynamic call graphs of
    - Ordinary method calls
    - ICC calls

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td># benign apps</td>
<td>1,530</td>
<td>2,019</td>
<td>2,053</td>
<td>1,748</td>
<td>3,127</td>
<td>1,333</td>
<td>1,548</td>
<td>2,093</td>
</tr>
<tr>
<td># malware</td>
<td>2,029</td>
<td>2,431</td>
<td>2,225</td>
<td>1,230</td>
<td>1,493</td>
<td>1,667</td>
<td>2,171</td>
<td>1,937</td>
</tr>
</tbody>
</table>

Every characterization metric is a percentage of certain kinds of calls over all calls in a larger class.
Results and findings

• Cross-layer calling relationship: benign apps
  • both benign apps and malware had decreasing calls within user code and increasing calls within the SDK
  • malware had more calls to SDK from third-party libraries, benign apps had more such calls within the Android framework

• Cross-layer calling relationship: malicious apps
Results and findings

Percentage of callsites targeting a sink

Percentage of calls targeting a vulnerable sink

Context – Problem – Sustainable Defense Manifesto – Current Results – Looking Ahead
Defining, assessing, improving sustainability

• Defining - sustainability
  the accuracy of a classifier trained on apps of year $x$ and tested against apps of year $y$, $y\geq x$

• Developing – DroidSpan
  Characterizing evolution to discover evolution-resilient features for classification

• Comparing sustainability
  • 5 state-of-the-art malware detectors
  • 10k+ benign apps and 10k+ malware
  • spanning 8 years

• Sustainability improvement
Sustainability – a new quality metric

- **Sustainability**
  the accuracy of a classifier trained on apps of year $x$ and tested against apps of year $y$, $y \geq x$

- **Reusability**
  the accuracy of a classifier trained on apps of year $x$ and tested against apps of year $y$, $y = x$

- **Stability**
  - the accuracy of a classifier trained on apps of year $x$ and tested against apps of year $y$, $y > x$
  - $y - x$

Accounting for how the classifier sustains with retraining

Accounting for how the classifier sustains without retraining or other model updates
DroidSpan – a detector based on SAD profiles

App evolution characterization

Evolution-resilient feature discovery

Sensitive Access Distribution (SAD)

Sustainable classification
DroidSpan – a detector based on SAD profiles

- Extent of sensitive access
  - E.g., percentage of total source/sink callsites and call instances
- Categorization sensitive data and operations accessed
  - E.g., percentage of source/sink callsites retrieving network info
- Vulnerable method-level control flows
  - E.g., percentage of call instances to sources accessing Account data that reach at least a sink

App evolution characterization

Evolution-resilient feature discovery

Sensitive Access Distribution (SAD)

Sustainable classification
DroidSpan – a detector based on SAD profiles

Constructing the SAD profile of a given Android app

Context – Problem – Sustainable Defense Manifesto – **Current Results** – Looking Ahead
Assessing/Comparing sustainability

• Assessment datasets

<table>
<thead>
<tr>
<th>Benign apps</th>
<th>Malware</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Year</td>
</tr>
<tr>
<td>B10</td>
<td>2010</td>
</tr>
<tr>
<td>B11</td>
<td>2011</td>
</tr>
<tr>
<td>B12</td>
<td>2012</td>
</tr>
<tr>
<td>B13</td>
<td>2013</td>
</tr>
<tr>
<td>B14</td>
<td>2014</td>
</tr>
<tr>
<td>B15</td>
<td>2015</td>
</tr>
<tr>
<td>B16</td>
<td>2016</td>
</tr>
<tr>
<td>B17</td>
<td>2017</td>
</tr>
<tr>
<td>total</td>
<td>15,008</td>
</tr>
</tbody>
</table>

5 state-of-the-art malware detectors: MA*MA*DROID, DROIDSIEVE, RE*VEAL*DROID, MUDFLOW, AFONSO (A DYNAMIC DETECTOR)

(On 10k+ benign apps and 10k+ malware spanning 8 years)

• Metrics
  • Sustainability
  • Reusability
  • Stability
  • Efficiency

Context – Problem – Sustainable Defense Manifesto – Current Results – Looking Ahead
Results – reusability

• Each dataset: 1/3 hold-out (& 10-fold CV)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>DroidSpan</th>
<th>MamaDroid</th>
<th>DroidSieve</th>
<th>Afonso</th>
<th>RevealDroid</th>
<th>Mudflow</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>P</td>
<td>R</td>
<td>F1</td>
<td>P</td>
<td>R</td>
<td>F1</td>
</tr>
<tr>
<td>B10+M10</td>
<td>0.9376</td>
<td>0.9360</td>
<td>0.9362</td>
<td>0.8424</td>
<td>0.8357</td>
<td>0.8367</td>
</tr>
<tr>
<td>B11+M11</td>
<td>0.9432</td>
<td>0.9417</td>
<td>0.9423</td>
<td>0.9893</td>
<td>0.9893</td>
<td>0.9793</td>
</tr>
<tr>
<td>B12+M12</td>
<td>0.9424</td>
<td>0.9424</td>
<td>0.9423</td>
<td>0.8378</td>
<td>0.8378</td>
<td>0.8377</td>
</tr>
<tr>
<td>B13+M13</td>
<td>0.9554</td>
<td>0.9529</td>
<td>0.9525</td>
<td>0.9141</td>
<td>0.9076</td>
<td>0.9060</td>
</tr>
<tr>
<td>B14+M14</td>
<td>0.9302</td>
<td>0.9272</td>
<td>0.9250</td>
<td>0.8462</td>
<td>0.8467</td>
<td>0.8449</td>
</tr>
<tr>
<td>B15+M15</td>
<td>0.9061</td>
<td>0.9042</td>
<td>0.9036</td>
<td>0.8450</td>
<td>0.8440</td>
<td>0.8442</td>
</tr>
<tr>
<td>B16+M16</td>
<td>0.9352</td>
<td>0.9342</td>
<td>0.9339</td>
<td>0.9021</td>
<td>0.8969</td>
<td>0.8955</td>
</tr>
<tr>
<td>B17+M17</td>
<td>0.9723</td>
<td>0.9720</td>
<td>0.9720</td>
<td>0.9126</td>
<td>0.9093</td>
<td>0.9098</td>
</tr>
<tr>
<td>Average</td>
<td>0.9408</td>
<td>0.9393</td>
<td>0.9388</td>
<td>0.8835</td>
<td>0.8810</td>
<td>0.8794</td>
</tr>
</tbody>
</table>

DroidSpan achieved reusability of 94% with small variations across years, outperforming all the five baselines considered (by 6–32%).
Results – stability

- 28 experiments
- Each experiment: benign+malware of year \( x \) for training, benign+malware of year \( x+n \) for testing, \( 1 \leq n \leq 7 \)
DroidSpan outperformed all the five baselines, achieving 21% to 37% higher F1.
Results – stability

• Overall stability

• Significance of improvements

<table>
<thead>
<tr>
<th>Contrast Group</th>
<th>Reusability p-value</th>
<th>Reusability Effect size</th>
<th>Stability p-value</th>
<th>Stability Effect size</th>
</tr>
</thead>
<tbody>
<tr>
<td>DroidSpan vs MamaDroid</td>
<td>4.23E-02</td>
<td>0.75</td>
<td>4.00E-06</td>
<td>1</td>
</tr>
<tr>
<td>DroidSpan vs DroidSieve</td>
<td>1.43E-02</td>
<td>1</td>
<td>4.00E-06</td>
<td>1</td>
</tr>
<tr>
<td>DroidSpan vs Afonso</td>
<td>1.43E-02</td>
<td>1</td>
<td>4.00E-06</td>
<td>1</td>
</tr>
<tr>
<td>DroidSpan vs RevealDroid</td>
<td>1.43E-02</td>
<td>1</td>
<td>8.51E-06</td>
<td>0.86</td>
</tr>
<tr>
<td>DroidSpan vs Mudflow</td>
<td>1.43E-02</td>
<td>1</td>
<td>5.84E-05</td>
<td>0.64</td>
</tr>
</tbody>
</table>
## Results – efficiency

<table>
<thead>
<tr>
<th></th>
<th>feature time</th>
<th>ML training time</th>
<th>total time</th>
<th>storage cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>DroidSpan</td>
<td>351.1s</td>
<td>0.01s</td>
<td>351.1s</td>
<td>21.2K</td>
</tr>
<tr>
<td>MamaDroid</td>
<td>430.9s</td>
<td>41.9s</td>
<td>471.9s</td>
<td>1736.01K</td>
</tr>
<tr>
<td>DroidSieve</td>
<td>75.2s</td>
<td>3.5s</td>
<td>78.7s</td>
<td>0.4KB</td>
</tr>
<tr>
<td>Afonso</td>
<td>521.4s</td>
<td>0.015s</td>
<td>521.4s</td>
<td>32.5K</td>
</tr>
<tr>
<td>Revealdroid</td>
<td>78.4s</td>
<td>18.3s</td>
<td>96.7s</td>
<td>1156.81K</td>
</tr>
<tr>
<td>Mudflow</td>
<td>698.7s</td>
<td>0.2978s</td>
<td>698.9s</td>
<td>46.67K</td>
</tr>
</tbody>
</table>

DroidSpan achieved clearly advantageous sustainability at reasonable time and storage space cots
Lessons and insights

- Concept drift causes learning-based malware detector to deteriorate in classification performance

- The deterioration may not be monotonic over years

- App evolution study enables the design of sustainable malware classifier

- Delaying but not eliminating retraining
Future challenges

- Sustainability for some spans, not for ever
  - Not feasible
  - May not be necessary
- Even short-term can be hard
  - Natural concept drift
  - Adversarial evasions
- Consequence: zero-days slipping through…
Future solutions

• Model design
  • Deep-semantic features
  • Learning beyond examples

• Data carpentry
  • ‘Ethical’ malware sample generation
    • Generating based on the distribution
    • Evolutionary computation

• Outcome: further slow down technique deterioration…
Summary

Sustainable Defenses against Evolving Malware in the Android Ecosystem: A Manifesto

- Ecosystem evolution: the status quo
- The problem with the evolution
- Sustainable defenses
- Results (TIFS’19, TOSEM’20, TSE’21)
- Forwardlooking

Thanks for your attendance and attention!