KANSAS STATE
College of Engineering
Computing and Information Sciences

Scalable Security for Millions of Apps:

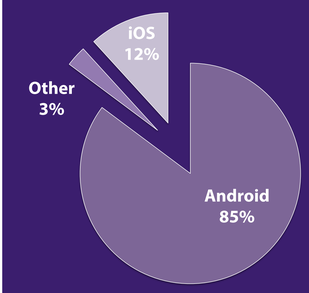
Utilizing Tiered-Machine Learning to Scale to Market-wide Solutions

Jordan DeLoach, Xinming Ou, Doina Caragea

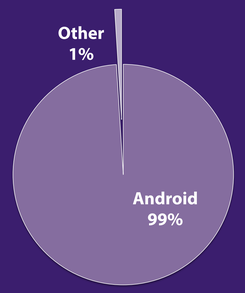
Background

- Android is the most popular platform, both for users (84.7%) and malware (99%)
- Many reasons make it the most common for malware: low protective measures taken by Google, more exploitable APIs, and third-party app stores.
- Google's malware detection tool, Bouncer, performs minimal checks before allowing apps onto the Play Store
- Third-party stores often have no protective measures

Users By Platform



Malware By Platform



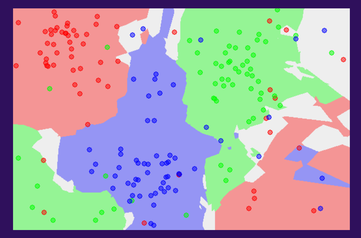
Problem

- The massive amounts of apps developed everyday makes existing solutions ineffective
- Static Analysis Tools are too computationally expensive and not scalable
- Lightweight syntactic approaches are too simple to work with high certainty
- Android Malware continue to evolve and quickly render signature-based systems irrelevant
- A proactive combination of the scalability of lightweight syntactic features with rich semantic features are necessary to provide for scalable security.

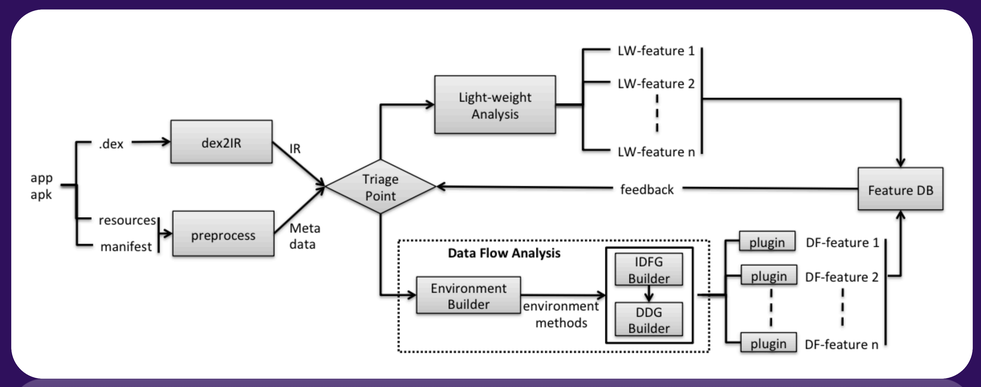
Objectives

- Develop an architecture that can provide the quality and precision of in-depth static analysis, at the speed and scalability of light weight analysis.
- Evaluate the system on realistic datasets symbolic of the magnitude and prevalence of malware in the wild.

Our Approach



- Start with over a million apps classified as either "malware" or "benign"
- Train our classifier on what characteristics relate to maliciousness
- Combine several light features to create a fast and less precise initial point of triage
- Apply more sophisticated and semantic-rich analyses where lightweight analysis were inconclusive
- Bottom line: fast and scalable malware detection with the insights of in-depth analysis where needed

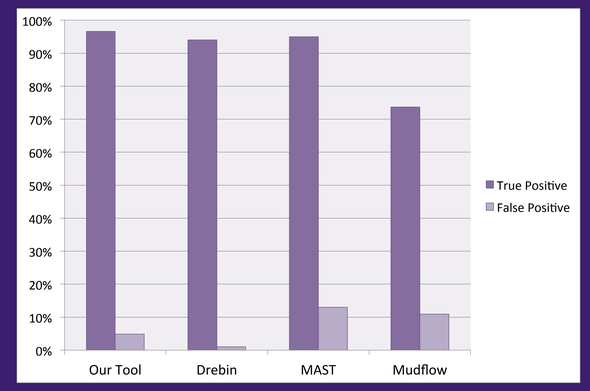


Implementation

- To scale, we utilize Kansas State's high performance computing cluster, Beocat, to extract lightweight feature vectors from millions of apps
- We then feed these characteristics, or feature vectors, to our machine learning classifier to teach it the characteristics of malicious and benign apps
- The system can then detect and classify new apps as either malicious or benign with high certainty
- If the app is marked as possibly malicious, we will apply slower and more precise analyses through our Amandroid framework to make a more accurate decision.

Results

- Our lightweight implementation is able to successfully detect malware 96.6% of the time
- We misclassify benign applications as malicious 4.9% of the time.



Conclusions

- Dataset selection can highly affect results
- Using small, old, or "popular" apps can yield very misleading results
- Our large-scale approach makes our results more real-world but also accuracy

Future Work

- Future work includes devising even newer and more indicative lightweight features
- Examining how detection models change and evolve as the Android ecosystem advances
- Tying in our custom, in-house static analysis framework, Amandroid, to provide more semantic-rich features to learn on

Acknowledgements

This research is supported by the National Science Foundation under Grant No. 0644288, 0954138 and 1018703, and the U.S. Air Force Office of Scientific Research under award no. FA9550-09-1- 0138. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the sponsors.