

# Detecting Malware in Android Applications

Harrison Mansour  
Seaver Thorn  
Prof. Fu  
Hani Alshahrani

# Our Project

As mobile computing has become more and more pervasive in today's world, so has malware that specifically targets our mobile devices. The newest iterations of malware are smart enough to evade most static analysis techniques and even some dynamic techniques. To combat this, we plan to expand upon the ideas and techniques presented by Andromaly, DroidDetector, MARVIN and DroidScribe, among others, that utilize machine learning techniques to detect malicious applications.



# Related Works



# Andromaly (Detection)

- On-device malware detection software utilizing machine learning algorithms such as:
  - K-Means
  - Logistic Regression
  - Histograms
  - Decision Trees
  - Bayesian Networks
  - Naive Bayes
- Collected 88 features from which the most highly ranked were:
  - Anonymous\_Pages
  - Garbage\_Collections
  - Battery\_Temp, Total\_Entities
  - Active\_Pages and Running\_Processes



# Andromaly Limitations

- Did not train on real world malware; they developed 4 apps that performed DoS and information theft attacks.
- Omitted system call features.
- Only used supervised machine learning algorithms.



# DroidDetector (Detection)

- Off-device malware detection software utilizing deep learning techniques, specifically Deep Belief Networks (DBN).
- Extracted 192 features from static and dynamic analysis, including:
  - Required permissions (static)
  - Sensitive APIs (static)
  - Dynamic behaviors (dynamic via Droidbox emulator)
- Showed detection accuracy of 96.76% after analyzing 20,000 trusted and 1,760 malicious applications.
- Limitations:
  - Does not work on-device
  - Static analysis was done manually
  - Used Droidbox instead of a real device to run dynamic analysis.

# MARVIN (Classification)

- Collected 490,000 features from 135,000 benign and 15,000 malicious apps through static and dynamic analysis.
  - F-score feature selection was utilized to select the most important features.
- Fed pruned feature set to Linear Classifier and Support Vector Machine (SVM) machine learning models.
  - Found that linear classification was faster and at least as accurate as SVM.
  - Showed an accuracy of 98.24%
- Limitations:
  - Dynamic analysis done off-device.
  - Excluded system calls from feature set.




# DroidScribe (Classification)

- Uses machine learning and dynamic analysis to classify malware into families
  - Extracted 254 system call features.
  - Used Support Vector Machines (SVM) to classify malware.
  - When insufficient information was available, Conformal Prediction was used to improve the SVM classifier to deliver a group of potential classifiers.
- Showed an accuracy of 94% after analyzing 5,560 malicious applications.
- Limitations
  - Inherits all of CopperDroid's limitations
  - Extracted features in an emulated environment; some malware can detect and evade emulators.
  - Only evaluate system call features.



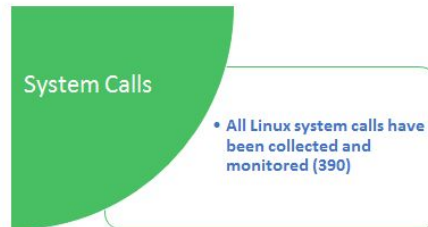
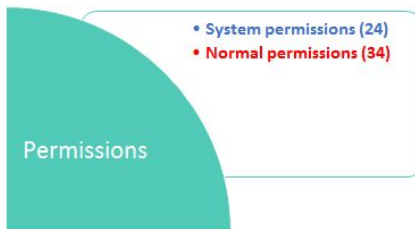
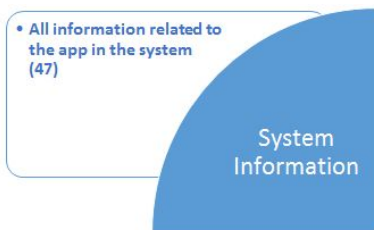


# PrivacyGuard (Information Leakage Detector)

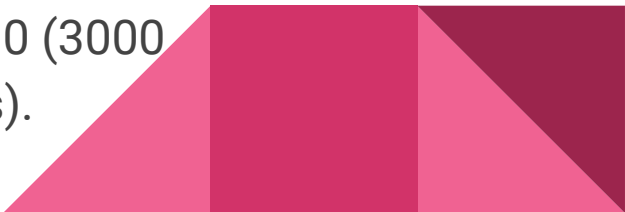
- Attempts to monitor the network using a local VPN to screen all outgoing packets for sensitive information such as phone number, deviceId, and location.
  - Has plugin API so others can write their own information trackers. It will be useful to track network information to detect malware, since malware sends malicious information.
  - While not a Malware detector/classifier, PrivacyGuard may prove to be beneficial for our cause. Apps that leak information are often malware.
  - Shortcomings
    - Cannot distinguish between intentional functionality and illegitimate leakage.
    - Does not track gyroscope, camera, or microphone.
    - Cannot detect information leakage if the app encrypts information before transit.
- 

# Our Improvements

- We are running apps on-device, and therefore do not face any of the limitations that are inherent with the use of emulators.
- Focuses on System calls, permissions, and other hardware features.



# Our Project

- Collect a feature set of benign and malicious applications via on-device dynamic analysis.
  - Design a machine learning framework that effectively utilizes these features to identify malicious applications.
    - Try many different machine learning techniques and pick the best
    - SVM
    - Neural Network
    - Logistic Regression
    - etc...
  - Teach our framework with the feature vectors gathered.
  - Using MARVIN and Drebin datasets for a total of 4000 (3000 Benign applications and 1000 malicious applications).
- 

# Android App Architecture

- Computer works with device to automate the process of installing new apps, and monitoring them.
- Runs in the background on the device, and checks what apps are doing
  - To obtain system call information
  - Battery life, cpu, memory used.
  - Network information
  - App permissions, and developer certificate.
- Sends results to a database, machine learning algorithms will use this information for training, testing, and validation techniques.



# Current Issues

- Feature collection for malicious applications is proving problematic:
  - Some malicious applications crash frequently, making it difficult to collect information about the device.
- Collecting features takes a long time. Been collecting features all week, still not done.
- Network information is not being monitored.



# This Week

- Finish collecting features.
- Create and test machine learning models in the cloud.
- Attempt to fix some problems in our app that does the scanning.
- Create PrivacyGuard plugin that works with our app to understand network traffic.



# References

- Gerardo Canfora, Eric Medvet, Francesco Mercaldo, and Corrado Aaron Visaggio. 2016. Acquiring and Analyzing App Metrics for Effective Mobile Malware Detection. In Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics (IWSPA '16). ACM, New York, NY, USA, 50-57.
  - S. K. Dash et al., "DroidScribe: Classifying Android Malware Based on Runtime Behavior," 2016 IEEE Security and Privacy Workshops (SPW), San Jose, CA, 2016, pp. 252-261.
  - M. Lindorfer, M. Neugschwandtner and C. Platzer, "MARVIN: Efficient and Comprehensive Mobile App Classification through Static and Dynamic Analysis," 2015 IEEE 39th Annual Computer Software and Applications Conference, Taichung, 2015, pp. 422-433.
  - Song, Yihang, and Urs Hengartner. "Privacyguard: A vpn-based platform to detect information leakage on android devices." Proceedings of the 5th Annual ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices. ACM, 2015.
- 