Title: Poster: NATICUSdroid: A malware detection framework for Android using native and custom permissions

Authors: Akshay Mathura, Laxmi Mounika Podila, Keyur Kulkarni, Quamar Niyaz, Ahmad Y. Javaid

aThe University of Toledo, 2801 W Bancroft St, Toledo, OH 43606, USA
bPurdue University Northwest, 2200 169th St, Hammond, IN 46323, USA

Date: Available online 13 January 2021


Abstract: The rapid growth of Android apps and its worldwide popularity in the smartphone market has made it an easy and accessible target for malware. In the past few years, the Android operating system (AOS) has been updated several times to fix various vulnerabilities. Unfortunately, malware apps have also upgraded and adapted to this evolution. The ever-increasing number of native AOS permissions and developers’ ability to create custom permissions provide plenty of options to gain control over devices and private data. Therefore, newly created permissions could be of great importance in detecting current malware. Previous popular works on malware detection used apps collected during 2010–2012 to propose malware detection and classification methods. A majority of permissions used in those apps are not as widely used or do not exist anymore. In this work, we present a novel malware detection framework for Android called NATICUSdroid, which investigates and classifies benign and malware using statistically selected native and custom Android permissions as features for various machine learning (ML) classifiers. We analyze declared permissions in more than 29,000 benign and malware collected during 2010–2019 to identify the most significant permissions based on the trend. Subsequently, we collect these identified permissions that include both the native and custom permissions. Finally, we use feature selection techniques and evaluate eight ML algorithms for NATICUSdroid to distinguish benign apps from malware. Experimental results show that the Random Forest classifier-based model performed best with an accuracy of 97%, a false-positive rate of 3.32%, and an f-measure of 0.96.

Link to PDF (available until March 04, 2021): https://authors.elsevier.com/c/1cPOP7tT2CiC-L

DOI: https://doi.org/10.1016/j.jisa.2020.102696
1. INTRODUCTION

- Malware infections increasing with popularity of Android devices
- Use of dated datasets and obsolete features in recent malware detection frameworks is alarming
- Need for a scalable system based on robust and significant features
- Permissions used as features before, but only “native” permissions are insufficient to classify good vs. bad

2. CONTRIBUTIONS

- Proposed and built Android malware detection framework, NATICUSdroid (NATive and CUStom permissions analysis for Android)
- Utilized native and custom (created by third-party app vendors) permissions of 29k+ apps
- Built additional baseline malware detection framework using native permissions
- Achieved better results compared to the state-of-the-art techniques
- Explained achieved results leveraging XAI (eXplainable Artificial Intelligence) [7]

3a. NATICUSdroid WORKFLOW

- Application Database
- Feature Extraction
- Feature Selection
- Classification
- Model Development

3b. METHODOLOGY

**Application Database:**
- Benign Apps:
  - 14630 API level 23+ apps from Androzoo,
  - rated benign by VirusTotal
- Malware Apps:
  - 14700 apps from Arguslab Android Malware Dataset (AMD)

**Feature extraction and dataset generation:**
- Extracted permissions using Androguard
- Generated two datasets:
  - only native permissions (Native)
  - native + custom permissions (Naticus)

**Feature Selection:**
- Frequency Counting
- Permission occurrences in apps
- Backward Elimination
- Insignificant permissions removed
- Multicollinearity Removal
- Only one of highly correlated permissions kept

<table>
<thead>
<tr>
<th>Step</th>
<th>Naticus permissions</th>
<th>Native permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Feature extraction</td>
<td>6751</td>
<td>125</td>
</tr>
<tr>
<td>Permission frequency counting</td>
<td>86</td>
<td>52</td>
</tr>
<tr>
<td>Backward elimination</td>
<td>58</td>
<td>39</td>
</tr>
<tr>
<td>Collinearity check</td>
<td>55</td>
<td>39</td>
</tr>
</tbody>
</table>

Permissions remaining after each selection step in the datasets

3c. CLASSIFICATION

- Single Learners
  - Logistic Regression (LR)
  - Random Forests (RF)
  - Extra Trees (ET)
  - XGBoost (XG)
  - AdaBoosting (AB)
- Ensemble Learners
  - Support Vector Machines (SVM)
  - Bagging (BG)
  - Accuracy + F-Score

4a. EXPERIMENTAL RESULTS

- ROC Curves for Naticus Dataset
- ROC Curves for Native Dataset

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Validation accuracy (%)</th>
<th>Detection accuracy (%)</th>
<th>F-Score</th>
<th>Training time (s)</th>
<th>Detection time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naticus</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Native</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SVM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LR</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RF</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ET</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AB</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>BG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Classification Results for Naticus and Native Datasets

4b. EXPERIMENTAL RESULTS

- Feature Importance in Train and Test sets of Naticus Dataset

5. CONCLUSION

- NATICUSdroid exhibits accuracy of 96.9% and F-Score of 0.97 in 0.11 seconds in detecting test dataset malware
- Capable of regular updates with newer native and custom permissions from newer Android versions

REFERENCES