“Hey, You, Get off of My Market”
Detecting Malicious Apps in Official and Alternative Android Markets

Paper Presented by: Przemysław Warias
Overview

- Evaluate Android Markets (Official and Unofficial)
  - Health of Markets
- DroidRanger
Motivation

- Android was exploding in popularity
- No comprehensive study of Android Markets was done yet
- Malware detection was signature based
Technical Details

- DroidRanger
  - Permission based behavioral footprinting
  - Heuristics based filtering scheme

a. Scalable, efficient and catches known malware
   - Distill known malware to a footprint: API call graph + Permissions
   - Data Flow Analysis: Detect fixed function parameters
   - Example: App has permission to android.provider.Telephony.SMS_RECIEVED and API calls abortBroadcast

b. Used to catch unknown malware
   - Detects dynamic loading of new (untrusted) code, either java binary or native machine code
Technical Details Diagrams

Permission Based Behavioral Footprinting

- Permissions
- API call Graph
- Data Flow Analysis (Fixed Function Parameters)
- Class Structure

Dynamic Code Loading

- Bytecode
- Native (Check if loaded into non standard directory)

Dynamic Execution Monitor

- Syscalls (native)
- API Calls
Evaluation

- Collected 200,000+ applications from 5 different markets
  - Android Marketplace (75% of samples)
  - eoeMarket
  - alcatelclub
  - Gfan
  - Mmoovv
- 10 known Malwares used
Results

- **171 infected apps found (119 unique)**
  - Android Market: 32 apps ~0.02% infection rate
  - Other markets 0.20%-0.47%
- **Two zero day malware found: Plankton (11 apps on AM) and DroidKungFu**
  - Plankton found using dynamic code loading (jar file)
  - DroidKungFu found because of suspicious syscalls

Table 7: The missed known malware families by Lookout Security & Antivirus software (T, D, and M represent the total, detected, and missed number of samples, respectively.)

<table>
<thead>
<tr>
<th></th>
<th>ADRD</th>
<th>Bgserv</th>
<th>jSMHider</th>
<th>BaseBridge</th>
<th>Pjapps</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>version 6.3 (r8472)</strong></td>
<td>T</td>
<td>D</td>
<td>M</td>
<td>T</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>8</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>31</td>
<td>15</td>
<td>16</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Table 8: Two zero-day malware families detected by DroidRanger

<table>
<thead>
<tr>
<th>Malware</th>
<th>Official Android Market</th>
<th>Alternative Markets</th>
<th>Total</th>
<th>Distinct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plankton</td>
<td>11</td>
<td>0 0 0 0</td>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>DroidKungFu</td>
<td>0</td>
<td>9 10 1 9</td>
<td>29</td>
<td>18</td>
</tr>
<tr>
<td>Total</td>
<td>11</td>
<td>9 10 1 9</td>
<td>40</td>
<td>29</td>
</tr>
</tbody>
</table>
Contribution

- Showed Android Market has an order of magnitude lower infection rate
- First comprehensive study of the Android Marketplace, and first comparison to alternative markets
- 2 Zero Day Malwares
- Showed there is a need for a more vigorous app vetting process in both official and unofficial marketplaces
Limitations

- Only sampled free apps
- Not applicable to other app stores such as IOS
- Only used two basic heuristics to uncover zero day malware
References

- Enck et al.
Thank you!

Questions?