ANDetect: A Third-party Ad Network Libraries Detection Framework for Android Applications

Xinyu Liu\textsuperscript{1,2}, Ze Jin\textsuperscript{1,2}, Jiaxi Liu\textsuperscript{1,2}, Wei Liu\textsuperscript{1,2}, Xiaoxi Wang\textsuperscript{1,2}, Qixu Liu\textsuperscript{1,2}

\textsuperscript{1}Institute of Information Engineering, Chinese Academy of Sciences
\textsuperscript{2}School of Cyber Security, University of Chinese Academy of Sciences

December 5, 2023
Advertising Networks

1. Provide Ad Resources
2. Register
3. Download
4. Request
5. Send Ads

Host APP  →  Ad Networks  →  Ad Publisher

Meta  ▷  APPLOVIN  ▷  Google AdMob  ▷  Yandex
Why Detect Ad Network Libs?

A1: Protect user's privacy.

Tracking → Analyzing → User Portrait

APPs

Read Identifier → Display Ads
Upload Identifier, Track User → Targeted Advertising

IMEI
IMSI
MAC ...

...
Why Detect Ad Network Libs?

A2: Identify adware effectively.

**Adware** is a type of malicious software that secretly installs itself on your device and displays unwanted advertisements and pop-ups.
APP Obfuscation and Encryption

- Identifier Renaming
- String Encryption
- Package Flattening
- Dead Code Removal
- Dex Encryption

APP Obfuscation

APP Encryption

- allatori
- Dasho
- DexGuard
- ProGuard
- Qihoo
- APKProtect
Recognize TPL
- LibRadar (ICSE’ 16)
- LibScout (CCS’ 16)
- LibD (ICSE’ 17)
- LibLoad (TMC’ 22)

The essence of these works is whitelist-based method.

None of these works can detect the ad libs in an encrypted APK.
Challenges

C1: Some integrated ad network libraries may be incorporated by many single ad libraries. Traditional whitelist-based methods fail.

Google AdMob  INMOBI  com.qq.e.ads  Chartboost
C2: The encryption of APP is always used by developers to enhance security and deter reverse engineering.
**C1:** Some integrated ad network libraries may be incorporated by many single ad libraries. Traditional whitelist-based methods fail.

- Extract the features of system API.
  - API count
  - Distribution across classes
  - Distribution across methods
- Train 3 Xgboost models with Dataset\textsubscript{AN} and Dataset\textsubscript{NAN}.
  - 833 ad libraries
  - 2758 non-ad libraries
- Integrate the models and classify every third-party module.
C2: The encryption of APP is always used by developers to enhance security and deter reverse engineering.

- Utilize the **resource features** of encrypted APPs.

### Resource Features
- **AndroidManifest.xml**
- files in **res**
- files in **assets**

### Key-value Mapping
- resource file
- elements path
- attribute
Q: How to choose suitable resource features and how to match?

• **Universality.**
  \[
  U_k = \frac{\sum l \cdot I_{n_k,k_i}}{l}, \quad I_{n_k,k_i} = \begin{cases} 
  1, & k \in K_i \\
  0, & k \not\in K_i 
  \end{cases}
  \]

• **Differentiation.**
  \[
  d_{o,i} = \max_{e \in o} \left\{ \frac{f_{e,i} \cdot \ln l}{\sum_{e' \in e} f_{e',i} + \sum_{j=1}^{l} I_{n_e,e}} \right\}
  \]
  \[
  D_k = \frac{\sum_{l=1}^{l} \sum_{o \in M_{k,i}} d_{o,i}}{\sum_{l=1}^{l} |M_{k,i}|}
  \]

• **Insensitivity.**
  \[
  IS_k = \frac{\sum_{l=1}^{l} |a_{n_k}|}{|a_{n_k}|}
  \]

---

**Table 5: The key, universality, differentiation and insensitivity of each feature.**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Key</th>
<th>(U_k)</th>
<th>(D_k)</th>
<th>(IS_k)</th>
</tr>
</thead>
<tbody>
<tr>
<td>F1</td>
<td>AndroidManifest.xml_manifest_package</td>
<td>1.0000</td>
<td>0.7294</td>
<td>0.7913</td>
</tr>
<tr>
<td>F2</td>
<td>AndroidManifest.xml_manifest uses-permission android:name</td>
<td>0.5385</td>
<td>0.1122</td>
<td>0.3215</td>
</tr>
<tr>
<td>F3</td>
<td>AndroidManifest.xml_manifest application provider androidauthorities</td>
<td>0.1186</td>
<td>0.0517</td>
<td>0.7953</td>
</tr>
<tr>
<td>F4</td>
<td>AndroidManifest.xml_manifest application receiver intent-filter action android:name</td>
<td>0.1602</td>
<td>0.0648</td>
<td>0.7222</td>
</tr>
<tr>
<td>F5</td>
<td>AndroidManifest.xml_manifest application activity android:name</td>
<td>0.3960</td>
<td>0.4886</td>
<td>0.4251</td>
</tr>
<tr>
<td>F6</td>
<td>AndroidManifest.xml_manifest application service android:name</td>
<td>0.1609</td>
<td>0.3541</td>
<td>0.5326</td>
</tr>
<tr>
<td>F7</td>
<td>AndroidManifest.xml_manifest application receiver android:name</td>
<td>0.1943</td>
<td>0.3261</td>
<td>0.6667</td>
</tr>
<tr>
<td>F8</td>
<td>res/values/values.xml_resources dimen_name</td>
<td>0.1350</td>
<td>0.1666</td>
<td>0.1018</td>
</tr>
<tr>
<td>F9</td>
<td>res/values/values.xml_resources string_name</td>
<td>0.3770</td>
<td>0.2886</td>
<td>0.2112</td>
</tr>
<tr>
<td>F10</td>
<td>res/values/values.xml_resources declare-styleable attr name</td>
<td>0.1302</td>
<td>0.3541</td>
<td>0.1628</td>
</tr>
<tr>
<td>F11</td>
<td>res/values/values.xml_resources style_name</td>
<td>0.2529</td>
<td>0.1062</td>
<td>0.3273</td>
</tr>
</tbody>
</table>

---

**Address the Challenge of Encrypted APPs**

**Global search**

**Full-path match**

**No-path match**

**Image pool**

**Comprehensive scoring**
Architecture of ANDetect

Decoupling

Ad Recognition

Profiling

SW Similarity

Profiling

Encryption Detection

Sub-Modules

Advertising Behavior

Class Profile

Class Profile

Resources Files

Resource Profile

Resource Profile

Resources Analysis

Profiling

Profiling

Resources

Result

Global search

Full-path limitation

No-path limitation

Image pool limitation

traverse

generate

input

generate

input

generate

pass

fail

pass

fail

Encryption

Detection

Decrypted APK

Res

Encrypted APK

F₁

F₂

F₃

...

F₁₂

Resources Analysis

Profiling

Profiling

AN

AN

AN

F₁

F₂

F₃

...
Architecture of ANDetect

Decoupling
- Louvain

Ad Recognition
- Xgboost

Profiling
- Edge
- Field
- Twig
- Method

SW Similarity
- Global search
- Full-path limitation
- No-path limitation
- Image pool limitation

Encryption Detection

Encrypted APK
- Resources Files
- Resource Profile
- Result

Resources Analysis
- Profiling

Input
- Generate
- Traverse
- Generate

Generate
- Result
- Pass
- Fail
Architecture of ANDetect

Decoupling
- Sub-Modules: [Louvain]

Advertising Behavior
- Ad Recognition: [Xgboost]

Class Profile
- Profiling
  - Edge
  - Field
  - Method

SW Similarity
- Scoring

Resources Files
- Resources Analysis
  - Resources Parameter:
    - $F_1$
    - $F_2$
    - $F_3$
    - ...

Resource Profile
- Profiling
  - Global search
  - Full-path limitation
  - No-path limitation
  - Image pool limitation

Result
- Profiling
- Class Profile

Encryption Detection
- Encrypted APK
- Resources Files
- Resource Profile
- Profiling

AN
- Injection

Encryption
- Decryption
- Encryption

Ad Recognization
- Decoupling
- Ad Recognition
- Profiling

Decoupling
- Decoupling
- Profiling
Architecture of ANDetect

Decoupling

Ad Recognition

Profiling

Class Profile

SW Similarity

Resources Analysis

Profiling

Encryption Detection

Encrypted APK

Resources Files

Resource Profile

Result

Global search
Full-path limitation
No-path limitation
Image pool limitation

Resources Analysis

Profiling

Scoring

Profiling

AN

unencrypted

encrypted
## Datasets

### Evaluation metrics
- Fine-grained True Positive (TP-FG)
- Coarse-grained True Positive (TP-CG)
- Precision
- Recall

### Datasets
- Android Application Dataset (20,000 adwares & 125,401 unlabeled APPs)
  - Low-confidence labeled datasets
  - High-confidence labeled datasets
- Ad Networks Dataset (833 Ad libs & 2758 non-ad libs)
Evaluation

Performance of ANDetect:

- Leveraging a 2:1 mix of non-encrypted and encrypted applications.
- Compare with LibD, LibRadar and LibScout.
- Low Precision: extra ad libs
- High Recall: competitors’ misclassification of \texttt{com.google.android.gms.ads}

<table>
<thead>
<tr>
<th>Tools</th>
<th>TP-CG</th>
<th>TP-FG</th>
<th>Precision</th>
<th>Recall</th>
<th>Time(s)/AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LibD</td>
<td>46.08%</td>
<td>78.31%</td>
<td>99.72%</td>
<td>78.31%</td>
<td>58.14</td>
</tr>
<tr>
<td>LibRadar</td>
<td>48.04%</td>
<td>77.86%</td>
<td>99.50%</td>
<td>77.86%</td>
<td>29.37</td>
</tr>
<tr>
<td>LibScout</td>
<td>25.49%</td>
<td>72.64%</td>
<td>99.54%</td>
<td>72.64%</td>
<td>74.54</td>
</tr>
<tr>
<td>ANDetect</td>
<td>95.10%</td>
<td>95.97%</td>
<td>95.84%</td>
<td>95.97%</td>
<td>34.07</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Tools</th>
<th>TP-CG</th>
<th>TP-FG</th>
<th>Precision</th>
<th>Recall</th>
<th>Time(s)/AP</th>
</tr>
</thead>
<tbody>
<tr>
<td>LibD</td>
<td>47.40%</td>
<td>62.67%</td>
<td>98.73%</td>
<td>62.67%</td>
<td>153.45</td>
</tr>
<tr>
<td>LibRadar</td>
<td>43.35%</td>
<td>59.69%</td>
<td>98.26%</td>
<td>59.69%</td>
<td>65.11</td>
</tr>
<tr>
<td>LibScout</td>
<td>32.37%</td>
<td>53.47%</td>
<td>99.79%</td>
<td>53.47%</td>
<td>246.51</td>
</tr>
<tr>
<td>ANDetect</td>
<td>93.64%</td>
<td>93.08%</td>
<td>97.29%</td>
<td>93.08%</td>
<td>69.24</td>
</tr>
</tbody>
</table>
Evaluation

Robustness in encrypted APP detection:

- Randomly select 100 non-encrypted APPs with resource confusion.
- Tagged by the combined outputs of LibScout, LibD and LibRadar (Low-confidence)
- Identify the ad libs through ANDetect’s encrypted APP detection module

- **Resource confusion properties:**
  - `res` directory is absent;
  - over half of the file names in `res` comprise less than 5 characters;
  - over half of the file names in `color` or `xml` comprise less than 5 characters;

<table>
<thead>
<tr>
<th>Dataset</th>
<th>size</th>
<th>TP-CG</th>
<th>TP-FG</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>AP\textsubscript{adware}</td>
<td>100</td>
<td>94.00%</td>
<td>86.24%</td>
<td>81.84%</td>
<td>86.24%</td>
</tr>
<tr>
<td>AP\textsubscript{truth}</td>
<td>100</td>
<td>94.74%</td>
<td>91.50%</td>
<td>72.27%</td>
<td>91.50%</td>
</tr>
</tbody>
</table>

Table 3: The result of encrypted application detection in resource confused applications.
Evaluation

Novel ad libraries:

- ANDetect’s non-encrypted application detection module
- Discern the true ad libs:
  - define ad association weight
  - search new lib name in Google
  - compute associate weight
- Allow for false positive to identify more novel ad libs
- 16 novel ad libs

Figure 5: The novel ad libraries detected by ANDetect. The number of genuine ALs and false positives from $AP_{\text{adware}}$ and $AP_{\text{truth}}$ are given respectively.
Novel adwares:
- Label 20,000 adwares using VT
- Relationship between ad libs and adware

Table 4: The candidates of malicious ALs.

<table>
<thead>
<tr>
<th>Candidate ALs</th>
<th>VirusTotal’s Label</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.adsmogo, net.youni.android, com.mopub,mobileads, com.inmobi, ads, com.nd.dianjin, cn.smartmad, cn.domob, android, com.kuguo.ad, com.jirbo.adcolony, com.sinth.woodroid, org.iqiyi, video, com.revmob</td>
<td>Adware, ANDROID, AdMogo, Android, Adware, Youmi, A, AdLibrary, MoPub, Android, Adware, iOS, AdMob, Android, MobileAds, AdMogo, Adware, Kuguo, A, Ad, AdColony! Android, AdWARE/ ANDRAS, AdW, Android, OGS, FAN, Gen</td>
<td>0.50, 0.47, 0.45, 0.80, 0.50, 0.12, 0.26, 1.00, 0.50, 0.43, 1.00, 0.20</td>
</tr>
</tbody>
</table>

Table 7: The brand-new adware detected by ANDectect. The sha256 of each adware and malicious ALs it contains is provided.

<table>
<thead>
<tr>
<th>Malicious ALs</th>
<th>Brand-new ALs</th>
</tr>
</thead>
<tbody>
<tr>
<td>com.mopubmobiles</td>
<td>com.immobilads, com.mopubmobiles</td>
</tr>
<tr>
<td>com.AdW</td>
<td>com.immobilads, com.mopubmobiles</td>
</tr>
<tr>
<td>com.AdwareMobiles</td>
<td>com.immobilads, com.mopubmobiles</td>
</tr>
<tr>
<td>com.Adw</td>
<td>com.immobilads, com.mopubmobiles</td>
</tr>
</tbody>
</table>

59 brand new adware

Evaluation

Incorrect answer: There is no incorrect answer as the evaluation is not provided in the image.
Summary

• Limitation
  • Resource features do not have a clear differentiation effect like class features, which is reflected in different versions of ad libraries.
  • Not all advertising libraries have meaningful resource features.

• Conclusion
  • We introduce ANDetect, a pioneering tool designed to efficiently identify ad libraries in Android applications, overcoming the limitations of existing methods especially when dealing with encrypted apps. The experiment of ANDetect to over 140,000 apps has resulted in the detection of 16 novel ad libraries and 53 new adwares.

https://sites.google.com/view/andetect
Thank you!

Q&A