Statistical Deobfuscation for Android Applications

Benjamin Bichsel
Veselin Raychev
Petar Tsankov
Martin Vechev

Department of Computer Science
Why De-obfuscate?

Android binaries (APKs) (no code available)

Number of APKs on Google Play

Google Play

2.4M APKs
Non-descriptive names

Names provide key semantic information

Some names remain
Layout Obfuscation in Android

Names provide key semantic information

Security Challenges
- Code Inspection
- Third-party Library Detection
... many others

Non-descriptive names

package com.example.dbhelper

class DBHelper extends SQLiteHelper {
    SQLiteDatabase db;
    public DBHelper(Context ctx) {
        db = getWritableDatabase();
    }
    Cursor execSQL(String str) {
        return db.rawQuery(str);
    }
}

package a.b.c

class a extends SQLiteHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDatabase();
    }
    Cursor c(String str) {
        return b.rawQuery(str);
    }
}

Some names remain...
Some names remain

Can we reverse layout obfuscation?

Names provide key semantic information

Non-descriptive names

package com.example.dbhelper

class DBHelper extends SQLiteHelper {
  SQLiteDatabase db;
  
  public DBHelper(Context ctx) {
    db = getWritableDatabase();
  }

  Cursor execSQL(String str) {
    return db.rawQuery(str);
  }
}

package a.b.c

class a extends SQLiteHelper {
  SQLiteDatabase b;
  
  public a(Context ctx) {
    b = getWritableDatabase();
  }

  Cursor c(String str) {
    return b.rawQuery(str);
  }
}
Layout Obfuscation in Android

package com.example.dbhelper

class DBHelper extends SQLiteHelper {
    SQLiteDatabase db;

    public DBHelper(Context ctx) {
        db = getWritableDatabase();
    }

    Cursor execSQL(String str) {
        return db.rawQuery(str);
    }
}

package a.b.c

class a extends SQLiteHelper {
    SQLiteDatabase b;

    public a(Context ctx) {
        b = getWritableDatabase();
    }

    Cursor c(String str) {
        return b.rawQuery(str);
    }
}

Non-descriptive names

Names provide key semantic information. Yes, with roughly 80% accuracy!
Released last week, so far:

> 5K users
> 5GB APKs

Reddit posts/comments

[-] evantatarka WillowTree 3 points 2 days ago
Nice! This should help with debugging issues in play services or other libs that are obfuscated just to make my life harder a bit easier.

[-] oleeEncantado 2 points 4 days ago
Works quite well, I've tried on some small games.

[-] Tycon712 3 points 2 days ago
Can someone tell me what the point of using Proguard is if there are tools out there like this?

[-] theheartbeatpug 6 points 2 days ago
As far as I know, this is brand new. I asked the creator of ProGuard a week ago how hard it is to unobfuscate code after it's run through proguard. He said it strips all the names out of the code so it's essentially impossible. I'm super impressed by what they've done here.

Tweets

D|ARS|IN @dharshin · Oct 17
Deobfuscate #proguard 'ed APKs: apk-deguard.com. The paper on the inner workings: srl.inf.ethz.ch/papers/deguard... #Android #MobileSecurity

Brian Carpenter @geeknik · 9h
Android Deobfuscation with Machine Learning reverses the effects of ProGuard [PDF] srl.inf.ethz.ch/papers/deguard...

rvivek @vivek_310 · Oct 17
*Android Deobfuscation with Machine Learning reverses the effects of ProGuard* #security #feedly

Android Deobfuscation with Machine... · /r/Reverse...
8 points and 1 comments so far on reddit
How Does DeGuard Work?
DeGuard: System Overview

Prediction Phase

Static analysis

MAP Inference

Transform

De-obfuscated Code

Obfuscated Code

Learning Phase

Static analysis

Training

Probabilistic model

Semantic representation

Open-source, unobfuscated applications

class a extends SQLiteHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDatabase();
    }
}

class DBHelper extends SQLiteHelper{
    SQLiteDatabase db;
    public DBHelper(Context ctx) {
        db = getWritableDatabase();
    }
}
Probabilistic Graphical Models
class a extends SQLiteHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDB();
    }
}

\[
P(\hat{\theta} | \tilde{\theta}) = \\
= P(a, b | SQLiteHelper, getWritableDB) \\
= \frac{1}{Z} \exp(0.3 \cdot f_1(SQLiteHelper, a) \\
+ 0.2 \cdot f_2(SQLiteHelper, a) + \cdots)
\]
**Probabilistic Graphical Models**

Next

How are the weights and features learned?

\[
P(\vec{\theta} | \vec{K}) = P(a, b | \text{SQLiteHelper, getWritableDB})
\]

\[
= \frac{1}{Z} \exp(0.3 \cdot f_1(\text{SQLiteHelper, a}) + 0.2 \cdot f_2(\text{SQLiteHelper, a}) + \cdots)
\]
Learning
Learning

Unobfuscated APKs

Feature templates

Static analysis

Dependency graphs

<table>
<thead>
<tr>
<th>name1</th>
<th>name2</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>$f_2$</td>
</tr>
<tr>
<td>$f_3$</td>
<td>$f_4$</td>
</tr>
<tr>
<td>$f_5$</td>
<td>$f_6$</td>
</tr>
<tr>
<td>$f_7$</td>
<td>...</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>name1</th>
<th>name2</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f_1$</td>
<td>SQLiteHelper</td>
<td>DBUtils</td>
</tr>
<tr>
<td>$f_2$</td>
<td>SQLiteHelper</td>
<td>DBHelper</td>
</tr>
<tr>
<td>$f_3$</td>
<td>getWritableDB</td>
<td>db</td>
</tr>
<tr>
<td>$f_4$</td>
<td>getWritableDB</td>
<td>instance</td>
</tr>
<tr>
<td>$f_5$</td>
<td>DBUtils</td>
<td>instance</td>
</tr>
<tr>
<td>$f_6$</td>
<td>DBHelper</td>
<td>db</td>
</tr>
<tr>
<td>$f_7$</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Features (with candidate names)

>100,000

Actual graphs have >1,000 nodes

Static analysis

Train Model

Compute weights that maximize $P(\hat{\theta} = \overrightarrow{o_i} | \overrightarrow{k} = \overrightarrow{k_i})$ for all training samples $(\overrightarrow{o_i}, \overrightarrow{k_i})$.
DeGuard: System Overview

```
class a extends SQLiteHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDB();
    }
}
```

```
class DBHelper extends SQLiteHelper {
    SQLiteDatabase db;
    public DBHelper(Context ctx) {
        db = getWritableDB();
    }
}
```

Prediction Phase

Static analysis

MAP Inference

Transform

Learning Phase

Static analysis

Training

Probabilistic model

\( P(C|A) \)

Obfuscated Code

De-obfuscated Code

Open-source, unobfuscated applications
DeGuard: System Overview

Obfuscated Code

class a extends SQLiteHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDatabase();
    }
}

De-obfuscated Code

class DBHelper extends SQLiteHelper {
    SQLiteDatabase db;
    public DBHelper(Context ctx) {
        db = getWritableDatabase();
    }
}

Static analysis

MAP Inference

Transform

Prediction Phase

Probabilistic model $P(\square | \square)$
class a extends SQLiteHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDatabase();
    }
}

Prediction Phase

Obfuscated Code
class a extends SQLiteOpenHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDatabase();
    }
}

MAP Inference

\[ \tilde{\sigma} = \arg\max \ P(\tilde{\sigma} = \sigma' \mid \tilde{K} = \tilde{k}) \]

\[ \sigma' \in \Omega \]

<table>
<thead>
<tr>
<th>Candidate assignment ( \tilde{\sigma} )</th>
<th>( P(\tilde{\sigma} \mid \tilde{k}) )*</th>
</tr>
</thead>
<tbody>
<tr>
<td>a = DBUtils</td>
<td>b = instance</td>
</tr>
<tr>
<td>a = DBHelper</td>
<td>b = db</td>
</tr>
<tr>
<td>a = DBUtils</td>
<td>b = db</td>
</tr>
<tr>
<td>a = DBHelper</td>
<td>b = instance</td>
</tr>
</tbody>
</table>

*Non-normalized
class a extends SQLiteHelper
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDB();
    }
}

MAP Inference
\[ \hat{\phi} = \text{argmax} P(\bar{\phi} = \phi' \mid \bar{K} = \bar{k}) \]
\[ \phi' \in \Omega \]

<table>
<thead>
<tr>
<th>Candidate assignment ( \hat{\phi} )</th>
<th>( P(\hat{\phi} \mid \bar{k}) )*</th>
</tr>
</thead>
<tbody>
<tr>
<td>a = DBUtils b = instance</td>
<td>1.2</td>
</tr>
<tr>
<td>a = DBHelper b = db</td>
<td>1.3</td>
</tr>
<tr>
<td>a = DBUtils b = db</td>
<td>0.8</td>
</tr>
<tr>
<td>a = DBHelper b = instance</td>
<td>1.2</td>
</tr>
</tbody>
</table>

*Non-normalized
Prediction Phase

Obfuscated Code

class a extends SQLiteHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDB();
    }
}

Deobfuscated Code

class DBHelper extends SQLiteHelper {
    SQLiteDatabase db;
    public DBHelper(Context ctx) {
        db = getWritableDB();
    }
}
Preserving Semantics

Freely renaming fields/variables/methods may change the program semantics

Syntactic constraints
  e.g. fields within a class must have distinct names

Semantic constraints
  e.g. method overloads must be preserved

class A
  int a
  Object b
  void a()

class B
  extends
  void b()
  void c(A a)

must have distinct names
must not override method a()
DeGuard: System Overview

Prediction Phase

- Obfuscated Code
- De-obfuscated Code

Learning Phase

- Open-source, unobfuscated applications
- Static analysis
- Training

Static analysis

MAP Inference

Transform

- Static analysis
- SQLiteHelper
- SQLiteDatabase
- a
- b
- getWritableDatabase

class a extends SQLiteHelper {
    SQLiteDatabase b;
    public a(Context ctx) {
        b = getWritableDatabase();
    }
}

class DBHelper extends SQLiteHelper {
    SQLiteDatabase db;
    public DBHelper(Context ctx) {
        db = getWritableDatabase();
    }
}
DeGuard Implementation

Static Analysis

- Static analysis framework for Java and Android

Learning and MAP Inference

- Scalable open-source framework for structured prediction
  - Open-source: http://nice2predict.org
- Training data: 2K open-source, unobfuscated Android applications
Evaluation

1. Can DeGuard reverse ProGuard?
2. Can DeGuard detect third-party libraries?
3. Is DeGuard useful for malware inspection?
ProGuard Experiment

Source Code → ProGuard → Compile → Non-obfuscated APK

Obfuscated APK → DEGUARD → De-obfuscated APK
After Obfuscation

<table>
<thead>
<tr>
<th></th>
<th>Fields</th>
<th>Methods</th>
<th>Classes</th>
<th>Packages</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of program elements</td>
<td>20</td>
<td>4</td>
<td>0</td>
<td>60</td>
<td>80</td>
</tr>
</tbody>
</table>

Only 13% known names
Can DeGuard reverse ProGuard?

- 80.6% correct names
- 1.6% known names
- 80% of the names are identical to the original ones

Package names are directly used to predict third-party libraries.
Can DeGuard Detect Third-Party Libraries?

ProGuard obfuscates library package names

Precision: 93.1%  Recall: 91%
Is DeGuard Useful for Malware Inspection?

De-obfuscating samples from the Android Malware Genome Project

Malware Sample

class d {
    String a = System.getProperty(..)
    char[] b;
    byte[] c;
    byte[] a(String) {}
}

De-obfuscated Malware Sample

class Base64 {
    String NL = System.getProperty(..)
    char[] ENC;
    byte[] DEC;
    byte[] decode(String) {}
}

- Reveals string decoders
- Reveals classes that handle sensitive data (e.g. Location)
- Hard to handle heavily-obfuscated code (e.g. reflection)
Summary

Try online: www.apk-deguard.com

More info: http://www.srl.inf.ethz.ch/spas