カメラを秘密裏に濫用するAndroidアプリの検出

Detection of Android apps that secretly abuse the camera

† 早稲田大学 基幹理工学部 情報理工学科

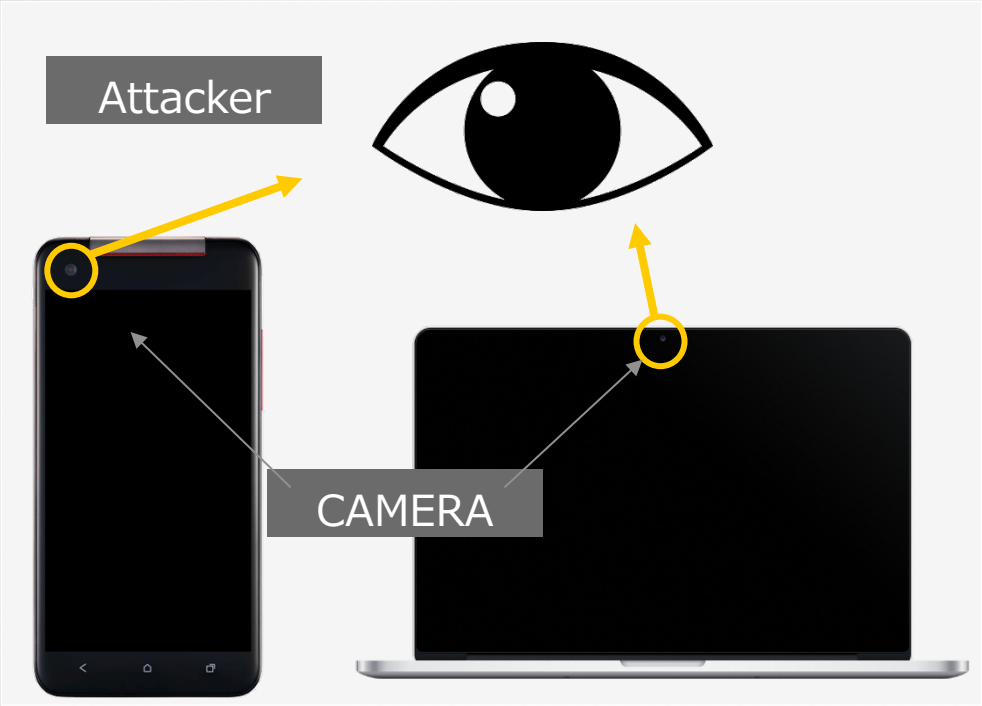
渡邉 卓弥† Takuya Watanabe
森 達哉† Tatsuya Mori
酒井 哲也† Tetsuya Sakai

2014/03/28 ICSS 情報通信システムセキュリティ研究会

研究の背景

Background

研究の背景　Background



スマートフォンやパソコンの
カメラからユーザーの生活が
覗かれてしまう！

気づかない内に

アプリケーションが映像を外部に送っても
ユーザーが気づくのは困難

実際の被害事例

・2010年 ドイツ　- 被害者：学生150人
・2013年 アメリカ - 被害者：ミス・ティーン

今後，世界的に拡大するのでは？

研究の背景　Background

◆関連研究

PlaceRaider

・NDSS Symposium 2013
・Robert Templeman, Zahid Rahman, David Crandall, Apu Kapadia

・スマートフォンカメラから取得した静止画の断片を再構築し，三次元空間を再現することができるマルウェアのProof of Concept.

・個人情報，企業機密の漏洩や，軍事的な利用まで，スマートフォンカメラ悪用がもたらす危険性を示している.

研究の背景　Background

◆Androidはなぜ危険か

・パーミッション確認が甘い
　インストール時に一度だけ確認

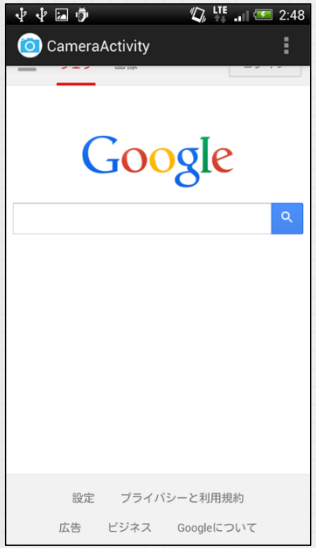
・既存アプリのリパッケージングが容易
　人気アプリに悪用ツールを同梱できる

・撮影音なしで撮影が可能
　秘密裏の撮影に向いている

Proof of Conceptとして
実際にアプリを試作

研究の背景　Background

実際に試作したAndroidアプリ



何も動いていないように見えるが…

サーバー上

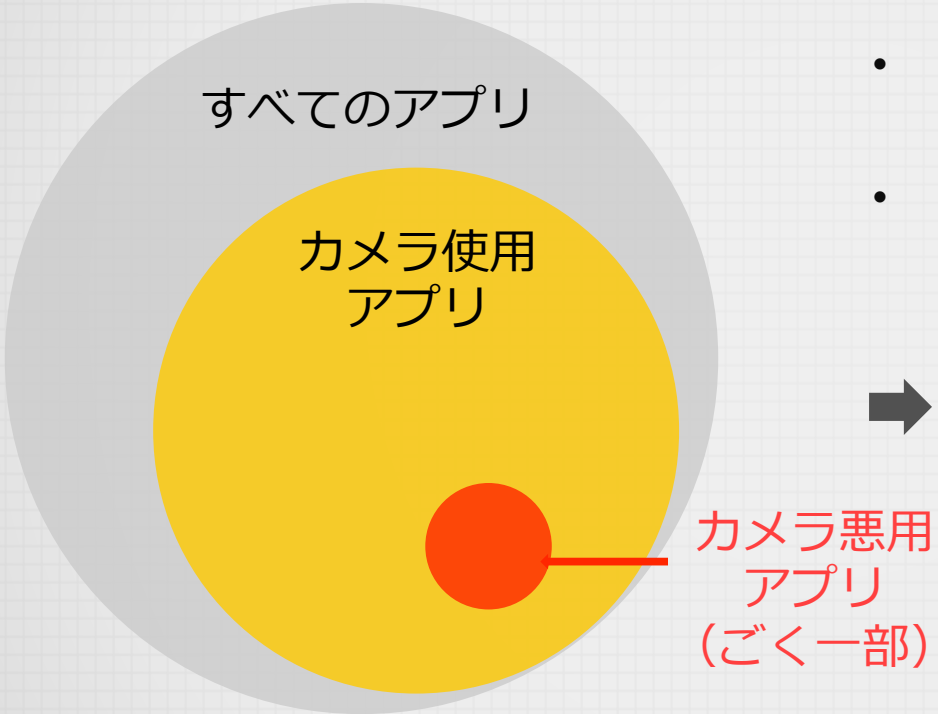
撮影した画像を順次アップロード

研究の目的

Purpose

1. 研究の背景
2. 研究の目的
3. アプローチ
4. 先行研究
5. 実動作の解析
6. 説明文の解析
7. 適用結果
8. まとめ

研究の目的　Purpose



- 世界にある膨大な数のアプリの中から，カメラを悪用するアプリを検出したい
- すべてのコードや動作を人の手や人の目で追うのは非現実的

➡ 自動的なスクリーニングが目的

アプローチ

Approach

1. 研究の背景
2. 研究の目的
3. アプローチ
4. 先行研究
5. 実動作の解析
6. 説明文の解析
7. 適用結果
8. まとめ

アプローチ　Approach

◆仮説

・疑わしいアプリを抽出するため，以下の仮説にもとづいてアプローチ

アプリには，カメラを利用するコードが含まれている

にもかかわらず

アプリの説明文にて，カメラを利用する旨が書かれていない

↓

秘密撮影の可能性が高い

アプローチ　Approach

- ２つの手法を組み合わせて，Androidアプリを解析する

逆アセンブルコード

```
.super Landroid/view/SurfaceView;
.source "CameraSurfaceView.java"

# interfaces
.implements Landroid/view/SurfaceHolder$Callback;


# static fields
.field private static final SDF:Ljava/text/SimpleDateFormat;


# instance fields
.field private final ReadPreviewImage:Landroid/hardware/Camera$Previe

.field camera:Landroid/hardware/Camera;

.field cameraParameters:Landroid/hardware/Camera$Parameters;

.field prevdata:[B
```

ディスクリプション

详细描述：

SuperSU（超级授权管理）是一款功能超强超级用户访问权限管理工具.Super
有应用的超级用户访问权限的高级管理,在另外一个超级用户管理工具的基础上,

主要功能：
- 超级用户访问权限提示
- 超级用户访问权限日志记录
- 超级用户访问权限通知
- 配置各应用的通知规则
- 永久取消ROOT权限
- 深度处理检测
- 可在Recovery中工作
- 可在Android系统未能正常启动时工作
- 可在不标准的Shell位置中工作
- 始终运行在幽灵模式中
- 提示时唤醒设备

アプローチ　Approach

CODE

実動作の解析
アプリの逆アセンブルコードから，利用されている関数を調査

DESC

説明文の解析
ディスクリプションがカメラ利用に言及しているか判別

以上の解析を組み合わせ，カメラについて
「言及してないのに使用している」
というアプリを抽出する
（秘密撮影のおそれあり）

		Description	
		言及あり	言及なし
CODE	使用あり		疑
	使用なし		

説明文に関する先行研究

Prior Research

先行研究　Prior Research

◆WHYPER

Rahul Pandita, Xusheng Xiao, Wei Yang,William Enck, and Tao Xie
22nd USENIX Security Symposium

- Why Permissionsの略
- 自然言語処理によって，ディスクリプションからパーミッションの利用を示唆するセンテンスを抽出.

本研究との違い

- 連絡先の読み取りや，マイクの録音，スケジュールの読み取りなどを対象
- 「言及していないのに利用している」アプリについては評価をしていない
- パーミッションだけに注目しており，コードの中身までは踏み込んでいない

先行研究　Prior Research

- パーミッションを特定するセンテンスの例

You can now turn recordings into ringtones.

Permission: RECORD AUDIO

これは，マイクを使用した音声取得のパーミッションである

実動作の解析

Code Analysis

1. 研究の背景
2. 研究の目的
3. アプローチ
4. 先行研究
5. 実動作の解析
6. 説明文の解析
7. 適用結果
8. まとめ

実動作の解析　Code Analysis

逆アセンブルコード (Smali)

```
.method public surfaceCreated(Landroid/view/SurfaceHolder;)V
    .locals 2
    .parameter "holder"

    .prologue
    .line 59
    const/4 v1, 0x1

    :try_start_0
    invoke-static {v1}, Landroid/hardware/Camera;->open(I)Landroid/hardware/Camera;

    move-result-object v1

    iput-object v1, p0, Lcom/iPentec/systemoverlay/CameraSurfaceV      >camera:Landroi

    .line 61
    iget-object v1, p0, Lcom/iPentec/systemoverlay/CameraSurfaceView;->camera:Landroi
```

カメラリソースの確保

open(I)Landroid/hardware/Camera

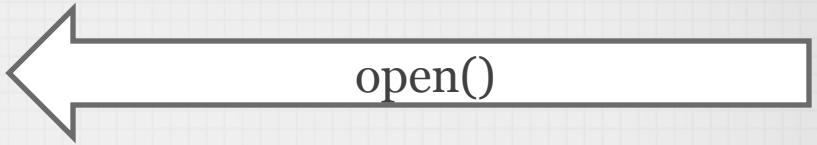
解析の流れ

- パーミッションでカメラの利用を許可しているものを対象
- Androidアプリの実行コードをbaksmaliというツールで逆アセンブル
- 関数の有無を検索し，カメラに関するコードが含まれているか判別

実動作の解析　Code Analysis

◆検索対象の関数について　（PoCアプリ作成によって得られた知見）

・ Androidでカメラデバイスを利用する際は，まずリソースの確保をおこなう

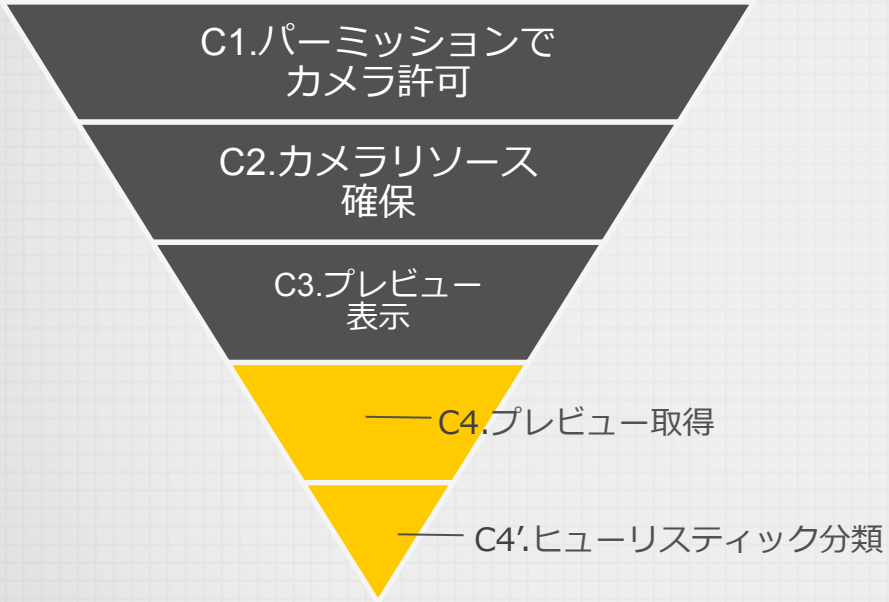


・カメラ撮影には，プレビュー画面を設置する必要がある
（このプレビュー画面を隠蔽するテクニックがある）

・利用者に気づかれないための無音撮影には，さらにプレビュー映像の取得をおこなう必要がある

実動作の解析　Code Analysis

◆呼び出される関数によるアプリの分類



C1
・パーミッションで宣言してるだけで，
実際にはカメラ使用なし

C2
・懐中電灯アプリなど　　open

C3
・撮影音ありの撮影　　setPreviewDisplay
・手鏡アプリなど

C4
・QRコードリーダー　　onPreviewFrame
・ビデオ通話
・秘密撮影

C4′
・C4のなかでも特に用途不明なもの
（クラス名などで判別）

実動作の解析　Code Analysis

◆C4’の抽出に用いたヒューリスティックの例

・C4には，秘密撮影アプリの他に，極めて多数の“QRコードリーダー機能”をもつアプリが含まれている

・QRコードリーダーには，”zxing”というライブラリが広く普及している

・カメラ機能のうち，パス名に”zxing”を含むものは，「カメラの用途が明確」と判断し，除外した

C1	・パーミッションで宣言してるだけで，実際にはカメラ使用なし
C2	・懐中電灯アプリなど
C3	・撮影音ありの撮影 ・手鏡アプリなど
C4	・QRコードリーダー ・ビデオ通話 ・秘密撮影
C4’	・C4のなかでも特に用途不明なもの（クラス名などで判別）

説明文の解析

Description Analysis

1. 研究の背景
2. 研究の目的
3. アプローチ
4. 先行研究
5. 実動作の解析
6. 説明文の解析
7. 適用結果
8. まとめ

説明文の解析　Description Analysis

ディスクリプション（説明文）

解析の流れ

- 中国のマーケットから検体を集めたため，中国語の説明文を対象
- 中国人学生４名の協力のもと，900個のディスクリプションをラベル付け
  - +1. カメラを使っていると言及されている
  - - 1. カメラを使っていると言及されていない
- これを教師データとし，ディスクリプションがカメラに言及しているかを自動判別するモデルを作成

説明文の解析 Description Analysis

ラベル付けのため作ったWEB画面

カメラへの言及ありと判断した人数

	0人	1人	2人	3人	4人
検体数	657	29	30	35	149

全員の回答が一致したのは 約 90 %

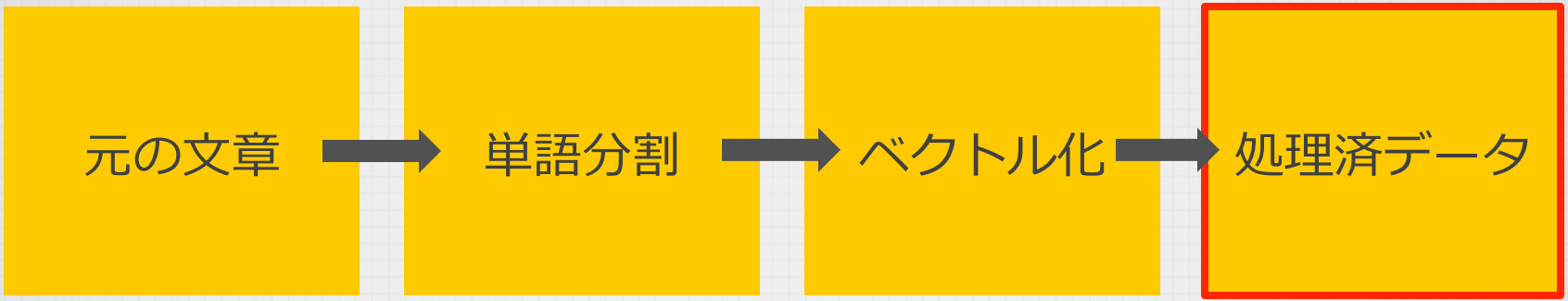
Fleiss のカッパ係数を計算すると，

$$\kappa = 0.83$$

非常に信頼度の高い回答結果といえる．

※カッパ係数とは，アンケート等における選択回答の一致率を示す指標．Fleiss方式では複数人の回答を評価でき，0.81以上は「ほぼ完全な一致」とされる．

ディスクリプションの前処理



マーケットから
収集した状態

KyTeaによる
単語分割

tf-idfを成分とした
ベクトル化

STOPWORD除去
DF Threshold

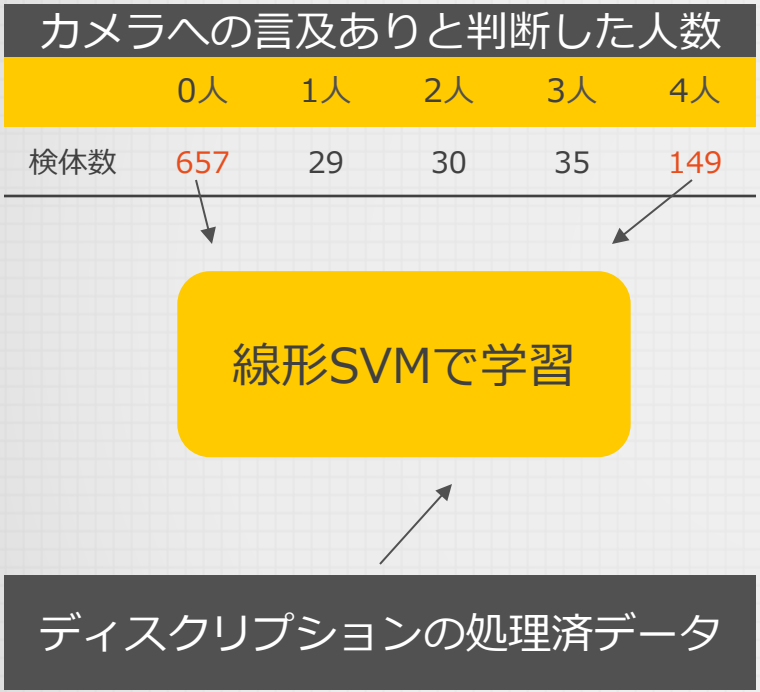
KyTea… 単語分割ツール

tf-idf…単語出現頻度による重み付け

STOPWORD…助詞のように意味を持たない語

DF Threshold…出現過多な単語の除去

説明文の解析 Description Analysis



- SVM（Support vector machine）は，教師あり学習を用いた識別手法
- SVMの実装はLIBLINEARを使用

チューニングの結果，

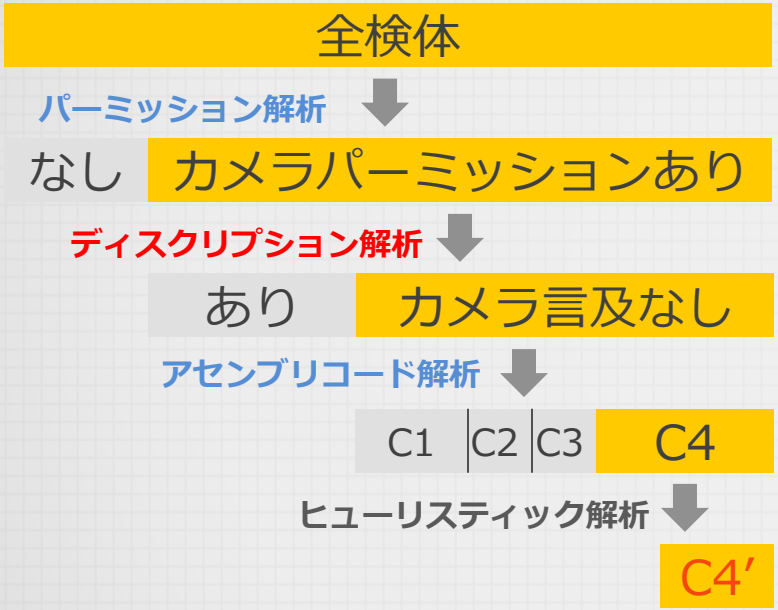
97%以上にまで精度が向上

これを，1万以上の検体の二値分類に用いる．

適用結果
Results

1. 研究の背景
2. 研究の目的
3. アプローチ
4. 先行研究
5. 実動作の解析
6. 説明文の解析
7. 適用結果
8. まとめ

適用結果 Results

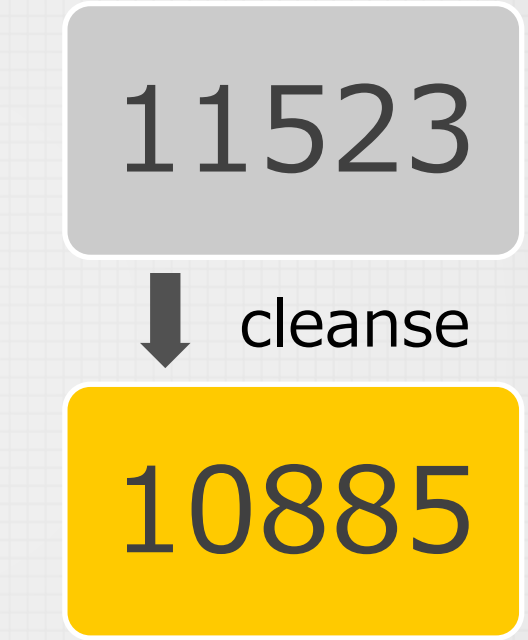


- 左図のフローでアプリをフィルタリングしていく
- 最終的に残ったC4’は，カメラ使用コードを含むのにその理由がわからないという，極めて疑わしいアプリである

適用結果 Results

◆適用検体について

- 中国のサードパーティアプリマーケットを対象
- 無作為に集めたアプリのうち，不完全なものやディスクリプションが存在しないものを除外



これを全検体とする

適用結果 Results

全検体
10,885
カメラパーミッションあり
2,922
カメラへの言及なし
2,408

c1	c2	c3	c4
1,264	61	115	968

c4'
43

c1:パーミッション
宣言してるだけの
カメラ不使用アプリ

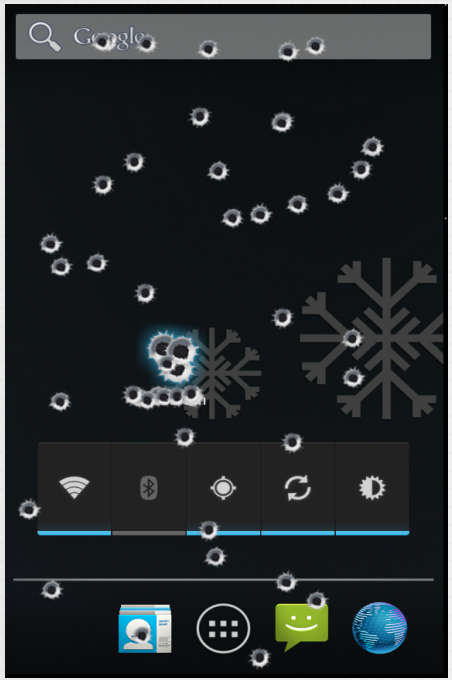
最も疑わしいC4‘の43検体について実際にアプリを動かして調査したところ,

- 28検体は正当な理由でカメラを使用
- 13検体は会員登録が必要で調査不能
- 2検体は完全に用途不明

また，Virus Totalにかけたところ,

18/43 検体がマルウェアとして検出された.

適用結果 Results



最終的に残ったアプリの例

抽出の結果，最後に残ったアプリの例．
画面をタッチするごとに，画面上に銃痕のようなものを刻み込む．
どこでカメラを利用するのか不明にも関わらず，カメラを利用するコードが含まれている．

精査が必要なアプリを抽出することに成功

なお，手動で詳細にコードを解析したところ，カメラで秘密裏に撮影したり，その画像を外部へと転送するような動作は見受けられなかった．

まとめ

Summary

1. 研究の背景
2. 研究の目的
3. アプローチ
4. 先行研究
5. 実動作の解析
6. 説明文の解析
7. 適用結果
8. まとめ

まとめ　Summary

CODE

実動作の解析
カメラを悪用するアプリが含む関数を，コードから特定

DESC

説明文の解析
ディスクリプションがカメラについて言及しているかを，97%の精度で自動判別

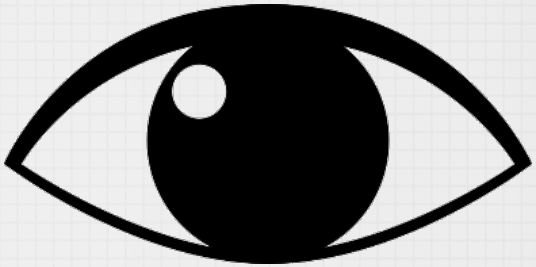
以上の解析を組み合わせ，1万以上のアプリから特に悪用の可能性をもつ43のアプリを抽出することができた.

今後の課題

◆コード解析を充実させ，より精密な悪用の検出を行う

◆カメラ以外の機能についても，同様の手法を適用する

おわり



補足資料

なぜAndroidなのか

◆市場的な理由

- 世界シェア１位のモバイルOS
  「カメラ悪用」と「モバイル」の親和性
- ハードウェア側の対策が未熟
  アクセスランプもついていない端末が多い
- サードパーティマーケットが豊富
  審査のゆるいアプリ配布体系

◆技術的な理由

- 撮影音なしで撮影が可能
  秘密裏の撮影に向いている
- パーミッション確認が甘い
  インストール時に一度だけ確認
- 既存アプリのリパッケージングが容易
  人気アプリに悪用ツールを同梱できる

→ 悪用アプリ検出方法の確立が急務

Kappa係数

n: 評価者数
N: 評価対象のdescription数
n_{ij}: i番目の descriptionを j∈{Yes,No}
と判定した評価者数

$$\bar{P} = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{n_{iY}^2 + n_{iN}^2}{n(n-1)}\right) - \frac{1}{n-1}, \quad \bar{P}_e = \left(\frac{1}{N}\sum_{i=1}^{N}\frac{n_{iY}}{n}\right)^2 + \left(\frac{1}{N}\sum_{i=1}^{N}\frac{n_{iN}}{n}\right)^2$$

$$\kappa = \frac{\bar{P} - \bar{P}_e}{1 - \bar{P}_e}$$

上式で定義されるのが，Fleiss のカッパ係数である.

- 複数の評価者による判定の客観的な一致度を測るための尺度
- 偶然による一致を勘案している分，単純な一致率よりも強固
- すべて一致した場合はκ=1，すべて偶然の場合はκ≦0
- 回答者が二者の場合は，cohenのKappa係数が用いられる

カメラに関するAndroid SDKの関数名

Camera.open()
　　リソースを確保する関数

Camera.setPreviewDisplay()
　　画面上にプレビューを表示する関数

onPreviewFrame()
　　プレビューの映像をデータとして取得する関数

C4’選別に用いたヒューリスティック

- 調査の結果，”zxing”という名前を含むクラスが極めて多いことがわかった
- zxingはQRコードの読み取りでよく使われるライブラリ
- 用途不明なカメラの利用をフィルタリングするため，zxingを含むアプリは除外した.
- 外部からインポートした機能は，自前で実装された機能よりも悪用の可能性が低いと断定
- アプリ本体のパッケージ名と，使用しているクラスのパッケージが異なる場合，除外した.

最終的なマルウェアの内訳

・Virus Totalでは,
18/43がマルウェアとして検出された

Adware ・・・11
Generic ・・・2
Trojan ・・・3
SMS ・・・2

最後に残ったもう一つのアプリ

- 説明文によると，仏教に関するアプリ
- カメラのコードが含まれるが，何に使っているのか不明
- 一部の機能は，会員登録が必要なため確認しきれず

SVMによる文書分類の詳細

- 実装はLIBLINEARという，線形SVMに特化したオープンソースソフト（台湾大学）
- L1- 拘束付きの L2- 損失関数を採用
- 文書分類のように，次元数が多く疎である場合には，線形SVMでも十分に精度が高く，非常に高速であるためパラメータの調整がしやすい

- SVMによる分類だけでは見逃すディスクリプションがいくつかあったため，手動で以下の語彙をホワイトリスト化し，カメラ利用ありの判定に用いた.

CAM，拍，拍照，照相，扫码，录像，二维

ディスクリプション分類精度の詳細

- 全体の正答率は97%
- 10回のクロスバリデーションによって評価

誤差	平均値	標準偏差
FPR	0.0208	0.0157
FNR	0.0617	0.0450