




Androidアプリケーションからの通信特徴抽出手法

日本電信電話株式会社
 NTTネットワーク基盤技術研究所
 中野雄介, 上山憲昭, 塩本公平

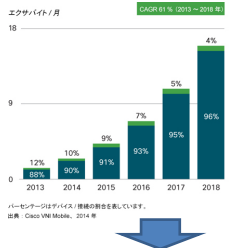
大阪大学
 サイバーメディアセンター
 長谷川剛
 大学院情報科学研究科
 村田正幸

Copyright © 2016 NTT corp. All Rights Reserved.



背景


- スマートフォンの普及による、スマートフォンのトラフィックの増加



スマートトラフィックによるトラフィックがモバイルトラフィック全体に占める割合が今後も増加傾向

モバイルネットワークに対する影響の増大

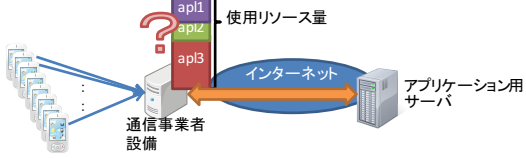
http://www.cisco.com/web/solution/ip/jipgn/literature/white_paper_c11-520862.html Copyright © 2016 NTT corp. All Rights Reserved. 2



課題


- 通信事業者以外の事業者やユーザによるアプリケーション作成
 - ネットワークへの負荷を意識しないアプリケーション

▶ 特定アプリケーションによるリソースの専有



▶ リソースを専有しているアプリケーションの特定が困難

Copyright © 2016 NTT corp. All Rights Reserved. 3



目的

Androidアプリケーションの通信特徴抽出

目的達成でできること


アプリケーション固有の通信の特徴を用いて、ネットワーク内で収集された、複数アプリケーションの packets が混在するキャプチャデータからアプリケーションごとの packets に分類

アプリケーションごとのネットワークにおける振る舞いを抽出

アプリケーションごとのインストール数を特定

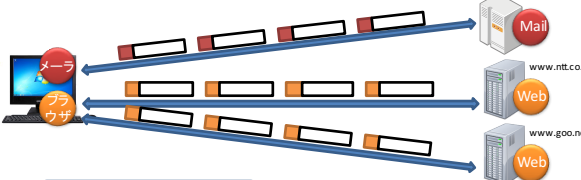
リソースを専有しているアプリケーションを特定

Copyright © 2016 NTT corp. All Rights Reserved. 4



既存技術: 5-tupleによる分類


- TCPヘッダ, IPヘッダの情報でアプリケーションごとの packets に分類



スマートフォンアプリケーションへの適用困難

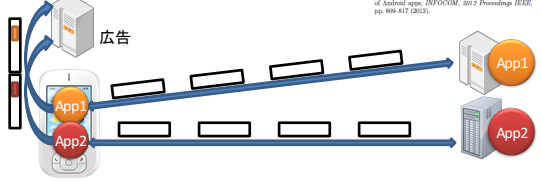
複数種類のアプリケーションが共通のサーバと通信する場合、アプリケーション毎の分類が困難
 (例) TwitterのAPIが実装されたアプリケーションは複数存在

Copyright © 2016 NTT corp. All Rights Reserved. 5



既存技術: NetworkProfiler

- Androidアプリケーションを解析することで通信のFingerprintを抽出
 - Fingerprint: 広告取得時のHTTPに付加するアプリケーションごとのIDなど
 - ネットワークに接続された端末における特定アプリケーションの起動を特定



パケットの分類には適用困難

アプリケーションの広告取得以外の packets の分類は対象外

Copyright © 2016 NTT corp. All Rights Reserved. 6

提案手法

- 各アプリケーションのソースコードを解析し、
通信の順序と各通信に含まれる文字列(通信の内容)を通信の特徴として抽出する

抽出される特徴の例

```

    graph TD
      UA[User-Agent: abc] --> GET[GET  
Accept-Encoding: gzip,compress  
Accept-Language: ja,en]
      GET --> POST[POST  
Accept-Language: ja]
  
```

共通のサーバを利用していても異なる特徴
特定(広告取得などの通信ではなく、全通信を分類できる特徴)

Copyright © 2015 NTT corp. All Rights Reserved. 7

0. ソースコードの抽出

- Sootを利用してAPKファイルからソースコード抽出
- 提案手法により、ソースコードから通信内容とその順序を抽出

APK file (APL) → Soot ※ → ソースコード (APL)

通信内容と順序 (User-Agent: abc, GET, POST)

※ Soot http://sable.github.io/soot/

Copyright © 2015 NTT corp. All Rights Reserved. 8

1. メソッドの前後関係抽出

前後関係があると判断されるソースコードの箇所

- メソッドの呼出
メソッドaからメソッドbを呼出し
- Intentの使用
画面Aから画面Bへ遷移
- イベントリスナによる実行
setOnXXXListener
→OnXXXListener.onXXX
- ActivityやAsyncTaskクラスなどのライフサイクル

ソースコード (APL) → メソッドの前後関係

前	後
Class X Method x	Class Y Method y
Class A Method a	Class B Method b
Class Y Method y	Class Y Method yy
Class B Method b	Class B Method bb
Class B Method b	Class X Method x

Copyright © 2015 NTT corp. All Rights Reserved. 9

2. 通信関連メソッドの呼出順序抽出

- 一番最初のメソッドを発見(「後」の行に書かれていないメソッド)
- 通信関連メソッドを探索
- 発見された順序で、通信関連メソッドの呼び出し順序を抽出

① メソッドの前後関係

通信関連メソッド一覧
Class Y Method y
Class B Method b
Class B Method bb

② 一番最初のメソッド
③ 通信関連メソッドの探索

分岐
最後だったら分岐箇所に戻って別のルートをたどる

凡例
★ 一番最初のメソッド
★ 通信関連メソッド
→ 通信関連メソッドの探索

Copyright © 2015 NTT corp. All Rights Reserved. 10

3. 通信内容の順序抽出

- 対象のメソッドから、メソッドの前後関係を逆にソースコードを探索し、対象メソッドの引数を見出し
- 発見された引数を通信内容として抽出

通信関連メソッドの呼出順序

メソッドの前後関係

ソースコード

```

    Class B {
        Method bb() {
            // ①
        }
    }
    Class A {
        Method a() {
            // ②
        }
    }
    User-Agent: abc;
  
```

Copyright © 2015 NTT corp. All Rights Reserved. 11

評価手法

- 評価対象アプリケーション: Spika
- オープンソースのインスタントメッセージアプリケーション

Spikaのみのパケット ← 一致確認 Spikaの通信内容の順序 (提案手法で抽出)

Nexus5 (Spika) → 無線LAN → キャプチャ用PC (Wireshark) → インターネット

Network Log (log) → アプリケーションごとのパケットのヘッダ情報 → Spikaのみのパケット ← 一致確認

Wireshark (pcap) → 混在したパケットキャプチャデータ → Spikaの通信内容の順序 (提案手法 ※) → APL SpikaのAPKファイル

Copyright © 2015 NTT corp. All Rights Reserved. 12

抽出された通信内容の順序

Copyright © 2015 NTT corp. All Rights Reserved. 13

一致確認の結果

- キャプチャデータと、抽出された通信内容の順序との一致を確認

操作しない状態でのキャプチャデータ

メッセージ送信時のキャプチャデータ

Copyright © 2015 NTT corp. All Rights Reserved. 14

まとめと今後の課題

- 複数のアプリケーションのバケットが混在するキャプチャデータから、アプリケーションごとのバケットに分類するための、アプリケーションごとの通信の特徴を抽出する手法を提案
- 通信の特徴として

通信の順序

と

各通信に含まれる文字列
(通信の内容)

 を抽出
- 今後
 - Spika以外の多様なアプリケーションで、提案手法の有効性を確認
 - 複数アプリケーションが特定のサーバと通信していても、異なる特徴が抽出されること
 - 抽出された特徴を用いて、混在するキャプチャデータからアプリケーションごとのバケットに分類する手法の検討

Copyright © 2015 NTT corp. All Rights Reserved. 15

既存技術 : Network Log

- スマートフォン内でアプリケーションごとに分類してバケットをキャプチャ

ネットワーク内でのキャプチャ結果の分類に適用困難

- ネットワークリソースを専有しているアプリケーション発見のためには、ネットワーク内でキャプチャされた複数端末からのバケットを分類する必要がある。

Copyright © 2015 NTT corp. All Rights Reserved. 16

既存技術 : AndroidLeaks

Copyright © 2015 NTT corp. All Rights Reserved. 17

既存技術 : TaintDroid

Copyright © 2015 NTT corp. All Rights Reserved. 18