NetworkProfiler: Towards Automatic Fingerprinting of Android Apps

Shuaifu Dai[1,3], Alok Tongaonkar[2], Xiaoyin Wang[3], Antonio Nucci[2], Dawn Song[3]

Presenter: Mario Baldi[2]

[1] Peking University, China
[2]Narus Inc, USA
[3] University of California, Berkeley, USA

Motivation

- Mobile Device *vs* PC
  - 488 million *vs* 415 million (2011)
- Mobile traffic is up to 5000% over the past three years
- Identifying applications critical for
  - Network Management
  - Security
  - Market Analysis

Challenges In Mobile App Identification

- Explosive growth rate of apps
  - 700,000 apps in Google Play (Oct. 2012)

- Bring Your Own Device (BYOD)
  - Network admins have no control over apps on personal device in enterprises

- Network operators need to be aware of all apps being used in their network

State of the Art Techniques

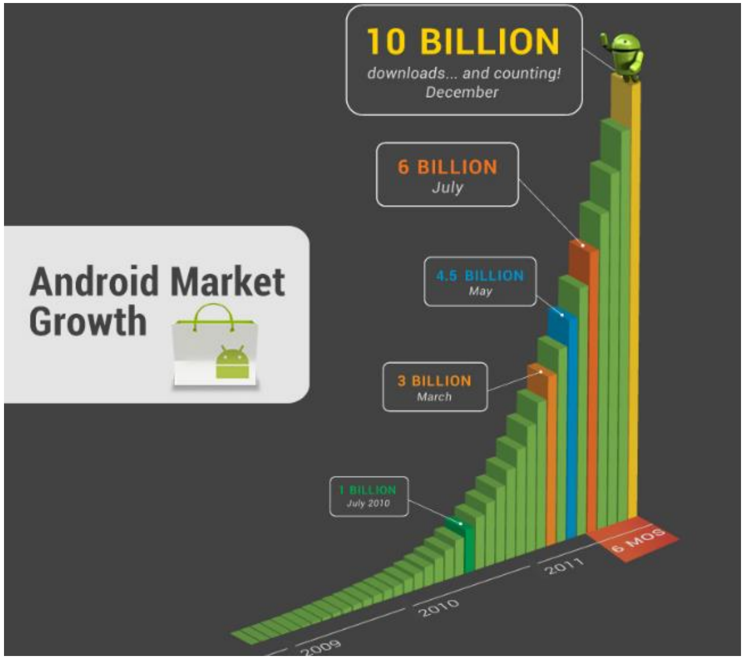
- User-Agent:
  - [Xu, Q. et al.] Identifying diverse usage behaviors of smartphone apps. IMC,2011.
  - Not strictly enforced on any mobile platform,
    - Android apps use generic strings in this field
- Host:
  - [Falaki, H. et al.] A first look at traffic on smartphones. IMC, 2010
  - May not be unique
    - Same host may serve multiple apps
- Manually running apps
  - [Wei,X., etc.] Profiledroid: Multi-layer profiling of android applications. MobiCom, 2012
  - Requires tremendous human labor

Key Idea: Network Profile of Apps

- Network profile of apps analogous to DNA profiles of people
  - Use unique characteristics of the network behavior of the app to identify the app
  - Each unique network behavior is called as "network fingerprint"
- Network fingerprint consists of
  - Host that the app connects to
  - A state machine representing the patterns over the strings that occur in the HTTP header of the requests made by the app to those servers

Objective

- Build network profiles of Android apps automatically
- Why Android?
  - More difficult to identify apps on Android platform
  - Growth rate of apps on Android is exponential



Design Considerations

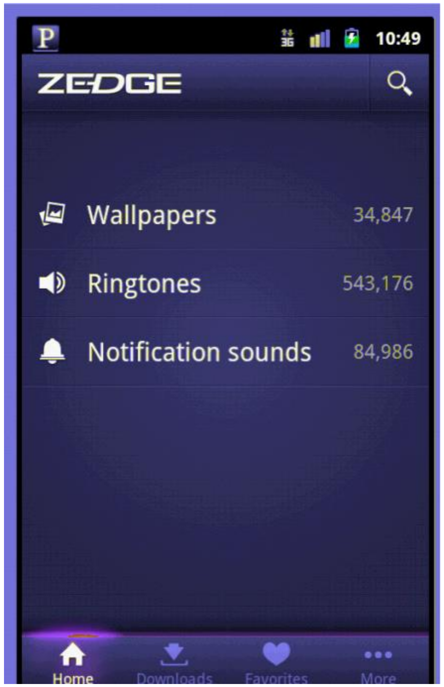
- Observation regarding 90K apps:
  - Most of the app use HTTP/HTTPS
  - Only 30% use HTTPS
  - HTTPS mainly used only for authentication
- HTTP app flow classification*
  - 1. Origin: e.g. app provider
  - 2. CDN+Cloud: e.g. Amazon AWS
  - 3. Third party: e.g. ads & analytics

* Wei, X. et al. Profiledroid: Multi-layer profiling of android applications. MobiCom'2012

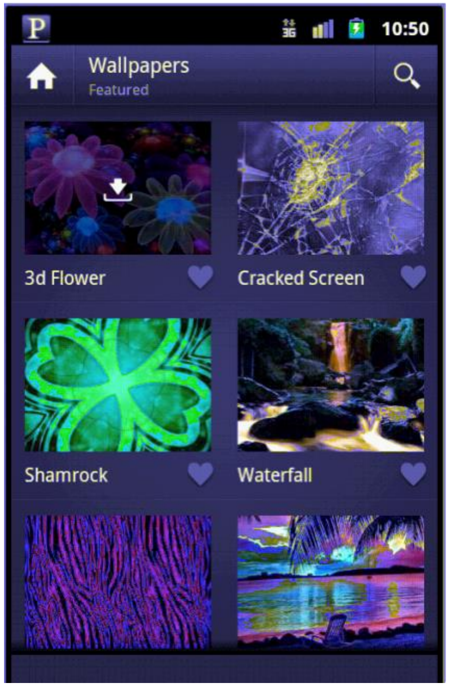
Our Solution: NetworkProfiler

- A system for automatic generation of Network Profiles for Android apps
  - Run Android app in automated fashion in emulator
  - Collect network traces for the app
  - Extract fingerprints from the traces
- Challenge
  - Thorough exploration of an app’s network behaviors
  - Extraction of meaningful fingerprints

Case Study: Zedge



Wallpapers	34,847
Ringtones	543,176
Notification sounds	84,986



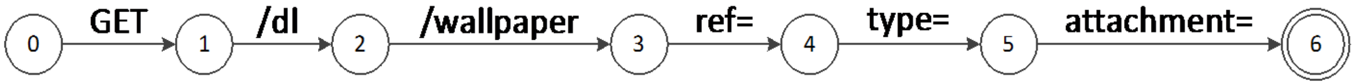
GET /dl/wallpaper/
9370c626058a0e01a0a45d1aff0b730c/
mountains.jpg?
ref=android&type=mc&attachment=1
HTTP/1.1
Host: fsa.zedge.net

GET /dl/wallpaper/
3dead9d0f52b1858bb028a974e2cd13f/
angry_birds.jpg?
ref=android&type=mc&attachment=1
HTTP/1.1Host: fsb.zedge.net

GET /dl/wallpaper/
b26473e40eb9bfd3c45c0aa44c33438a/
multi_zebra.jpg?
ref=android&type=mc&attachment=1
HTTP/1.1
Host: fsb.zedge.net

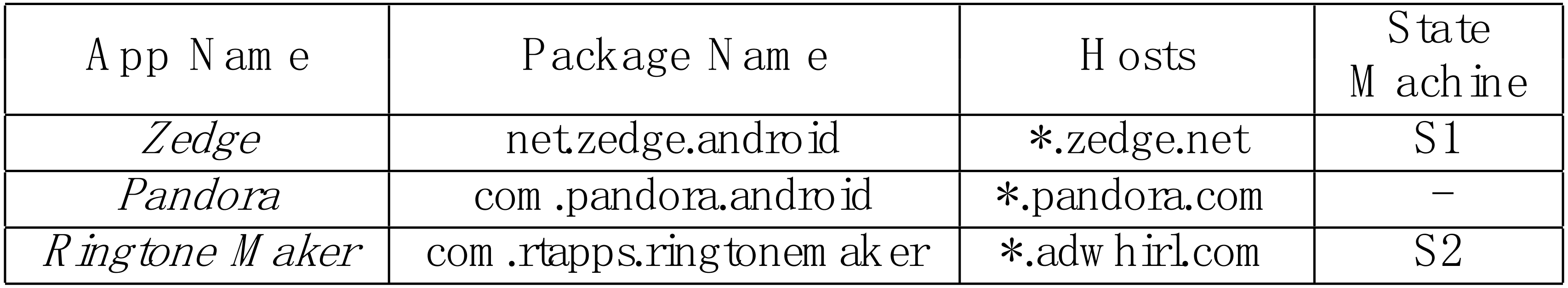
Host: (fsa|fsb).zedge.net

State machine:



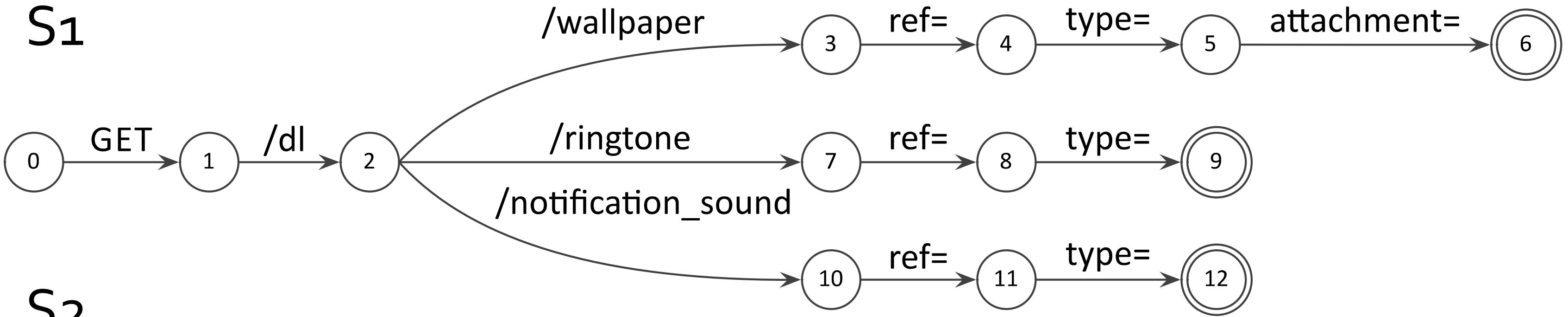
Examples of Network Profiles

Network Profile = host + state machine

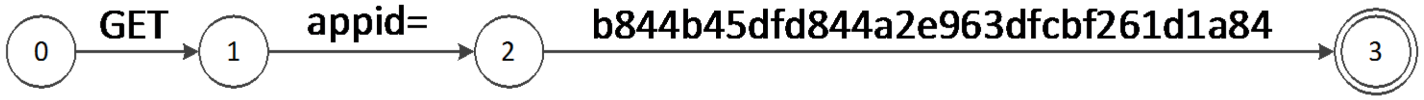


App Name	Package Name	Hosts	State Machine
Zedge	net.zedge.android	*.zedge.net	S1
Pandora	com.pandora.android	*.pandora.com	–
Ringtone Maker	com.rtapps.ringtonemaker	*.adwhirl.com	S2

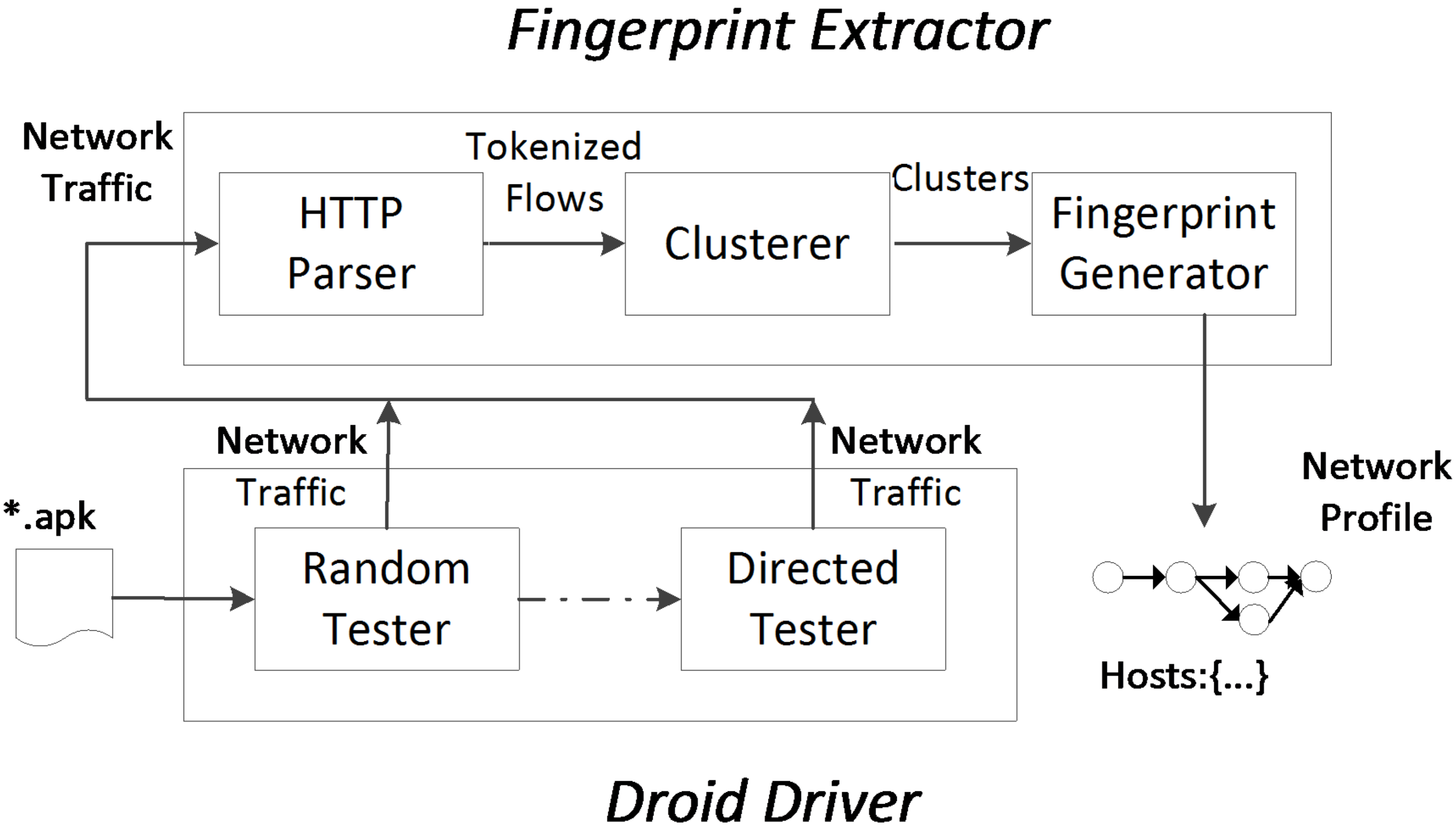
S1



S2



Network Profiler Overview



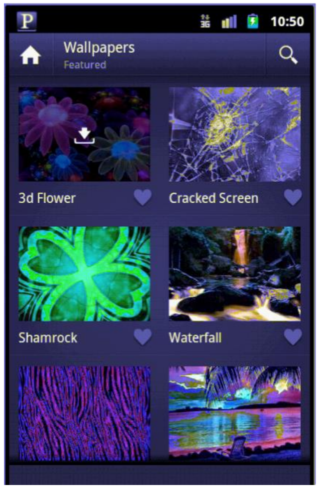
Droid Driver

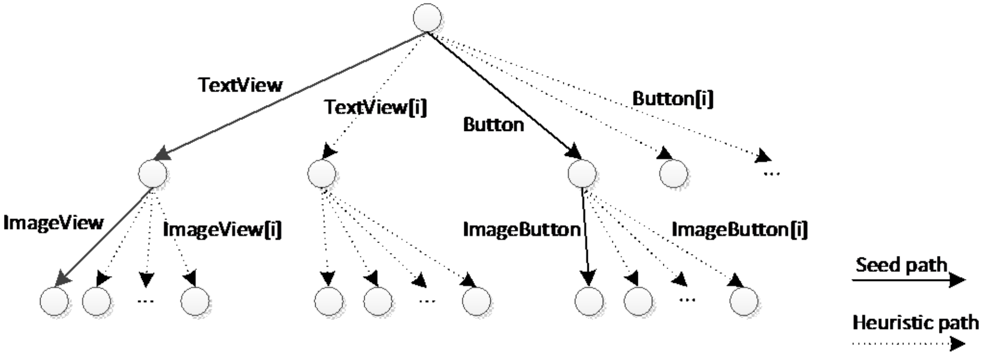
Goal: execute Android apps and collect the network traces

- Random Tester
  - efficient
  - collect traces that connect to ads or origin server
- Directed Tester
  - diverse
  - collect traces that need human interaction

Directed Tester

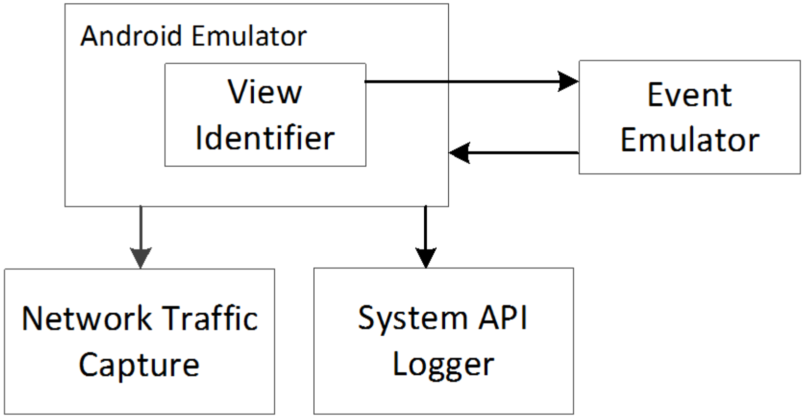
1. Path Recorder
   - Record human interaction as the seed path
2. Heuristic Path Generator
   - Generate heuristic path





Directed Tester(cont'd)

3. Path Replayer



- We use the information in system API logger to get the *pid* and its corresponding *ip* address to remove the noise in the traces

Fingerprint Extractor

Goal: extract meaningful fingerprints

Step 1: Tokenize

$$\overbrace{\text{GET}}^{m}\ \overbrace{\underbrace{\text{/dl/}}_{pc}\text{wallpaper/71481.../}\underbrace{\text{london\_streets.jpg}}_{fn}}^{p}\overbrace{\underbrace{\text{?ref}}_{k}\text{=}\underbrace{\text{android}}_{v}\text{\&type=mc\&attachment=1}}^{q}\ \text{HTTP/1.1}$$

m: method, p: page, q: query, pc: page-components, fn: file name, k: key, v: value

Step 2: Cluster

- distance $d_h(i,j)=(d_p(i,j) + d_q(i,j))/2$
- $d_{(p|q)}(i,j) = 1 - \textit{similarity}$
- *Similarity* measured by *Jaccard index*

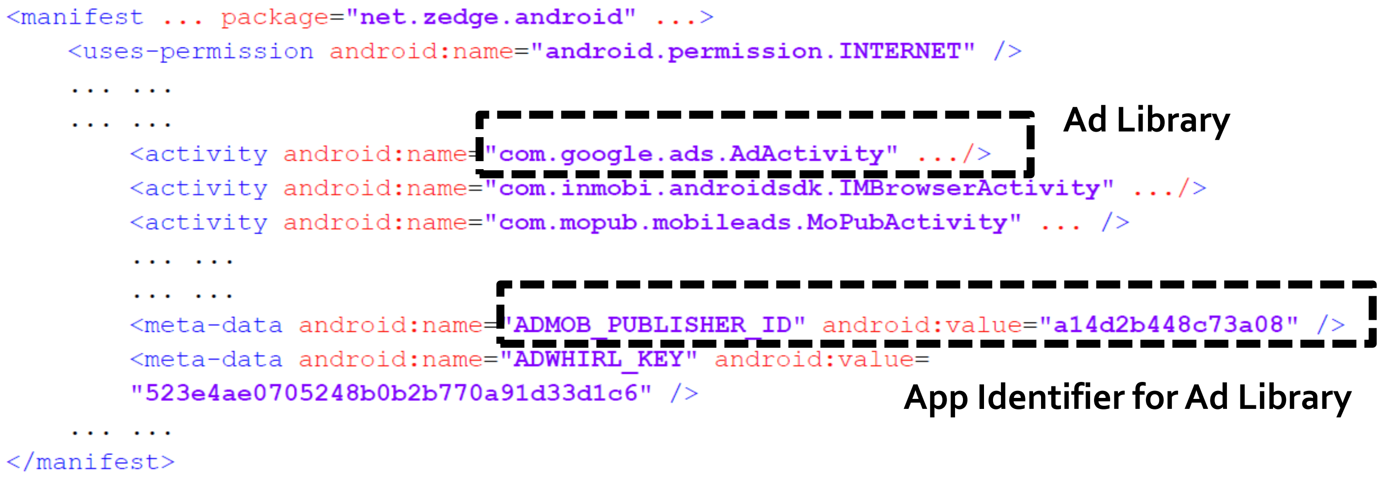
Evaluation

Goal: identify apps in traces for a cellular provider

- Ads traffic
    - 90K free apps
        - 70K(87%) ask for internet permission
            - 32k(46%) have ads library
- Non-ad traffic
    - 6 popular apps
        - Youtube, Flixster, ESPN Score Center, CNET news, Pandora, Zedge

Ad Information in Android Apps

- In manifest file for apps

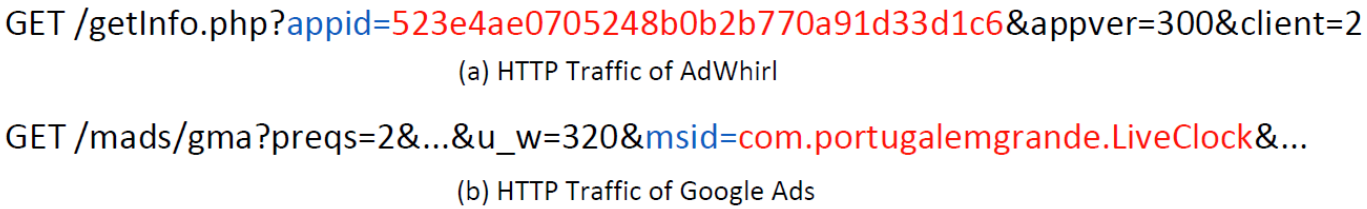


```
<manifest ... package="net.zedge.android" ...>
    <uses-permission android:name="android.permission.INTERNET" />
    ... ...
    ... ...
        <activity android:name="com.google.ads.AdActivity" .../>
        <activity android:name="com.inmobi.androidsdk.IMBrowserActivity" .../>
        <activity android:name="com.mopub.mobileads.MoPubActivity" ... />
        ... ...
        ... ...
        <meta-data android:name="ADMOB_PUBLISHER_ID" android:value="a14d2b448c73a08" />
        <meta-data android:name="ADWHIRL_KEY" android:value=
        "523e4ae0705248b0b2b770a91d33d1c6" />
    ... ...
</manifest>
```

Ad Library

App Identifier for Ad Library

- In traffic



GET /getInfo.php?appid=523e4ae0705248b0b2b770a91d33d1c6&appver=300&client=2

(a) HTTP Traffic of AdWhirl

GET /mads/gma?preqs=2&...&u_w=320&msid=com.portugalemgrande.LiveClock&...

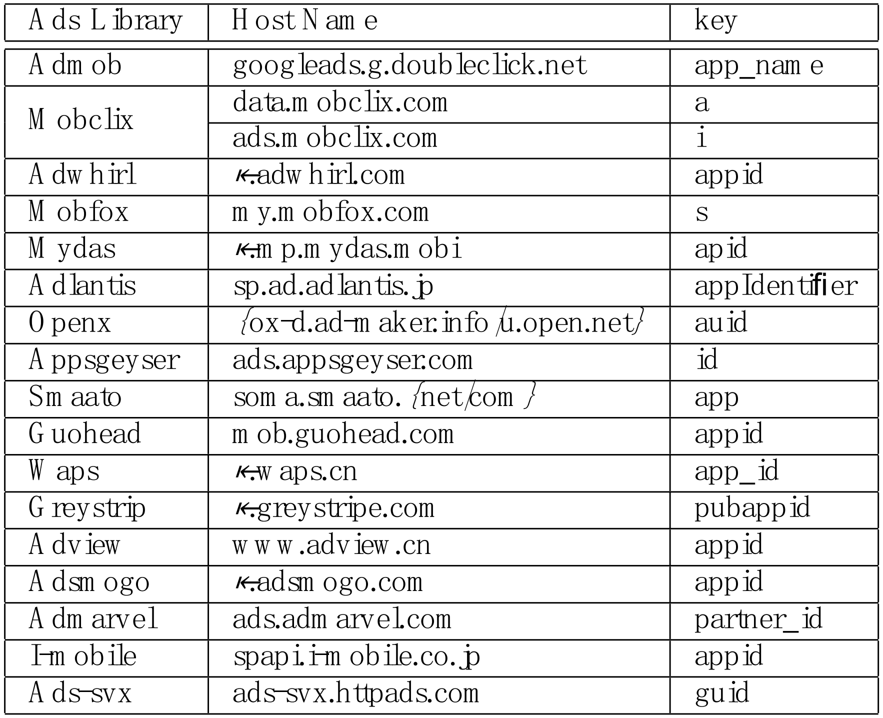
(b) HTTP Traffic of Google Ads

Ads Identifiers

- Explicit ID for Ad Libraries

Ads Library	Key in App Manifest
Admob	ADMOB_PUBLISHER_ID
Mobclix	com.mobclix.APPLICATION_ID
Adwhirl	ADWHIRL_KEY
Waps	WAPS_ID
Wooboo	Wooboo_PID
Domob	DOMOB_PID
Admarvel	ADMARVEL_PARTNER_ID
Admogo	ADMOGO_KEY
Madvertise	madvertise_site_token
Adwo	Adwo_PID
Nexage	NEXAGE_DCN
Flurry	flurry_key
Tapjoy	tapjoy_key
Aduru	ADURU_DEVELOPER_ID

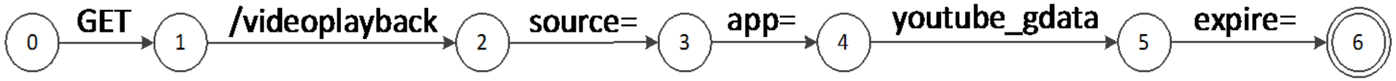
- Keys for Different Ads Libraries



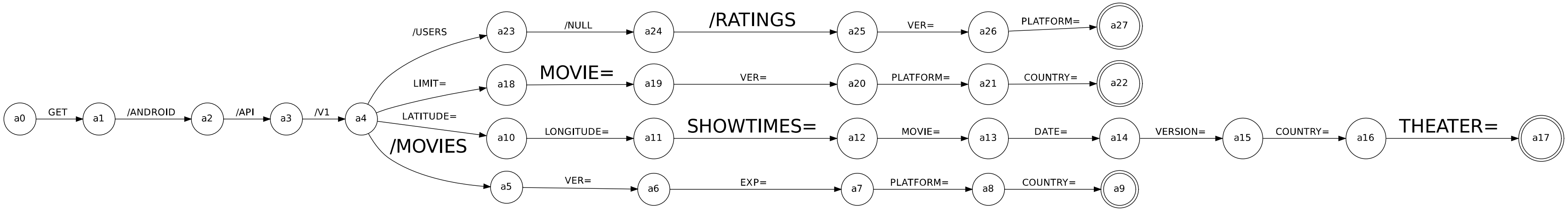
Ads Library	Host Name	key
Admob	googleads.g.doubleclick.net	app_name
Mobclix	data.mobclix.com	a
	ads.mobclix.com	i
Adwhirl	↚.adwhirl.com	appid
Mobfox	my.mobfox.com	s
Mydas	↚.mp.mydas.mobi	apid
Adlantis	sp.ad.adlantis.jp	appIdentifier
Openx	{ox-d.ad-maker.info/u.open.net}	auid
Appsgeyser	ads.appsgeyser.com	id
Smaato	soma.smaato.{net/com}	app
Guohead	mob.guohead.com	appid
Waps	↚.waps.cn	app_id
Greystrip	↚.greystripe.com	pubappid
Adview	www.adview.cn	appid
Adsmogo	↚.adsmogo.com	appid
Admarvel	ads.admarvel.com	partner_id
I-mobile	spapi.i-mobile.co.jp	appid
Ads-svx	ads-svx.httpads.com	guid

Non-Ad traffic

- Fingerprints
  - Youtube



  - Flixster



- The fingerprints never match traffic from any other app
- We succeeded in identifying all 6 apps

Future Work

- Explore automated test methods as well crowd-sourcing approaches for obtaining seed path
- Combine static analysis with the dynamic analysis to improve our coverage of execution paths
- Create large database of network profiles

Conclusion

- We proposed a novel system called Network-Profiler for the automated generation of network profiles for Android apps

- This technique provides a new perspective and our evaluation shows that we can identify the apps with high precision

Thanks!

Q&A?