Panoptispy:
Characterizing Audio and Video Exfiltration from Android Applications

Elleen Pan, Jingjing Ren, Martina Lindorfer*, Christo Wilson, and David Choffnes

Northeastern University, *UC Santa Barbara

Motivation

📷 🎙 + internet connectivity … 👀

Examples

ultrasonic beacons for cross-device linking

patents for recognizing user emotion

listening for unlicensed broadcasting

photos taken surreptitiously by shrinking preview to 1x1 pixel

Media surveillance, so far, has been anecdotal

Goals 🥅

- Identify & measure media (audio, images, video) exfiltration **at scale**
  - Large number of apps & broad coverage of app stores
- Focus on exfiltration over network
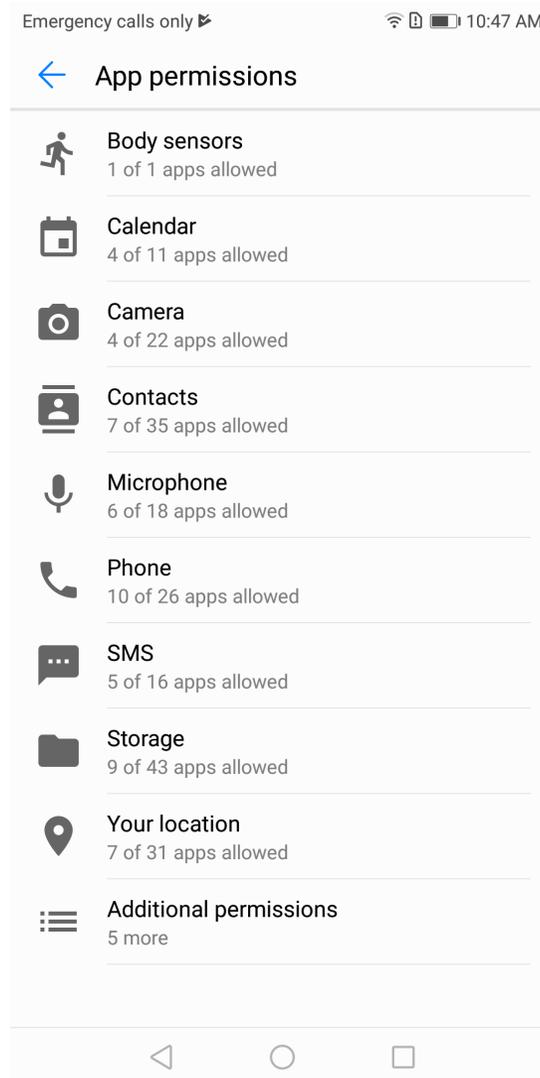- Is the exfiltration a **leak** (undisclosed/unexpected)?

- How do apps use sensors?
  - Permissions requested
  - APIs called
  - First or third-parties

Outline

- Motivation

- Threat Model

- Methodology

- Aggregate Results

- Case Studies

  - Photography apps

  - Screen recording

- Discussion

- Conclusion

Android Access Control

- Certain APIs require permissions in order for code to execute
- Protects sensors from being accessed by apps that don't need it
- Requested at install time for API level 22-, runtime for API level 23+

Android Permission Model
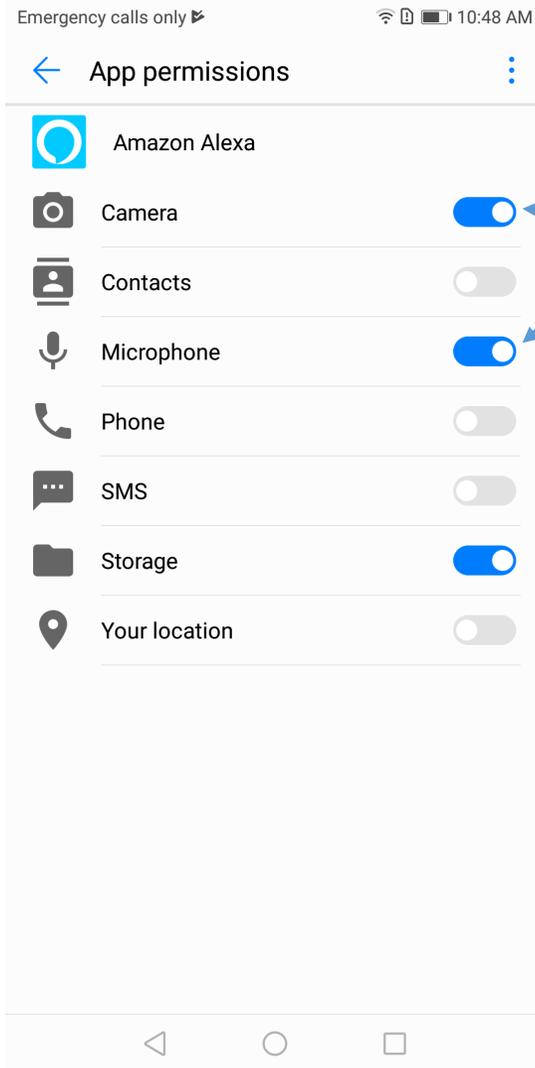
- Camera & mic hardware access

Why aren't permissions enough?

- Incomplete
  - No permissions required for capturing app screen

- Coarse-grained
  - Permissions granted at app level
    - Third-party libraries also get access
  - Users don't know when apps are using hardware

- Lack of visibility and control (may contain PII!)
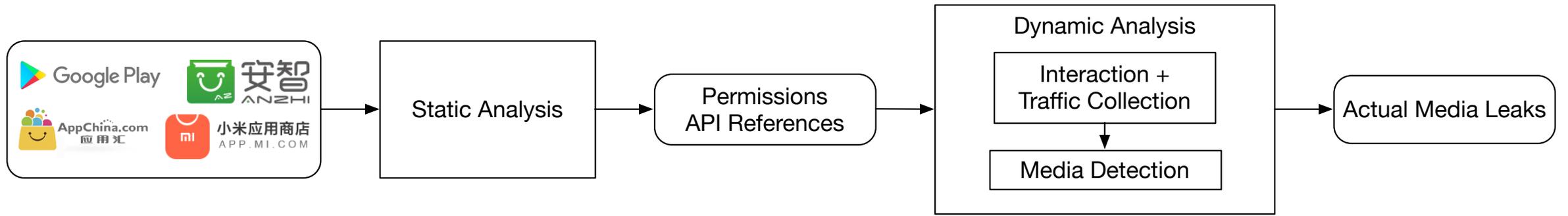 - as media is exfiltrated over the network
  - Background access
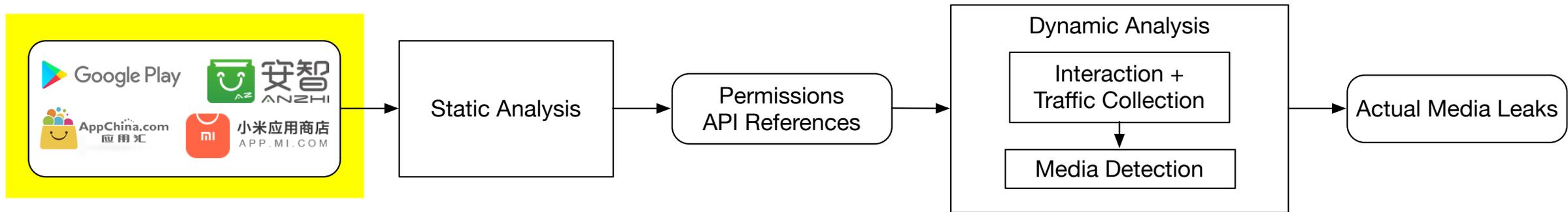
Definition of media leak

Suspicious or *unexpected*

🤔

1. Does it further the primary purpose of the app?
2. Is it disclosed to the user?
   - Privacy policies
3. Is it employed by similar apps?
4. Is it encrypted over the internet?

No? It's a **leak**

Static Analysis → Permissions / API References → Dynamic Analysis (Interaction + Traffic Collection → Media Detection) → Actual Media Leaks

App Selection

Popular + new from **Google Play**		
Popular + random from **AppChina**, **Mi.com**, **Anzhi**	→	Camera or audio permission →

| Store	# of apps
Google Play	15,627
AppChina	510
Mi.com	528
Anzhi	285
Total	17,260
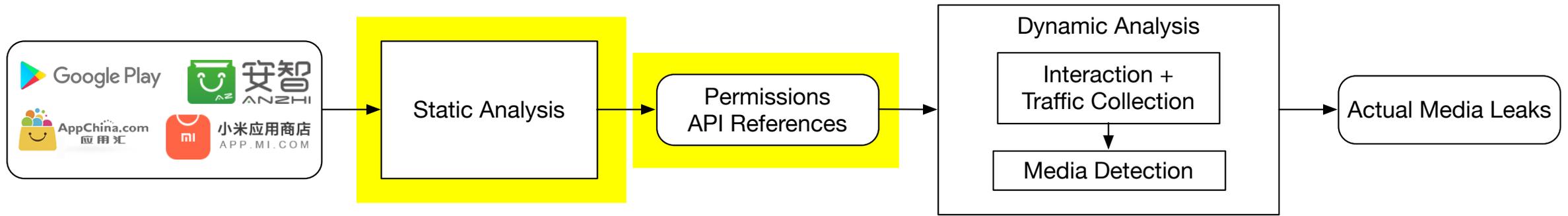
Static Analysis

- Permission analysis (camera, record audio)

- Media API references (camera, record audio, video, screen capturing)

- Media API references found in third-party libraries
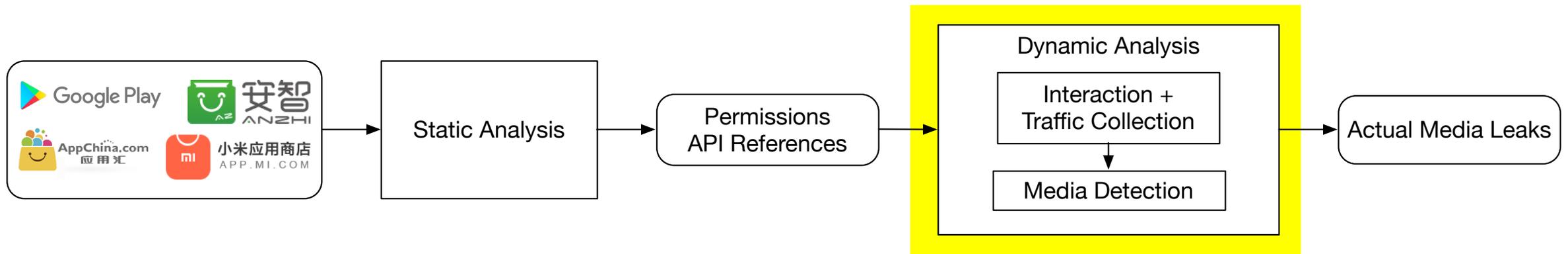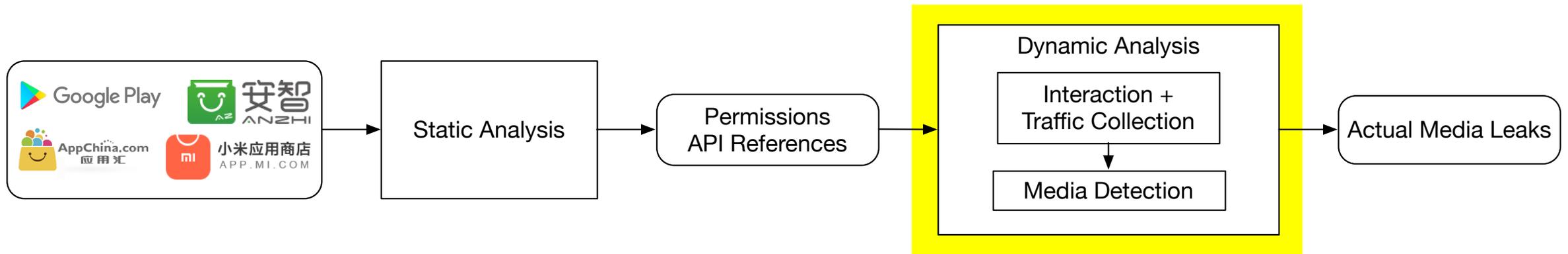
Dynamic Analysis

- Why is dynamic analysis necessary?
  - Detect whether media permissions are actually used
  - Media APIs may be in dead code paths
  - Detect dynamically loaded / obfuscated code
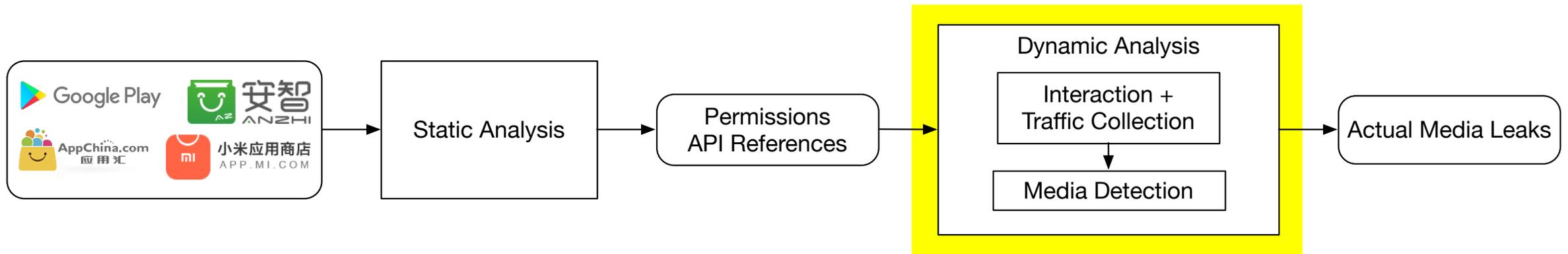
Dynamic Analysis

- Test environment

- Automated interaction
  - Monkey for 5,000 events

- Recording network traffic
  - Mitmproxy to intercept traffic

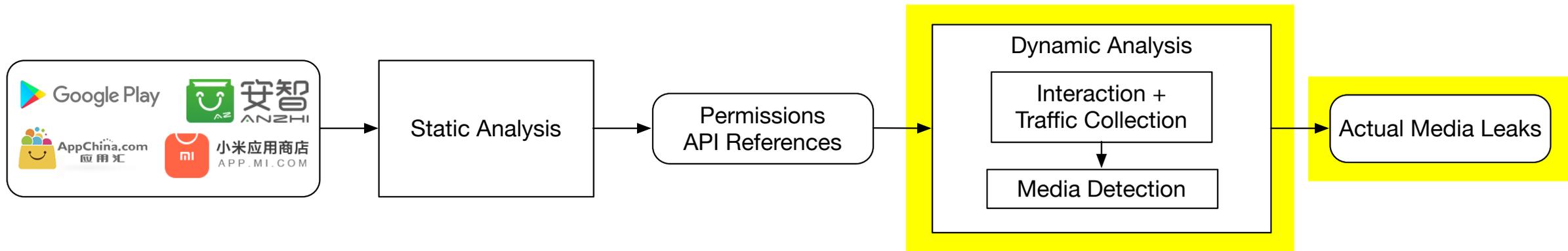# Detection of Media in Network Traffic

- Extraction
  - Mediaextract detection with **file magic numbers**
    - E.g. JPEG files: FF D8 FF …
    - False positives require manual checking

Category	Supported	Unsupported
Audio	3gp, aac, id3v2, m4a, ogg, wav	raw
Image	bmp, gif, jpg, png, webp	
Video	3gp, mp4, webm	

Detection of Media in Network Traffic

- Validation
  - Test app
  - Manual tests with known apps – imgur GIPHY SOUNDCLOUD
  - Verification of detected media by manually interacting with apps

Static: Permission vs. API
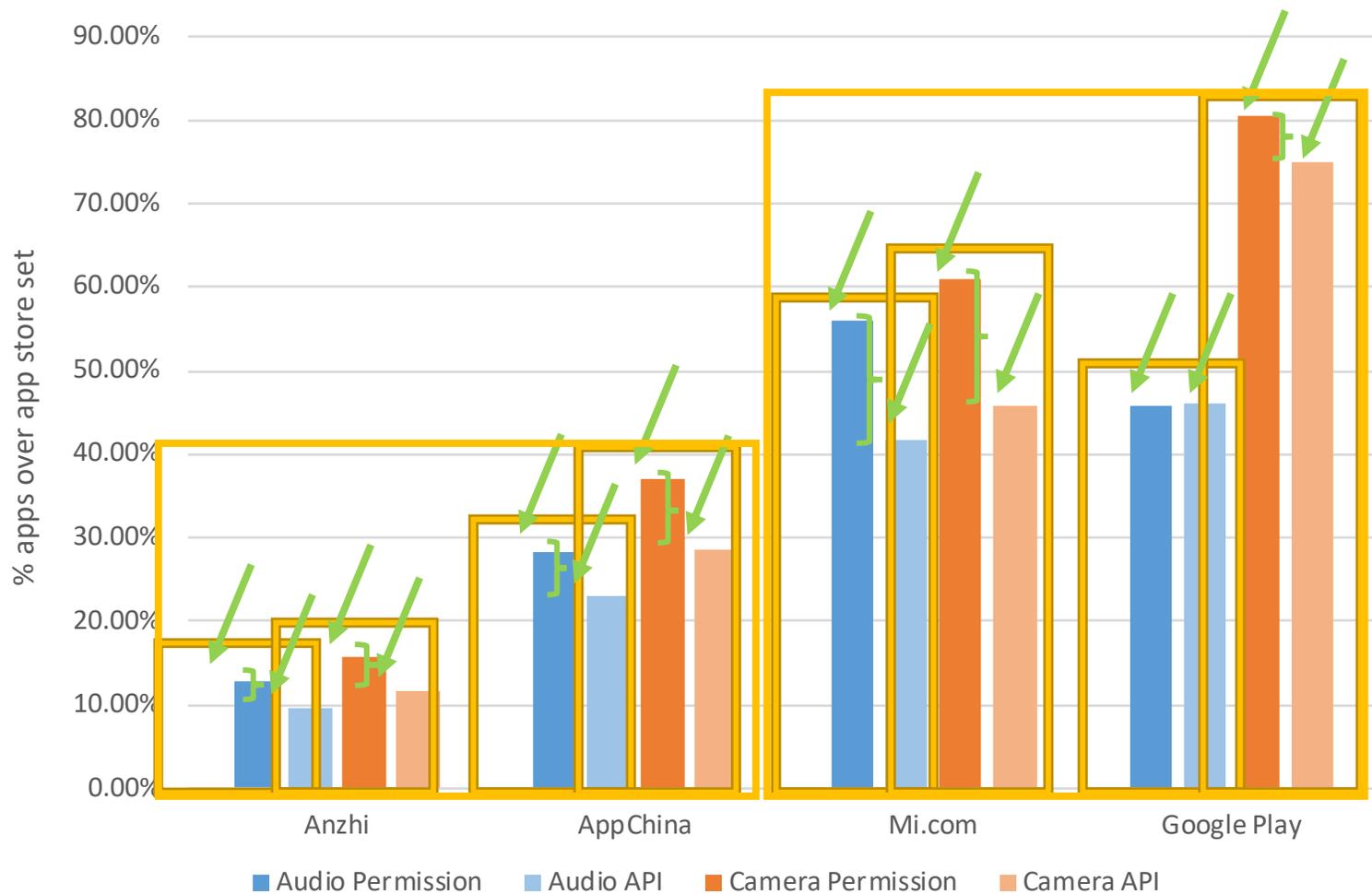
- Large fractions of audio (43.8%) and camera (75.6%) permission declarations

- Permissions > API calls

- Mi, Google > Anzhi, AppChina

- One exception: API > permission (audio in Play)

Legend: Audio Permission, Audio API, Camera Permission, Camera API

Dynamic: Media in Network Traffic

😱

- 21 cases of detected media – 12 considered **leaks**
  - Unexpected or unencrypted
- 9 shared with third parties

Case Study: Photography Apps

- Server-side photo editing
  - Photos are sent to servers
  - Users not notified

- App has no other functionality requiring internet connection

- Privacy policy vaguely disclosed (5 apps) or didn't mention (1 app)

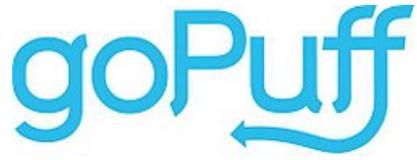# Case Study: Screen Recording

goPuff

- Screen recording of user interaction, where PII was exposed
  - Leaked to an Appsee domain

appsee

- Screen recording as a feature
- Developers are responsible for hiding sensitive screens
- Few apps use the API method to do so – 5/33 apps
  - Server-side way exists, unknown how many apps use it

Responsible Disclosure

- Pulled Appsee from Android & iOS builds
- Updated privacy policy

- Reviewed GoPuff & Appsee
  - "Google constantly monitors apps and analytics providers to ensure they are policy-compliant. When notified of our findings, they reviewed GoPuff and Appsee and took the appropriate actions."

 ¯_(ツ)_/¯

Limitations

- Translated media formats (audio being transcribed, etc.)
- Controlled experiments do not replicate environmental conditions
- Intentional obfuscation of traffic

These Academics Spent the Last Year Testing Whether Your Phone Is Secretly Listening to You

Kashmir Hill
7/03/18 1:00pm • Filed to: IT IS PARANOIA ✓

263.4K   144   8

Your phone isn't listening to you, researchers say, but it may be watching e

There's a new conspiracy theory in town

By Makena Kelly | Jul 3, 2018, 3:36pm EDT

No, your smartphone is not lis

But it may be watching you

By Cal Jeffrey on July 3, 2018, 7:17 PM | 25 comments

Smartphone apps don't listen to your conversations, but they do something equally creepy

The researchers found that while smartphone applications did not send audio clippings to third-party domains, they did send screenshots or screen recordings to them.

🐦 BusinessToday.In   New Delhi   Last Updated: July 4, 2018 | 22:14 IST

Uh-oh. Boffins say most Android apps can slurp your screen – and you wouldn't even know it

Fancy that

59 💬   SHARE ▼

may be spying on you
pect

Elizabeth Weise, USA TODAY   Published 12:04 p.m. ET July 5, 2018 | Updated 4:21 p.m. ET July 6, 2018

Your phone is probably spying on you

By Andy Meek, BGR

July 5, 2018 | 10:25am | Updated

Yes, your phone is spying on you...but not how you think it is

Yahoo Finance Video • July 5, 2018

Panoptispy   23

Recommendations

- Access to the screen should be protected by OS
  - Or, users should at least be notified & able to opt out
- Main app & third-party permissions should be separated

Conclusion

- Most apps have over-provisioned permissions
    - Susceptible for abuse from third parties
- 12 cases of unexpected or unencrypted media
    - 9 cases of third party sharing
- Screen recording video sent to a third party library
    - Sensitive input fields
    - No permissions or notification to the user

https://recon.meddle.mobi/panoptispy/