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

SilverPush 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 

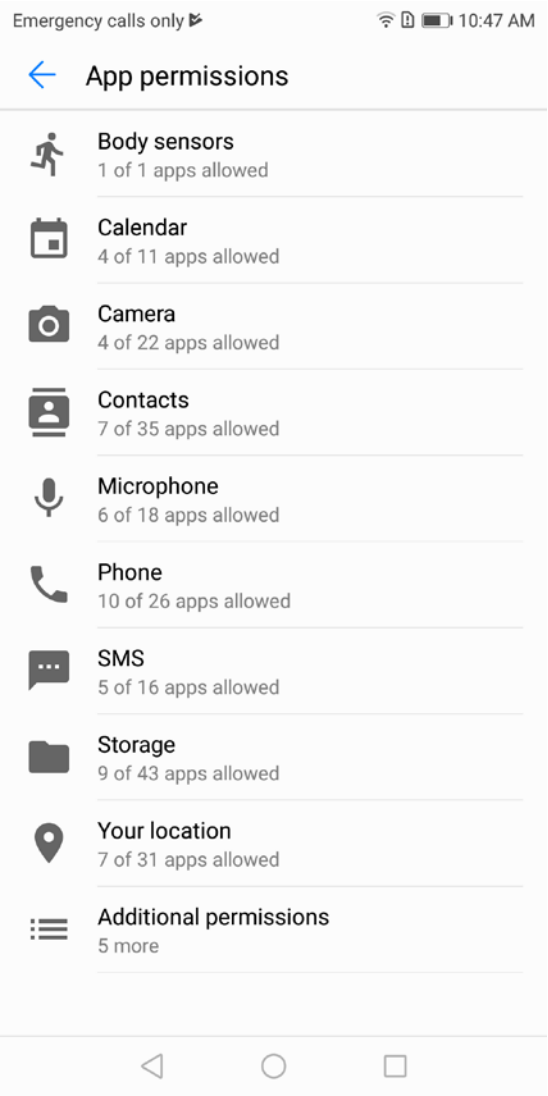
- Identify & measure media (audio, images, video) exfiltration **at scale**
  - Large number of apps & broad coverage of app stores
- Focus on exfiltration over network
- Is the exfiltration a **leak** (undisclosed/unexpected)?

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

Outline

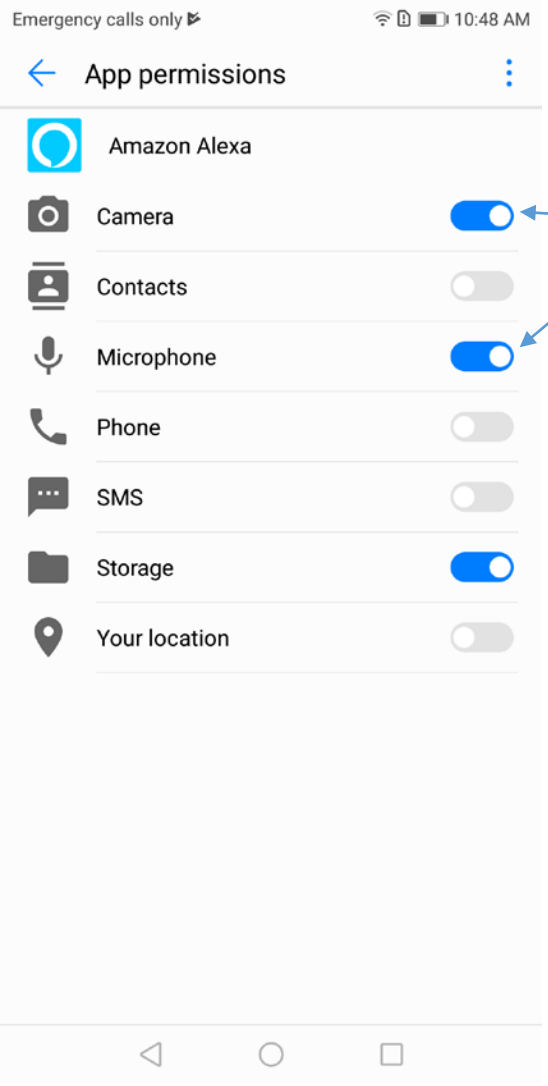
- Motivation
- Threat Model
- Methodology
- Aggregate Results
- Case Studies
  - Photography apps
  - Screen recording
- Discussion
- Conclusion

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

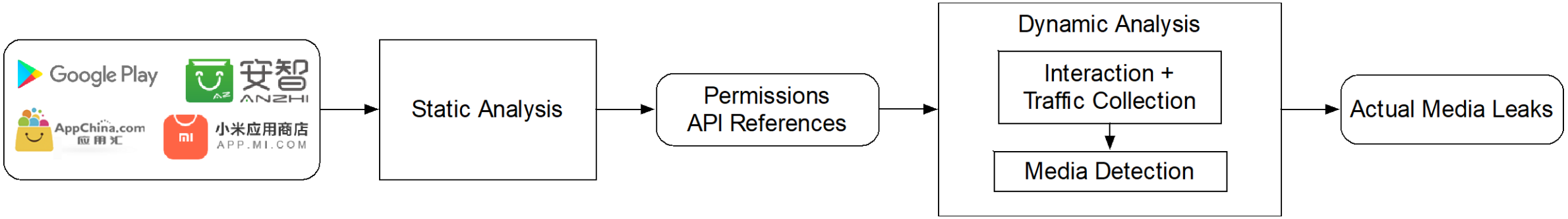
- Lack of visibility and control (may contain PII!)
  - as media is exfiltrated over the network
  - Background access

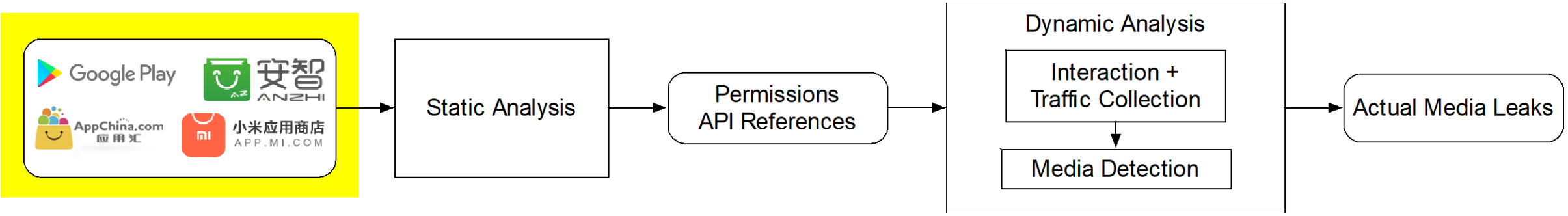
Definition of media leak

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**


Google Play
安智
ANZHI
AppChina.com
应用汇
小米应用商店
APP.MI.COM
Static Analysis
Permissions
API References
Dynamic Analysis
Interaction +
Traffic Collection
Media Detection
Actual Media Leaks



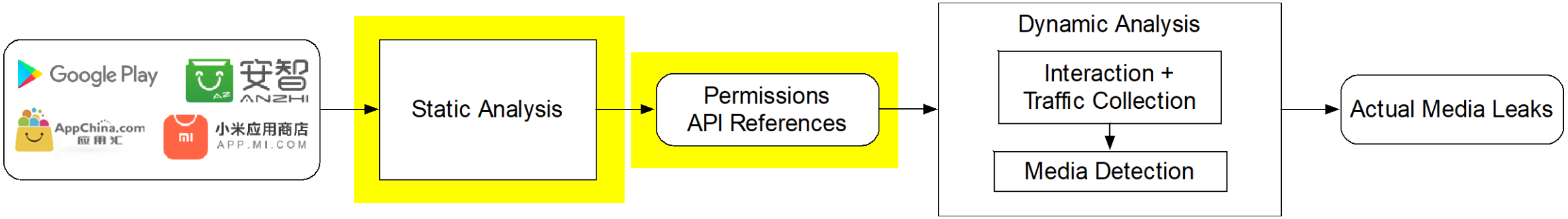
App Selection

Popular + new from **Google Play**

Popular + random from **AppChina**, **Mi.com**, **Anzhi**

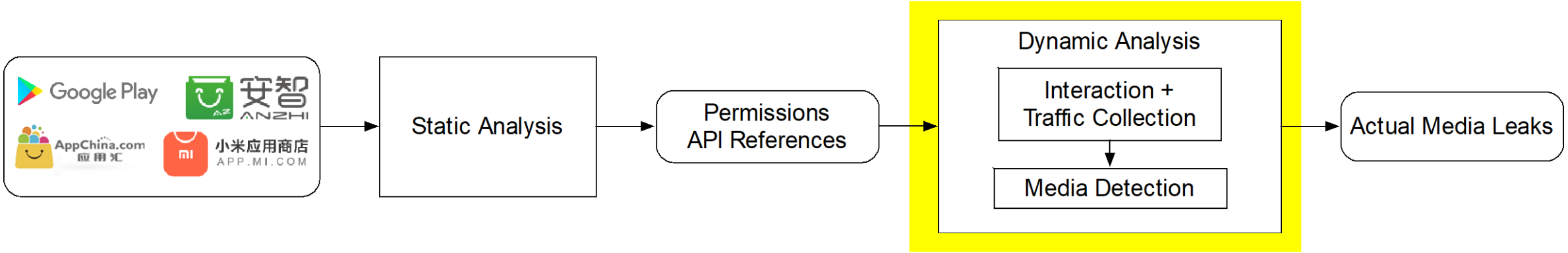
Camera or audio permission

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



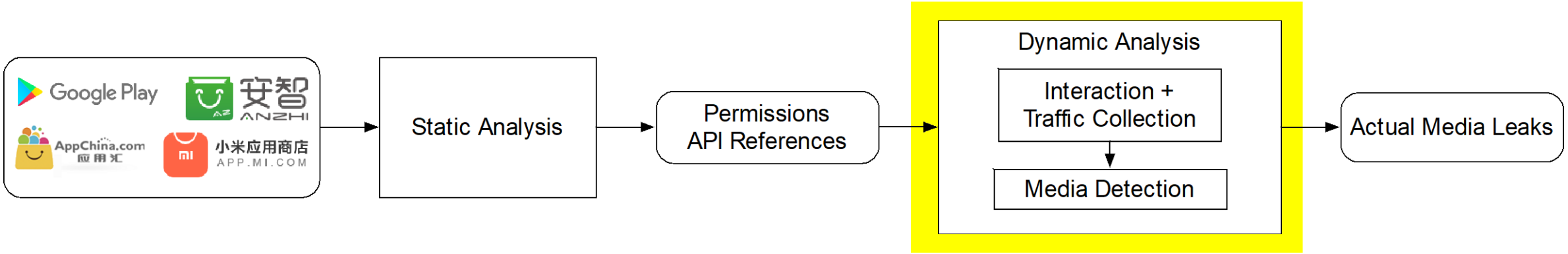
Static Analysis

- Permission analysis (camera, record audio)
- Media API references (camera, record audio, video, screen capturing)
- Media API references found in third-party libraries



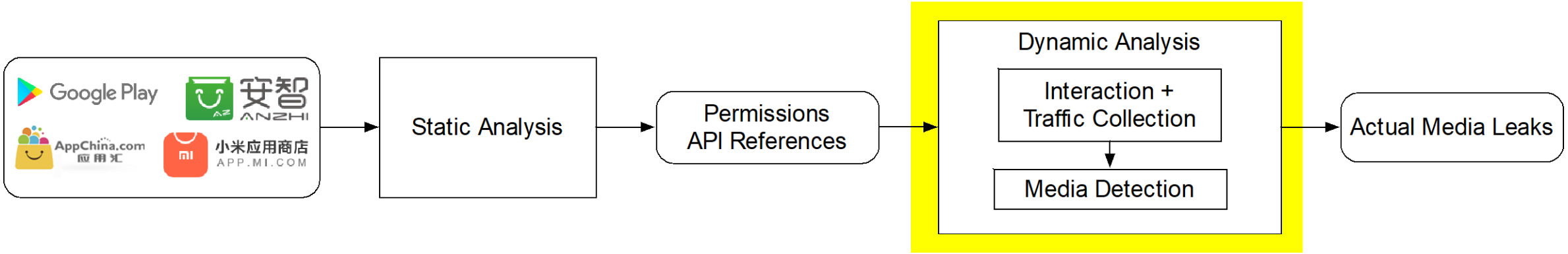
Dynamic Analysis

- 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

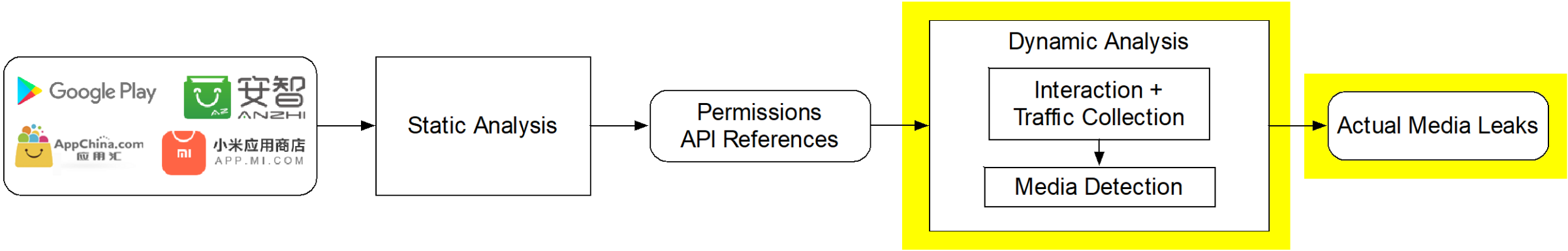
- Test environment
- Automated interaction
  - Monkey for 5,000 events
- Recording network traffic
  - Mitmproxy to intercept traffic



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

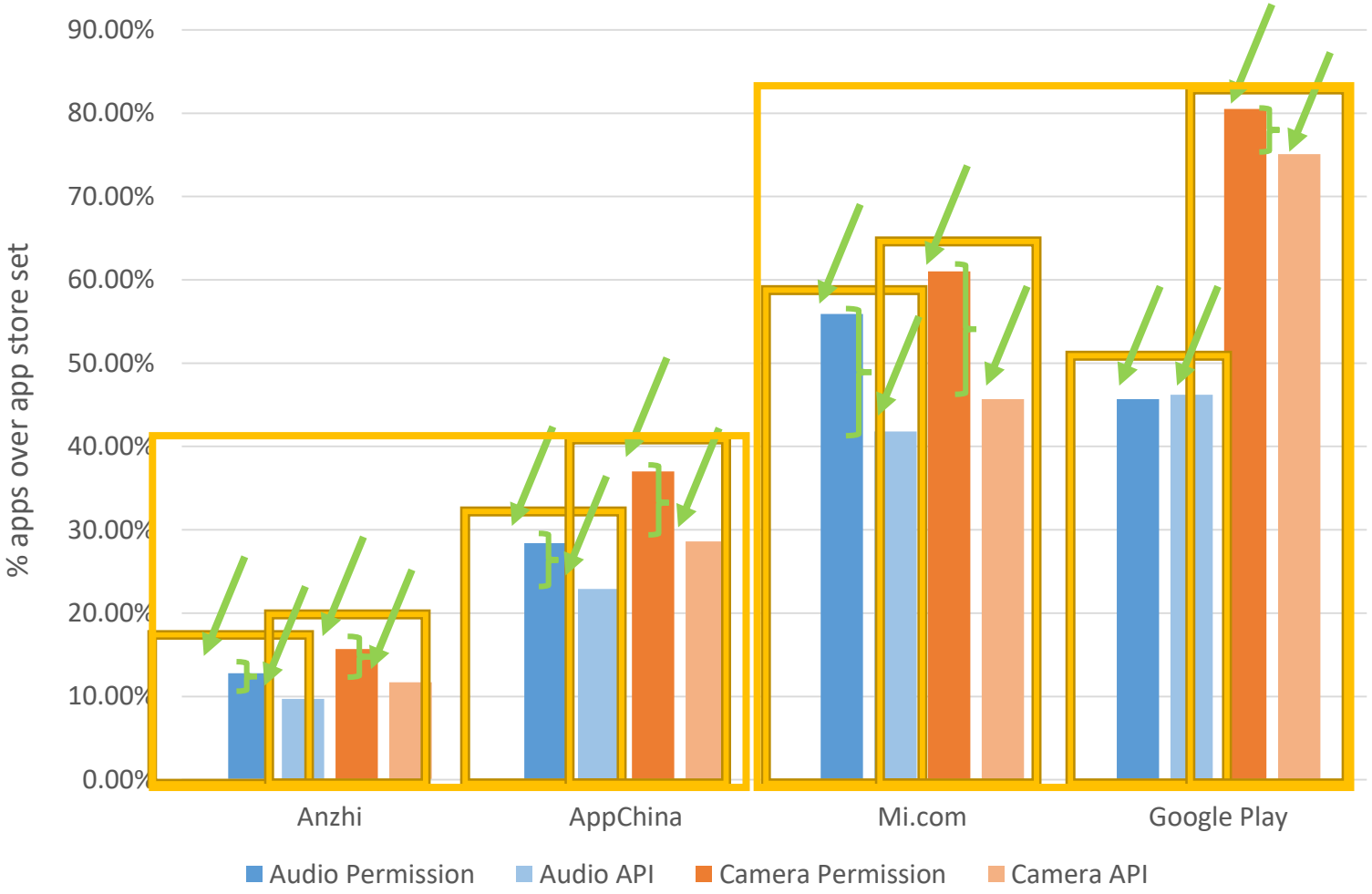
Category	Supported	Unsupported
Audio	**3gp**, **aac**, id3v2, **m4a**, **ogg**, wav	**raw**
Image	**bmp**, gif, **jpg**, **png**, **webp**	
Video	**3gp**, **mp4**, **webm**	



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)

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

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

Fancy that

59 SHARE

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

Your phone is probably spying on you

By Andy Meek, BGR

July 5, 2018 | 10:25am | Updated

No, your smartphone is not lis

But it may be watching you

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

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

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

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

Yahoo Finance Video • July 5, 2018

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

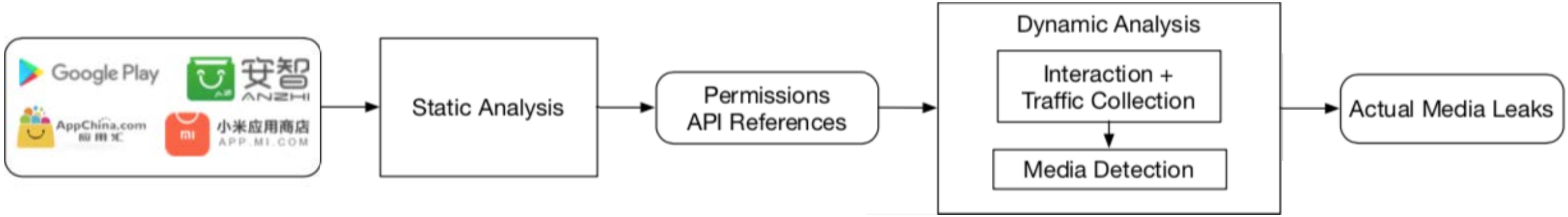
- Most apps have over-provisioned permissions
  - Susceptible for abuse from third parties
- 12 cases of unexpected or unencrypted media
  - 9 cases of third party sharing
- Screen recording video sent to a third party library
  - Sensitive input fields
  - No permissions or notification to the user

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

Threat Model

- Goal: media exfiltration from Android apps over the network
- Permissions
  - Not granted
  - Granted for a user-identifiable purpose
- Leaks: unexpected or suspicious

Experiment Design



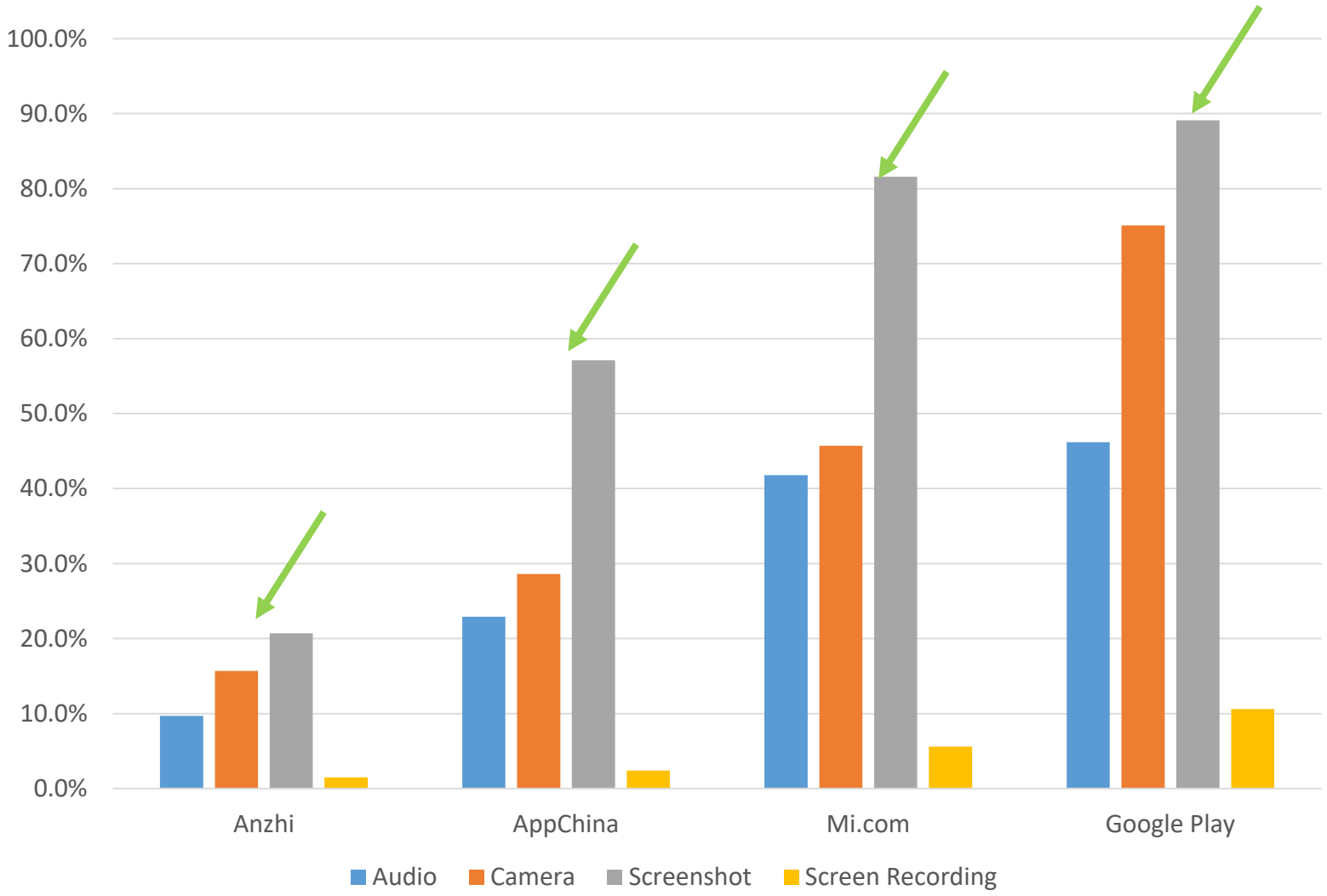
Permissions and API references

Store	# of Apps	Audio Permission	Audio API	Camera Permission	Camera API	Screenshot API	Screen recording API	External Storage Permission
Anzhi	883	12.8%	9.7%	15.7%	11.7%	20.7%	1.5%	23.4%
AppChina	468	28.4%	22.9%	37.0%	28.6%	57.1%	2.4%	94.0%
Mi.com	392	55.9%	41.8%	61.0%	45.7%	81.6%	5.6%	97.4%
Google Play	15,627	45.7%	46.2%	80.5%	75.1%	89.1%	10.6%	92.7%
All	17,260	43.8%	43.6%	75.6%	70.1%	84.6%	9.8%	89.9%

Permissions and API references

- Large percentages of apps request media permissions
  - Smaller percentage actually call methods that use those permissions
- Multipurpose APIs for screenshots and accessing external storage
  - High false positive rate
- Nontrivial inconsistency between permissions and API calls

Static: API References



March 26
- Initial disclosure to GoPuff

March 27
- Lawyer contacts NEU and accuses us of extortion
- No direct reply to our team

March 29
- After some back and forth, updated privacy policy – by removing it?

May 15
- Notified GoPuff of absent privacy policy

- ???

June 7
- Informed that the lawyer is no longer with company, but introduced to CTO
- Start talking about Appsee & the screen recording

June 21
- GoPuff pulls Appsee from iOS & Android builds and updates their privacy policy

March 26	• Initial disclosure to GoPuff
March 27	• Lawyer contacts NEU and accuses us of extortion • No direct reply to our team
March 29	• After some back and forth, updated privacy policy – by removing it?
May 15	• Notified GoPuff of absent privacy policy
	• ???
June 7	• Informed that the lawyer is no longer with company, but introduced to CTO • Start talking about Appsee & the screen recording
June 21	• GoPuff pulls Appsee from iOS & Android builds and updates their privacy policy

appsee

- ”verges on defamation”
- Provided info about screen recording
  - didn’t have to do with privacy concerns
- Asked us to remove Appsee / screen recording
- We replied to their points and clarified the privacy risk
- No reply

Google

- First reported as a security vulnerability
- Passed to privacy team
- “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.”

Screen Capturing

- Testfairy
  - Screenshots of app while in use
  - Library intended for beta testing
    - App was not a beta version in the Google Play store
    - User not informed of recording, not given a prompt to consent to beta testing