Panoptispy: Characterizing Audio and Video Exfiltration from Android Applications

Elleen Pan, Jingjing Ren, Martina Lindorfer*, Christo Wilson, and David Choffnes
Northeastern University, *TU Wien
Motivation

+ internet connectivity …

ultrasonic beacons for cross-device linking

patents for recognizing user emotion

listening for unlicensed broadcasting

photos taken surreptitiously by shrinking preview to 1x1 pixel
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
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
App Selection

• Apps from Google Play + 3 third-party app stores that requested camera and/or record audio permissions = 17,260 apps

Static Analysis

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

- Android phones w/ automated, random interaction
- Recorded network traffic

- Extracted media using **file magic numbers**
  - E.g. JPEG files: FF D8 FF ...

- Validation: test app, known apps, verified detected media
Results

• 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.”
• Removed additional apps beyond our findings

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

Fancy that.

Your phone is probably spying on you

By Andy Meek, BGR

July 5, 2018 | 10:25am | Updated

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.
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
• Need for independent, automated testing to audit apps
Conclusion

• 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
  – Could leak credit card numbers, passwords, unsent messages…
• More work needs to be done on iOS - screen recording behavior also found in major iOS apps

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