The Long-Standing Privacy Debate: Mobile Websites vs Mobile Apps

Elias P. Papadopoulos, Michalis Diamantaris, Panagiotis Papadopoulos, Thanasis Petsas, Sotiris Ioannidis, Evangelos P. Markatos

Presentation on CS-558 by

Bekos Paschalis pbekos@csd.uoc.gr
Plevridi Eleftheria plevridi@csd.uoc.gr

Mobile Services: Evolution through time

Almost two decades ago, the only way a user could access an online service was through **web browsers.**

This meant that the only way of interaction with a service through an mobile device was the use of the browser.

Mobile Services: Evolution through time

In the more recent years, most people have in their possession a mobile device with access to the internet

Moreover, the vast expansion of developer tools enable the creation mobile applications.

Having said that, service providers offer access to their services through their own **mobile applications (apps)**

Mobile Apps - Web Browsers

Both of the two access options have some significant advantages.More specifically :

- **Mobile Apps:**
  - Better support for specific functionalities
  - Providing extra functionalities (e.g. online multiplayer gaming)
- **Web Browsers:**
  - Pre-installed in most mobile devices
  - Provide easy access to a mobile friendly web page

Still, choosing between apps and browsers can be difficult.

A choice based on privacy

- There are many studies trying to compare those two access methods comparing them across different dimensions.
- The majority of those services provide both access option to the users.
- Guiding the choice through privacy-related characteristics:
  - Which one of the two options protects the user’s privacy in the best way?

The issue: Privacy leaks

Privacy leaks can be divided in two large categories:

- **Personally Identifiable Information(PII):**
  - Gender
  - Email address
  - Name
  - Username
  - Birth day

- **Device information:**
  - Installed applications
  - Known SSIDs (**S**ervice **S**et **Id**entifier : network’s name)
  - Connected Wi-Fi
  - Operating system’s build information

Who collects these data and why?

- **First & Third party web monitoring entities:**
  - Advertising companies
  - Web Analytics

- **User specific information can be used for:**
  - **Targeted advertising:** Providing adds that reflect the interests of the user provided by an advertising entity(ads are the main source of revenue for applications)
  - **Tracking:** Persistent tracking of the user by monitoring entities (e.g. cookies) to monitor the user's behavior through the internet (actions/interests/behavior)
  - **Device Fingerprinting**: Tracking entities can link **anonymous** with **eponymous sessions** or link app with web sessions.

Background: Third party tracking - Websites

- **Web sites track users and sessions by using cookies:**
  - **Cookies:** Small text files that hold information about the user, generated by a visited site and stored in the client’s web browser (identify user and improve browsing experience)

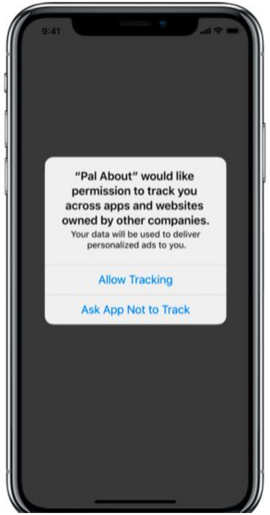
- By storing a cookie to the client side, and advertising or analytics company can identify a user along with his interests, preferences and past behavior.

- **Tracking mechanisms:**
  - **Web beacons:** A technique used on websites or emails to check (invisibly) if a user has accessed the same content.
  - **Cookie synchronization:** Is a process that enables all members of an ad transaction to o synchronize their cookies and share the incorporated user's data from different websites with each other.

Background: Third party tracking - Apps

- As we previously mentioned, advertising is the **main source of revenue** for mobile applications. In order to incorporate ads, the developers include **external libraries** in their application.
    - Those libraries are used to request ads at runtime, filling the ad slots.

- **With what permissions are those ads embedded?**
    - In order to facilitate delivery of personalized advertisement ad-libraries inherit all provided permission of the said app.
    - Such permissions can be access to the phone/contacts/device characteristics etc.
    - This way, ad-libraries can track the users.

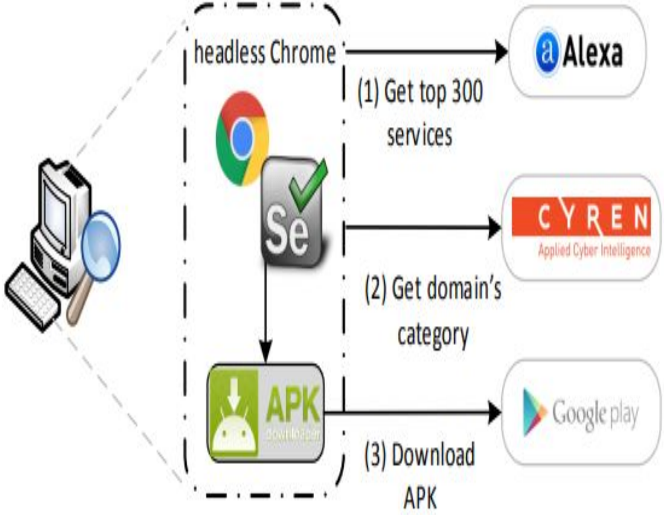


“Pal About” would like permission to track you across apps and websites owned by other companies.

Your data will be used to deliver personalized ads to you.

The Dataset: Collecting the data

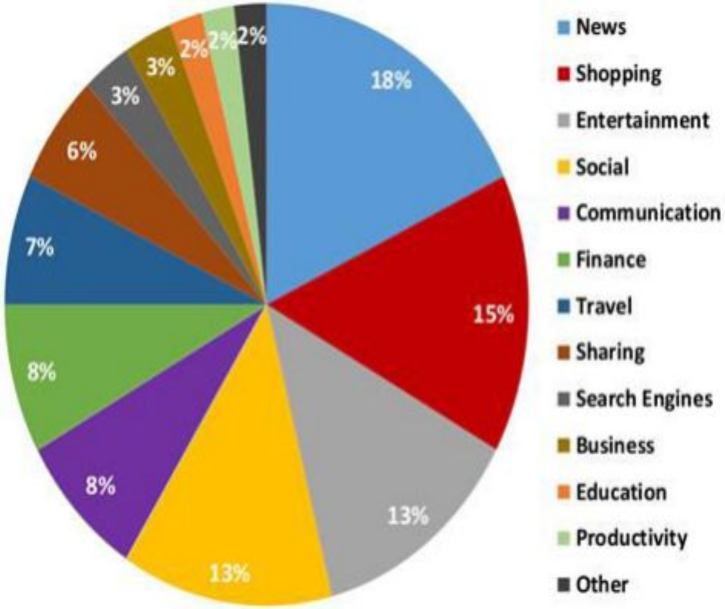
- **The data collection process:**
  - Requesting the top 300 online web services from Alexa
  - Finding the corresponding mobile app & its category (news, shopping, social etc.)
  - Download the full Android packages from Google Play (if any) with the help of Selenium suite (web browser automation tool).



App Categories

- Using **CYREN** intelligence services to extract the category of each service from the Dataset.

- The figure at the right, depicts the classification of each app based on the content category

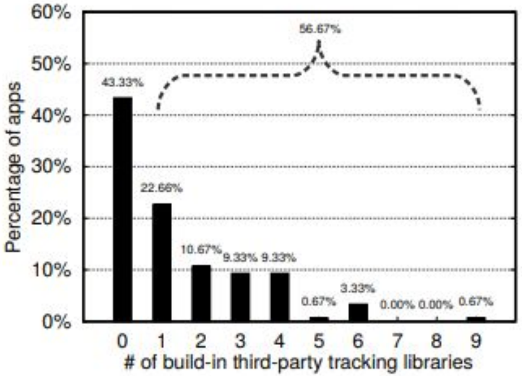


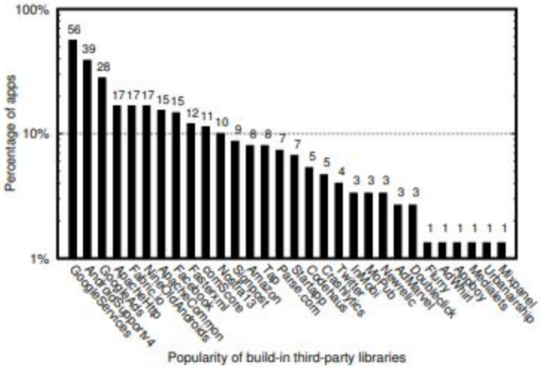
Third Party in-app libraries:

➢ Using LibRadar, an Android tool, to detect embedded libraries if any

❖ 56.67% of the apps include at least one third party library

❖ Popularity of the top 3rd party libraries with Google ads residing in 28% of the apps





Monitoring Outgoing Traffic (i)

- The monitoring process is described below:
  - Using a Nexus 6 smartphone run each online service:
    - From corresponding app
    - From web browser using Firefox(chosen due to its ability to support extensions)
  - Each experiment lasts 20 minutes performing the same following actions : login/share/registration/search/share etc.
  - Capturing HTTP and HTTPS traffic:
    - Using **raspberry PI2** device configured as access point running **mitmproxy** (SSL - capable monitoring proxy)

Monitoring Outgoing Traffic (ii)

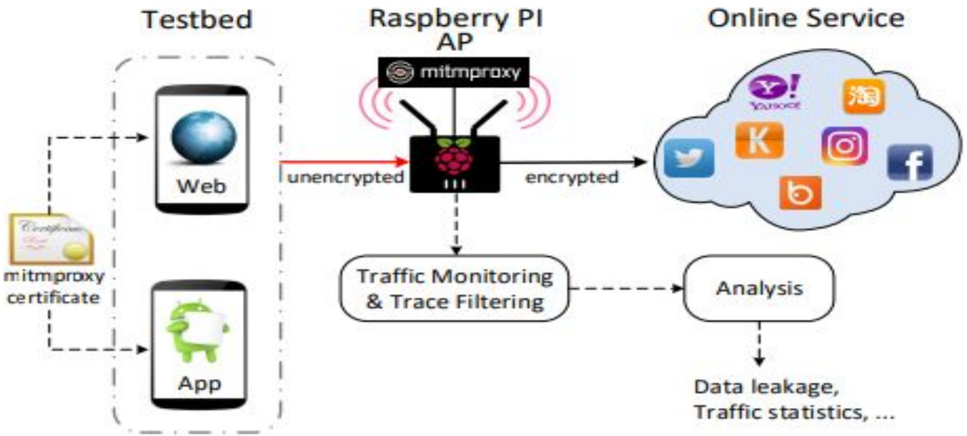
- All the captured traffic (both HTTP & HTTPS) traces are forwarded in the **Monitoring & Trace Filtering** module, where all tracking related requests are identified by using a blacklist for filtering.

- This blacklist is the **AdAway** mobile based blacklist extended with manual inspected entries.

- Finally, the identified tracking requests are forwarded to the **Analysis** module, where performing pattern matching (using a list of ID keywords) producing the statistics and privacy leak analysis results.

Monitoring Outgoing Traffic (iii)

The overall network monitoring process as described in the previous slides, depicted in the following figure:



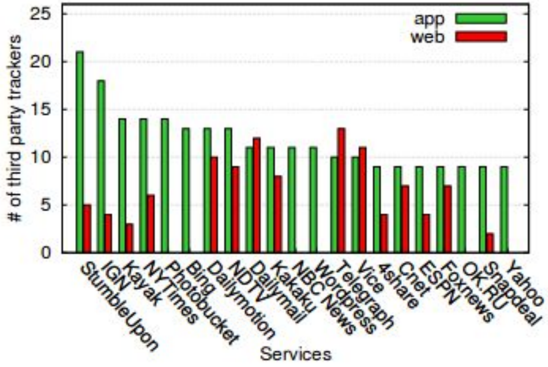
Privacy Leak Analysis (i)

- Measuring the quantity and type of information leaked as well as the diffusion of those leaks.
- A service might leak:
  - Personal data
  - Device Identifiers
- Those identifiers allow a tracking entity to follow a user inside a network without any deletable cookies or resettable Advertising IDs.

Privacy Leak Analysis (ii): Diffusion

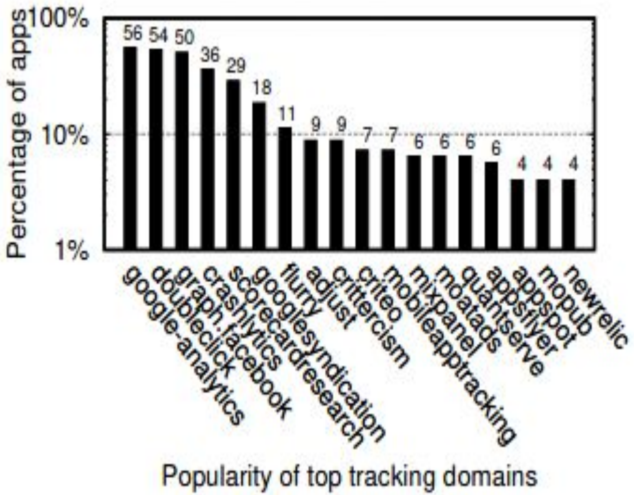
- ❖ The next step is identifying the third party entity that receives the leaked information

- ❖ The observation made was that apps send the leaked information to a larger number of trackers (on average):
    - ➢ **Apps :** sending information to an average of 15 trackers
    - ➢ **Browsers:** sending the information to an average of 5 trackers



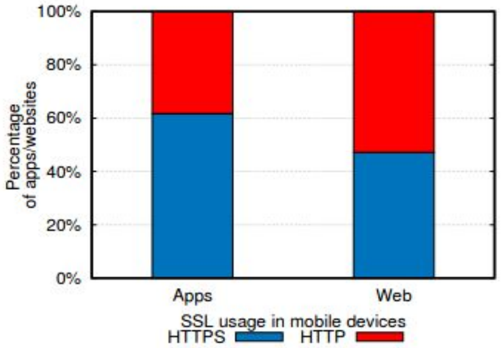
Privacy Leak Analysis (iii): Diffusion

- ❖ Moreover, we that the top two third parties that receive the leaked information both belong to google:
  - ➢ Google-analytics : **56%** of the apps leak information
  - ➢ Doubleclick (ad service bought by Google in 2008) : **54%** of the apps leak information
- ❖ And the third one belongs to Facebook
  - ➢ Graphs-facebook : **50%** of the apps leak information



Privacy Leak Analysis (iv):Encrypted Sessions

- The results of the analysis revealed that in mobile apps:
  - Only 18.97% of the apps use **exclusively** HTTPS
  - 2.58% use solely HTTP
  - 78.45% use a mixture of both.
- Compared to browsers, apps are more likely to use HTTPS
  - 62% of total apps
  - 47% of web browsers

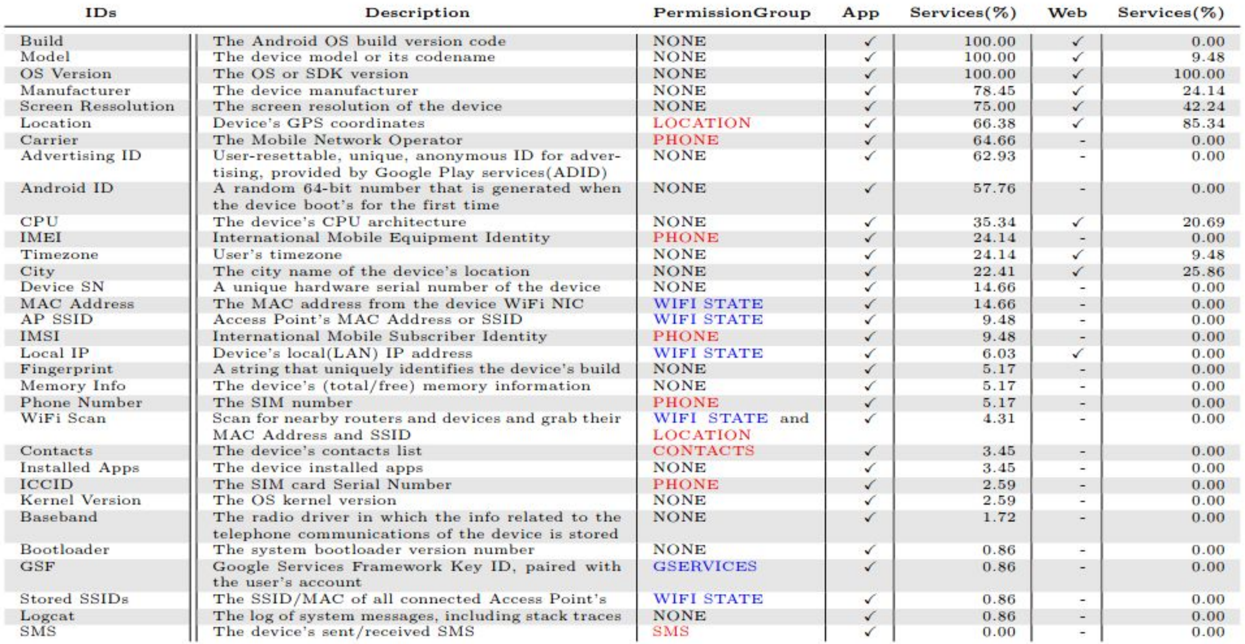


Privacy Leak Analysis (v): Identifiers Leaked

- Noticeable, 57.76% of the apps leak such identifiers (no browser leaks such information, not having access to it)

- Some remarkable leaked identifiers are:
  - **Android ID** (57.76%):Unique identifier of each device
  - **Location** (75%): Device's GPS coordinates
  - **Wifi Scan** (4.31%): Nearby routers (MAC addresses & SSID)
  - **Contacts** (3.45%): The contacts list

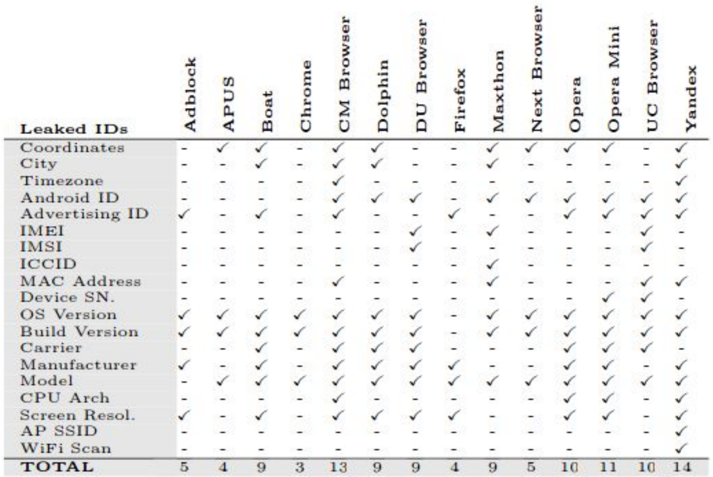
Privacy Leak Analysis (vi): Identifiers Leaked Table



IDs	Description	PermissionGroup	App	Services(%)	Web	Services(%)
Build	The Android OS build version code	NONE	✓	100.00	✓	0.00
Model	The device model or its codename	NONE	✓	100.00	✓	9.48
OS Version	The OS or SDK version	NONE	✓	100.00	✓	100.00
Manufacturer	The device manufacturer	NONE	✓	78.45	✓	24.14
Screen Ressolution	The screen resolution of the device	NONE	✓	75.00	✓	42.24
Location	Device's GPS coordinates	LOCATION	✓	66.38	✓	85.34
Carrier	The Mobile Network Operator	PHONE	✓	64.66	-	0.00
Advertising ID	User-resettable, unique, anonymous ID for advertising, provided by Google Play services(ADID)	NONE	✓	62.93	-	0.00
Android ID	A random 64-bit number that is generated when the device boot's for the first time	NONE	✓	57.76	-	0.00
CPU	The device's CPU architecture	NONE	✓	35.34	✓	20.69
IMEI	International Mobile Equipment Identity	PHONE	✓	24.14	-	0.00
Timezone	User's timezone	NONE	✓	24.14	✓	9.48
City	The city name of the device's location	NONE	✓	22.41	✓	25.86
Device SN	A unique hardware serial number of the device	NONE	✓	14.66	-	0.00
MAC Address	The MAC address from the device WiFi NIC	WIFI STATE	✓	14.66	-	0.00
AP SSID	Access Point's MAC Address or SSID	WIFI STATE	✓	9.48	-	0.00
IMSI	International Mobile Subscriber Identity	PHONE	✓	9.48	-	0.00
Local IP	Device's local(LAN) IP address	WIFI STATE	✓	6.03	✓	0.00
Fingerprint	A string that uniquely identifies the device's build	NONE	✓	5.17	-	0.00
Memory Info	The device's (total/free) memory information	NONE	✓	5.17	-	0.00
Phone Number	The SIM number	PHONE	✓	5.17	-	0.00
WiFi Scan	Scan for nearby routers and devices and grab their MAC Address and SSID	WIFI STATE and LOCATION	✓	4.31	-	0.00
Contacts	The device's contacts list	CONTACTS	✓	3.45	-	0.00
Installed Apps	The device installed apps	NONE	✓	3.45	-	0.00
ICCID	The SIM card Serial Number	PHONE	✓	2.59	-	0.00
Kernel Version	The OS kernel version	NONE	✓	2.59	-	0.00
Baseband	The radio driver in which the info related to the telephone communications of the device is stored	NONE	✓	1.72	-	0.00
Bootloader	The system bootloader version number	NONE	✓	0.86	-	0.00
GSF	Google Services Framework Key ID, paired with the user's account	GSERVICES	✓	0.86	-	0.00
Stored SSIDs	The SSID/MAC of all connected Access Point's	WIFI STATE	✓	0.86	-	0.00
Logcat	The log of system messages, including stack traces	NONE	✓	0.86	-	0.00
SMS	The device's sent/received SMS	SMS	✓	0.00	-	0.00

Privacy Leak Analysis (vii): Browser Leakage

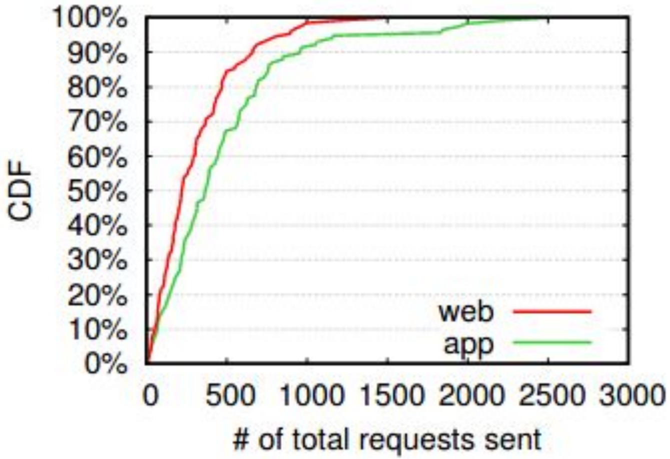
- It is important to note, that web browsers are also applications
- As a consequence, they might leak information to external entities too.
- There are cases where an identifier is being leaked from the web browser itself and not the visited page
    - Even the **AdBlock** browser send a request to a tracking domain



Leaked IDs	Adblock	APUS	Boat	Chrome	CM Browser	Dolphin	DU Browser	Firefox	Maxthon	Next Browser	Opera	Opera Mini	UC Browser	Yandex
Coordinates	-	✓	✓	-	✓	✓	-	-	✓	✓	✓	✓	-	✓
City	-	-	✓	-	✓	✓	-	-	✓	-	-	-	-	✓
Timezone	-	-	-	-	✓	-	-	-	-	-	-	-	-	✓
Android ID	-	-	-	-	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
Advertising ID	✓	-	✓	-	✓	-	-	✓	-	-	✓	✓	✓	✓
IMEI	-	-	-	-	-	-	✓	-	✓	-	-	-	✓	-
IMSI	-	-	-	-	-	-	✓	-	-	-	-	-	✓	-
ICCID	-	-	-	-	-	-	-	-	✓	-	-	-	-	-
MAC Address	-	-	-	-	✓	-	-	-	✓	-	-	-	✓	✓
Device SN.	-	-	-	-	-	-	-	-	-	-	-	✓	✓	-
OS Version	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
Build Version	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓	✓	✓	✓
Carrier	-	-	✓	-	✓	✓	✓	-	-	-	✓	✓	✓	-
Manufacturer	✓	-	✓	-	✓	✓	✓	✓	-	-	✓	✓	-	✓
Model	-	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
CPU Arch	-	-	-	-	✓	-	-	-	-	-	✓	✓	-	✓
Screen Resol.	✓	-	✓	-	✓	✓	✓	✓	-	-	✓	✓	-	✓
AP SSID	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
WiFi Scan	-	-	-	-	-	-	-	-	-	-	-	-	-	✓
TOTAL	5	4	9	3	13	9	9	4	9	5	10	11	10	14

Privacy Leak Analysis (viii): Performance cost of user tracking

- Tracking does not only cost on the privacy of the user
- The contented requested (ads) is:
  - Irrelevant to the initial content the user was browsing
  - Costing in MBytes consumption (fetching of the content)
- Moreover, the number of app requests is greater than the browsers requests
  - 367 for the 50% of the apps
  - 221 for the 50% of the browsers



Privacy Leak Analysis (ix): Summary

- To answer the question **“Which of the two protects the user’s privacy in the best way?”:**
  - **Mobile Browsers do** (leaking significantly less information than mobile apps)
- So choosing an access service based on privacy seems to have a straightforward answer.

- But this might not always be possible.
  - What can we do to fortify the mobile apps users’ privacy?

AntiTrackDroid

The solution for application users!

What is it ?

- A module that filters all **outgoing** requests
- Blocks the ones delivering tracking information

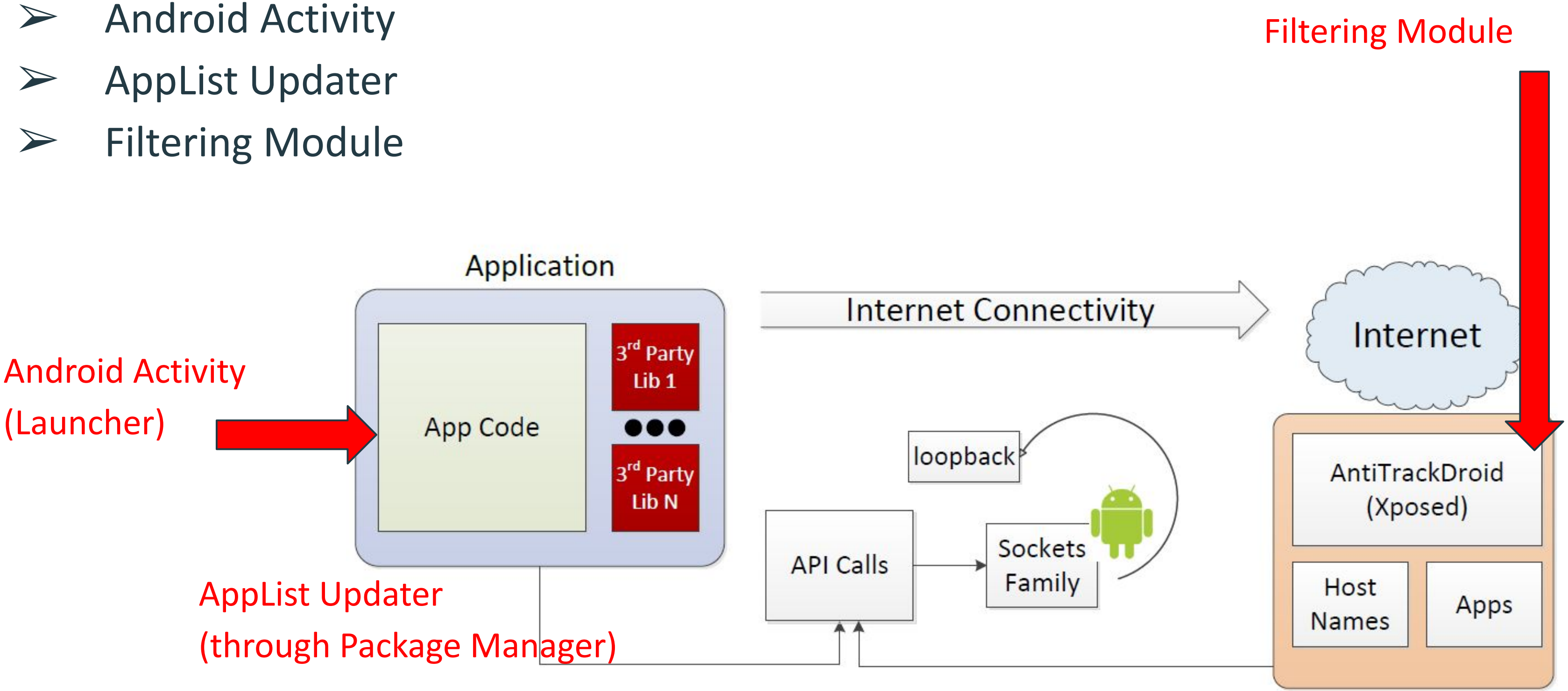
Why is it good ?

- Can operate for ALL apps.
- No need for extra infrastructure (VPN, Proxy)
- Uses Xposed Android Framework

By using this, AntiTrackDroid can check all outgoing requests, and see if the destination’s Domain name is on a BlackList of mobile trackers. If yes, the request is blocked!

How does it work ?

- ➢ Android Activity
- ➢ AppList Updater
- ➢ Filtering Module



How does each of the components work? (1)

Launcher Activity Module 🚀

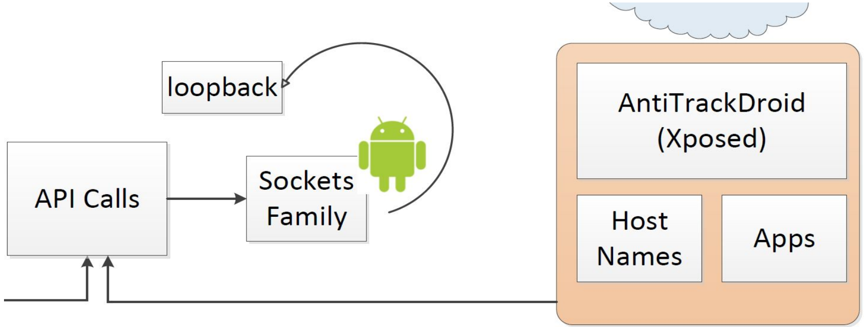
- Graphic UI for allowing users to configure the Filtering module.
- Users can Load different blacklists or exclude an app from the filtering
- Maintains 2 different Hash sets
  - The 1st contains tracking domain names loaded from blacklist
  - The 2nd contains the apps being monitored

Why? Lookups reduced at O(n) = per request latency reduced

How does each of the components work? (2)

Filtering Module

- Uses Xposed (check dst domain name if exists in blacklist)
- Apps send data using TCP sockets. The Apps on the hashset of monitored apps, open their own TCP sockets. **This** module, hooks on the constructor of the socket.
- Re-write the dest IP address with localhost in case of a blocked request.
- By redirecting to loopback, possible crashes are avoided!



How does each of the components work? (3)

AppList Updater

- ❖ Users may install or uninstall apps whenever they like.

- ❖ Through package manager and a broadcast receiver
  - ➢ Update lists for monitored apps (in the background)

Evaluation - Privacy Performance

How? Inspect the identifiers leaked to the network

- AntiTrackDroid is able to reduce the number of leaked identifiers about 27,41%
- Note that the module blocks 3rd party trackers.
- The rest of the leaking exists due to 1st party domains and content providers (e.g. CDNs)

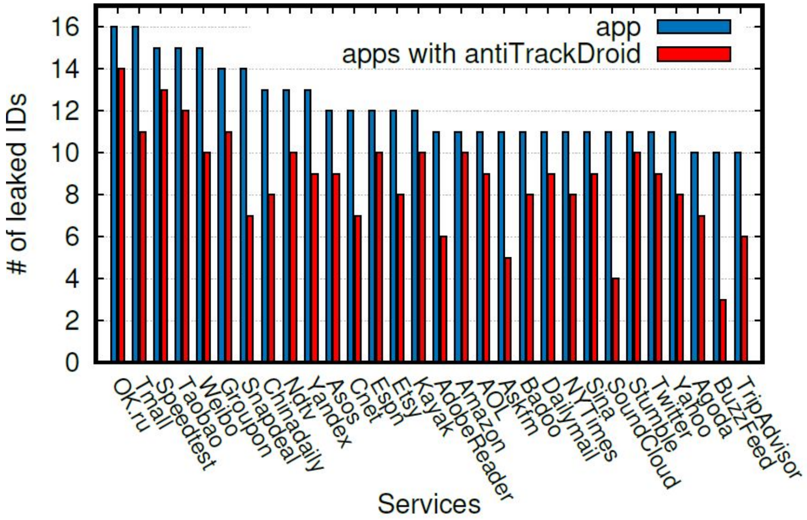


Fig. # of leaked ID's for the 30 more leaking apps.

Evaluation - Latency overhead(1)

Privacy...? Improved! What about Latency?

- Extra checks on the Blacklist of Trackers (66k entries) increases latency

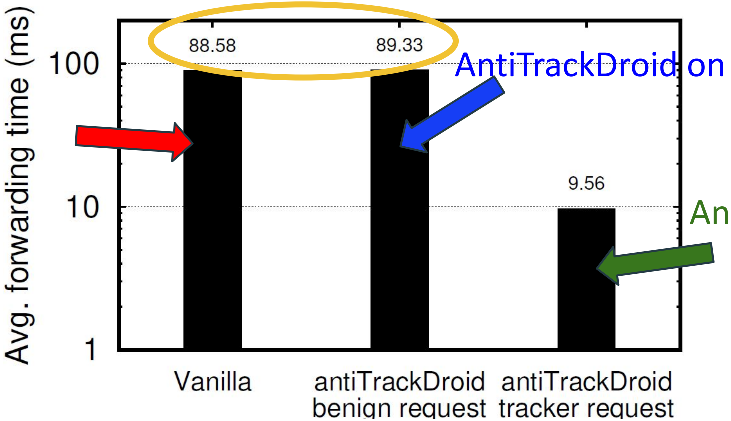
But..

- Latency imposed by tracker’s requests is blocked so, their latency will be none. Right ?

Evaluation - Latency overhead(2)

Run some tests...

- ❖ Created 1000 requests of 15KB each and send them to the same server
  - ➢ With ATD switch off
  - ➢ With ATD enabled and domain of the server **not blacklisted**
  - ➢ With ATD enabled and domain of the server **blacklisted**



AntiTrackDroid off

AntiTrackDroid on but server not on BlackList

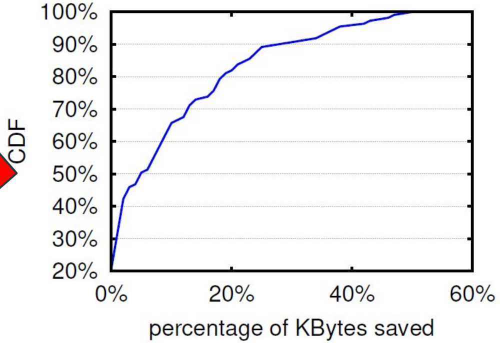
AntiTrackDroid on and Tracker Request

Latency DROPS to <10ms
=> performance improved!

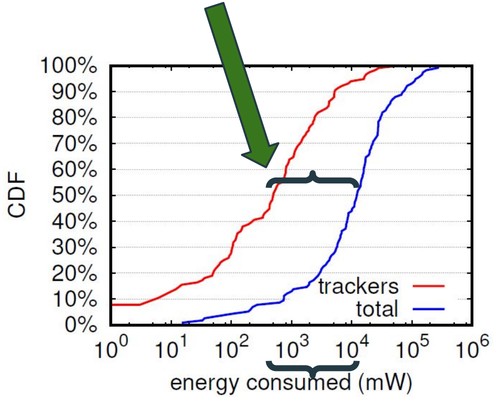
Evaluation - Benefits from usage

- Other Benefits ?
- They run the same tests to calculate the outgoing bytes of requests and the incoming bytes of their responses.
- Noticed that Blocking the Tracking Requests also reduced :
  - Transferred Bytes (MB from data plan)
  - Energy consumption

For 50% of the Apps 7.5%mW are saved



(a) Percentage of KBytes saved.



(b) Watts saved due to the less tracking request sent.

For 50% of the Apps the volume of transferred bytes is reduced by ~8%

Related Work

- Numerous studies have been conducted trying to analyze the privacy leaks of application and browsers, without though attempting the direct head to head comparison that was done in this specific study.
- Most of those studies focus on privacy leaks in mobile applications
- Users can now decide what is best for their own privacy.

Conclusion

- Initial Question: Which of the two (browsers or apps) protect the user the best way ? Conducted a comparative study to answer this question.
- Monitored all outgoing traffic and analyzed the leakage of identifiers
- The study showed that **web browsers leak less information than apps!**
- People are still using apps. How to protect them ?
- The authors proposed : AntiTrackDroid
- The evaluation shows that it can reduce privacy leaks by 27%
- Future work on iOS.

Thank you !

Any Questions?