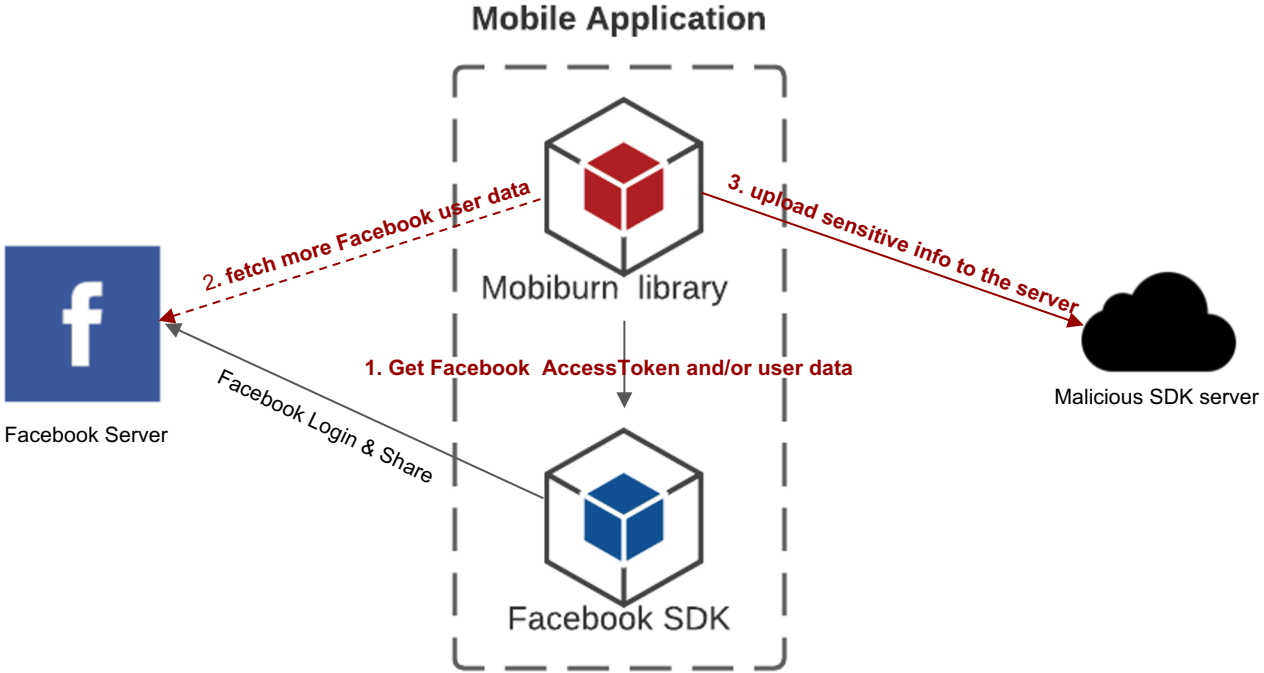
Understanding Malicious Cross-library Data Harvesting on Android

Jice Wang*, Yue Xiao*, Xueqiang Wang, Yuhong Nan, Luyi Xing, Xiaojing Liao, JinWei Dong, Nicolas Serrano, Haoran Lu, XiaoFeng Wang, Yuqing Zhang
(* co-first author)

University of Chinese Academy of Sciences
INDIANA UNIVERSITY BLOOMINGTON
PURDUE UNIVERSITY
XIDIAN UNIVERSITY

Our research found a new attack vector has long been ignored yet with serious privacy impacts



We call this attack Cross-Library Data Harvesting (XLDH)

1. Real-world example

Identify sensitive cross-library call

```
public class a {
    public static String getAccessToken() {
        Class[] param = new Class[0];
        Class clz = Class.forName("com.facebook.AccessToken");
        Method meth1 = clz.getDeclaredMethod(" getCurrentAccessToken ", param)
        Object curToken = meth1.invoke(clz, null);
        Method meth2 = clz.
        getDeclaredMethod(" getToken ", param)
        return meth2.invoke(curToken, null);
    }
}

public class i extends d {
    private static final Uri url = Uri.parse( " https :// graph . facebook . com / v2 .5/ me ");
    public a queryFacebook(String token) {
        HashMap hashMap = new HashMap();
        hashMap.put(" access_token ", token);
        hashMap.put(" fields ", "id,first_name, gender,last_name, link, locale, name, timezone, updated_time, verified, email ");
            return new a(g.getUri(url, hashMap)());
        }
    }

public class p extends a {
        public String a() {
            // send facebook user 's Profile to
            // " api . oneaudience . com / api / devices "
            String token = a.getAccessToken();
            new com.oneaudience.sdk.b().httpSend(new i().queryFacebook(token));
        }
    }
```

Check whether violate Facebook's data sharing policies

Don't transfer user data that you rece m us (including Facebook ID, gender, email, timezone) to any ad network, d broker, influencer network, or other advertising or monetization-related s

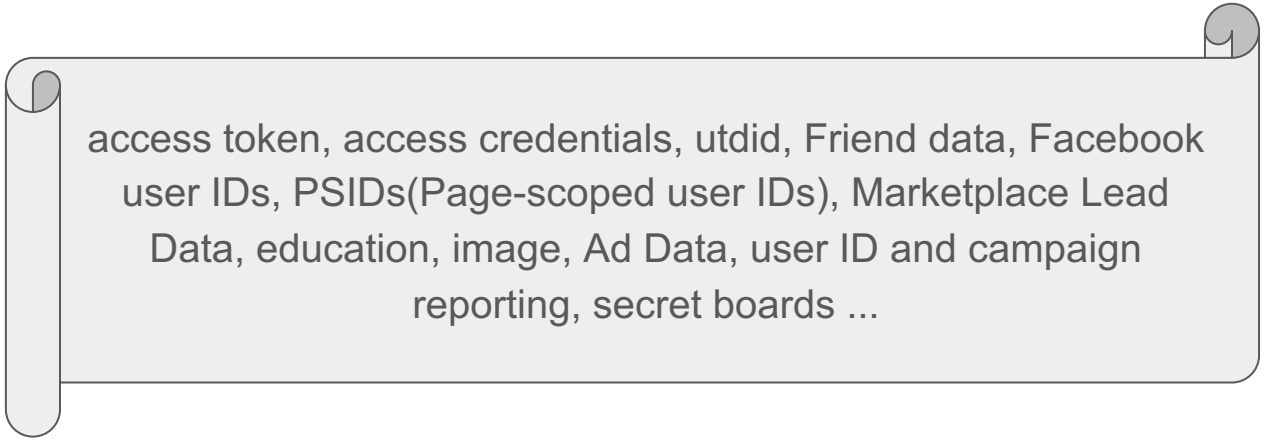
(user data,
No third-party access)

OneAudience

2. Challenges of detection

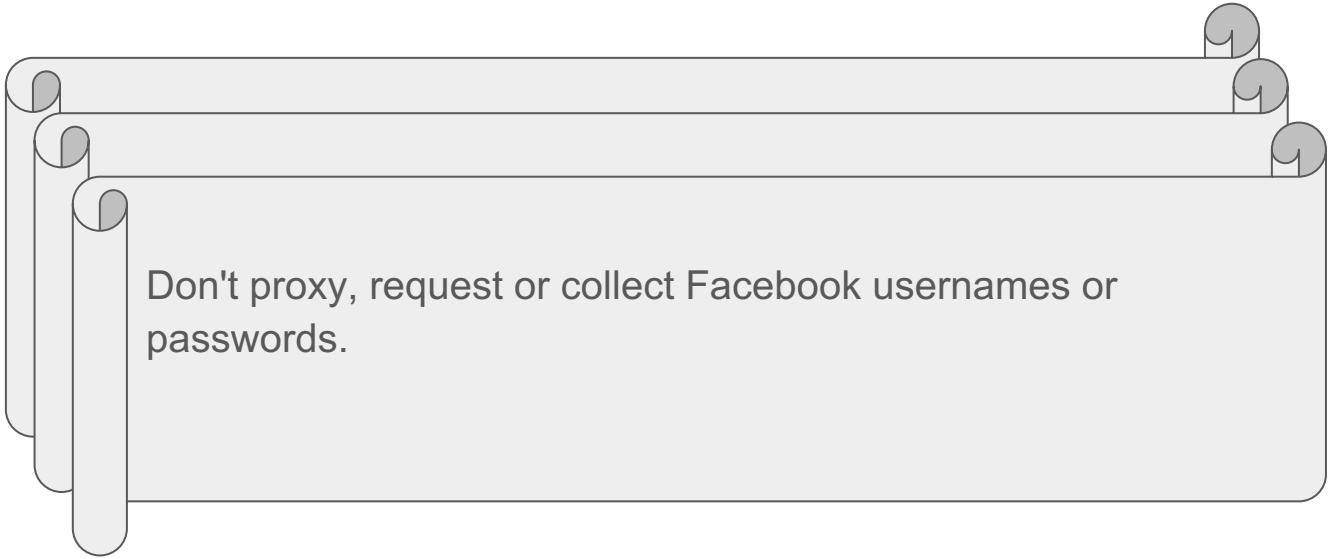
Challenges

1. Data protected under privacy policy is different from that covered by ToS, which failed to be extracted by existing tool like (Polisis / policyLint).



Challenges

2. data sharing policies are complicated, which is nontrivial to analyze



Previous Works

Prior research showed that malicious SDKs could collect users' sensitive data from the host apps

e.g., Razaghpanah, A, et al. "Apps, trackers, privacy, and regulators: A global study of the mobile tracking ecosystem." NDSS. 2018

e.g., Demetriou, Soteris, et al. "Free for All! Assessing User Data Exposure to Advertising Libraries on Android." NDSS. 2016.

Prior research proposed different fine-grained mechanisms to isolate third-party SDKs

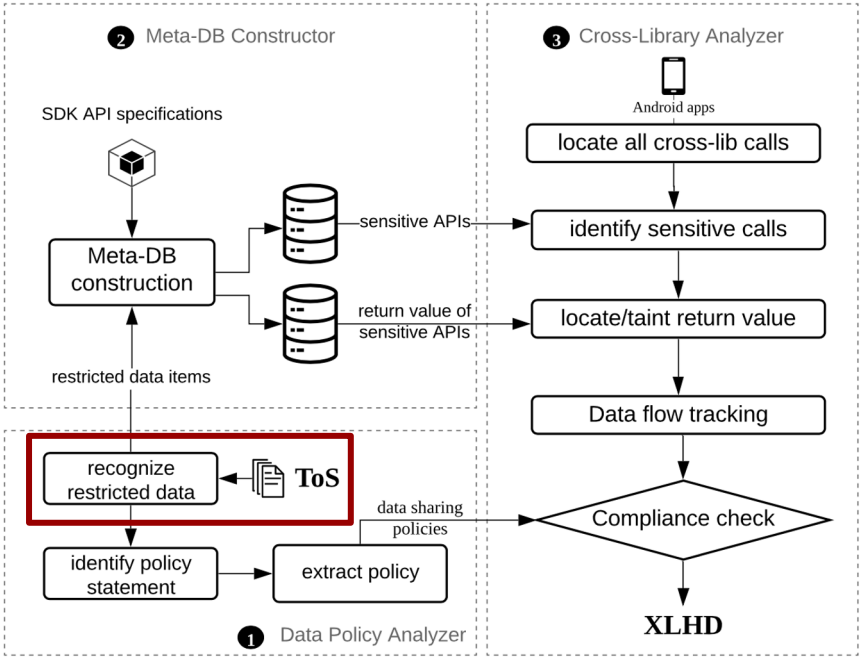
e.g., Nikos Vasilakis, Ben Karel, et al. Breakapp: Automated, flexible application compartmentalization. In NDSS, 2018

Existing tools can not detect such malicious behavior

VirusTotal and Google Play failed to detect such malicious libraries and the apps integrating them

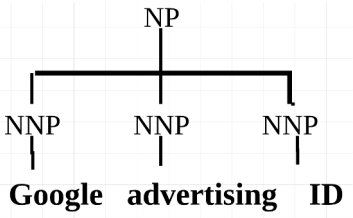
3. Methodology

Methodology Overview



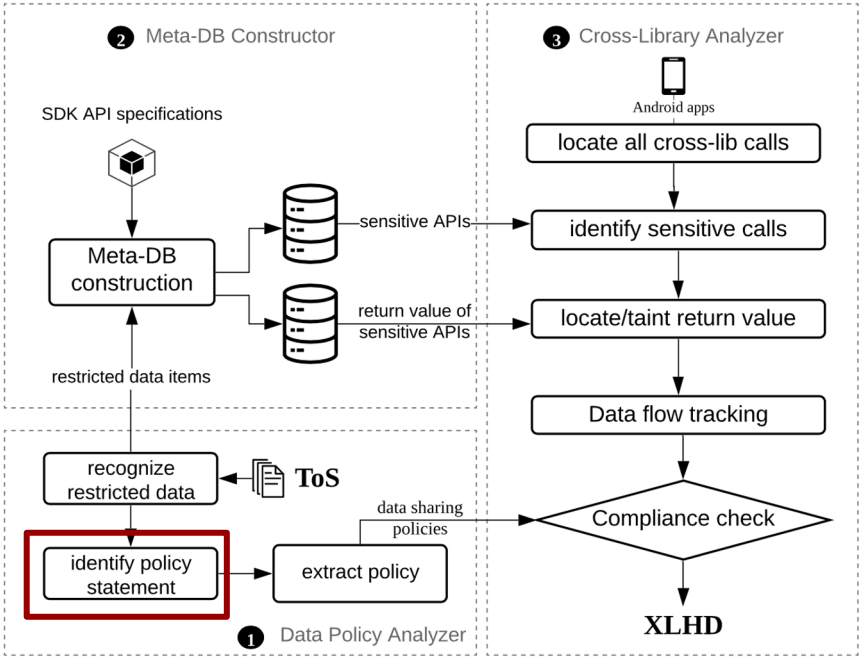
You must have legally valid consent from a Member before you store that **Member's Profile Data**

Customized Name Entity Recognition



- Craft several new features
- Built these features into Stanford NER

Methodology Overview



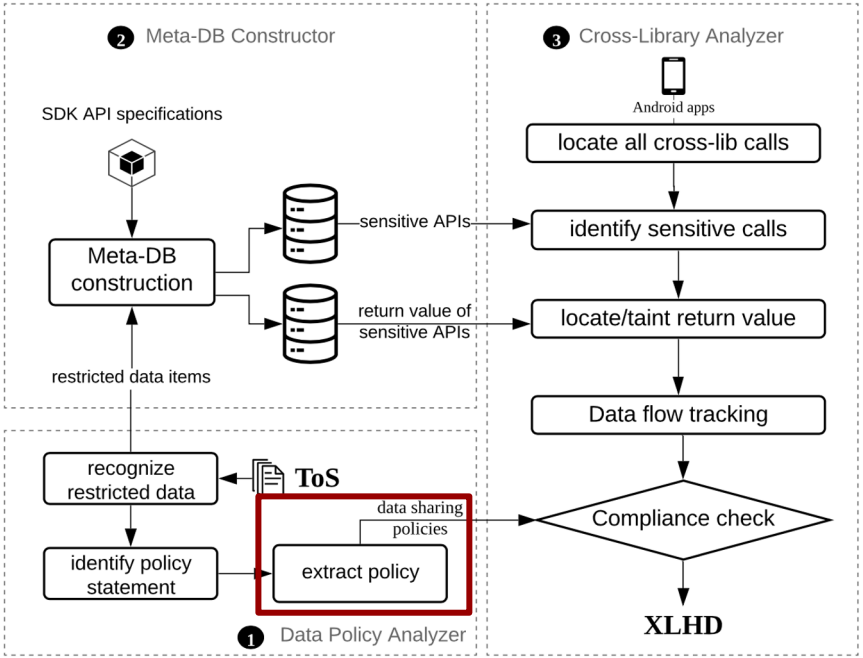
You must have legally valid consent from a Member before you **store** that Member's Profile Data

Identify Policy Statement

- check whether the sentence describing data collection and sharing (e.g., use, collect, transfer, etc.)
- check that the sentence subject is not the SDK itself but library developer

We may also **disclose** name and logo (with or without a link to Your Application) on Our services

Methodology Overview



You must **have legally valid consent** from a Member before you store that **Member's Profile Data**

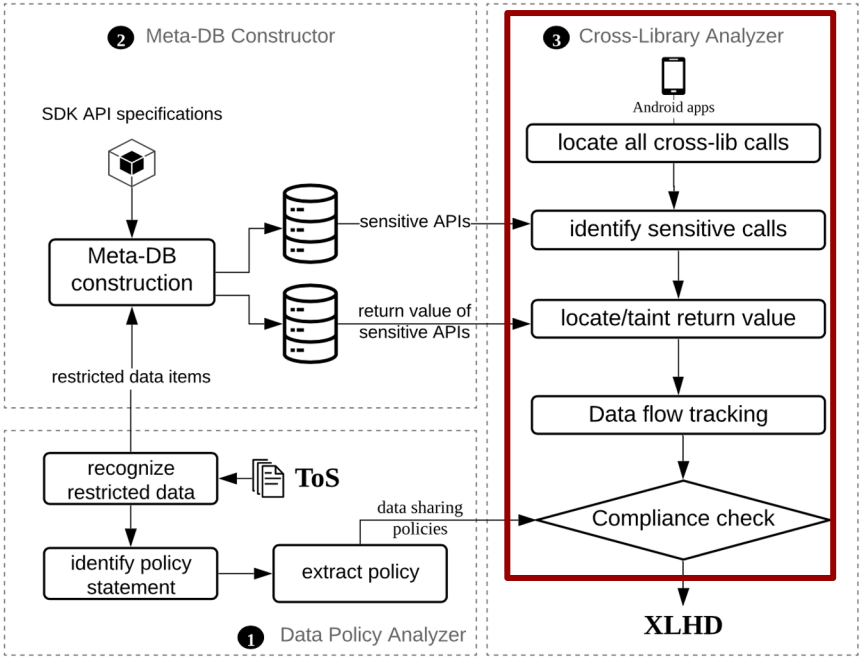
Data sharing policy identification

- transform a sentence into a dependency parsing tree
- leverages the restricted data object as known anchors to locate the condition

(object, condition)

(**Member's Profile Data**, **have legally valid consent**)

Methodology Overview



Compliance check

- Locating cross-library API calls
- Identifying cross-library leaks
- Checking policy non-compliance

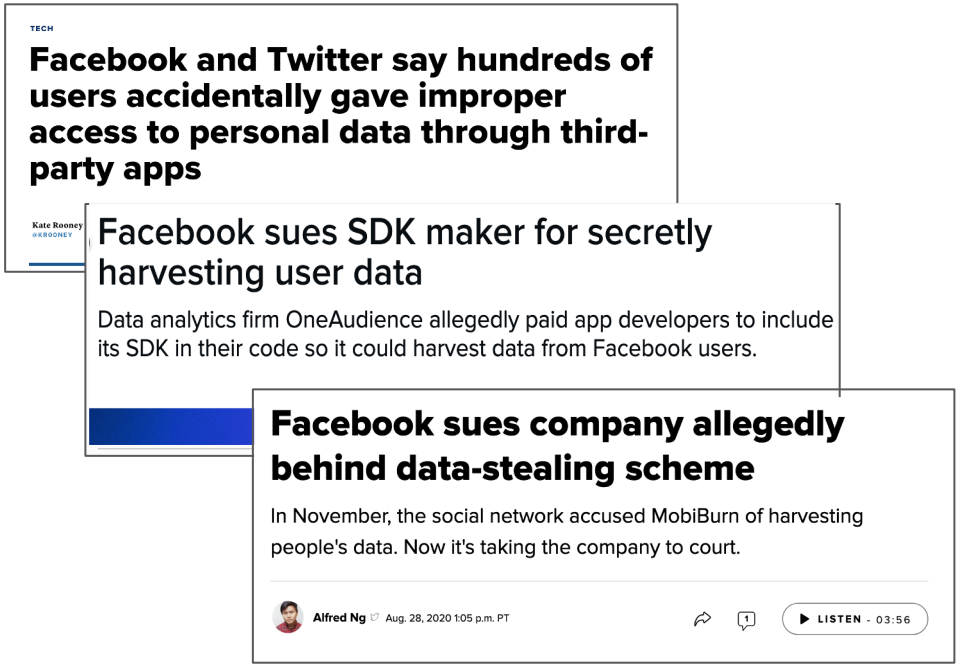
4. Results

Results

- we were able to investigate 1.3 million Google Play apps
- leading to the discovery of 42 distinct libraries stealthily harvesting data from 16 popular SDKs
- affect more than 19K apps with a total of 9 billion downloads.

Affected vendors include but not limited to the following...

More than 30+ medias report our findings



$$$ Awards

Facebook awarded us $30,000 USD through their white hat/bug bounty program, which they told us is one of their largest awards ever;

Google awarded us$5,000 USD and solicited from us the list of affected apps.

Twitter awarded us $576 USD for finding this risk to twitter user privacy.

Thank you!