USENIX Security 2024

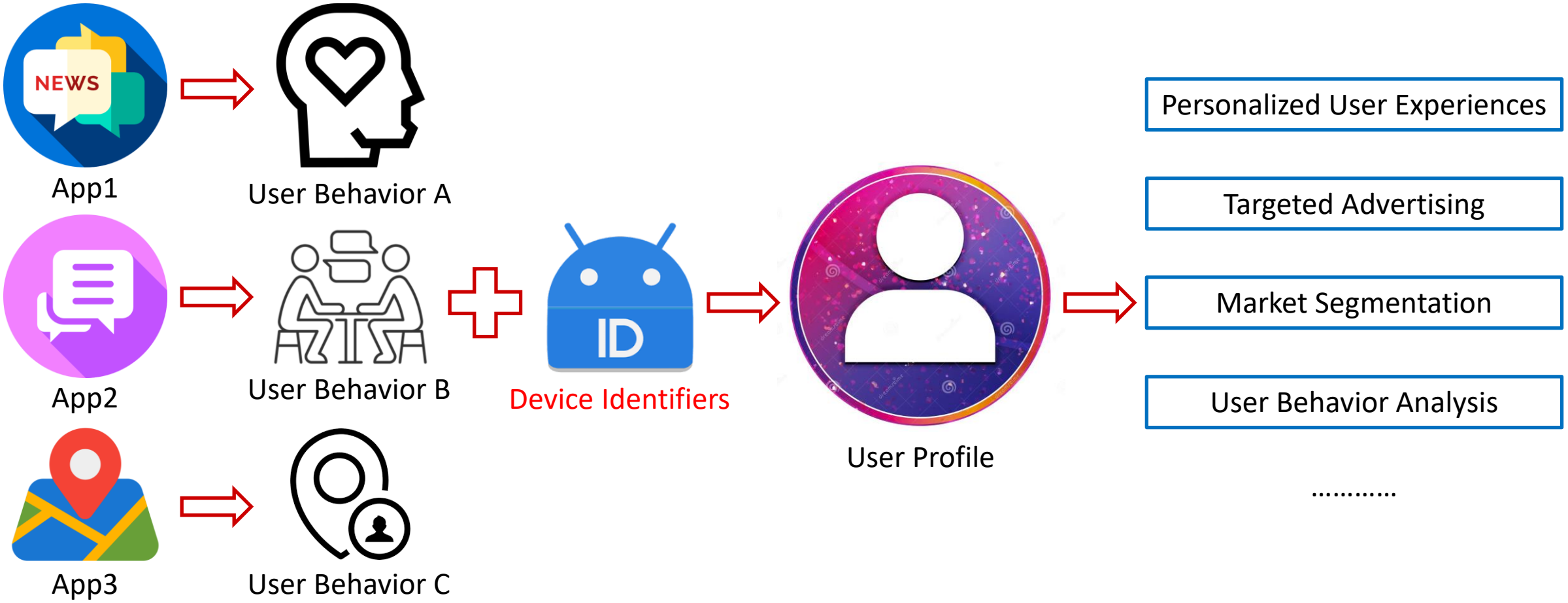
Exploring Covert Third-party Identifiers through External Storage in the Android New Era

Zikan Dong, Tianming Liu*, Jiapeng Deng, Haoyu Wang, Li Li,*

Minghui Yang, Meng Wang, Guosheng Xu and Guoai Xu

User Tracking & Identifiers

- **User tracking & identifiers play a vital role in the mobile ecosystem**



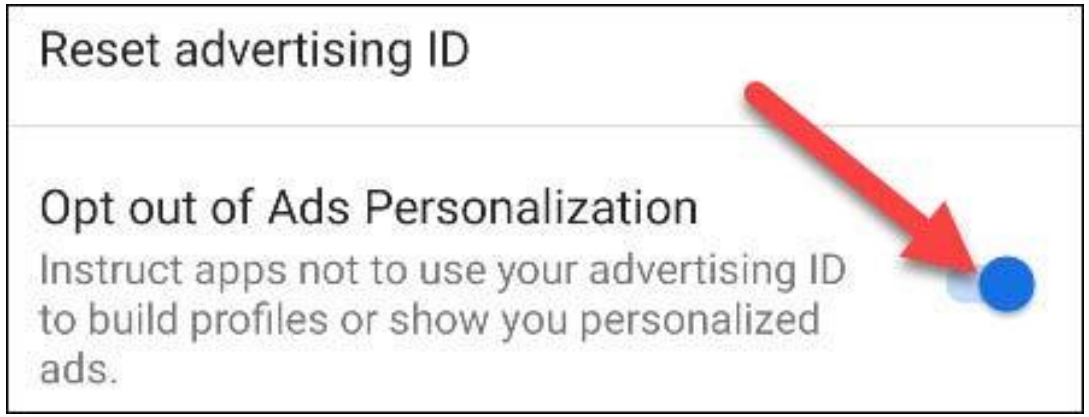
Restrictions on Identifiers

- **More privacy concerns and more restrictions on Hardware Identifiers**
  - Before Android 6: Unrestricted
    - Any app can access, IMEI, MAC Address, etc.
    - Remains unchanged even after a factory reset.
  - Android 6+ (2015): Require Runtime User Consent
  - Android 10+ (2019): Completely Inaccessible to Third-party Apps

    Introduce Mac Address Randomization

Restrictions on Identifiers

- **More privacy concerns and more restrictions on user tracking**
  - Currently, the only available device identifier is **Google Advertising ID**.
    - User friendly: a unique, user-resettable, and user-deletable ID.
    - Unstable: User can easily RESET or OPT OUT.



  - **In this new era, any circumventions?**

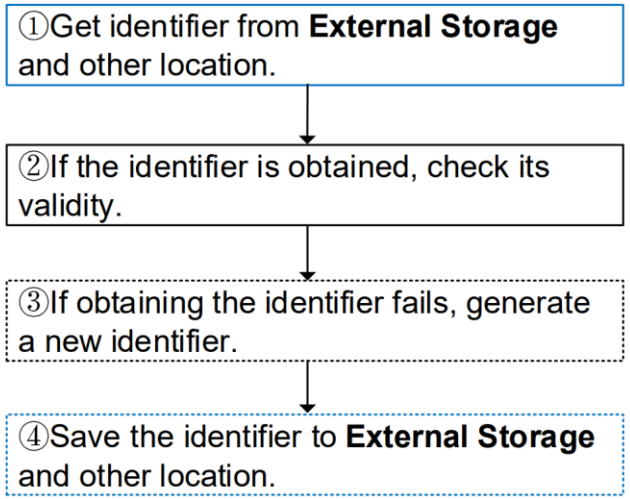
Motivating Example

- **Device identifier in external storage**
  - Unusual files found on external storage, accessible by all apps.
  - Turned out to be identifier files generated by SDK A belonging to a Tech Giant.

Code Example

```
String getUtdid(){
    String identifier;
    identifier = getIdentifierFromSystemSettings();
    if (isIdInvalid(identifier))
        identifier = getIdentifierFromSharedPreference();
        if (isIdInvalid(identifier))
            identifier = getIdentifierFromExternalStorage();
            if (isIdInvalid(identifier))
                identifier = generateUtdid();
    saveToOtherLocation(identifier);
    return identifier;
}
```

General Procedures




```
/storage/emulated/0/.UTSystemConfig/Global/Alvin2.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<long name="S" value="2466452439" />
<string name="UTDID">Y3g7KhjlJioDAEeXJjY4GYnd</string>
<string name="UTDID2">Y2IsVjdxE78DAEeXJjaqxwvI</string>
<long name="t2" value="1667216384445" />
<long name="timestamp" value="1668823850809" />
</map>
```

The Content of identifier files generated by SDK A

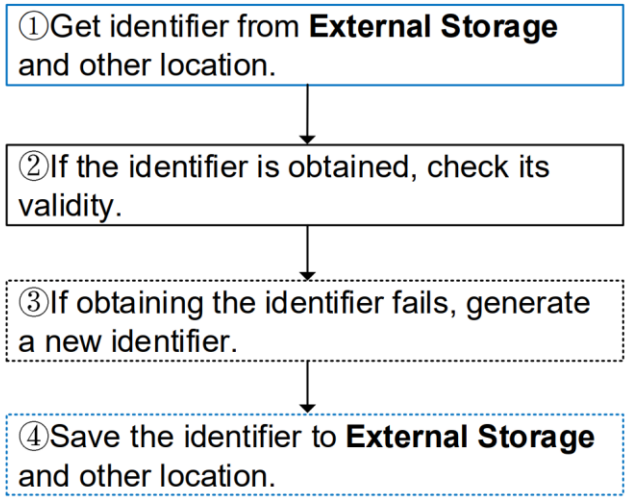
Motivating Example

- **Device identifier in external storage**
  - Unusual files found on external storage, accessible by all apps.
  - Turned out to be identifier files generated by SDK A belonging to a Tech Giant.

Code Example

```
String getUtdid(){
    String identifier;
    identifier = getIdentifierFromSystemSettings();
    if (isIdInvalid(identifier))
        identifier = getIdentifierFromSharedPreference();
        if (isIdInvalid(identifier))
            identifier = getIdentifierFromExternalStorage();
            if (isIdInvalid(identifier))
                identifier = generateUtdid();
    saveToOtherLocation(identifier);
    return identifier;
}
```

General Procedures




```
/storage/emulated/0/.UTSystemConfig/Global/Alvin2.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<long name="S" value="2466452439" />
<string name="UTDID">Y3g7KhjlJioDAEeXJjY4GYnd</string>
<string name="UTDID2">Y2IsVjdxE78DAEeXJjaqxwvI</string>
<long name="t2" value="1667216384445" />
<long name="timestamp" value="1668823850809" />
</map>
```

The Content of identifier files generated by SDK A

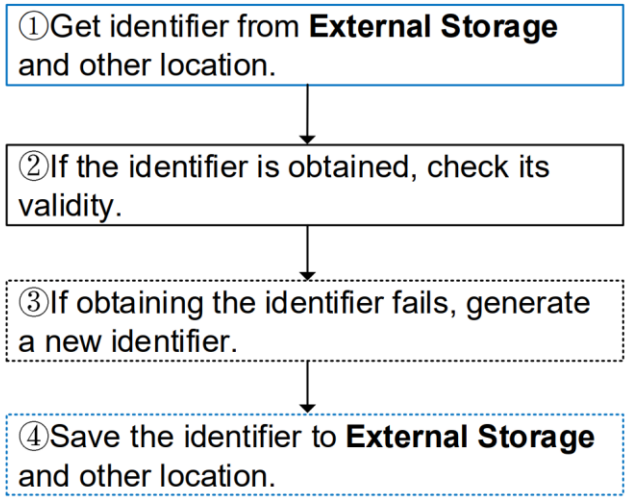
Motivating Example

- **Device identifier in external storage**
  - Unusual files found on external storage, accessible by all apps.
  - Turned out to be identifier files generated by SDK A belonging to a Tech Giant.

Code Example

```
String getUtdid(){
    String identifier;
    identifier = getIdentifierFromSystemSettings();
    if (isIdInvalid(identifier))
        identifier = getIdentifierFromSharedPreference();
        if (isIdInvalid(identifier))
            identifier = getIdentifierFromExternalStorage();
            if (isIdInvalid(identifier))
                identifier = generateUtdid();
    saveToOtherLocation(identifier);
    return identifier;
}
```

General Procedures




```
/storage/emulated/0/.UTSystemConfig/Global/Alvin2.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<long name="S" value="2466452439" />
<string name="UTDID">Y3g7KhjlJioDAEeXJjY4GYnd</string>
<string name="UTDID2">Y2IsVjdxE78DAEeXJjaqxwvI</string>
<long name="t2" value="1667216384445" />
<long name="timestamp" value="1668823850809" />
</map>
```

The Content of identifier files generated by SDK A

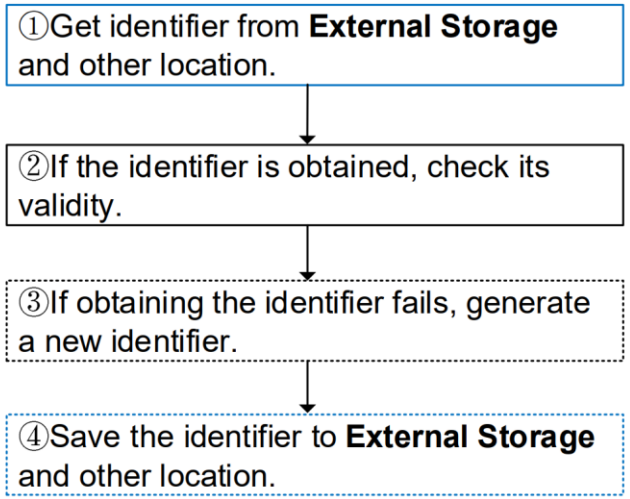
Motivating Example

- **Device identifier in external storage**
  - Unusual files found on external storage, accessible by all apps.
  - Turned out to be identifier files generated by SDK A belonging to a Tech Giant.

Code Example

```
String getUtdid(){
    String identifier;
    identifier = getIdentifierFromSystemSettings();
    if (isIdInvalid(identifier))
        identifier = getIdentifierFromSharedPreference();
        if (isIdInvalid(identifier))
            identifier = getIdentifierFromExternalStorage();
            if (isIdInvalid(identifier))
                identifier = generateUtdid();
    saveToOtherLocation(identifier);
    return identifier;
}
```

General Procedures




```
/storage/emulated/0/.UTSystemConfig/Global/Alvin2.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<long name="S" value="2466452439" />
<string name="UTDID">Y3g7KhjlJioDAEeXJjY4GYnd</string>
<string name="UTDID2">Y2IsVjdxE78DAEeXJjaqxwvI</string>
<long name="t2" value="1667216384445" />
<long name="timestamp" value="1668823850809" />
</map>
```

The Content of identifier files generated by SDK A

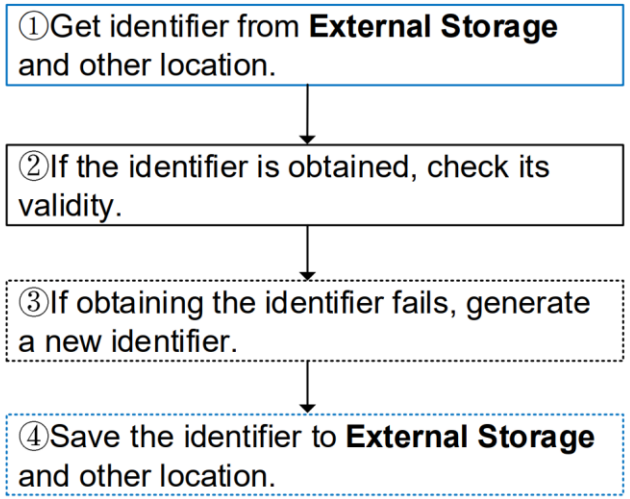
Motivating Example

- **Device identifier in external storage**
  - Unusual files found on external storage, accessible by all apps.
  - Turned out to be identifier files generated by SDK A belonging to a Tech Giant.
  - User tracking: cross-app user identification.

Code Example

```
String getUtdid(){
    String identifier;
    identifier = getIdentifierFromSystemSettings();
    if (isIdInvalid(identifier))
        identifier = getIdentifierFromSharedPreference();
        if (isIdInvalid(identifier))
            identifier = getIdentifierFromExternalStorage();
            if (isIdInvalid(identifier))
                identifier = generateUtdid();
    saveToOtherLocation(identifier);
    return identifier;
}
```

General Procedures




```
/storage/emulated/0/.UTSystemConfig/Global/Alvin2.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<long name="S" value="2466452439" />
<string name="UTDID">Y3g7KhjlJioDAEeXJjY4GYnd</string>
<string name="UTDID2">Y2IsVjdxE78DAEeXJjaqxwvI</string>
<long name="t2" value="1667216384445" />
<long name="timestamp" value="1668823850809" />
</map>
```

The Content of identifier files generated by SDK A

Potential Threat

- **Infringing upon user's privacy choice**
  - Breach regulation of app markets (e.g., Google Play and Xiaomi app store).
    - **Google Play's** requirements for the use of device identifiers.

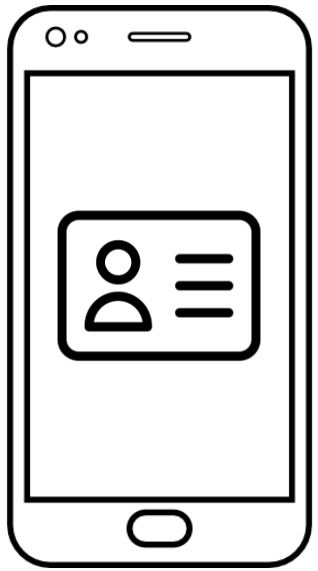
Policy requirements

The Google Play Developer Program Policy requires that all updates and new apps uploaded to Google Play use the advertising ID (when available on a device) in place of any other device identifiers for any advertising purposes. You're responsible for ensuring that your apps are in compliance with policies regarding its usage, as well as all Play policies.

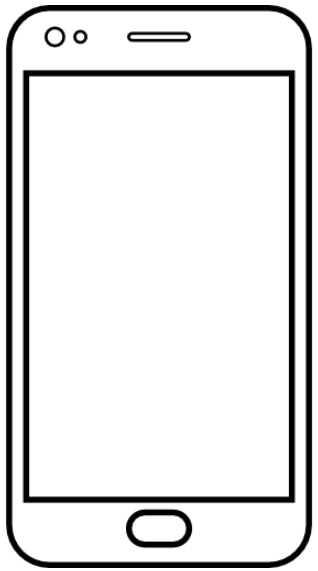
Potential Threat

- **Infringing upon user's privacy choice**
  - Breach regulation of app markets (e.g., Google Play and Xiaomi app store).

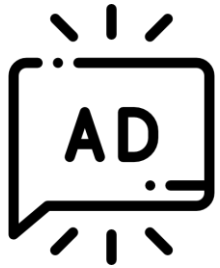
- **Open doors to potential attacks**
  - Any app can steal or modify the identifier with ease.



Victim's Smartphone



Attacker's Smartphone



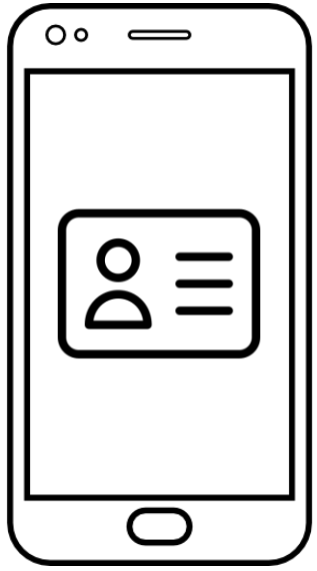
Advertisements
Personalized for Victims.

Potential Threat

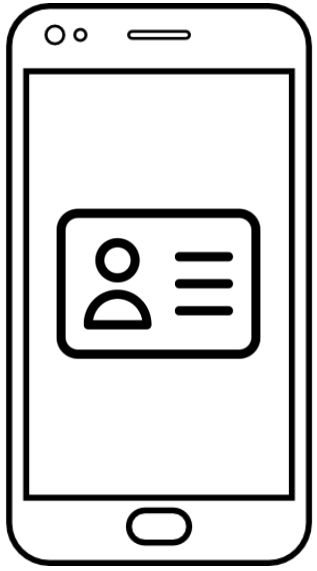
- **Infringing upon user's privacy choice**
  - Breach regulation of app markets (e.g., Google Play and Xiaomi app store).

- **Open doors to potential attacks**
  - Any app can steal or modify the identifier with ease.

Advertisements Customized by Attackers.



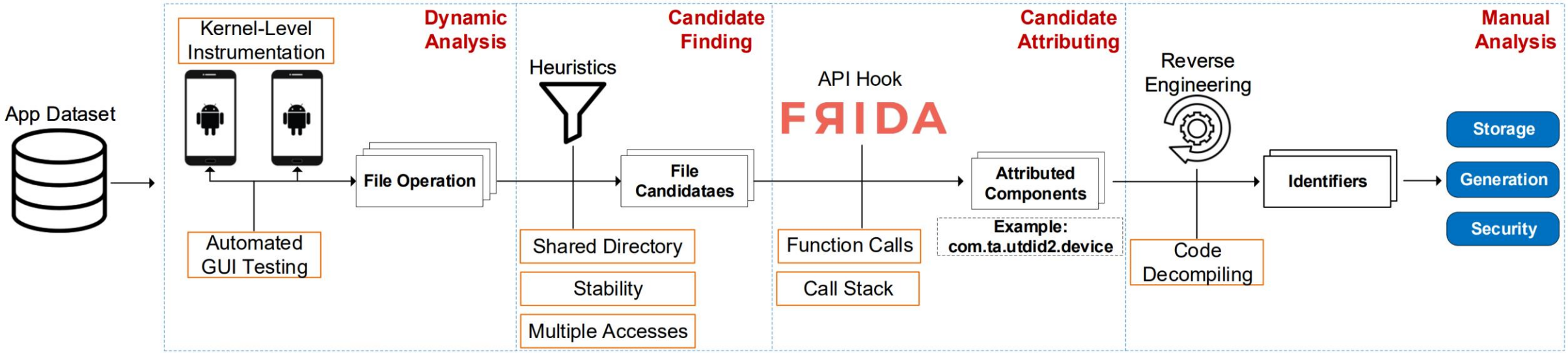
Victim's Smartphone



Attacker's Smartphone

Approach

- **A large-scale investigation of such identifiers**
  - To find SDKs with such tracking behaviors.
  - To understand how the identifiers are generated, stored, and (not) secured.



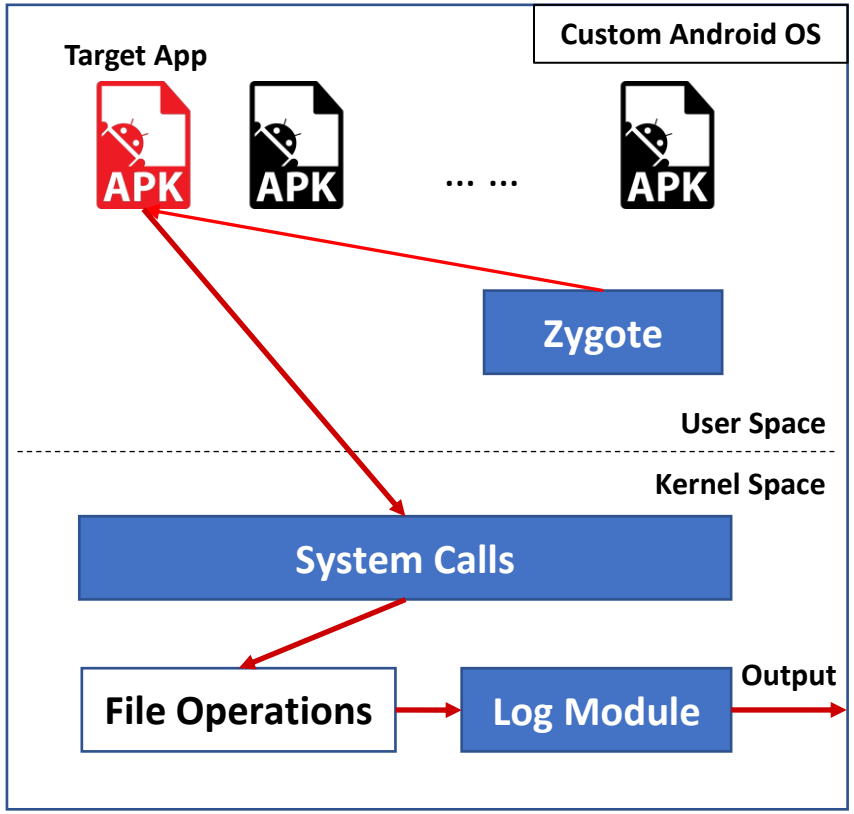
Approach – Dynamic Analysis

- **Testing Device**
  - Android Phones running our Customized Android OS.

1. Marking the target app during the Zygote initialization process.

2. Utilizing the kernel debug interface to record file operations, including file path, and operation type (deletion, read, etc.), conducted by the target app.

3. Using our developed logging module to output the records to a file.



Approach – Dynamic Analysis

- **Testing Device**
  - Android Phones running our Customized Android OS.
- **App Dataset**
  - 8,000 popular apps from Google Play and third-party markets.
- **Run these apps one by one automatically**
  - Each for 5 minutes
- **Collect our recorded file operations**

Approach – Candidate Finding

- **To find file candidates that could potentially contain such identifiers**

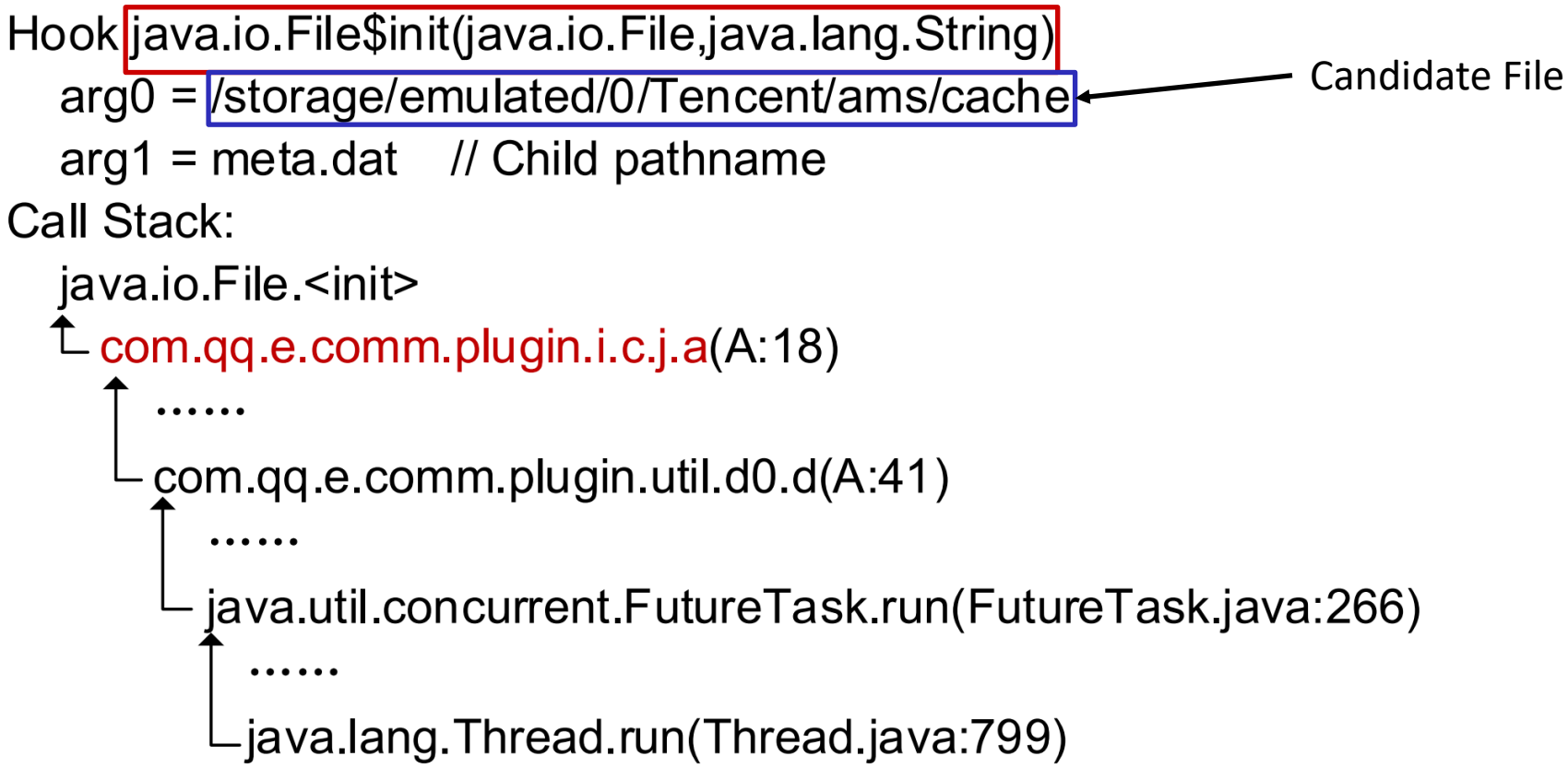
- **Heuristics**
  - File Location: External Storage, or otherwise
    - Other apps cannot access
    - Will be deleted upon app uninstallation
  - Relatively stable
    - No frequent deletion
  - Accessed by many apps
    - Sampling files, heuristically set 100 as threshold

- **30 File Candidates (differentiated by file path)**

Approach – Candidate Attributing

- **To find the app component that accessed the candidate file**

- **Rerun some apps, and hook file operation APIs to obtain Call Stack**



Approach – Candidate Attributing

- **To find the app component that accessed the candidate file**

- **Rerun some apps, and hook file operation APIs to obtain Call Stack**

```
Hook java.io.File$init(java.io.File,java.lang.String)
    arg0 = /storage/emulated/0/Tencent/ams/cache
    arg1 = meta.dat    // Child pathname
Call Stack:
    java.io.File.<init>
    └─com.qq.e.comm.plugin.i.c.j.a(A:18)
        ……
        └─com.qq.e.comm.plugin.util.d0.d(A:41)
            ……
            └─java.util.concurrent.FutureTask.run(FutureTask.java:266)
                ……
                └─java.lang.Thread.run(Thread.java:799)
```

Approach – Manual Analysis

- **Reverse engineering the culprit SDK**

- **Start with the captured file operation methods, look for how identifier is:**
  - Stored: How they hide themselves?
  - Generated: Unique? Stable?
  - Secured:
    - Confidentiality: Encrypted? How?
    - Integrity: Check for tampering? How?

Experimental Results

- **Collect in total 640K File Operations for the 8,000 apps**
- **30 File Candidates corresponding to 13,735 File Operations**
  - 24 Contain Third-party Identifier

Experimental Results

SDK Name	# Involved Apps	Generation Method	Stability	Uniqueness	Security	Storage Method
Alibaba ID	1,647	System-provided Identifier	○	●	○	/storage/emulated/0/.UTSystemConfig/Global/Alvin2.xml
					◎	/storage/emulated/0/.DataStorage/ContextData.xml
ByteDance AD	1,206	Random	○	●	○	/storage/emulated/0/Android/data/com.snssdk.api.embed/cache/clientudid.dat
Tencent AD	722	Random	○	●	◎	/storage/emulated/0/Tencent/ams/cache/meta.dat
						/storage/emulated/0/Android/data/com.tencent.ams/cache/meta.dat
Baidu Mobstat	542	System-provided Identifier	◎	●	◎	/storage/emulated/0/backups/.SystemConfig/.cuid2
Baidu Map						/storage/emulated/0/backups/.SystemConfig/.cuid
Amap	294	Remote Server	○	/	◎	/storage/emulated/0/backups/.adiu
Mob Share	171	System-provided Identifier	◎	●	○	/storage/emulated/0/Mob/comm/dbs/.duid
Mob SMS					◎	/storage/emulated/0/Android/data/.mn_1006862472
DCloud	173	Random	○	●	○	/storage/emulated/0/.imei.txt
						/storage/emulated/0/.DC4278477faeb9.txt
Umeng*	453	System-provided Identifier	○	●	●	/sdcard/Android/obj/.um/sysid.dat
						/sdcard/Android/data/.um/sysid.dat
Alibaba Quick Login	330	Random	○	●	○	/storage/emulated/0/.pns/.uniqueId/<id>
Kuaishou	265	Remote Server	○	/	◎	/storage/emulated/0/.oukdtft
Getui Push	212	Remote Server	○	/	○	/storage/emulated/0/libs/com.igexin.sdk.deviceId.db
Jiguang	175	Random	○	●	○	/storage/emulated/0/data/.push_deviceid
iFLYTEK	160	System-provided Identifier	○	◎	○	/storage/emulated/0/msc/.2F6E2C5B63F0F83B
Linkedme AD	29	Remote Server	◎	●	○	/storage/emulated/0/.lm_device/.lm_device_id
						/storage/emulated/0/LMDevice/lm_device_id
Shuzilm ID	56	Random	○	●	◎	The earliest created screenshot or photo file

17 SDKs

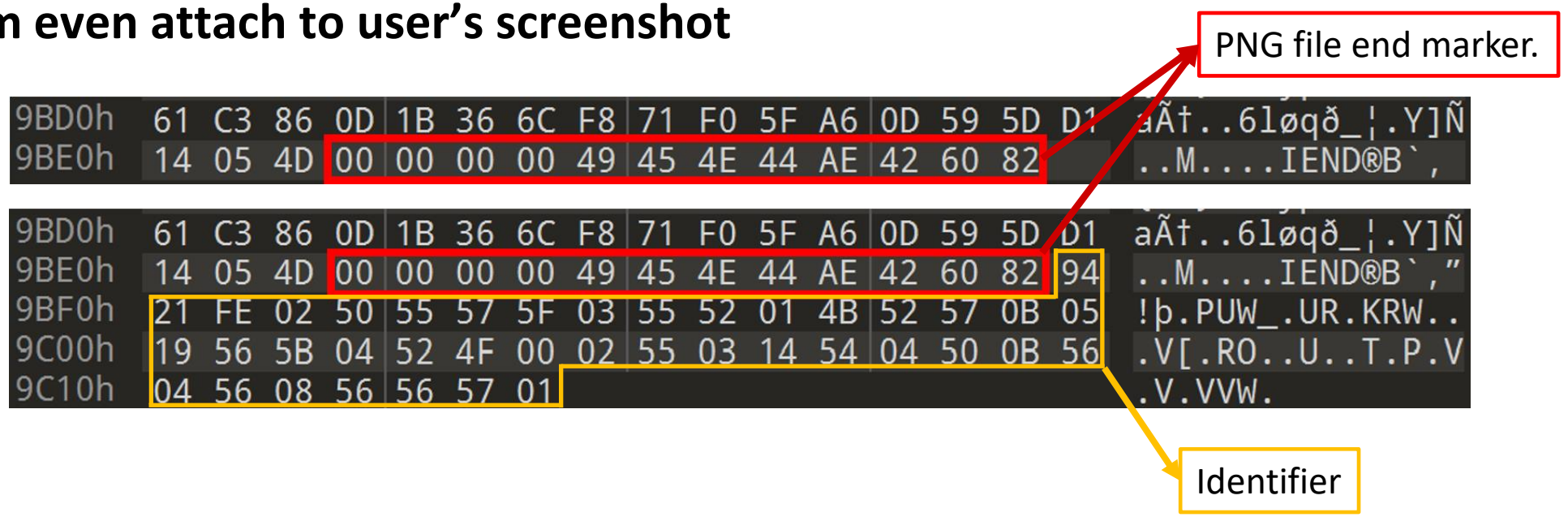
3,339 (40%) Apps involved, Cumulative Downloads nearly 100 billion

Storage

- **13/17 SDKs use hidden files to avoid being noticed**

- **Getui SDK hides identifier files among hundreds of .db files**
  - Under /storage/emulated/0/libs/com.igexin.sdk.deviceId.db

- **Shuzilm even attach to user's screenshot**



Generation and Security

- **Generation**
  - 6 SDKs just use random value
  - Others first try accessing Android's hardware identifier, if failed then random

```
String generateIdentifier(){
  String systemProvidedIdentifier = "";
  if (APILevel <= 23) {
    systemProvidedIdentifier = getIMEI();
    if (isEmpty(systemProvidedIdentifier)){
      systemProvidedIdentifier = getMacAddress();
      if (isEmpty(systemProvidedIdentifier)){
        systemProvidedIdentifier = getAndroidId();
        if (isEmpty(systemProvidedIdentifier)){
          systemProvidedIdentifier =
              getSerialNumber();
        }
      }
    }
  } else if(APILevel >= 29) {
    systemProvidedIdentifier = getAdvertisingId();
    if (isEmpty(systemProvidedIdentifier)){
      systemProvidedIdentifier = getAndroidId();
      if (isEmpty(systemProvidedIdentifier)){
        systemProvidedIdentifier = getSerialNumber
            ();
        if (isEmpty(systemProvidedIdentifier)){
          systemProvidedIdentifier = getMacAddress
              ();
        }
      }
    }
  } else {
    ......
  }
  String identifier = MD5(systemProvidedIdentifier
      );
  return identifier;
}
```

Hardware identifier

Umeng SDK's identifier generation method

Generation and Security

- **Generation**
- **Confidentiality**
  - 8 with no encryption at all
  - Others use hard-coded keys in SDK, vulnerable to attackers

Generation and Security

- **Generation**
- **Confidentiality**
- **Integrity**
  - 7 with no validation at all
  - 6 only check format
  - Others check hash, also vulnerable to attackers

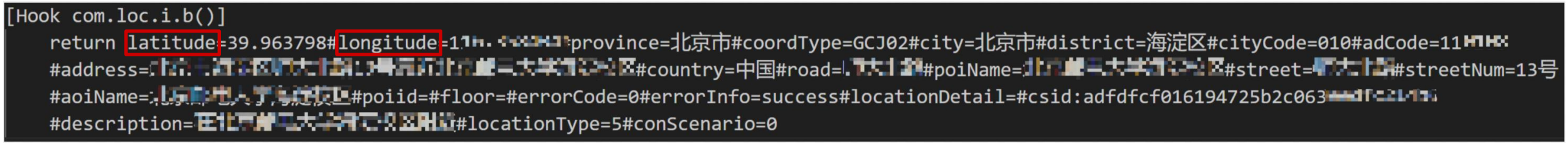
```
// Example of file content: {"v":1,"u":"d4c22e74-4762-4ad1-afce-3ee2b7525308","t":1667216542434,"m":"C9863EB3B042C74A1374EC82F231A869"}
boolean integrityCheck(String fileContent){
  JSONObject fileContentJson = new JSONObject(fileContent);
  String fileVersion = fileContentJson.getString("v");
  String identifier = fileContentJson.getString("u");
  String timestamp = fileContentJson.getString("t");
  String fileSignature = fileContentJson.getString("m");
  String fileSignatureCalculated = MD5(identifier + timestamp + "CB6;8PHJtpsNJwTW");
  if (fileSignature == fileSignatureCalculated) {
    return true;
  }
  return false;
}
```

Identifier

Check Hash

Other Interesting Findings

- **A location SDK shares location with external storage**
  - Apps with no location permission also access it.
  - Attacker can do the same.



Implication

- **Since Android 10, Scoped Storage is introduced**
  - By default, apps can only access files they have created themselves.
  - Access to files created by other apps is strictly controlled.

- **When Scoped Storage is effective, All 17 SDKs fail**

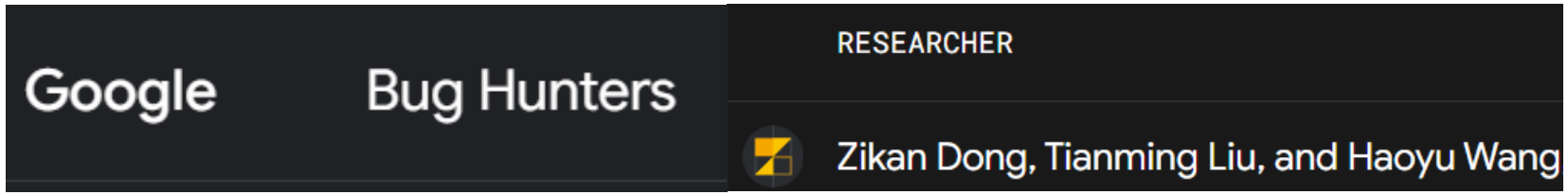
	# Apps Collected	# Violating Apps	% Violating Apps
Google Play	3,000	102	3.40%
Huawei	1,000	704	70.40%
Xiaomi	2,000	1,382	69.10%
Wandoujia	2,000	1,042	52.10%
Total	8,000	3,230	40.38%

Implication

- **However, developer can claim their app target Android 10 or lower**
  - Scoped Storage will not take effect
  - Google Play require targeting Android 13, other Markets like Samsung and HUAWEI not
  - Android Fragmentation

	# Apps Collected	# Violating Apps	% Violating Apps
Google Play	3,000	102	3.40%
Huawei	1,000	704	70.40%
Xiaomi	2,000	1,382	69.10%
Wandoujia	2,000	1,042	52.10%
Total	8,000	3,230	40.38%

- **Scoped Storage can also be exploited**
  - Reported to Google, received acknowledgement



Mitigation

- **Third-party markets expanding security requirements**
  - Android 8.0 in 2019 for Chinese App Markets as a Collective Effort
- **Permission separation on Android**
  - Google' Beta Program: Privacy Sandbox
- **Use our approach to find and clean these files**
  - Open-sourced
  - Google Play and other app markets

Take Away Message

Motivating Example

- **Device identifier in external storage**
  - Unusual files found on external storage, accessible by all apps.
  - Turned out to be identifier files generated by SDK A belonging to a Tech Giant.

Code Example

```
String getUtdid(){
    String identifier;
    identifier = getIdentifierFromSystemSettings();
    if (isIdInvalid(identifier))
        identifier = getIdentifierFromSharedPreference();
        if (isIdInvalid(identifier))
            identifier = getIdentifierFromExternalStorage();
            if (isIdInvalid(identifier))
                identifier = generateUtdid();
    saveToOtherLocation(identifier);
    return identifier;
}
```

General Procedures

①Get identifier from **External Storage** and other location.

②If the identifier is obtained, check its validity.

③If obtaining the identifier fails, generate a new identifier.

④Save the identifier to **External Storage** and other location.

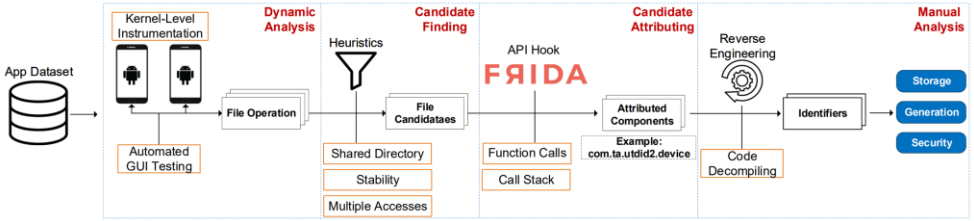
```
/storage/emulated/0/.UTSystemConfig/Global/Alvin2.xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<map>
<long name="S" value="2466452439" />
<string name="UTDID">Y3g7Khj1JioDAEeXJjY4GYnd</string>
<string name="UTDID2">Y2IsVjdxE7BDAEeXJjaqxwvI</string>
<long name="t2" value="1667216384445" />
<long name="timestamp" value="1668823850809" />
</map>
```

The Content of identifier files generated by SDK A

Highlight covert tracking practice

Approach

- **A large-scale investigation of such identifiers**
  - To find SDKs with such tracking behaviors.
  - To understand how the identifiers are generated, stored, and (not) secured.



Develop an analysis pipeline

Experimental Results

SDK Name	# Involved Apps	Generation Method	Stability	Uniqueness	Security	Storage Method
Alibaba ID	1,647	System-provided Identifier	○	●	○	/storage/emulated/0/.UTSystemConfig/Global/Alvin2.xml
					◎	/storage/emulated/0/.DataStorage/ContextData.xml
ByteDance AD	1,206	Random	○	●	○	/storage/emulated/0/Android/data/com.snssdk.api.embed/cache/clientudid.dat
Tencent AD	722	Random	○	●	◎	/storage/emulated/0/Tencent/ams/cache/meta.dat
						/storage/emulated/0/Android/data/com.tencent.ams/cache/meta.dat
Baidu Mobstat	542	System-provided Identifier	◎	●	◎	/storage/emulated/0/backups/.SystemConfig/.cuid2
Baidu Map						/storage/emulated/0/backups/.SystemConfig/.cuid
Amap	294	Remote Server	○	/	◎	/storage/emulated/0/backups/.adiu
Mob Share	171	System-provided Identifier	◎	●	○	/storage/emulated/0/Mob/comm/dbs/.duid
Mob SMS					◎	/storage/emulated/0/Android/data/.mn_1006862472
DCloud	173	Random	○	●	○	/storage/emulated/0/.imei.txt
						/storage/emulated/0/.DC4278477faeb9.txt
Umeng*	453	System-provided Identifier	○	●	●	/sdcard/Android/obj/.um/sysid.dat
						/sdcard/Android/data/.um/sysid.dat
Alibaba Quick Login	330	Random	○	●	○	/storage/emulated/0/.pns/.uniqueId/<id>
Kuaishou	265	Remote Server	○	/	◎	/storage/emulated/0/.oukdtft
Getui Push	212	Remote Server	○	/	○	/storage/emulated/0/libs/com.igexin.sdk.deviceId.db
Jiguang	175	Random	○	●	○	/storage/emulated/0/data/.push_deviceid
iFLYTEK	160	System-provided Identifier	○	◎	○	/storage/emulated/0/msc/.2F6E2C5B63F0F83B
Linkedme AD	29	Remote Server	◎	●	○	/storage/emulated/0/.lm_device/.lm_device_id
						/storage/emulated/0/LMDevice/lm_device_id
Shuzilm ID	56	Random	○	●	◎	The earliest created screenshot or photo file

17 SDKs

3,339 (40%) Apps involved, Cumulative Downloads nearly 100 billion

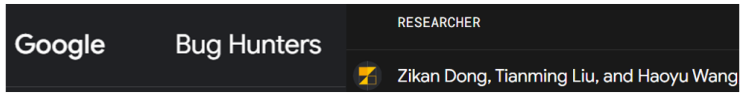
Identify and analyze 17 SDKs

Implication

- **However, developer can claim their app target Android 10 or lower**
  - Scoped Storage will not take effect
  - Google Play require targeting Android 13, other Markets like Samsung and HUAWEI not
  - Android Fragmentation

	# Apps Collected	# Violating Apps	% Violating Apps
Google Play	3,000	102	3.40%
Huawei	1,000	704	70.40%
Xiaomi	2,000	1,382	69.10%
Wandoujia	2,000	1,042	52.10%
Total	8,000	3,230	40.38%

- **Scoped Storage can also be exploited**
  - Reported to Google, received acknowledgement



Investigate their implication

Thanks!

Q&A

Contact: haoyuwang@hust.edu.cn