



Dangerous Hare: Hanging Attribute References Hazards Due to Vendor Customization

Speaker: Nan Zhang

System Security Lab, Indiana University

Who are we?



- System Security Lab, Indiana University
- Focus on novel problems in system security
- High-impact publications on IEEE S&P, ACM CCS, Usenix Security, NDSS



Android Customization

Google



Manufacturer



Carrier



Not just UI, Components too



“Among all the apps pre-installed by the major smartphone vendors (Samsung, HTC, LG, Sony) on their phones, only about 18% come from AOSP.”

Android Customization Process

- Unregulated, Decentralized
- Android Compatibility Program (ACP)
- Fails when proper precautions have not been taken

“Hare” Problem

- Hanging Attribute REferences
- An attribute is used while the party defining it has been removed
- A type of problem never investigated before

Attributes of Hares

- Package, Activity, Action names
- Content Provider
- Permission

An example

Phone Version



SMS/MMS
Content Provider



Instant Messaging
App



New
Message

VoIP Message
Content Provider



An example

Tablet Version



SMS/MMS
Content Provider

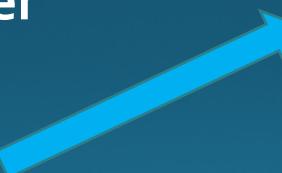


Instant Messaging
App



New
Message

VoIP Message
Content Provider



An example

Tablet Version

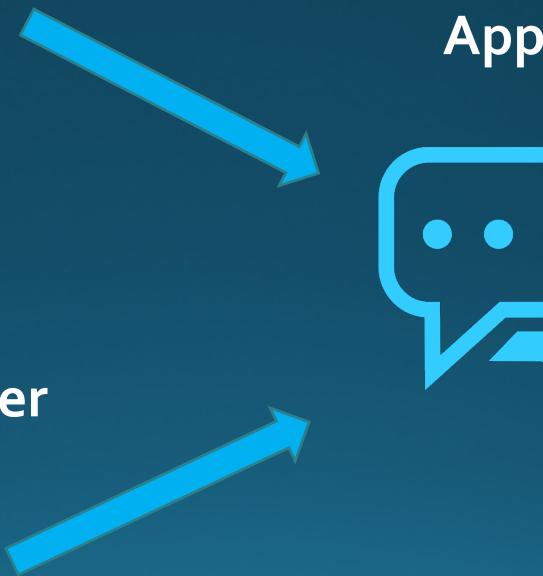
SMS/MMS
Content Provider



VoIP Message
Content Provider



Instant Messaging
App



Not All Hare are Exploitable

- Certain apps are aware of the fact that a referenced resource might not exist on the image.
 - Before invoking a certain resource within an app, some apps will first **check its signature**.
 - Some apps will **check the image properties** before invoking a non-existing app.
- Not all hares can be easily protected.

Signature Guard Example

```
1 public boolean extendAccessToken(Context context,
2         ServiceListener serviceListener) {
3     Intent intent = new Intent();
4     try{
5         PackageInfo pi =
6             context.getPackageManager().getPackageInfo(
7                 "com.facebook.katana",
8                 PackageManager.GET_SIGNATURES);
9         // Compare signature to the legitimate Facebook
10        // app Signature
11        if (!compareSignatures
12            (pi.signatures[0].toByteArray())){
13            return false;
14        } else{
15            intent.setClassName("com.facebook.katana",
16                "com.facebook.katana.platform.
17                TokenRefreshService");
18            return context.bindService(intent, new
19                TokenRefreshServiceConnection(context,
20                    serviceListener), 1); }
21        }catch(PackageManager.NameNotFoundException e){
22            return false;
23        }
24    }
```

Property Guard Example

```
1 private void ViewVideo(Uri uri){  
2     Intent intent = new  
3         Intent("android.intent.action.VIEW", uri);  
4     if (getPackageManager().hasSystemFeature  
5         ("com.google.android.tv")){  
6         intent.setPackage("com.google.android.youtube.googletv");  
7     } else{  
8         intent.setPackage("com.google.android.youtube");  
9     } startActivity(intent);  
10 }
```

Hares are

- Not random, isolated bugs.
- Caused by the **fundamental conflict** between the under-regulated Android Customization process and the **complicated relationship** among apps.
- Lack **verification** between two components.

Trust, but verify.



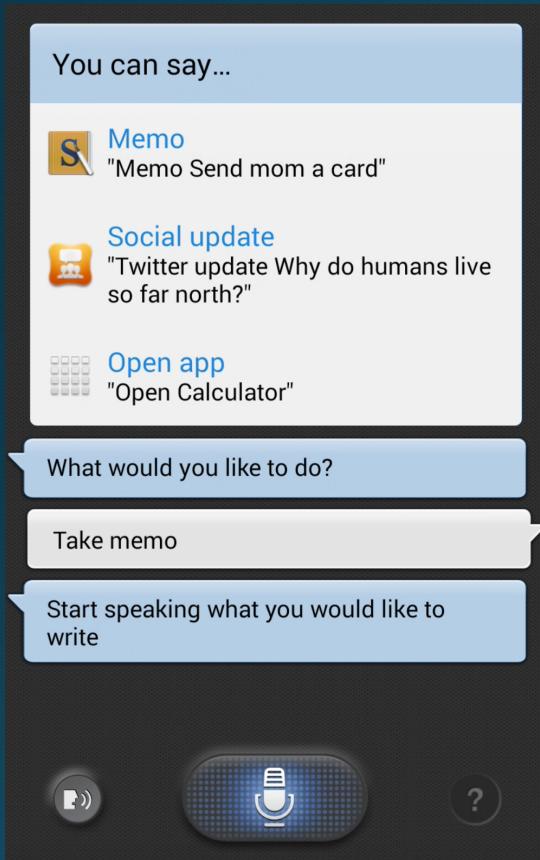
Exploiting Hares

➤ Missing Packages, Activities, Actions

Missing Content Providers

Missing Permissions

Stealing Voice Notes



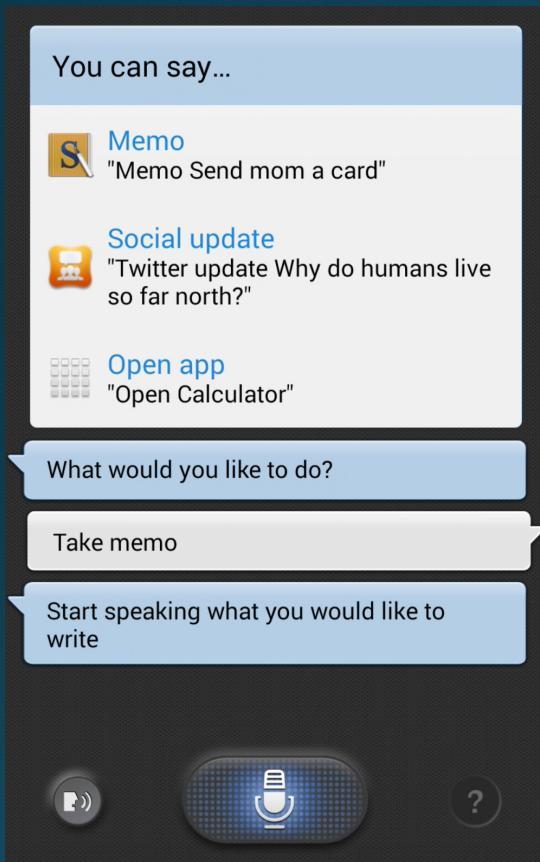
If **com.samsung.android.app.memo** exists



If **com.sec.android.app.smemo** exists



Stealing Voice Notes



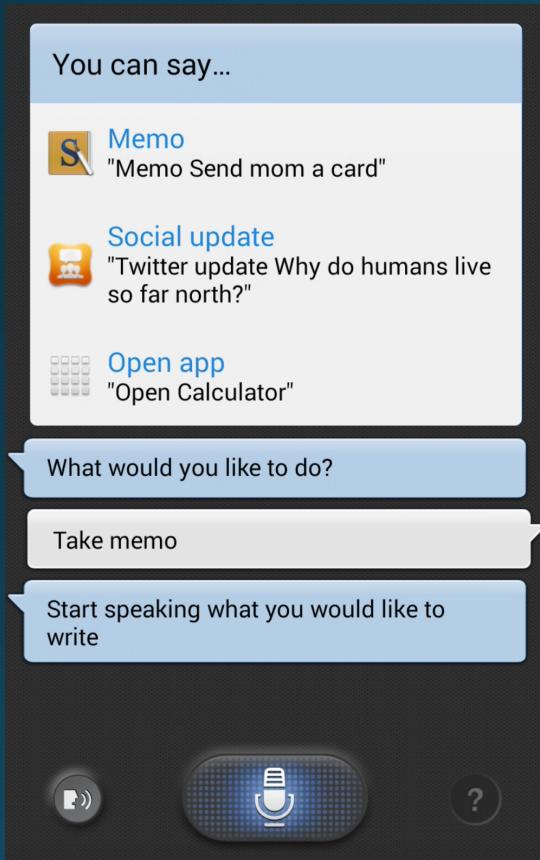
If **com.samsung.android.app.memo** exists



If **com.sec.android.app.smemo** exists



Stealing Voice Notes



If **com.samsung.android.app.memo** exists



com.samsung.android.app.memo

If **com.sec.android.app.smemo** exists



Demos



- Stealing voice note



- Faking Dropbox on LG



- Replacing official voice recorder



- Hulu on Watch

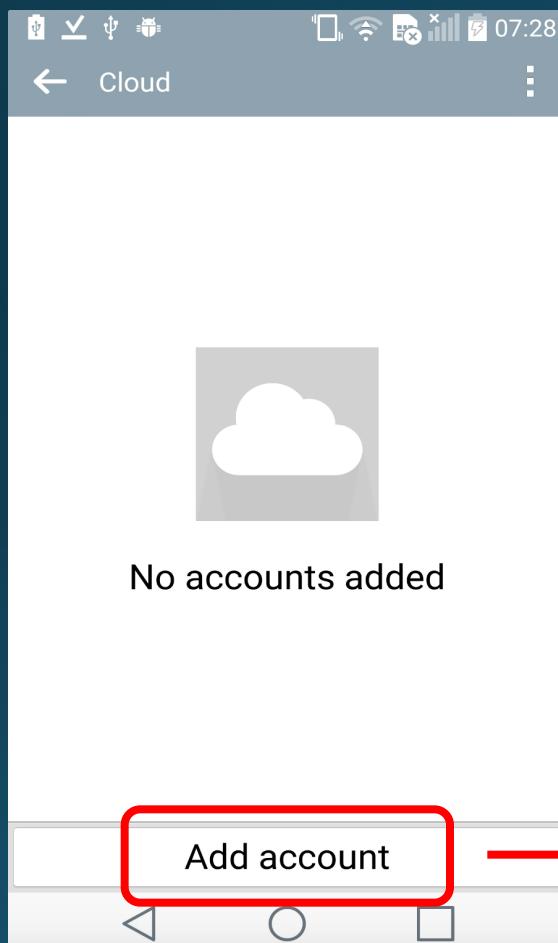
Exploiting Hares

Missing Packages, Activities, Actions

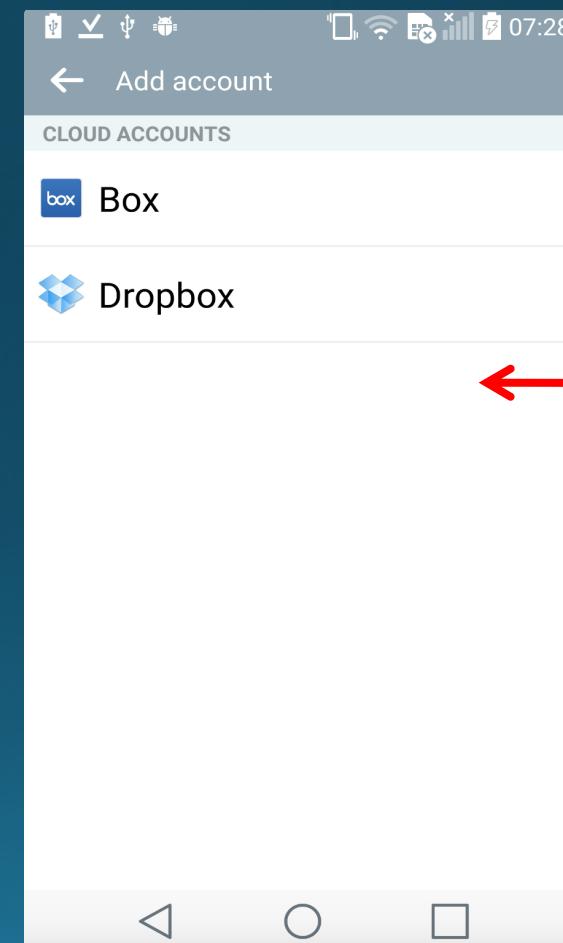
➤ **Missing Content Providers**

Missing Permissions

LG Cloud Scam

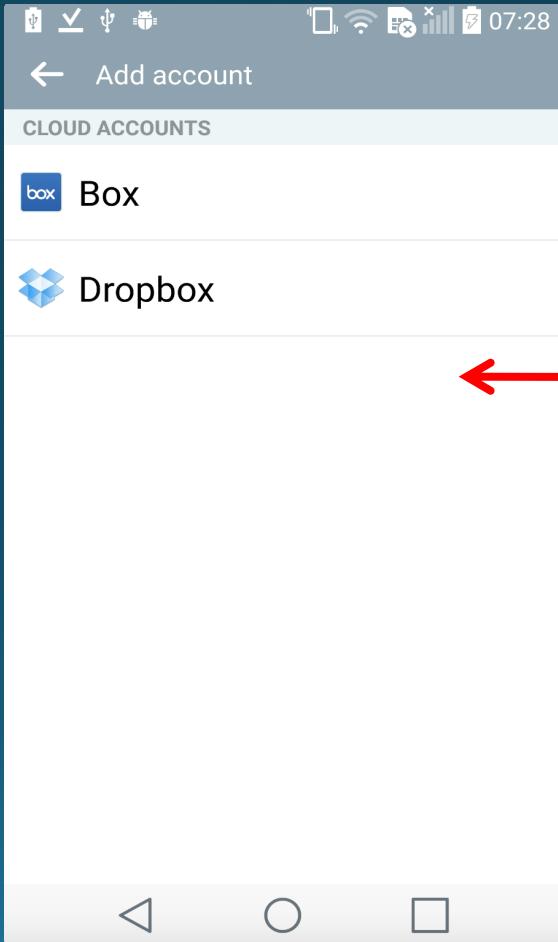


LG CloudHub App



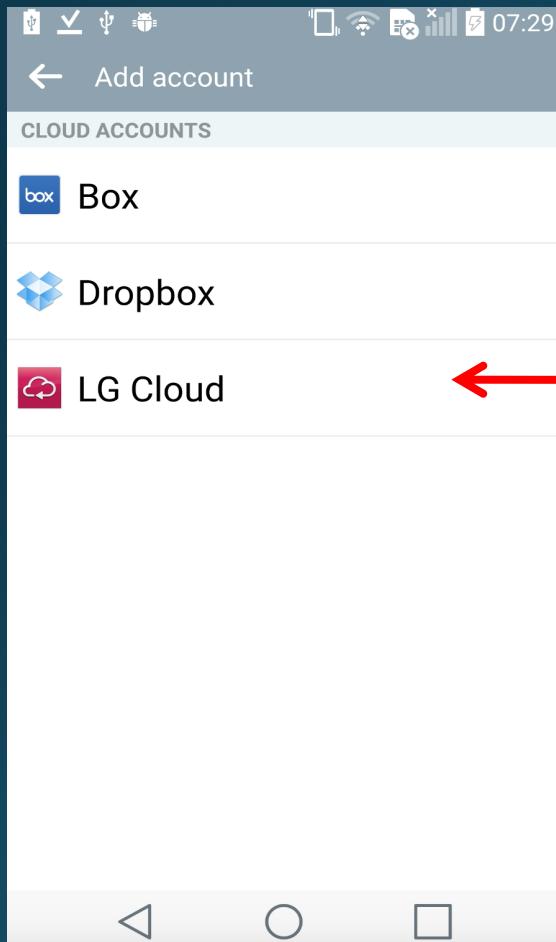
Query
“com.lge.lgaccount.provider”
for additional
Cloud Services
and add an item

LG Cloud Scam



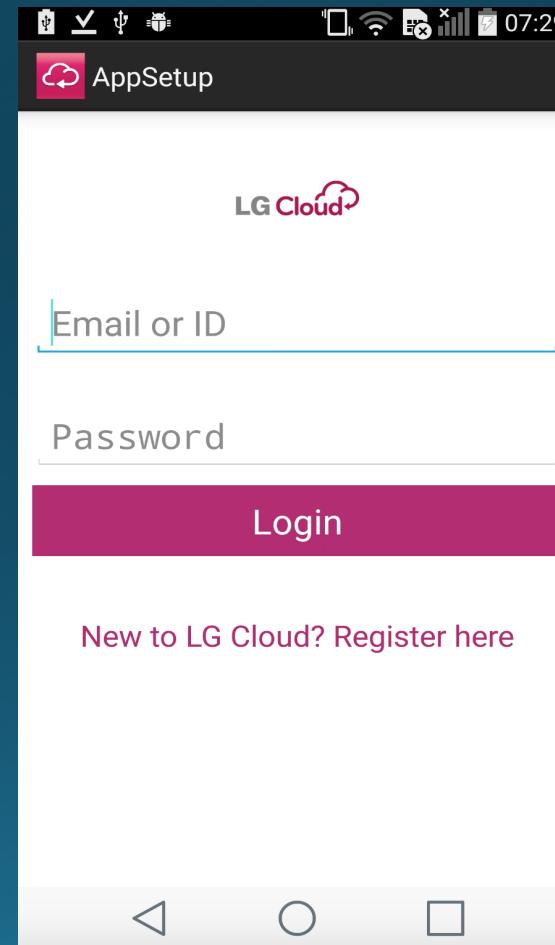
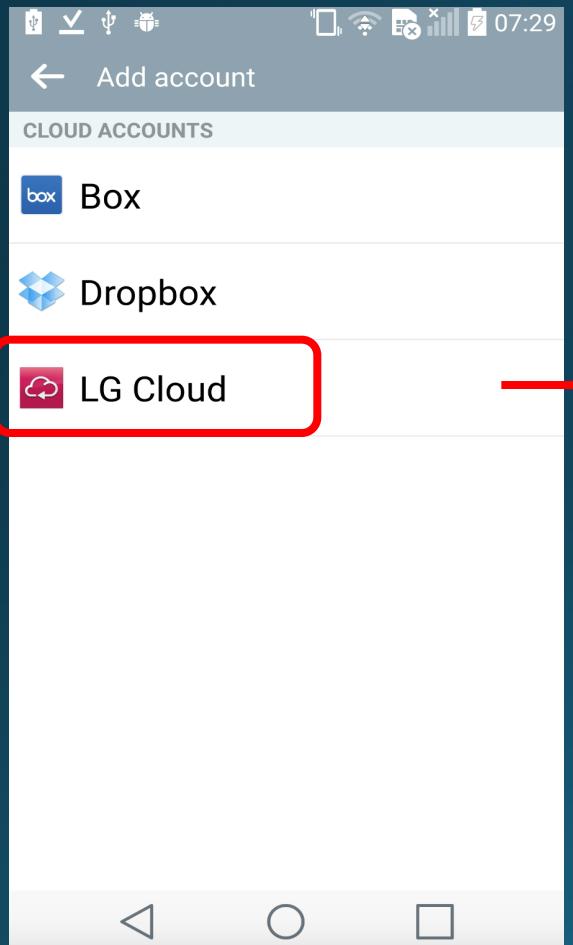
Query
“com.lge.lgaccount.provider”
for additional
Cloud Services
and add an item

LG Cloud Scam



Own
com.lge.lgaccount.provider

LG Cloud Scam

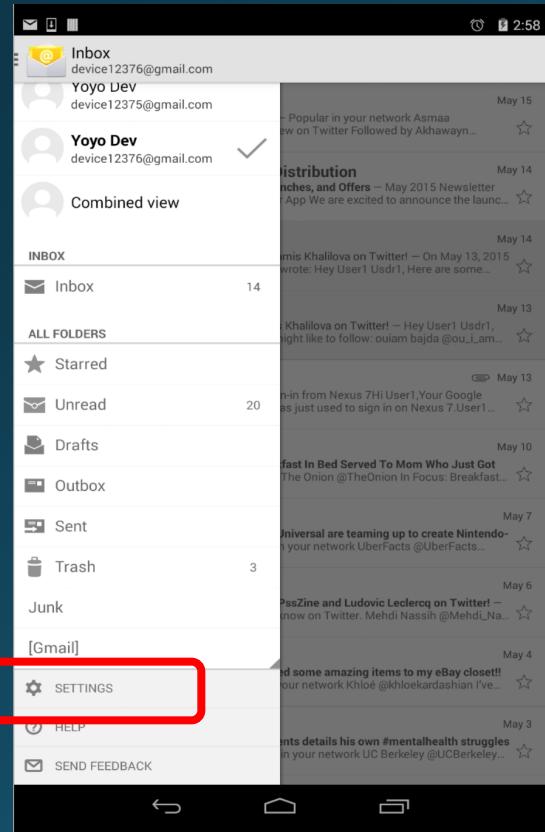


Handle
com.lge.ADD_ACCOUNT

Intent Hijacking

Google Email App

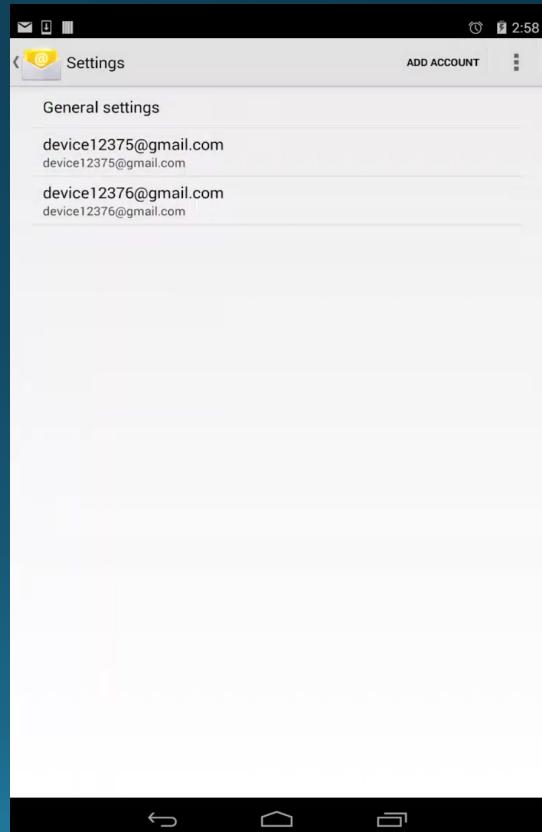
version 6.3-1218562



Action: `android.intent.action.EDIT`
Data: `content://ui.email.android.com/settings?account=id`

```
<!-- Account Settings Intent Filters-->
<activity
    android:name=".activity.setup.AccountSettings"
    android:exported="true">
<intent-filter>
    <action android:name="android.intent.action.EDIT"/>
    <category android:name=
        "android.intent.category.DEFAULT"/>
    <data android:scheme="content"
        android:host="ui.email.android.com"
        android:pathPrefix="/settings"/>
</intent-filter>
```

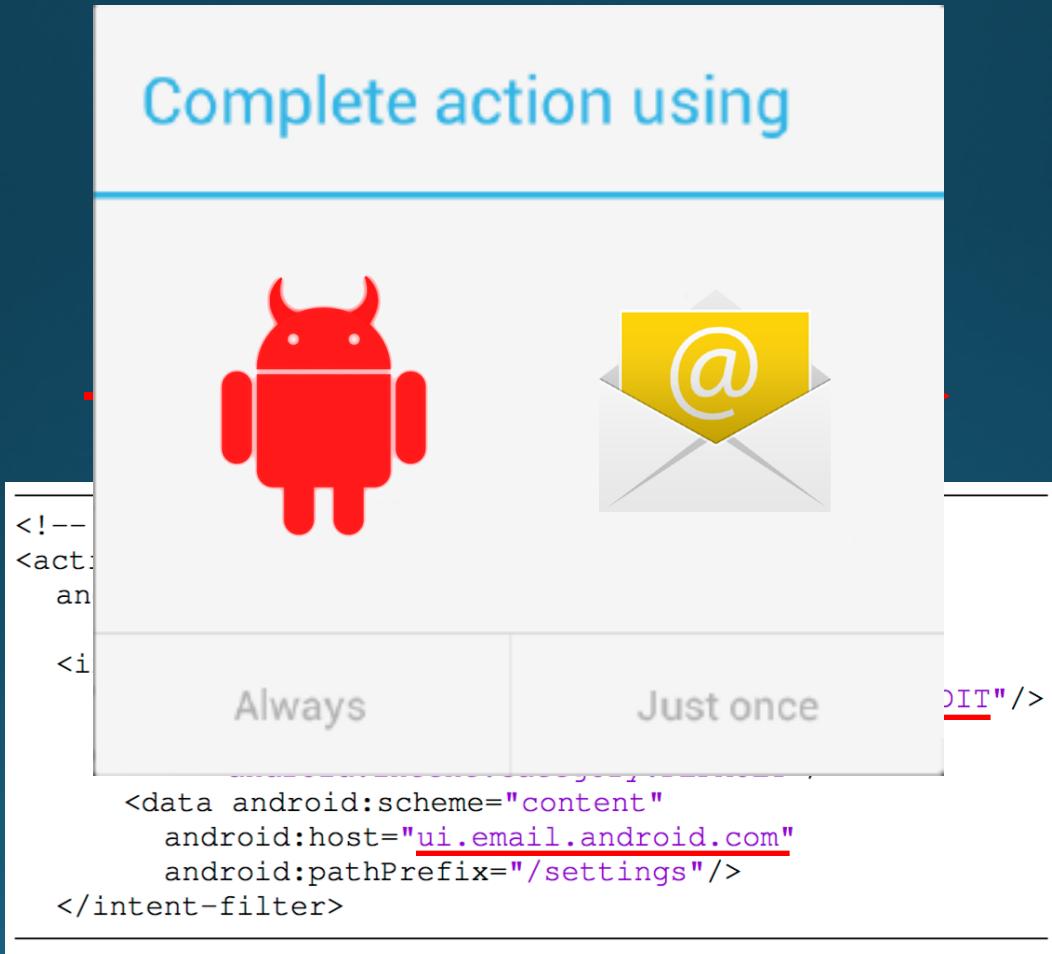
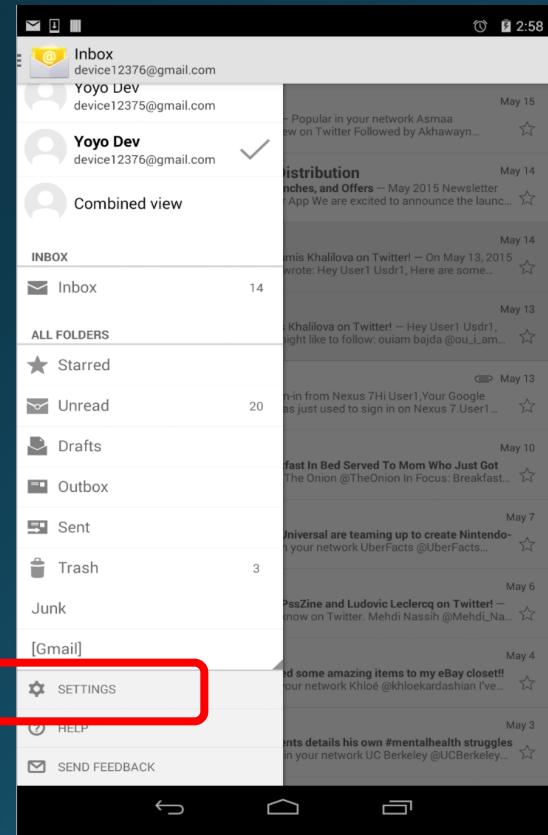
AccountSetting Activity Within Google Email App



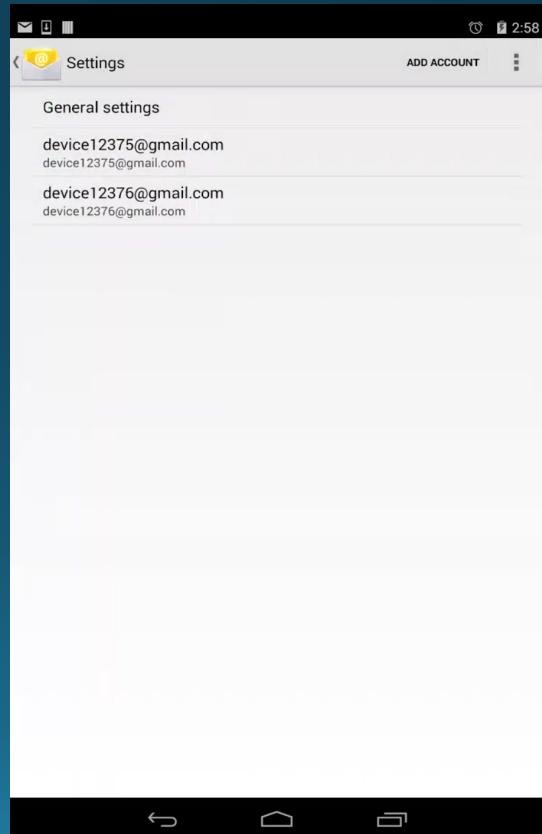
Intent Hijacking

Google Email App

version 6.3-1218562



AccountSetting Activity Within Google Email App

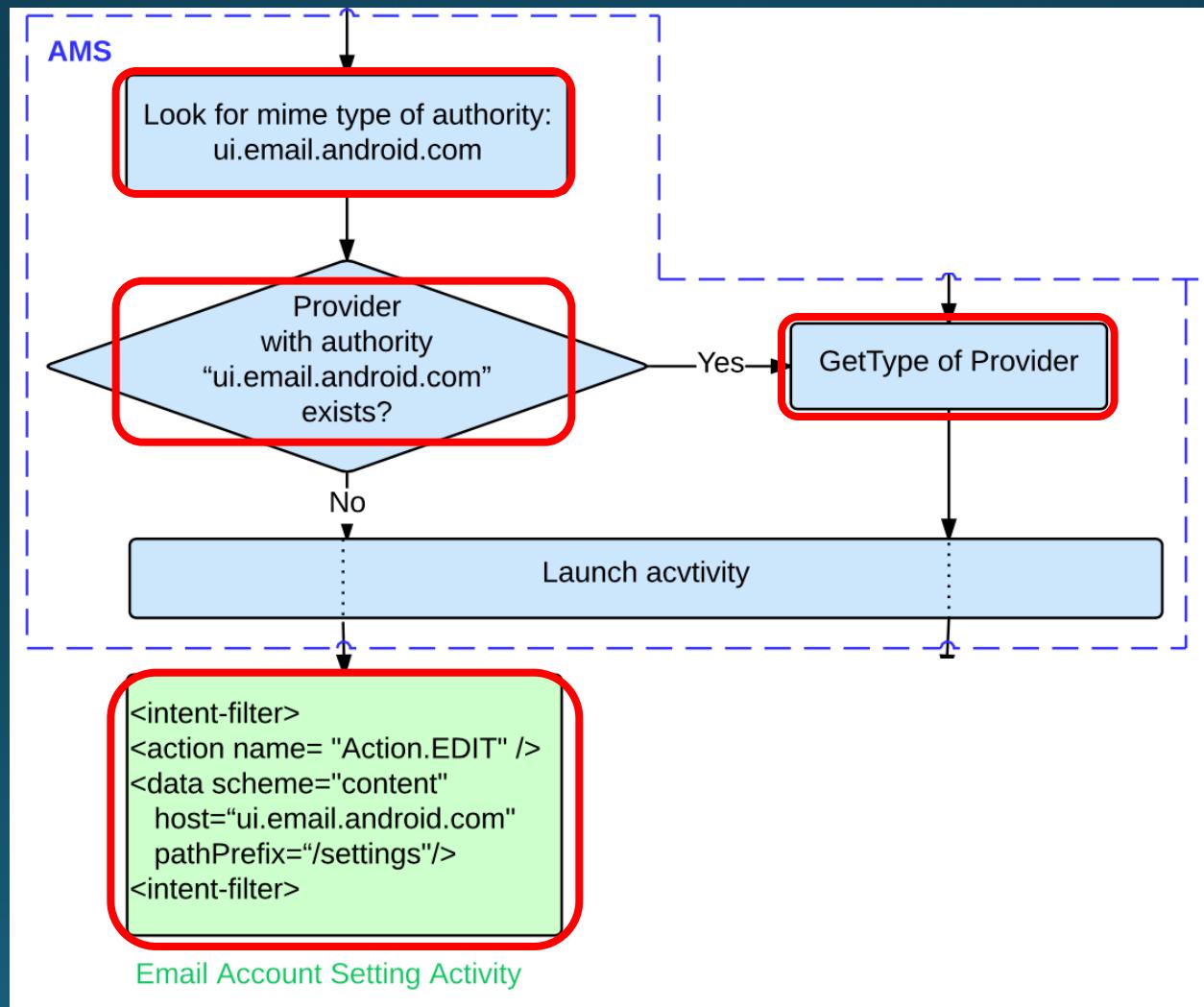


Intent Hijacking

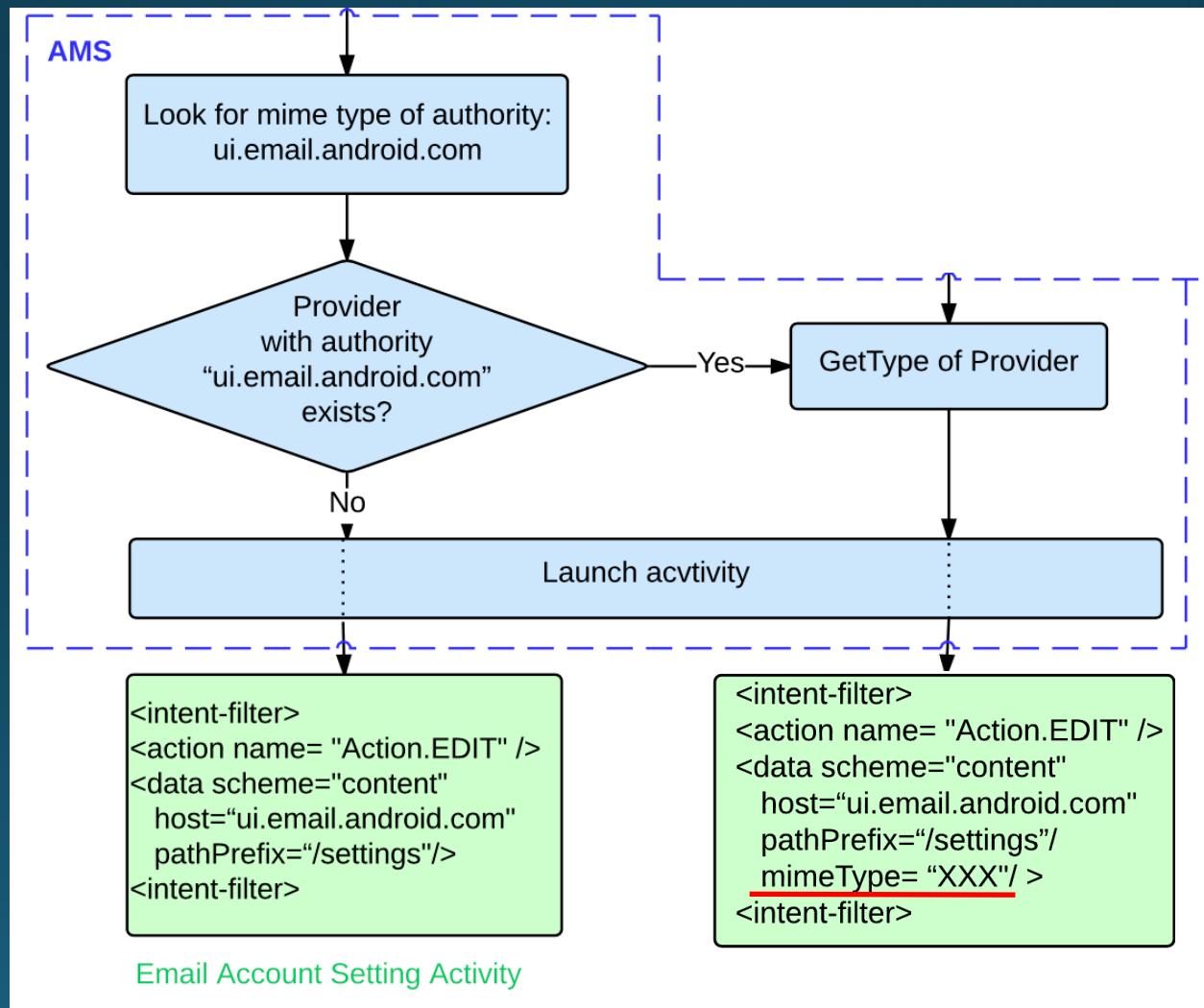
The authority of content provider “`ui.email.android.com`” does not exist.

How do we leverage this to be the **only** one to receive the intent?

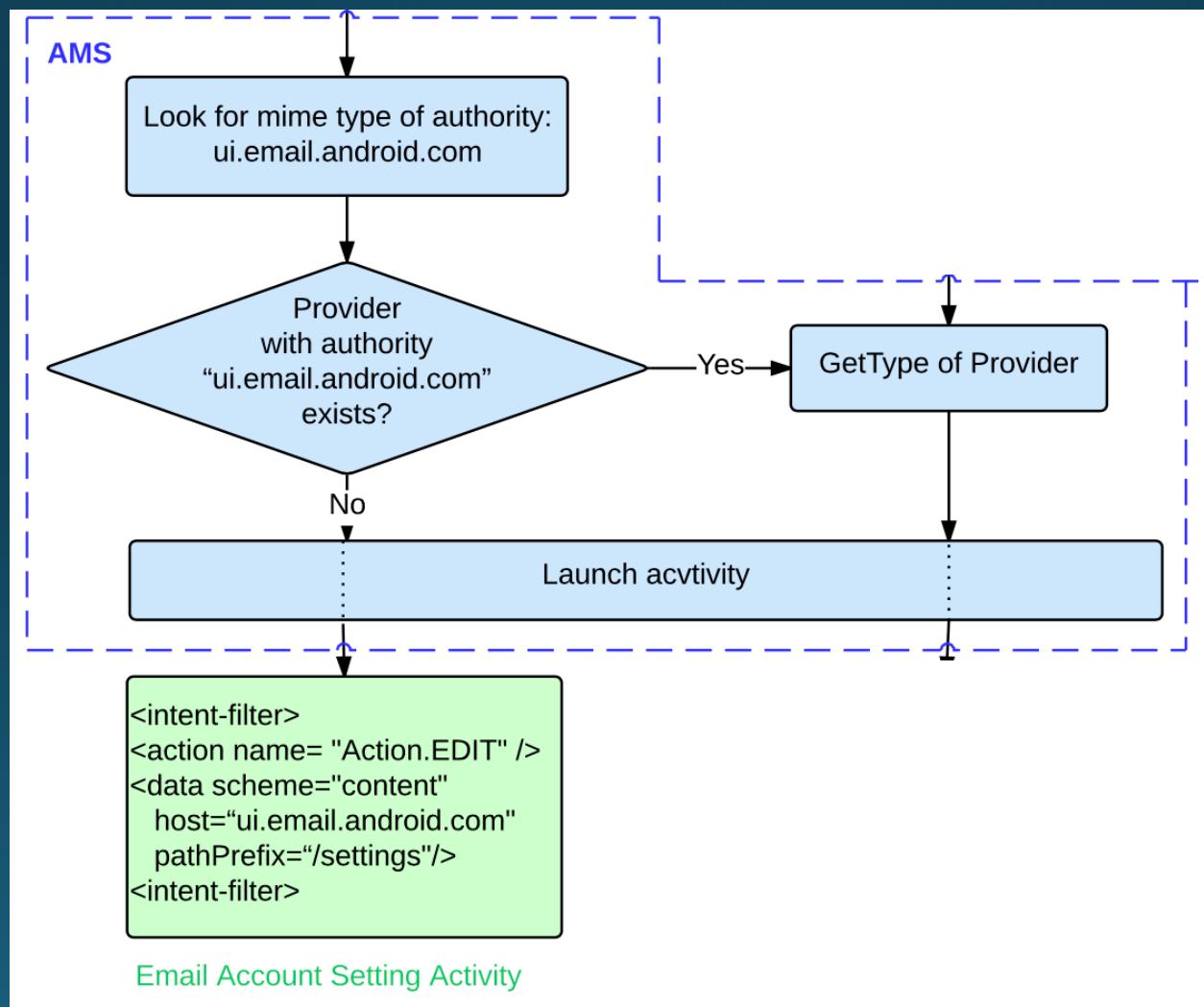
Intent Hijacking



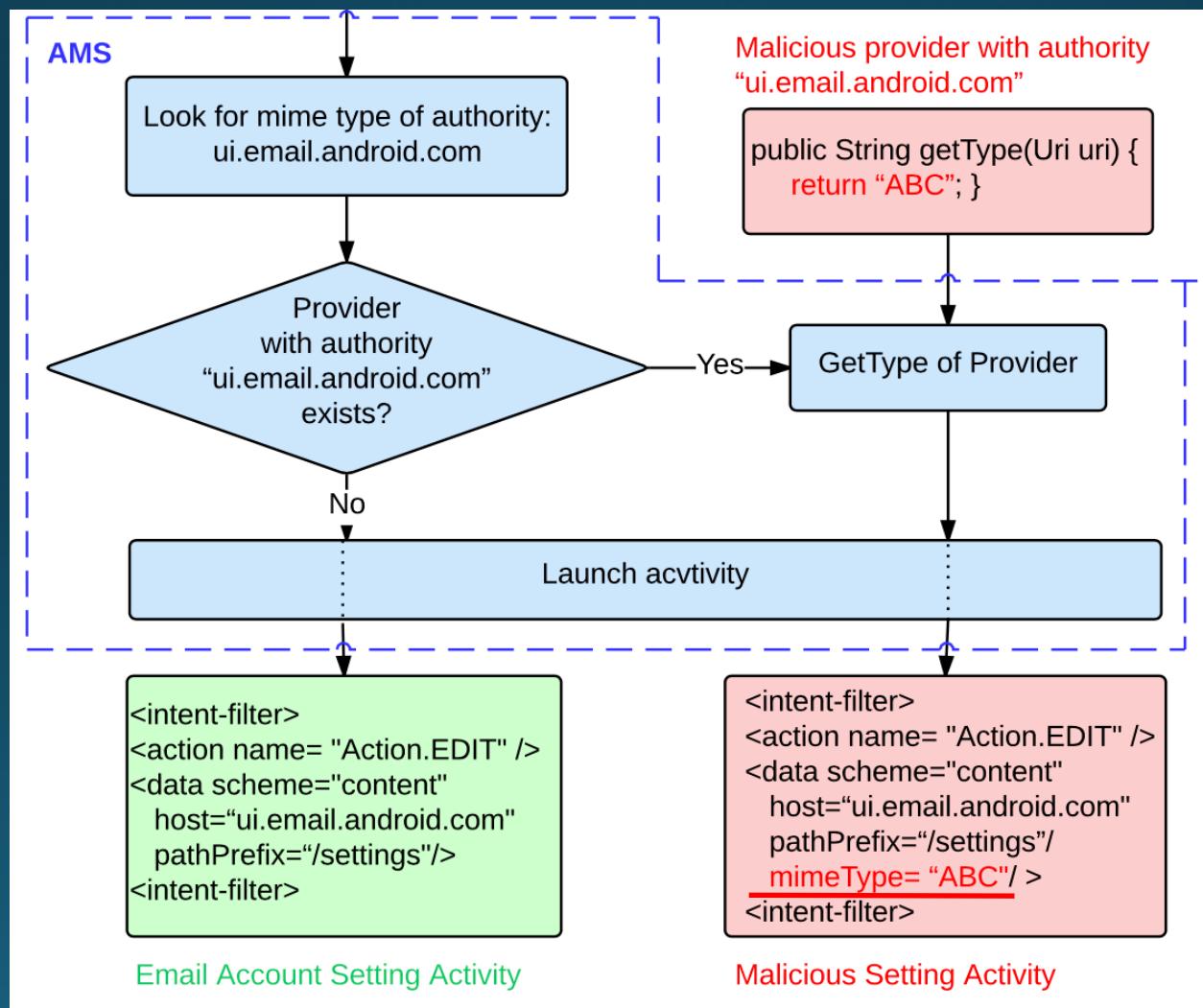
Intent Hijacking



Intent Hijacking



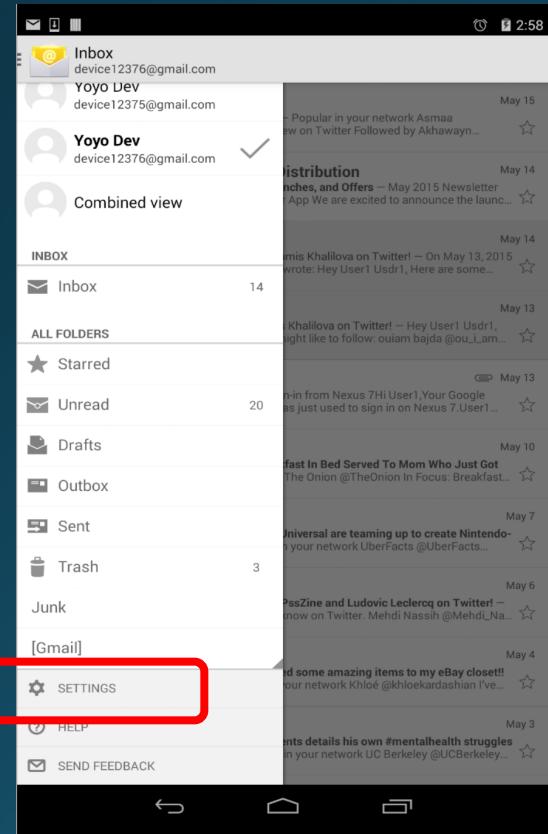
Intent Hijacking



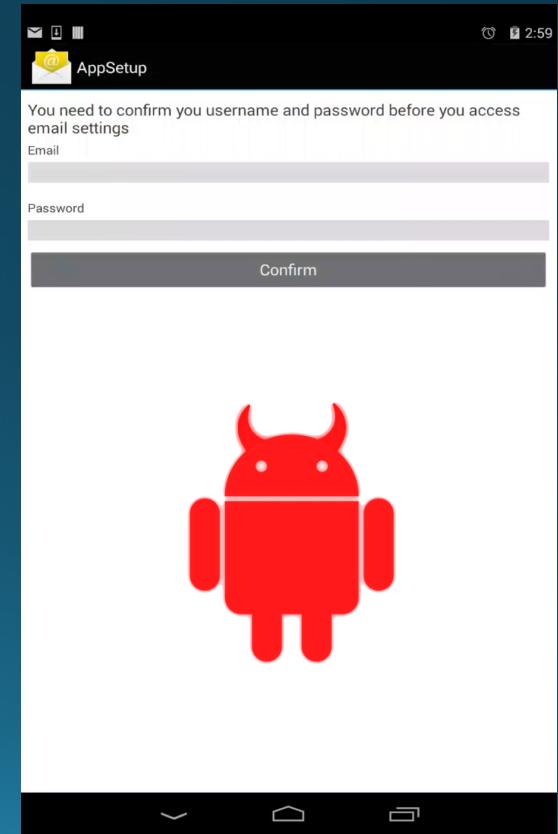
Intent Hijacking

Google Email App

version 6.3-1218562



Malicious App



Demos



- Google Email Attack



- LG CloudHub Attack



- Facebook Service Confusion



- Skype Service Confusion

All deomos can be found at <https://sites.google.com/site/androidharehunting/>
and <https://sites.google.com/site/perplexedmsg/>

Exploiting Hares

Missing Packages, Activities, Actions

Missing Content Providers

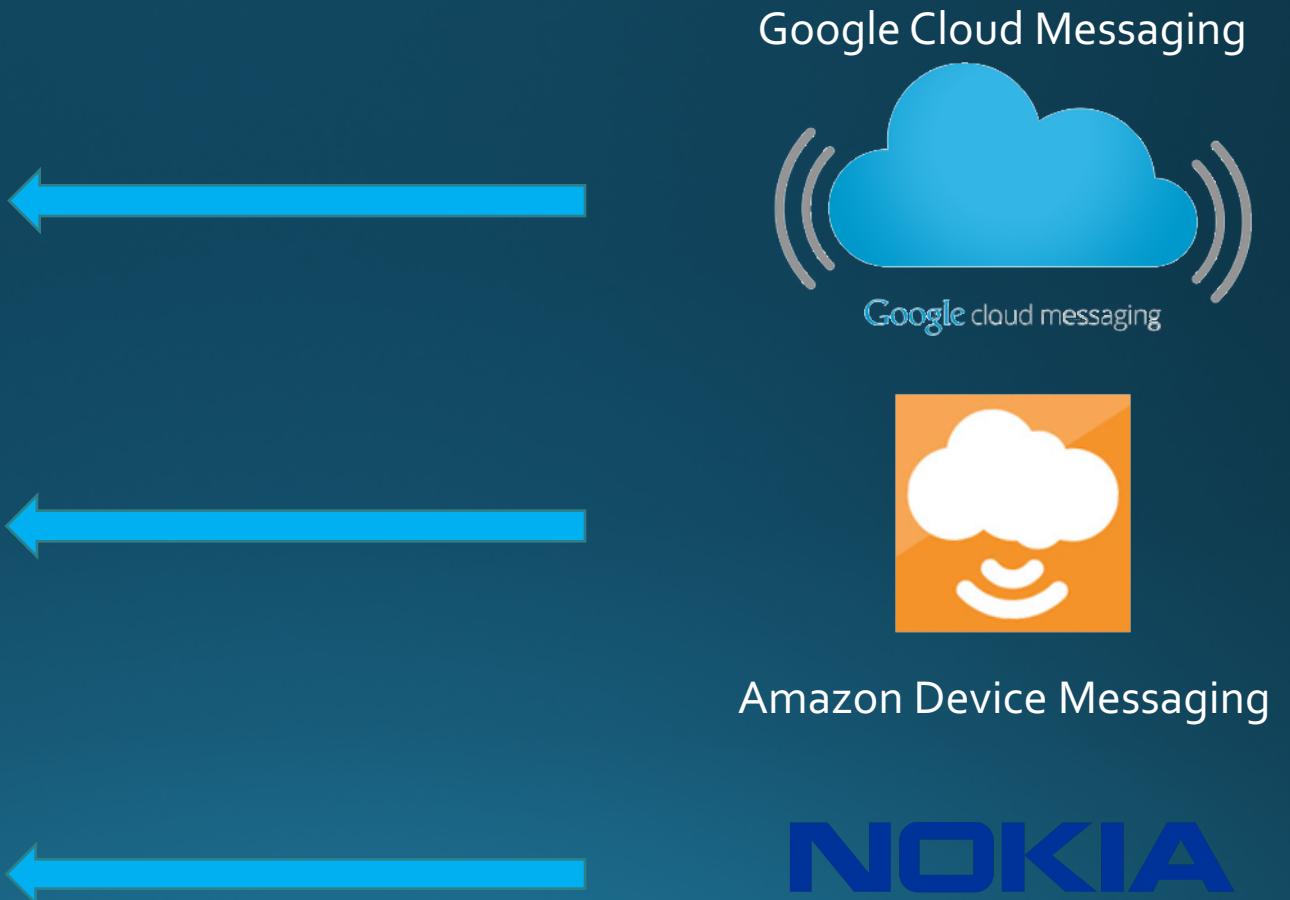
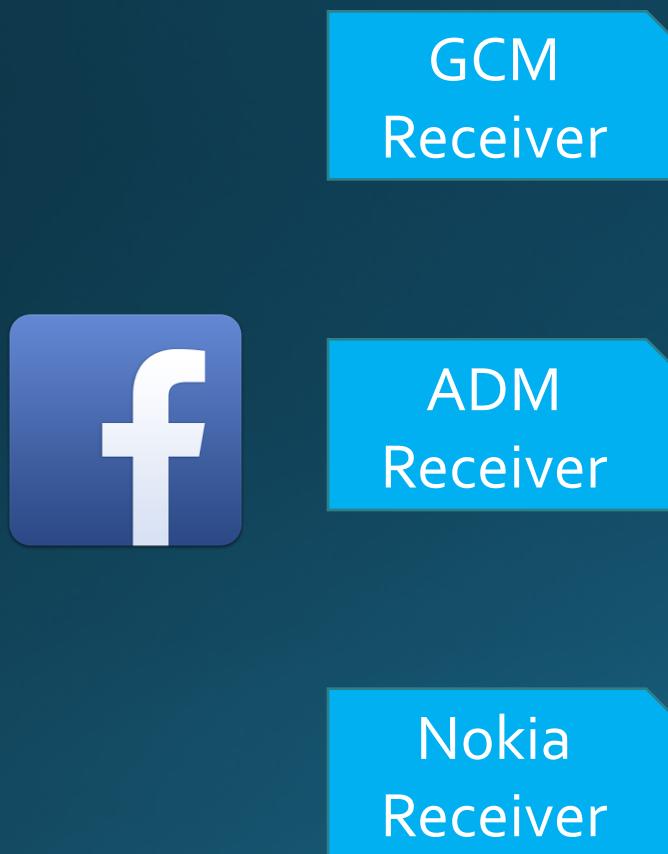
➤ **Missing Permissions**

Service Confusion

Apps use multiple push-messaging services, including

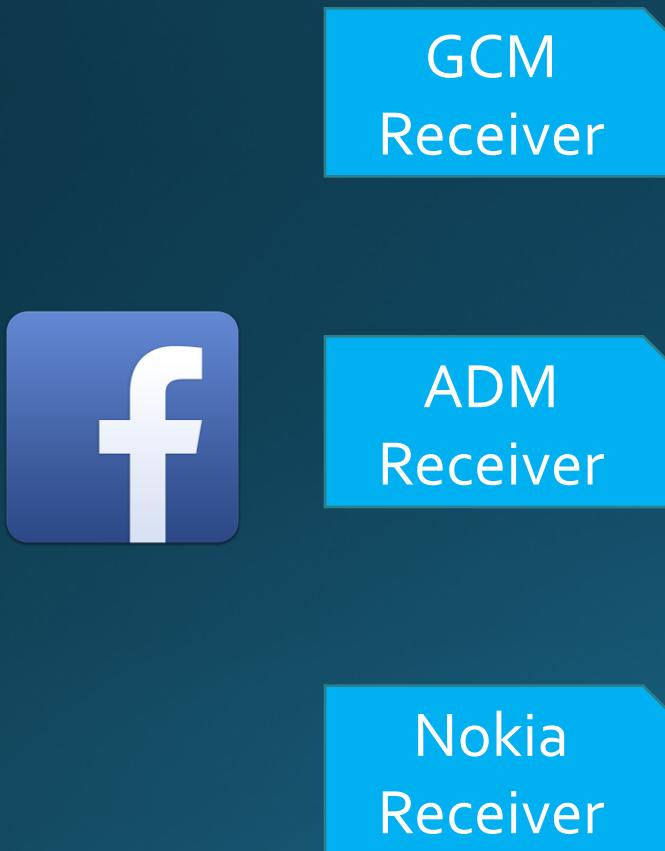
- Google Cloud Messageing (GCM)
`com.google.android.c2dm.permission.SEND`
- Amazon Device Messaging (ADM)
`com.amazon.device.messaging.permission.SEND`
- Nokia Notification
`com.nokia.pushnotifications.permission.SEND`

Service Confusion



Service Confusion

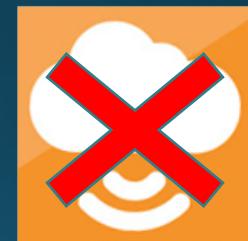
On Nexus



Google Cloud Messaging



Google cloud messaging



Amazon Device Messaging



Service Confusion

On Nexus

GCM
Receiver



Google Cloud Messaging



ADM
Receiver

Nokia
Receiver



Push Message



I have Nokia permission!

Demos



- Google Email Attack



- LG CloudHub Attack



- Facebook Service Confusion



- Skype Service Confusion

All deomos can be found at <https://sites.google.com/site/androidharehunting/>
and <https://sites.google.com/site/perplexedmsg/>

Messaging Stealing

On Kindle Fire



GCM
Receiver

ADM
Receiver

Nokia
Receiver

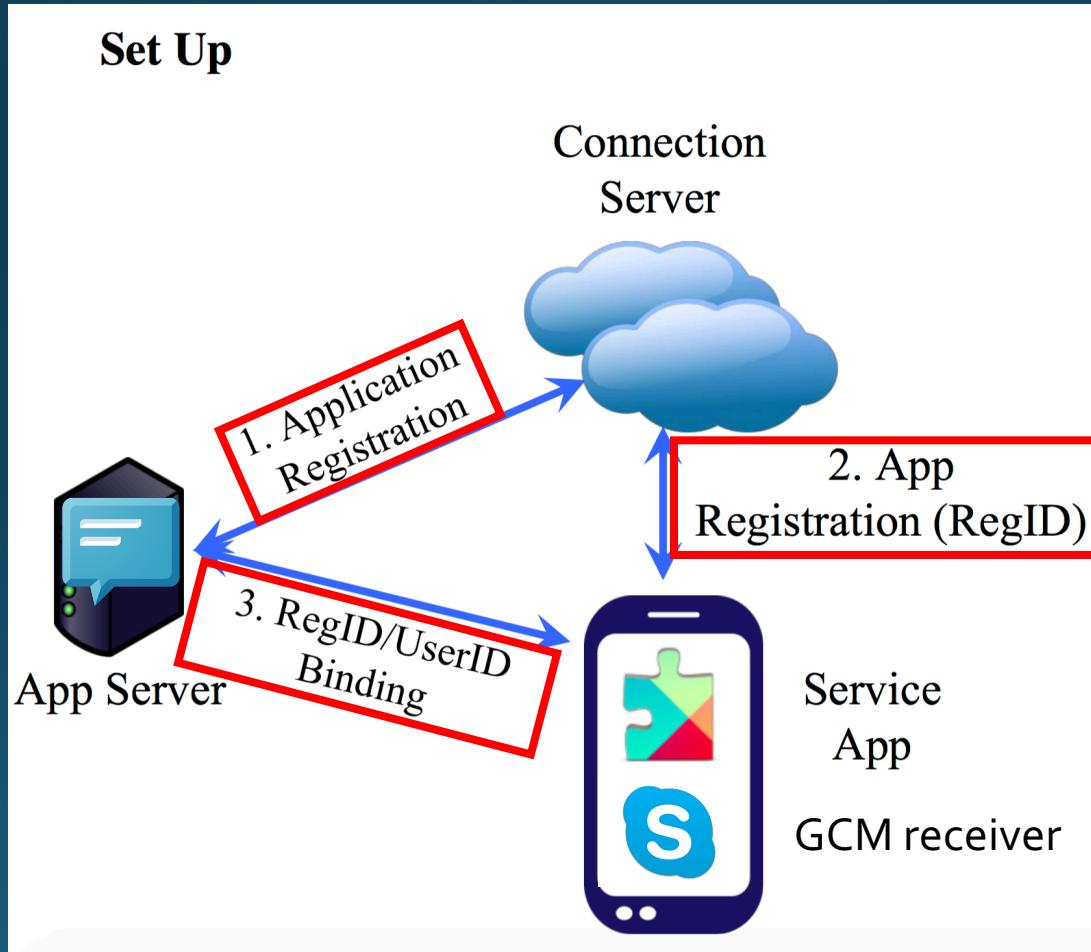


I have GCM permission!

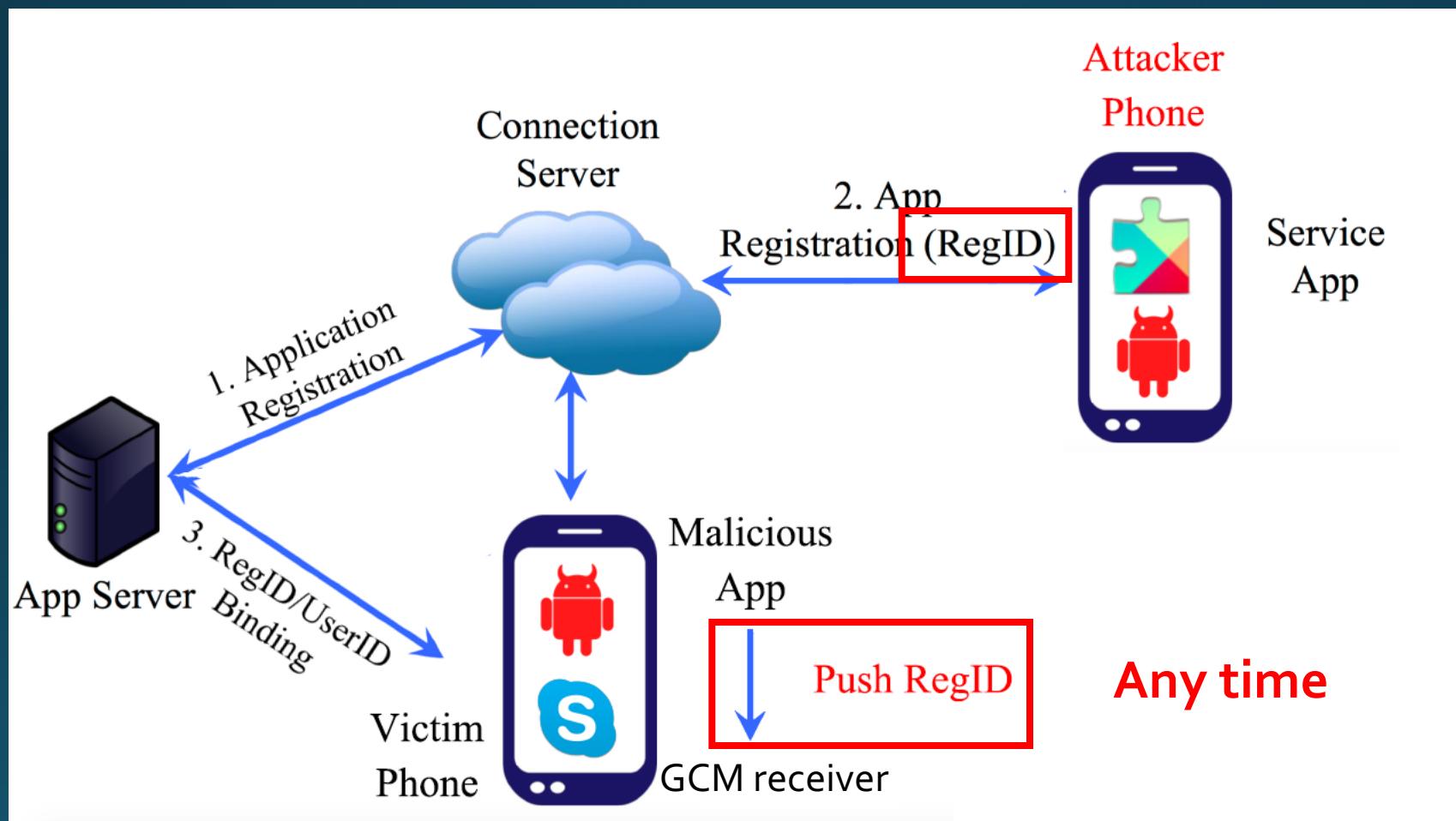


Amazon Device Messaging

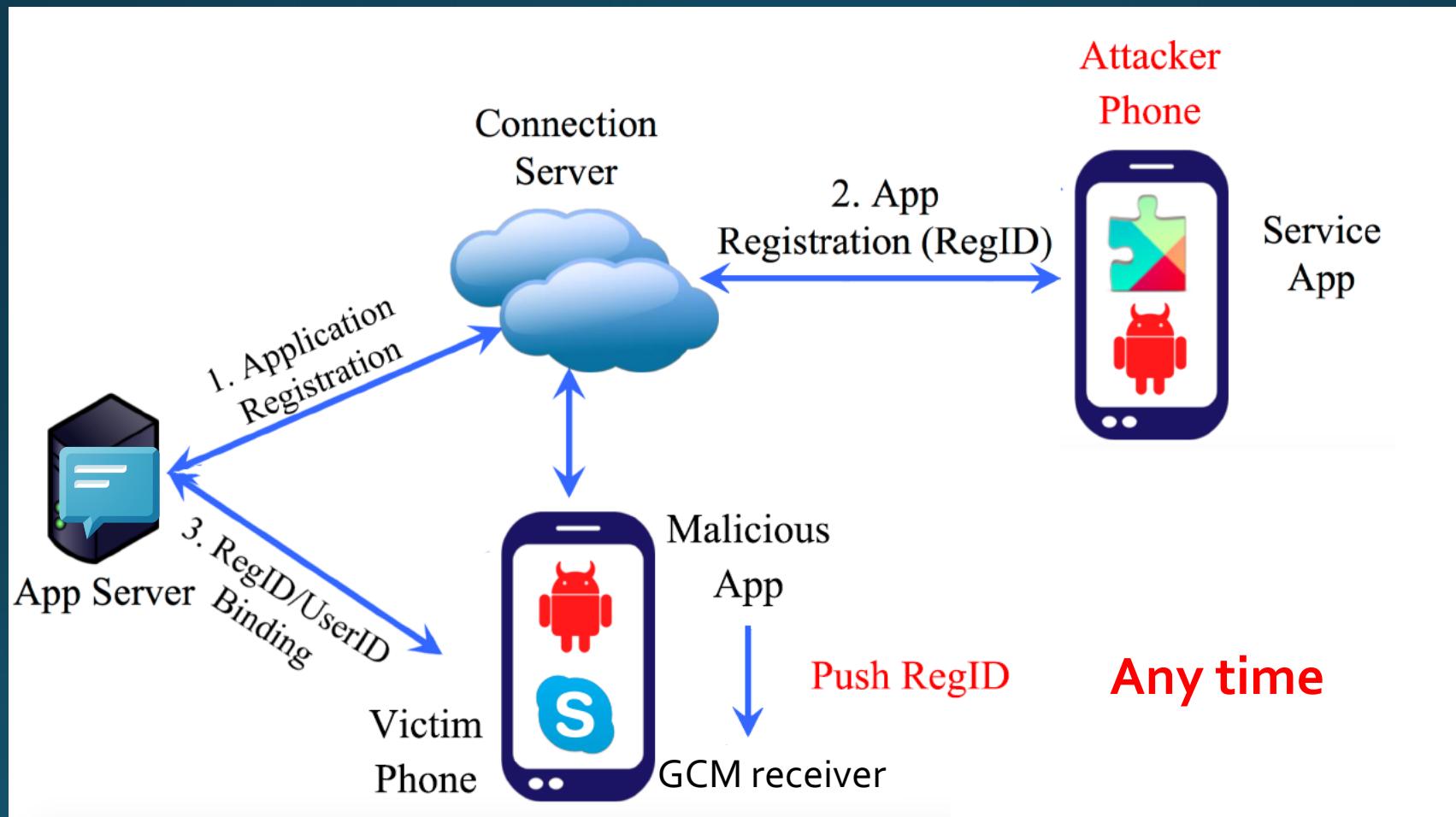
GCM Setup



Message Stealing



Message Stealing



Demos



- Google Email Attack



- LG CloudHub Attack



- Facebook Service Confusion



- Skype Message Stealing

All deomos can be found at <https://sites.google.com/site/androidharehunting/>
and <https://sites.google.com/site/perplexedmsg/>

App Store Submission

- We have submitted some of attack apps to several app stores (e.g. Google, Amazon, Samsung).
- Passed vetting process easily.
- Removed immediately after approval.

Responsible Disclosure

- All vulnerabilities we have discovered have submitted to vendors at least 3 months prior to publication.
- Most of them have been acknowledged and fixed.

Automatically Detect Hare

Attribute Reference

StartService(...)

StartActivity(...)

Query(...)

readPermission(...)



Attribute Definition

Package Names...

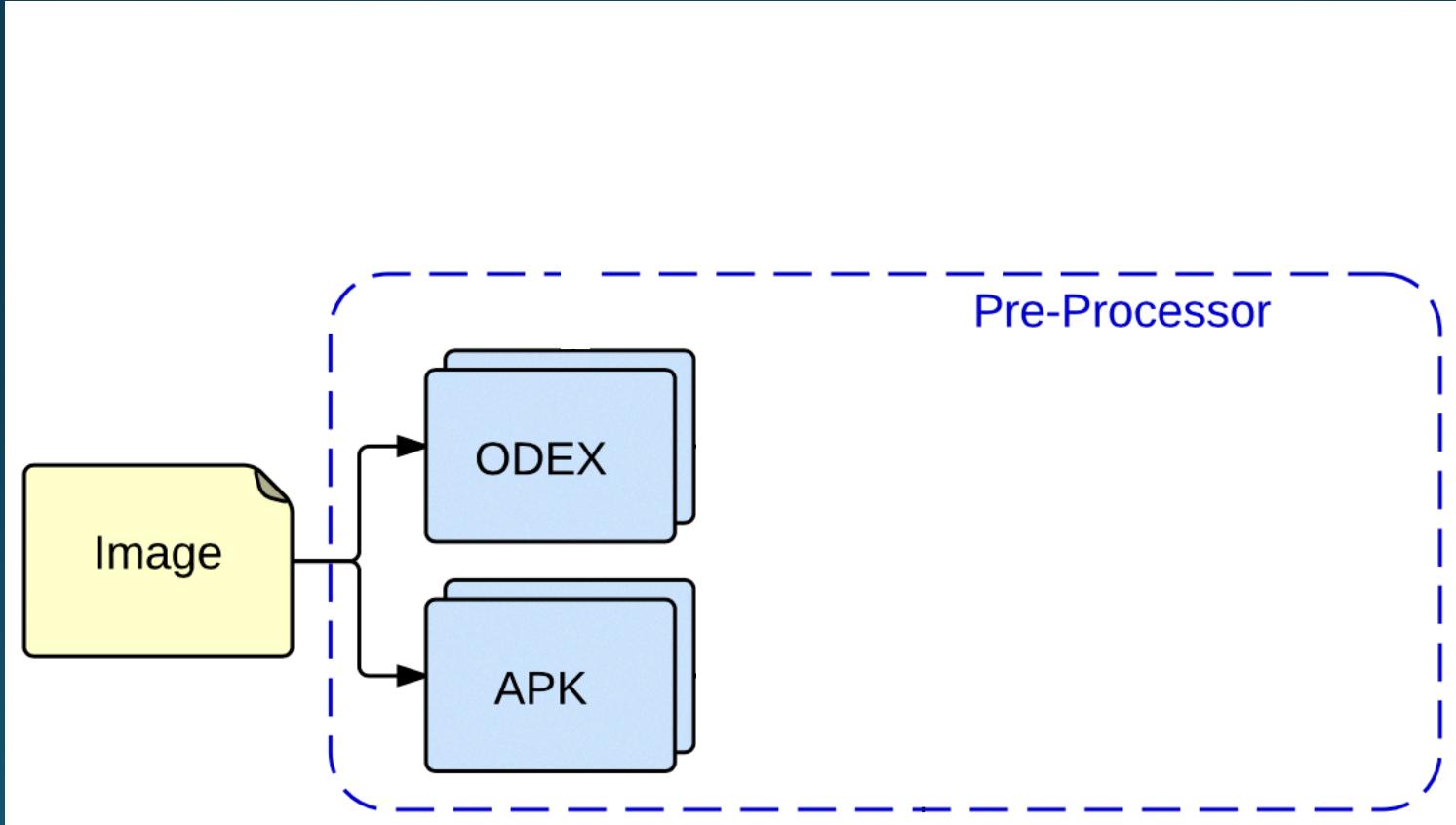
Actions...

Content Providers...

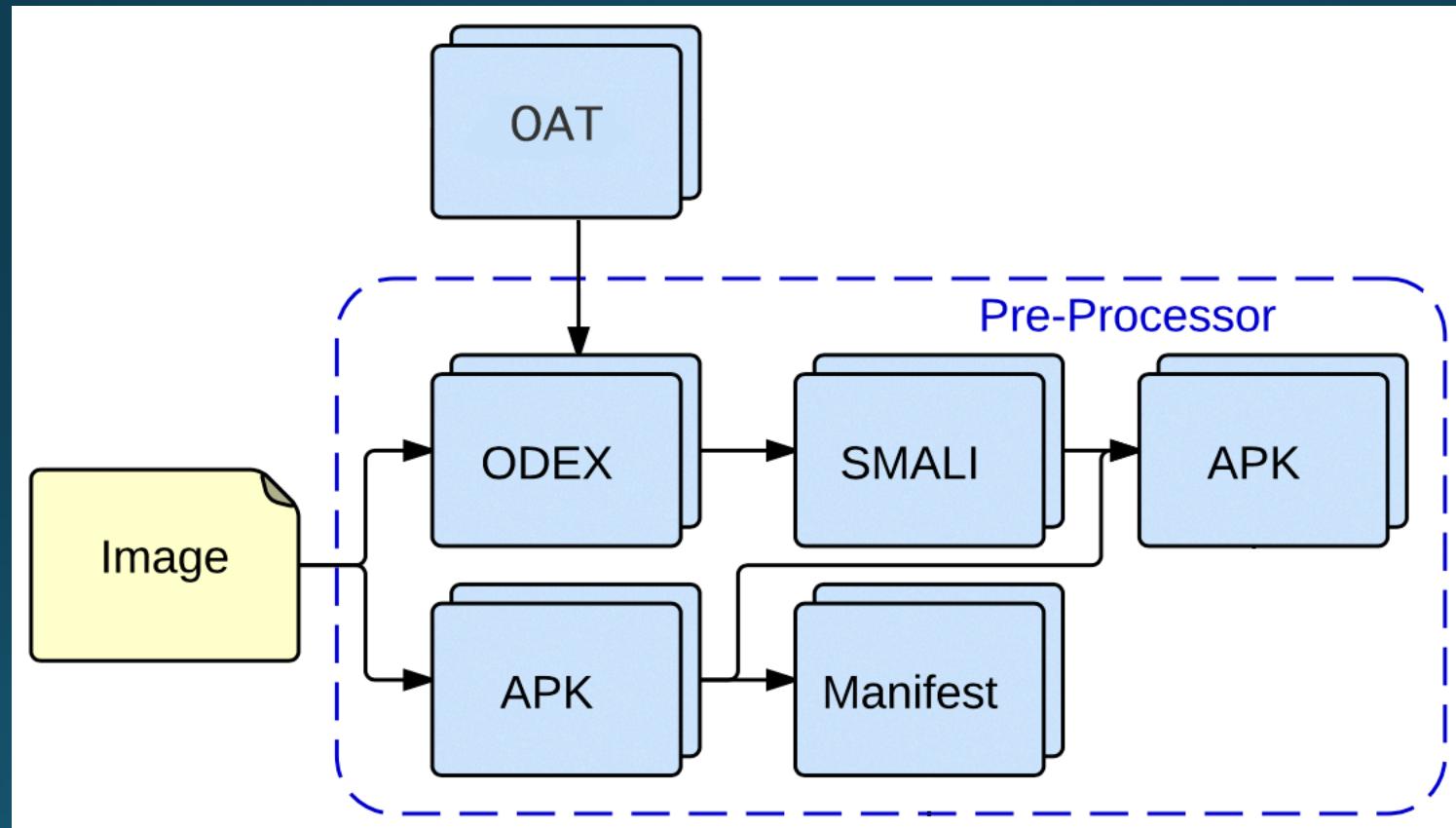
Permissions...



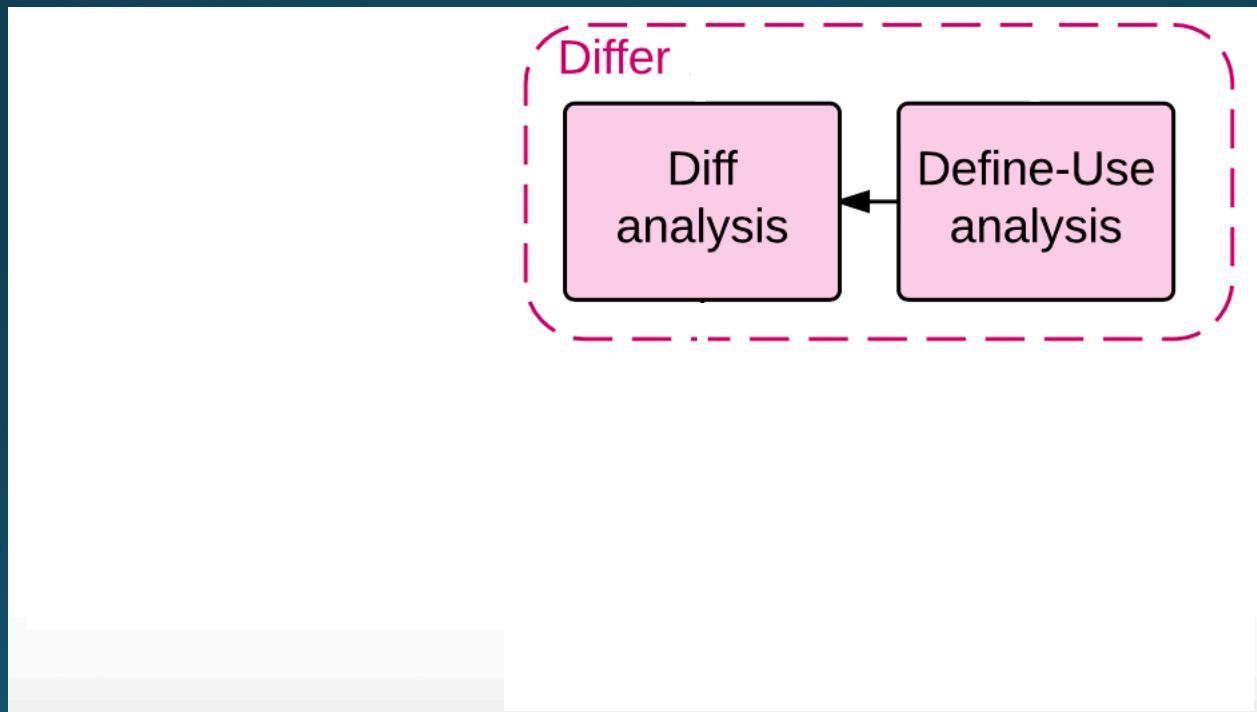
Construct APK



Construct APK



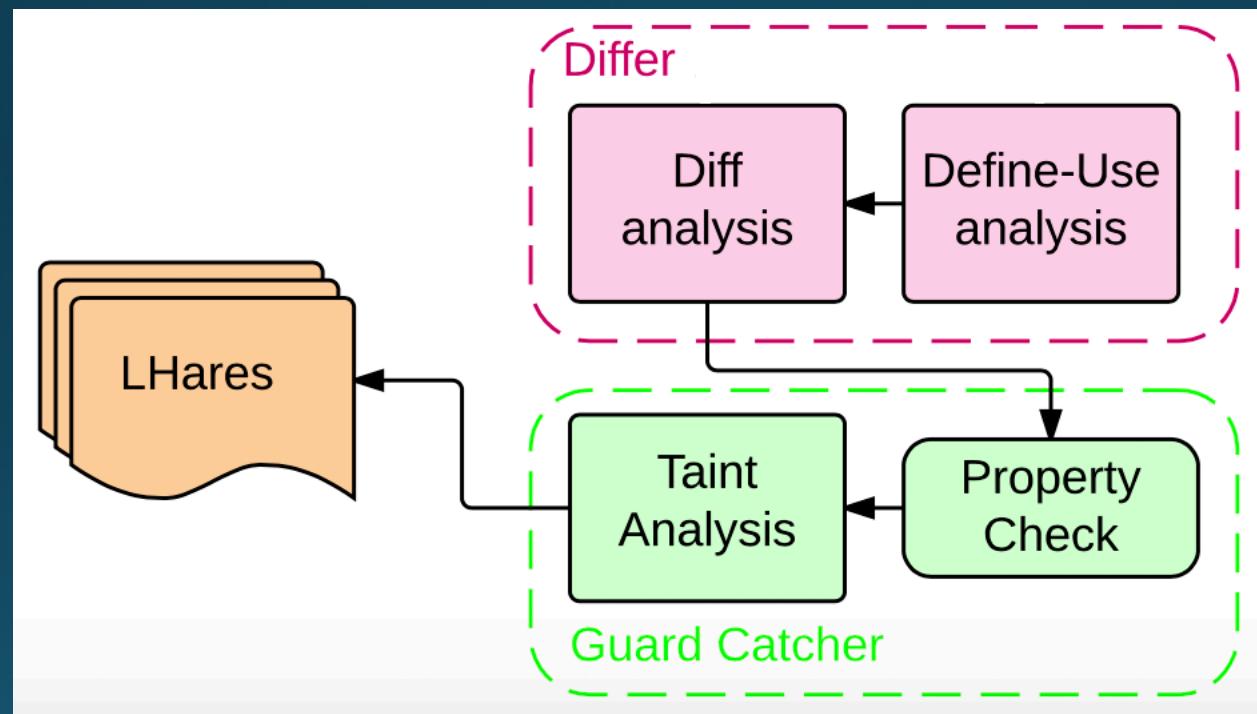
Generate Likely Hare



Guard Detection

- Taint Analysis
 - Source: guard API
 - Sink: Attribute reference API
- If a relationship is established between the source and the sink, then the Attribute Reference API is guarded.

Generate Likely Hare



Large Scale Measurement

Vendor	# of Images	# of System Apps	Avg # of System Apps Per Images	# of Countries	# of Carriers	# of OS Versions
A	83	21733	261	36	23	10
B	7	1561	223	1	1	4
C	1	174	174	1	1	1
D	4	398	99	1	1	3
E	2	319	159	2	1	2
Total	97	24185	183	36	23	10

Findings

Vendor	Hares in Android 4.X		Hares in Android 5.X		Avg Hares per Device	Min Hares per Device	Max Hares per Device
	# of Hares	# of Vulnerable Apps	# of Hares	# of Vulnerable Apps			
A	19279	3045 (18%)	608	99 (6%)	239	23	598
B	679	121 (13.3%)	425	85 (15.5%)	157	100	224
C	N/A	N/A	248	33 (21.5%)	248	248	248
D	107	31 (12.4%)	8	5 (5%)	29	8	45
E	187	23 (15.6%)	16	8 (12.1%)	101	16	187
Total	20252	3220 (14.3%)	1305	230 (11.7%)	153	8	598

Discussion

- In the absence of sufficient guidance and a proper enforcement mechanism, hanging references become inevitable.
- Systematic efforts should be made to eliminate hare flaw.

Eliminating Hare

- Secure each attributes reference.
 - explicit authentication
- Document interdependent relations between components.

Dangerous Hare

- Hare Hunting in the Wild Android: A Study on the Threat of Hanging Attribute References. (CCS '15)
[Yousra Aafer, Nan Zhang]*, Zhongwen Zhang, Xiao Zhang, Kai Chen, XiaoFeng Wang, Xiaoyong Zhou, Wenliang Du, and Michael Grace.
- Perplexed Messengers from the Cloud: Automated Security Analysis of Push-Messaging Integrations. (CCS '15)
 - [Yangyi Chen, Tongxin Li]*, XiaoFeng Wang, Kai Chen, and Xinhui Han. 2015.

* Co-first Author

Thank you!

Any questions?

System Security Lab, Indiana University

Sunk In Thought
System Security Lab, Indiana University



sit.soic.indiana.edu/