black hat®
ASIA 2017

MARCH 28-31, 2017
MARINA BAY SANDS / SINGAPORE

Tongbo Luo
Cong Zheng
Zhi Xu
Xin Ouyang
paloalto NETWORKS®

ANTI-PLUGIN: DON'T LET YOUR APP PLAY AS AN ANDROID PLUGIN

Bio

- Black Hat Veteran.
- Principle Security Researcher @ PANW.

Mobile Security
- Discover Malware
- Android Security

Web Security
- Exploit Kit Detection.
- Browser Security.

Explore & Exploit
- Fuzzing & CVEs.
- Attacks.

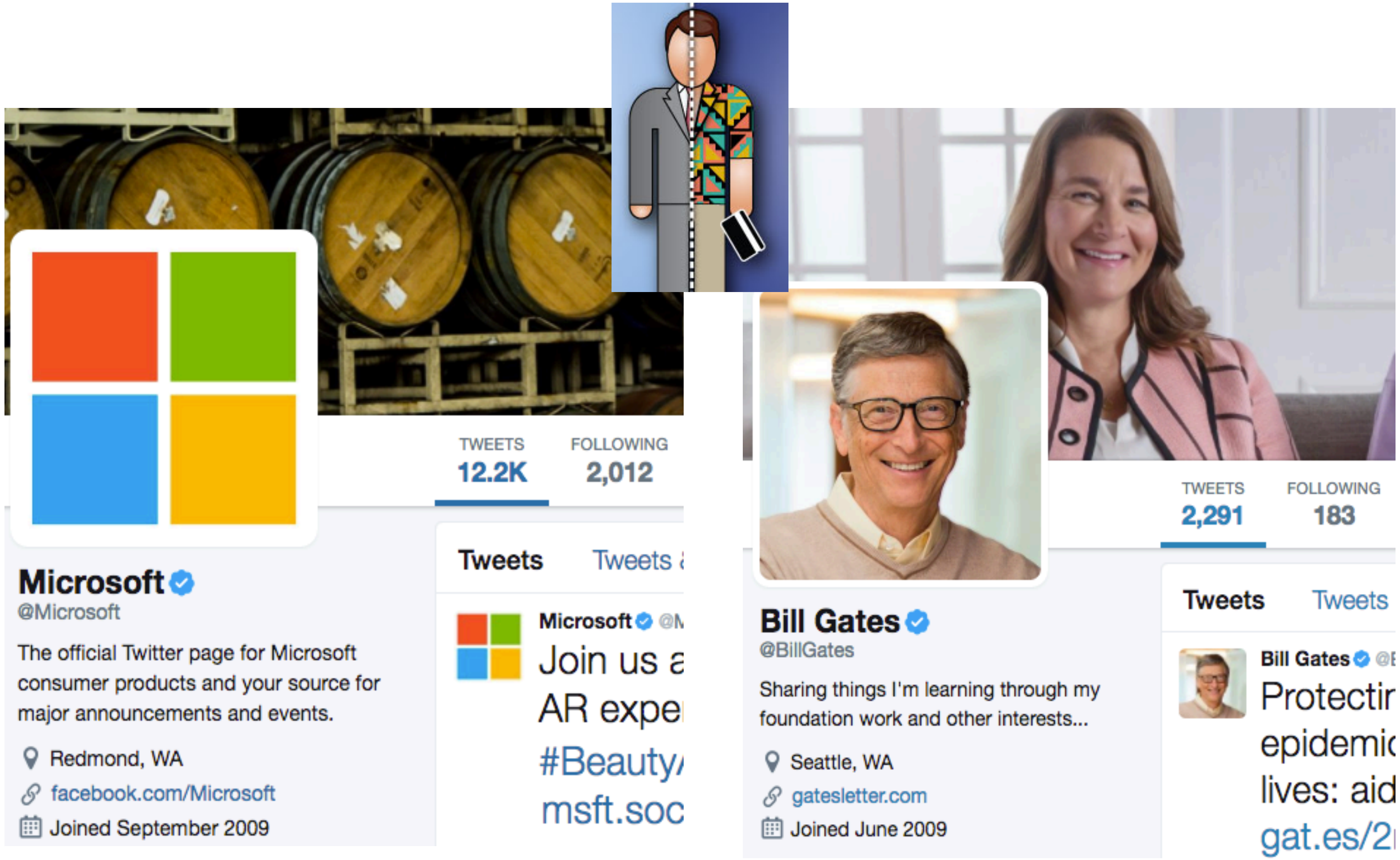
Agenda

- Plugin Technology Background
- Demystify Plugin Technology
- Abuse by Malwares
- Solution

Background of Plugin Technology

Why Plugin Technology is Popular?

Want to log in Multiple Accounts Simultaneously?



Microsoft
@Microsoft

The official Twitter page for Microsoft consumer products and your source for major announcements and events.

Redmond, WA
facebook.com/Microsoft
Joined September 2009

TWEETS 12.2K FOLLOWING 2,012

Tweets Tweets

Microsoft @M
Join us a
AR expe
#Beauty
msft.soc

Bill Gates
@BillGates

Sharing things I'm learning through my foundation work and other interests...

Seattle, WA
gatesletter.com
Joined June 2009

TWEETS 2,291 FOLLOWING 183

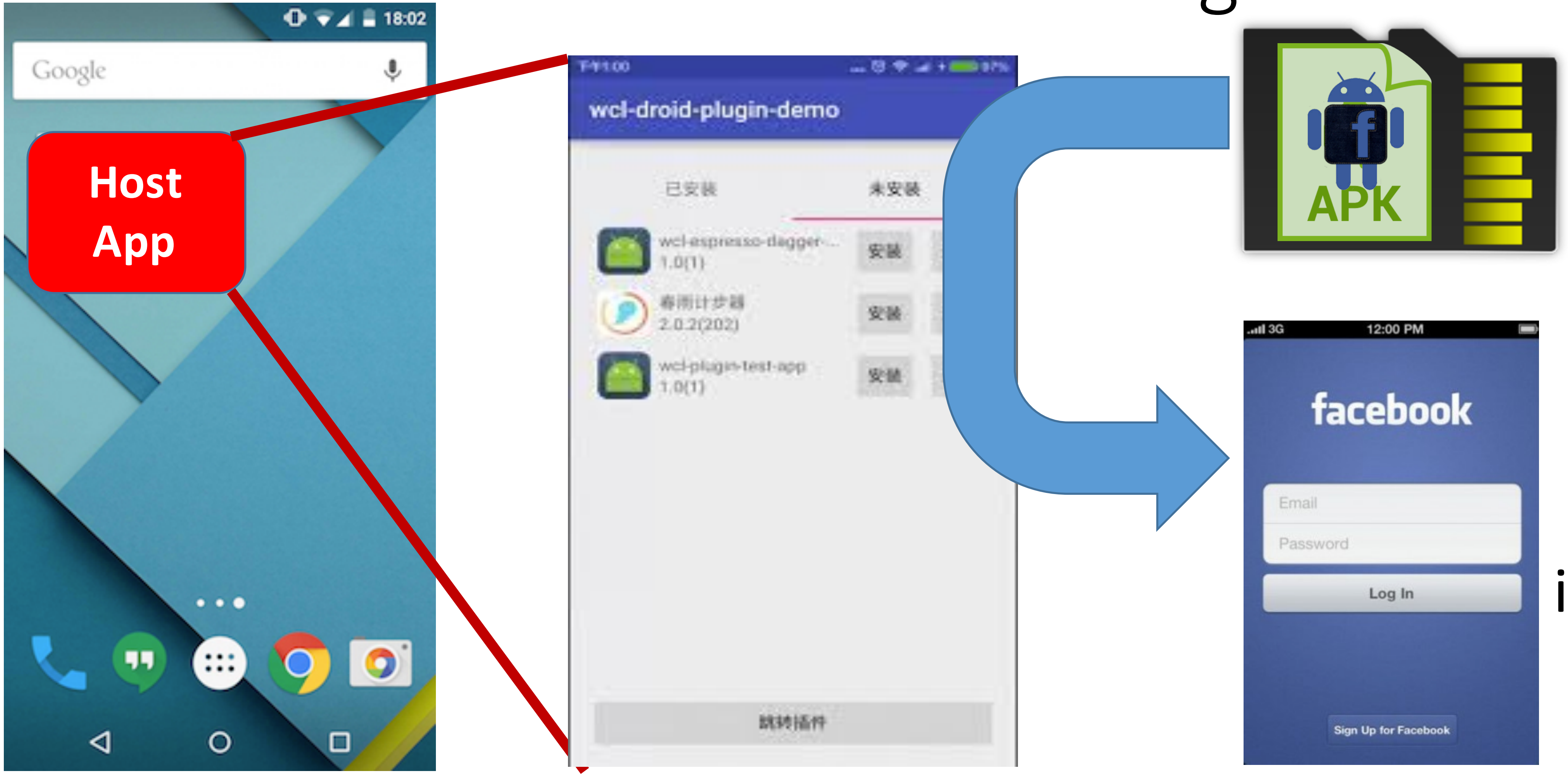
Tweets Tweets

Bill Gates @E
Protectir
epidemic
lives: aid
gat.es/2

What is Android Plugin Technology?

- Launch an APK file within an Android app.
- In the unrooted device.
- “Host App” = Android app
- “Plugin” = APK file.
- No need to install the plugin.

What is Android Plugin Technology?



Host App

Plugin

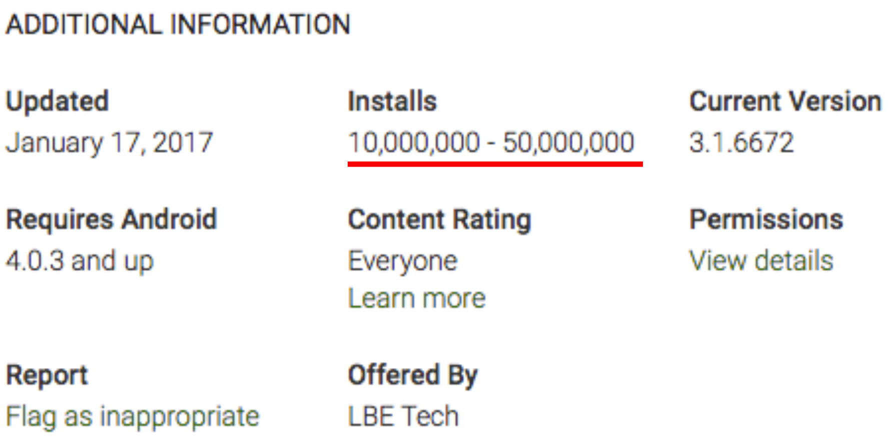
Launch APK w/o installation

vs Dynamic Code Loading (DCL)

- Load + Execute code at runtime.
- Not part of its initial static code base
- Use API like Java Class loader, Runtime.exec.
- Plugin technology is more advanced.

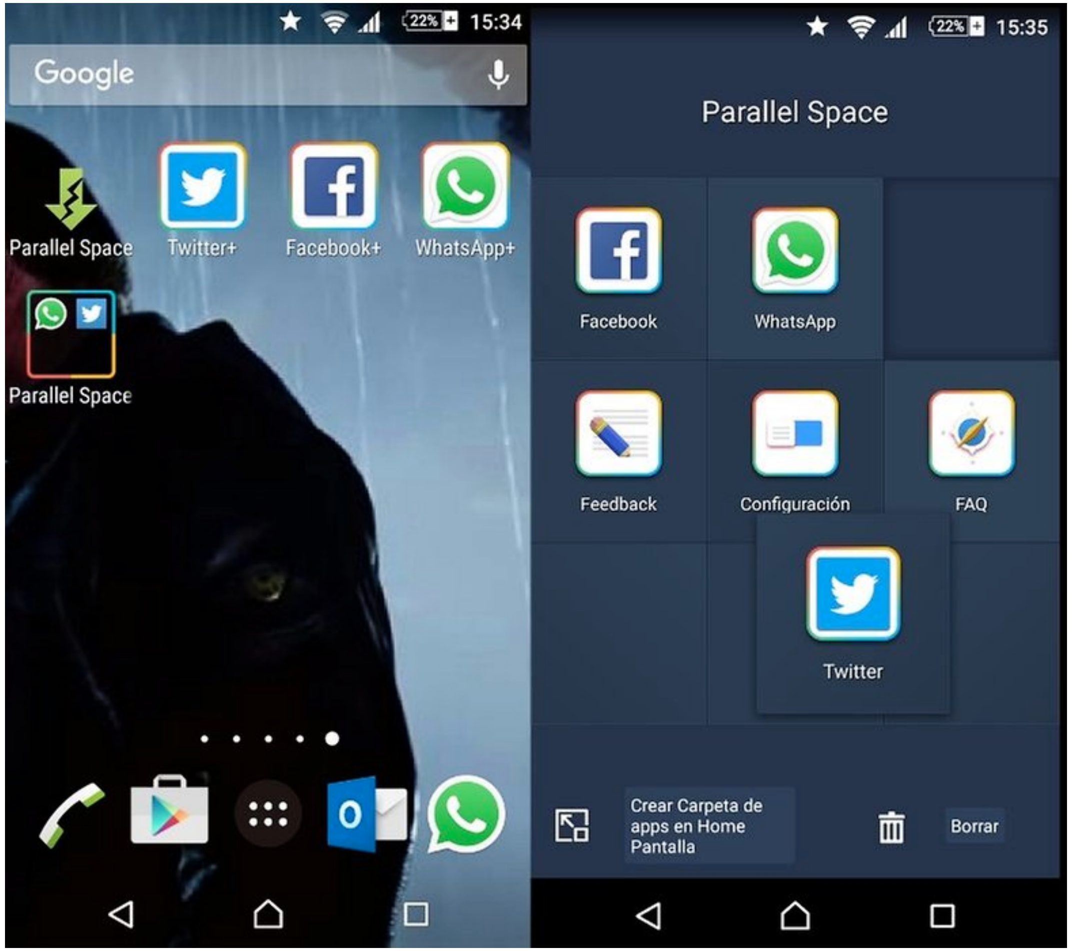
Parallel Space

The Most popular Plugin App



ADDITIONAL INFORMATION

Updated	Installs	Current Version
January 17, 2017	10,000,000 - 50,000,000	3.1.6672
Requires Android	**Content Rating**	**Permissions**
4.0.3 and up	Everyone Learn more	View details
Report	**Offered By**	
Flag as inappropriate	LBE Tech	

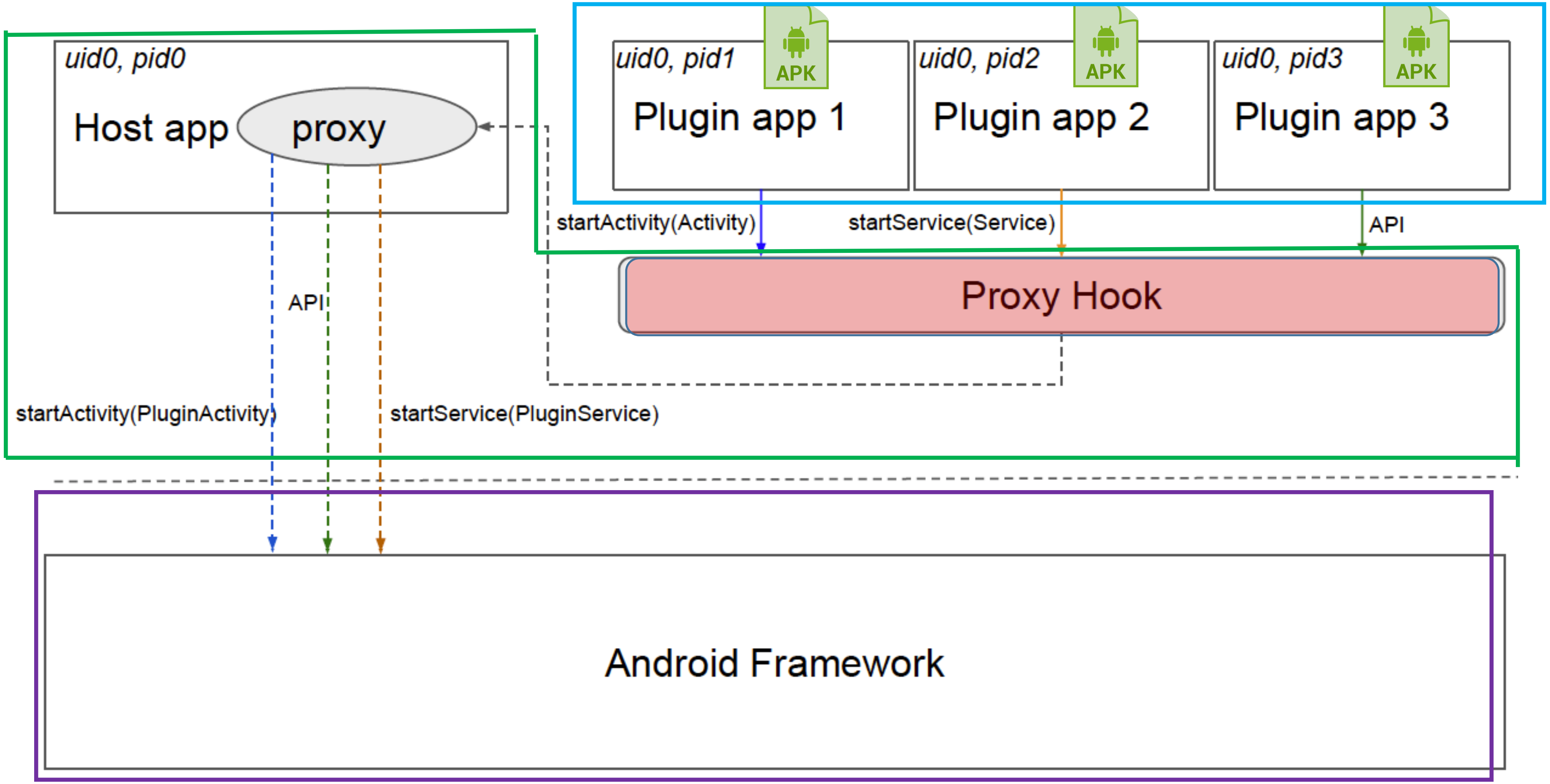


DroidPlugin

- The most popular SDK implemented Plugin technology.
- Open-Sourced.
- developed by Qihoo 360.

Demystify Plugin Technology

Droidplugin Overview



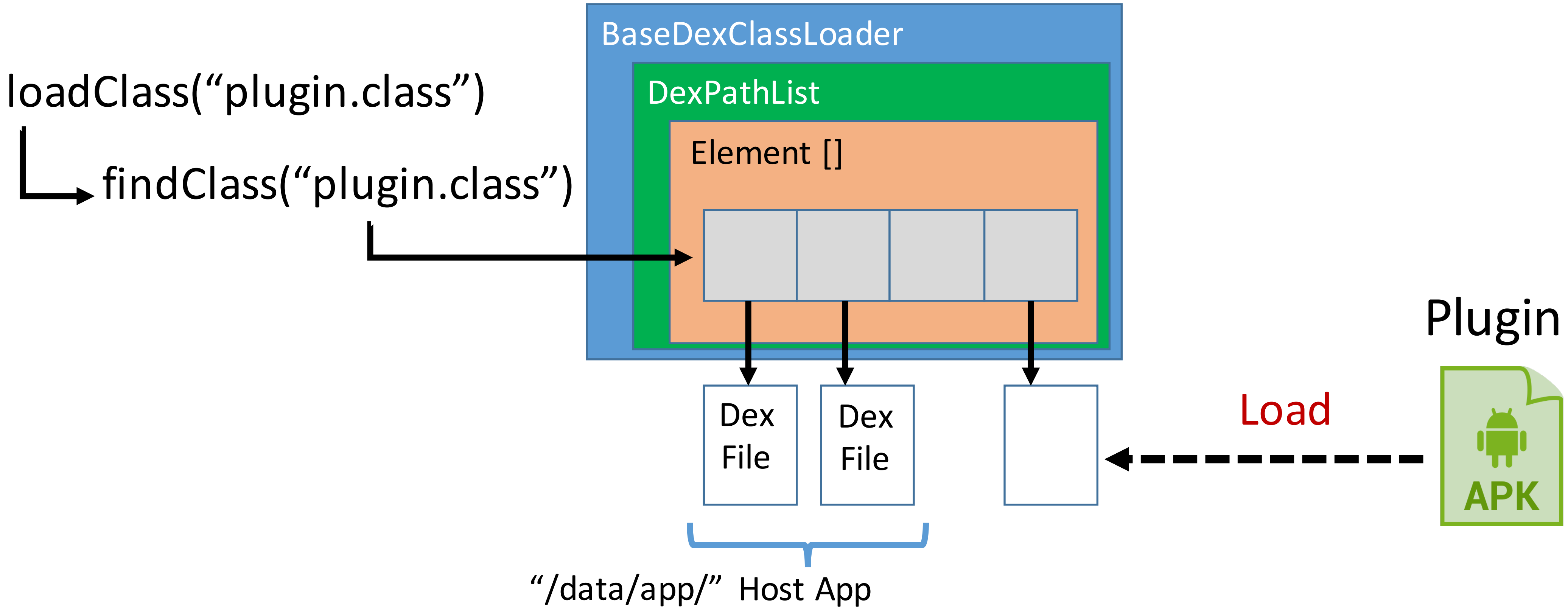
How to create a virtual environment?

- Hooking.
- How to hook API?
  - Java Dynamic Proxy API.
  - Java Reflection.
- What API to hook?

What API to hook?

- Load and launch plugin (APK) without installation.
- Manage the lifecycle of app components (activity, service, content-provide, broadcast-receiver.)
- Inter-plugin communication.
- Plugin management (download, update)

- Launch APK file without installation.



loadClass(“plugin.class”)

findClass(“plugin.class”)

BaseDexClassLoader

DexPathList

Element []

Dex File

Dex File

“/data/app/” Host App

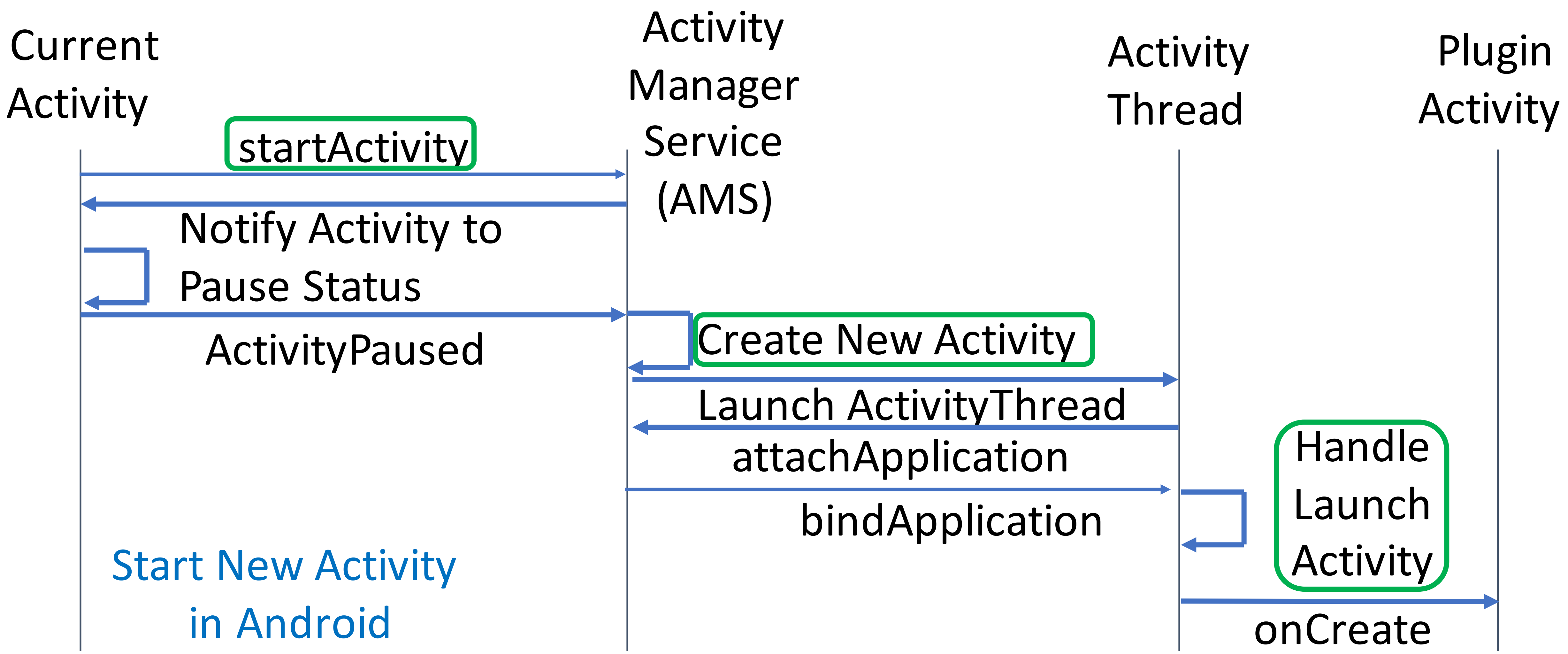
Load

Plugin

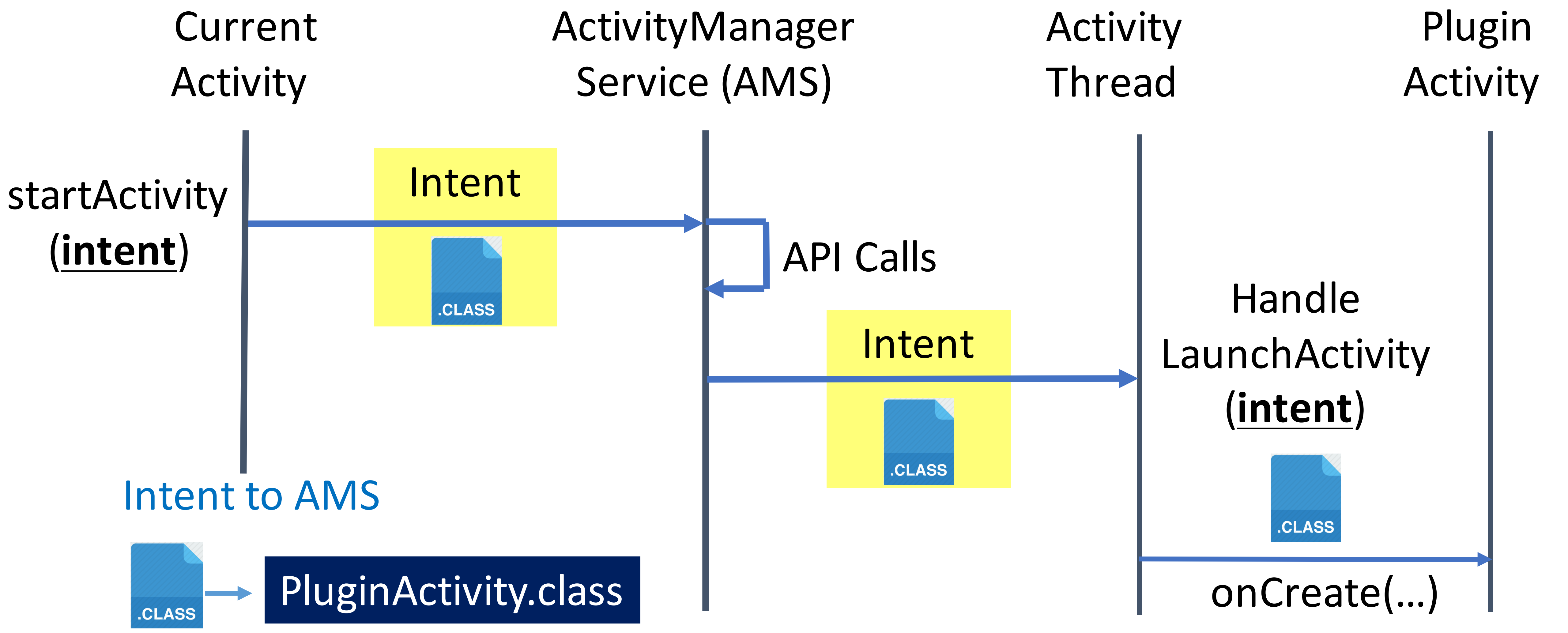
APK

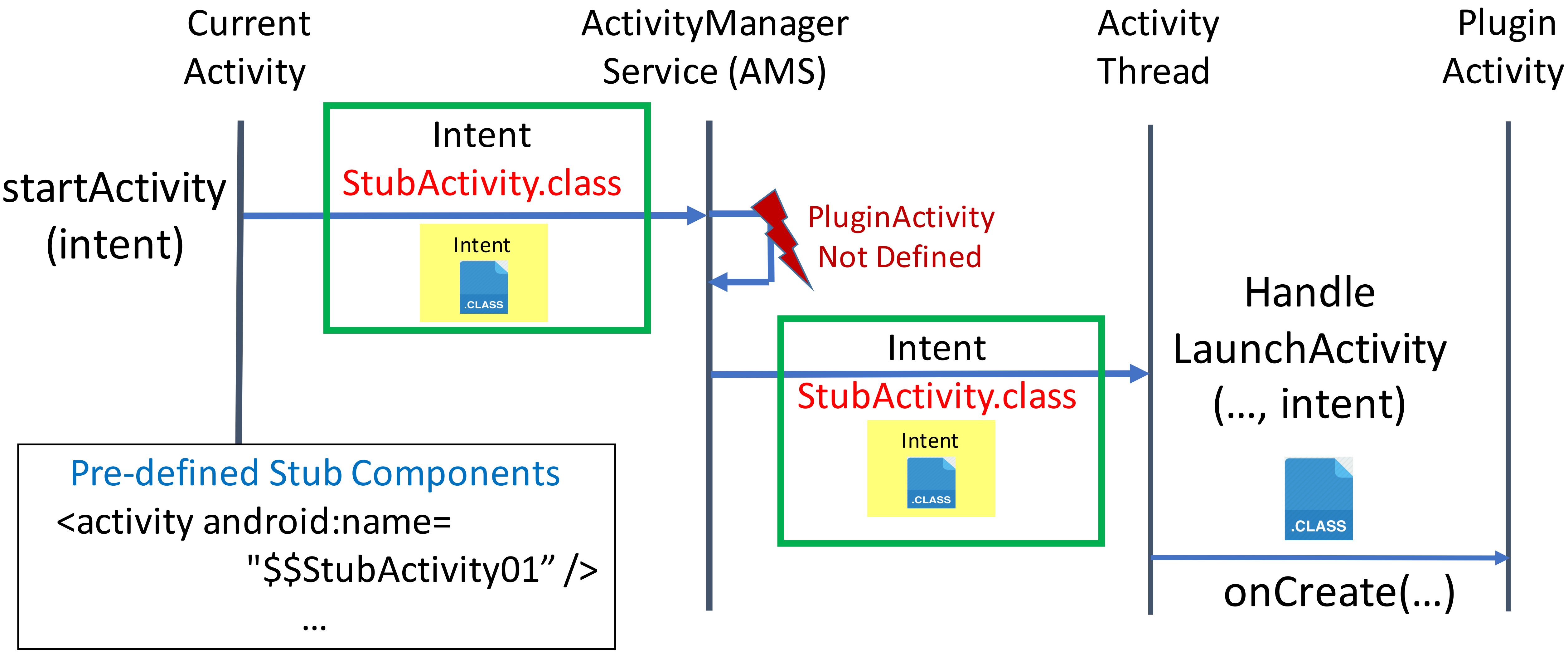
Manage the lifecycle of App components

- App Components
  - **Activity**
  - Service
  - Broadcast Receiver
  - Content Provider
- System maintain the lifecycle



Start New Activity in Android


Current Activity
ActivityManager Service (AMS)
Activity Thread
Plugin Activity
startActivity (**intent**)
Intent
.CLASS
API Calls
Intent
.CLASS
Handle LaunchActivity (**intent**)
.CLASS
onCreate(...)
Intent to AMS
.CLASS
PluginActivity.class



Current Activity

ActivityManager Service (AMS)

Activity Thread

Plugin Activity

startActivity (intent)

Intent

StubActivity.class

PluginActivity Not Defined

Intent

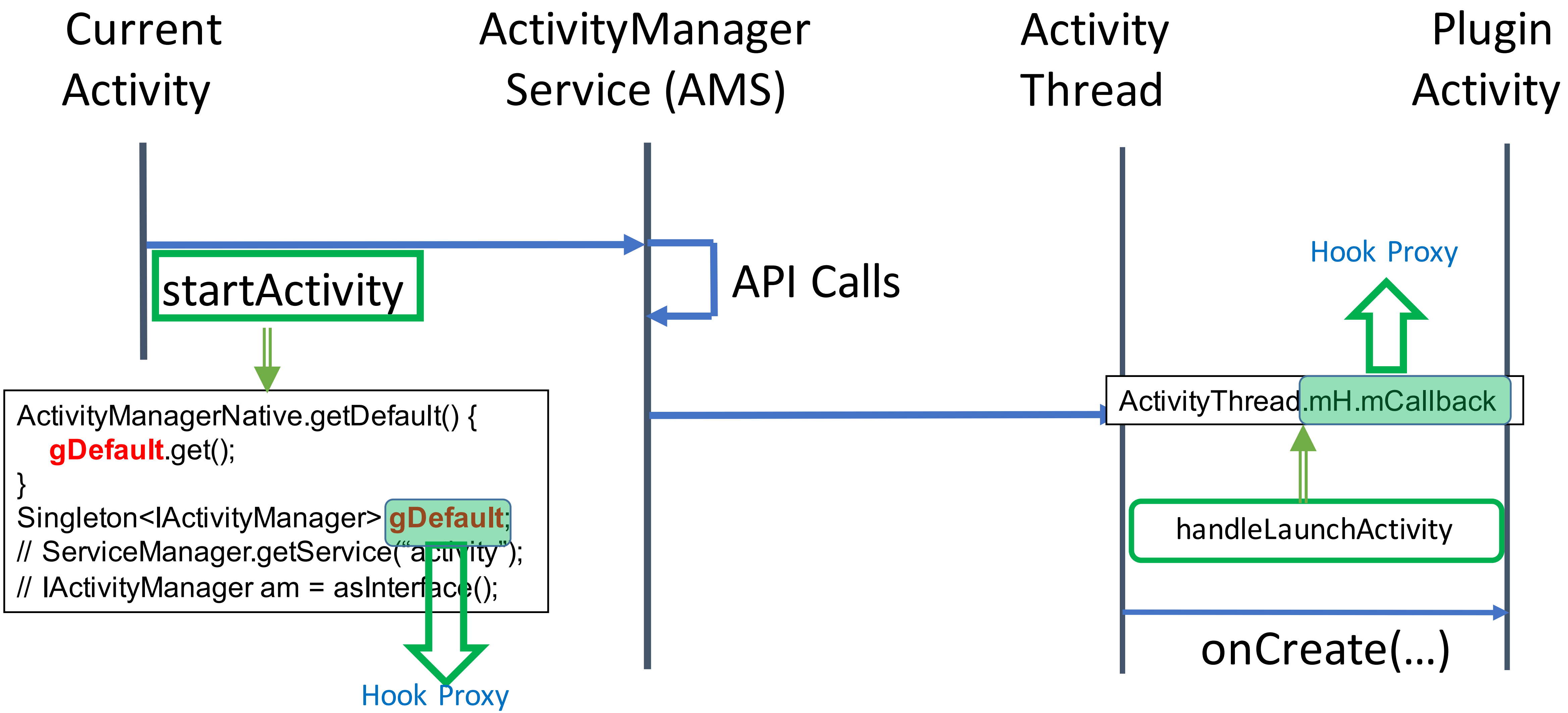
StubActivity.class

Handle LaunchActivity (…, intent)

onCreate(…)

Pre-defined Stub Components

<activity android:name=
"$$StubActivity01” />
…



Current Activity

ActivityManager Service (AMS)

Activity Thread

Plugin Activity

startActivity

API Calls

```
ActivityManagerNative.getDefault() {
    gDefault.get();
}
Singleton<IActivityManager> gDefault;
// ServiceManager.getService(“activity”);
// IActivityManager am = asInterface();
```

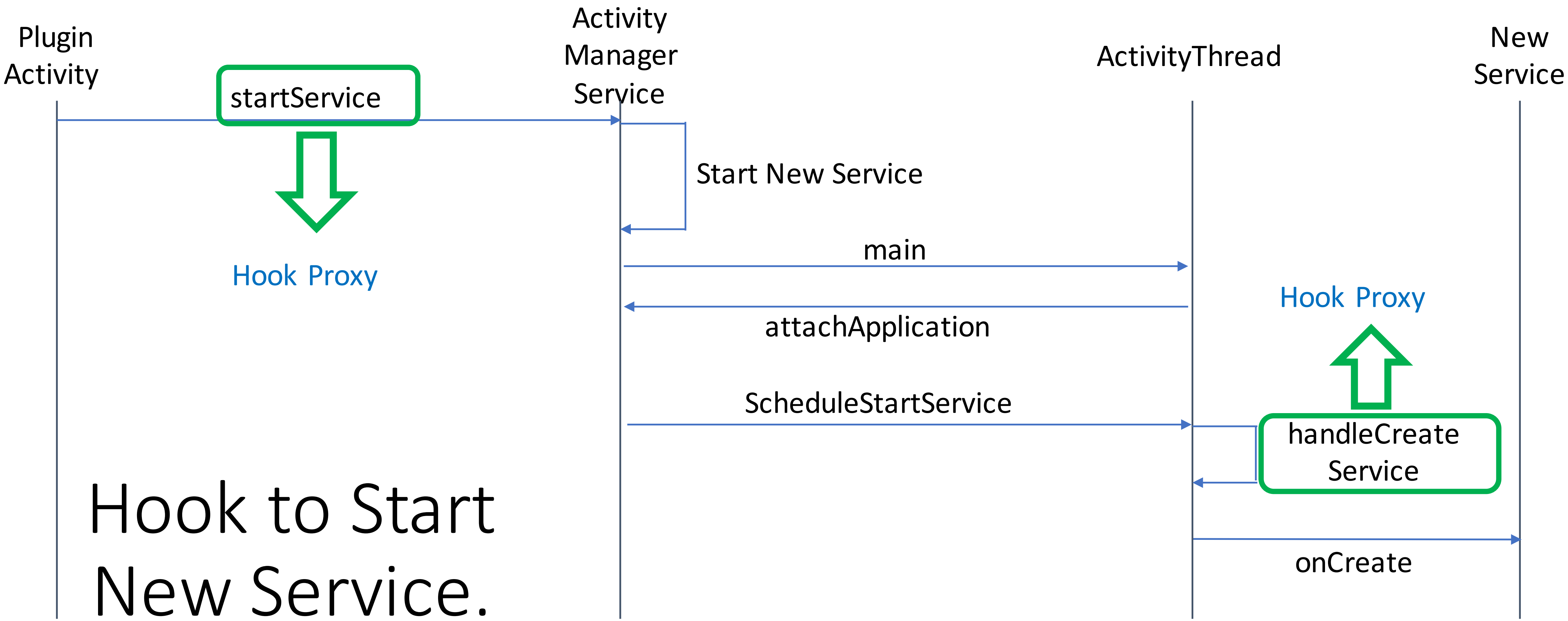
Hook Proxy

ActivityThread.mH.mCallback

Hook Proxy

handleLaunchActivity

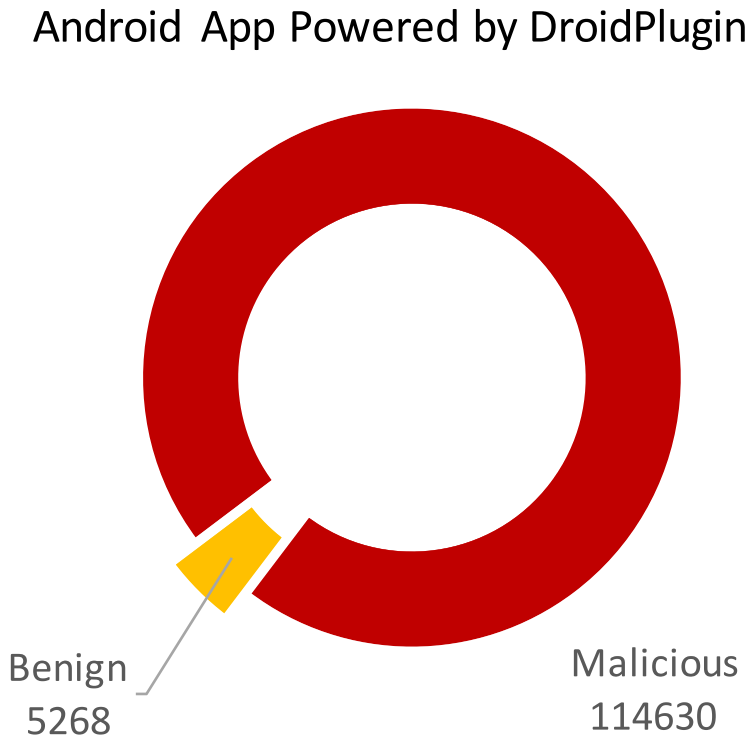
onCreate(...)

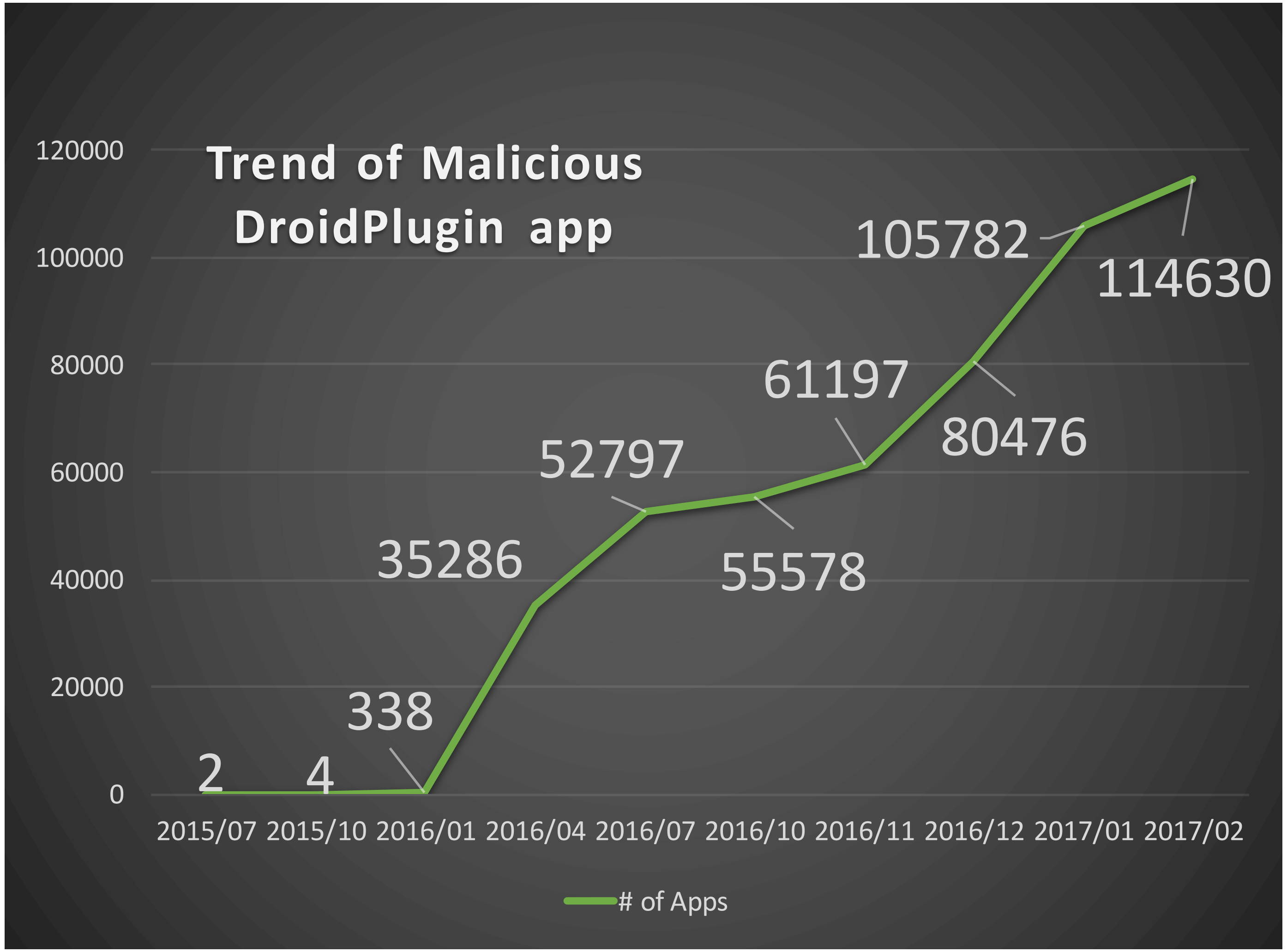


Hook to Start New Service.

Abusing Plugin Technology by Malware

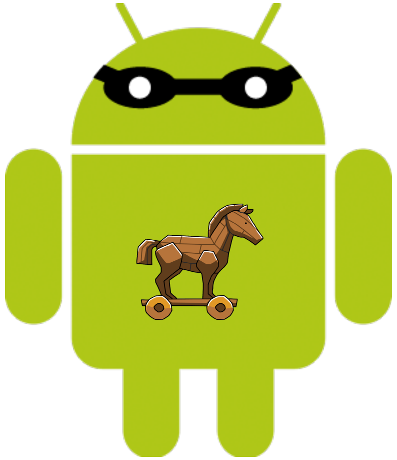
Abusing of DroidPlugin





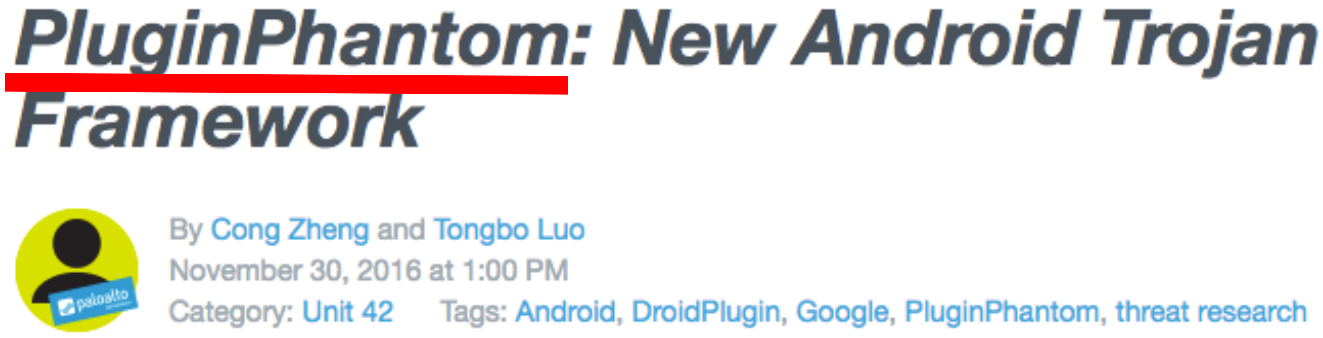
Benefit of Abusing DroidPlugin

Update/Install New Malware Without Rooting the Phone



Evade Static Detection

Phish on Authenticated App Without Repackaging



PluginPhantom: New Android Trojan Abuses "DroidPlugin" Framework

By Cong Zheng and Tongbo Luo
November 30, 2016 at 1:00 PM
Category: Unit 42 Tags: Android, DroidPlugin, Google, PluginPhantom, threat research

paloalto NETWORKS

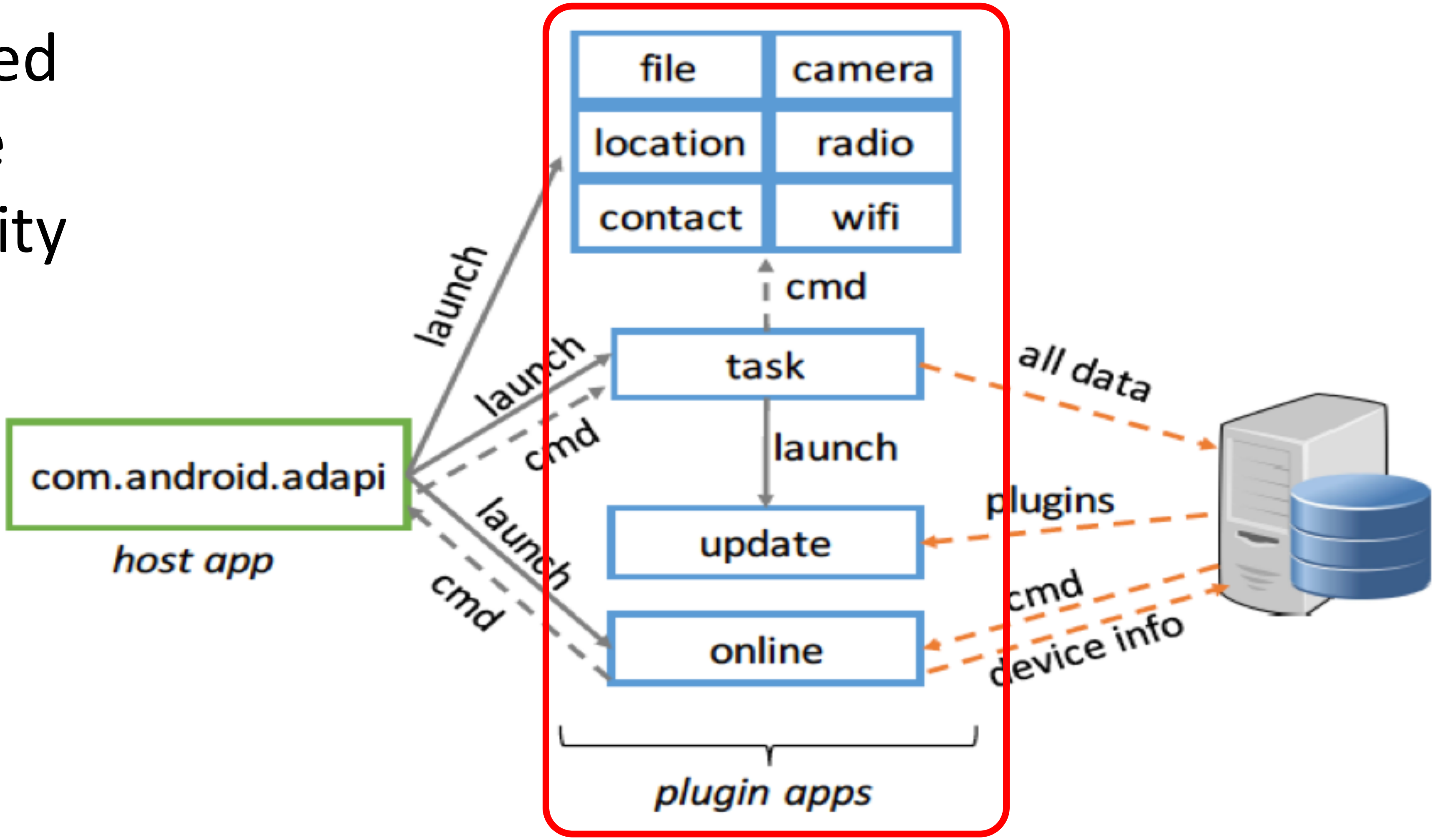
November 30, 2016

PluginPhantom trojan expoits Android plugins to snoop

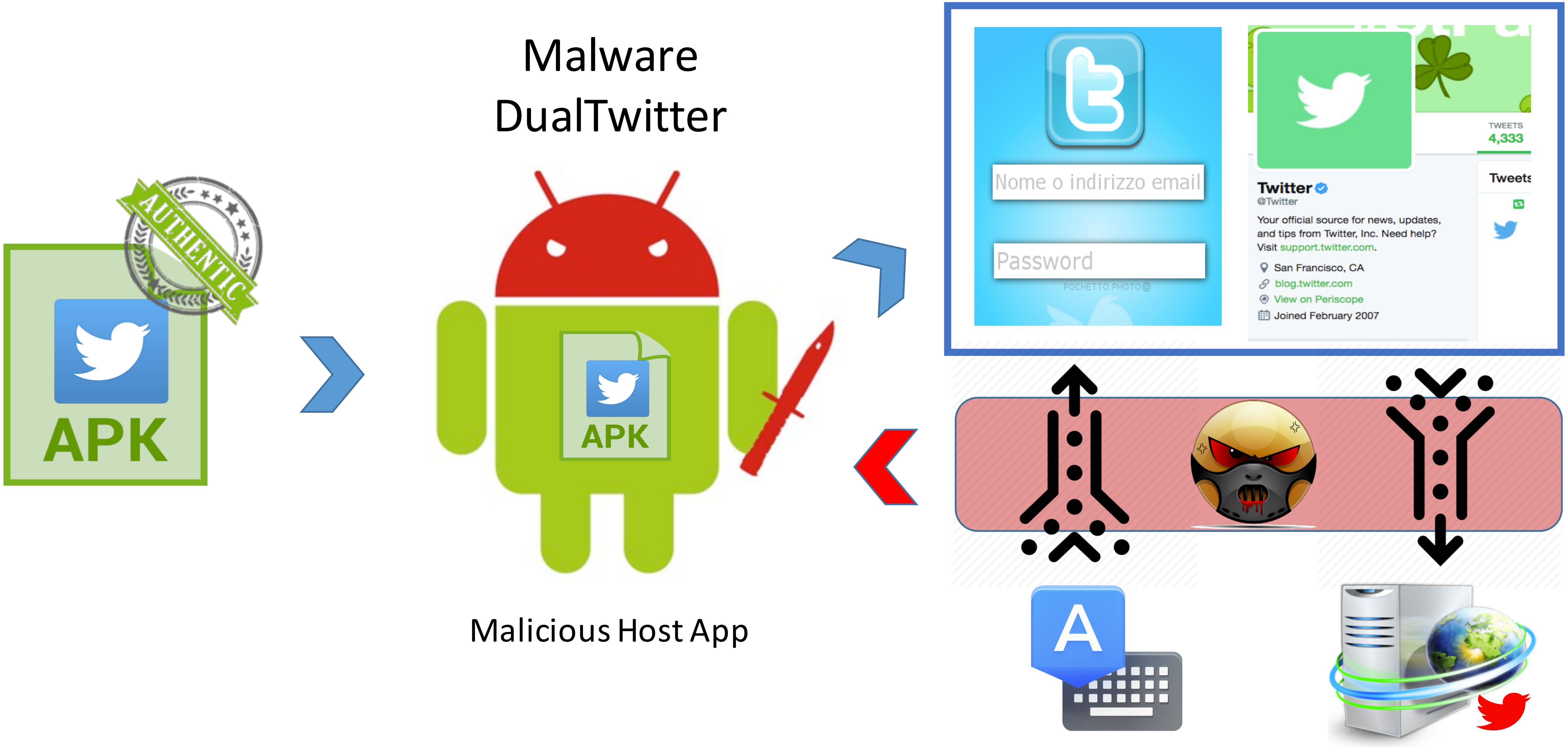
"PluginPhantom" Android Trojan Uses Plugins to Evade Detection

A new class of Trojan as it is the first to abuse Android Plugin technology

Modularized Malware Functionality



Malware DualTwitter



Nome o indirizzo email

Password

POCHETTO PHOTO ©

TWEETS 4,333

Tweets

Twitter
@Twitter

Your official source for news, updates, and tips from Twitter, Inc. Need help? Visit support.twitter.com.

San Francisco, CA

blog.twitter.com

View on Periscope

Joined February 2007

Malicious Host App

Our Solution: Plugin-Killer

Potential Solutions

- Block Plugin Technology.
- Support plugin by Android system.
- Improving Detection Technique.
- Opt-out options for APK file => PluginKiller.

Plugin Killer

- Protect legitimate app from running in malicious host app.
- App fails to be aware of being launched as a plugin.
- Our Solution: PluginKiller.
  - Lightweight Library.
  - Compatible to all Android versions.
- Mechanisms to detect the virtual environment.

Use PluginKiller

```
public class MainActivity extends Activity {
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        if( isLoadedAsPlugin() ) {        => Condition Statement
           TerminatesApp();               => Counter Action
        }
        … … …
    }
}
```

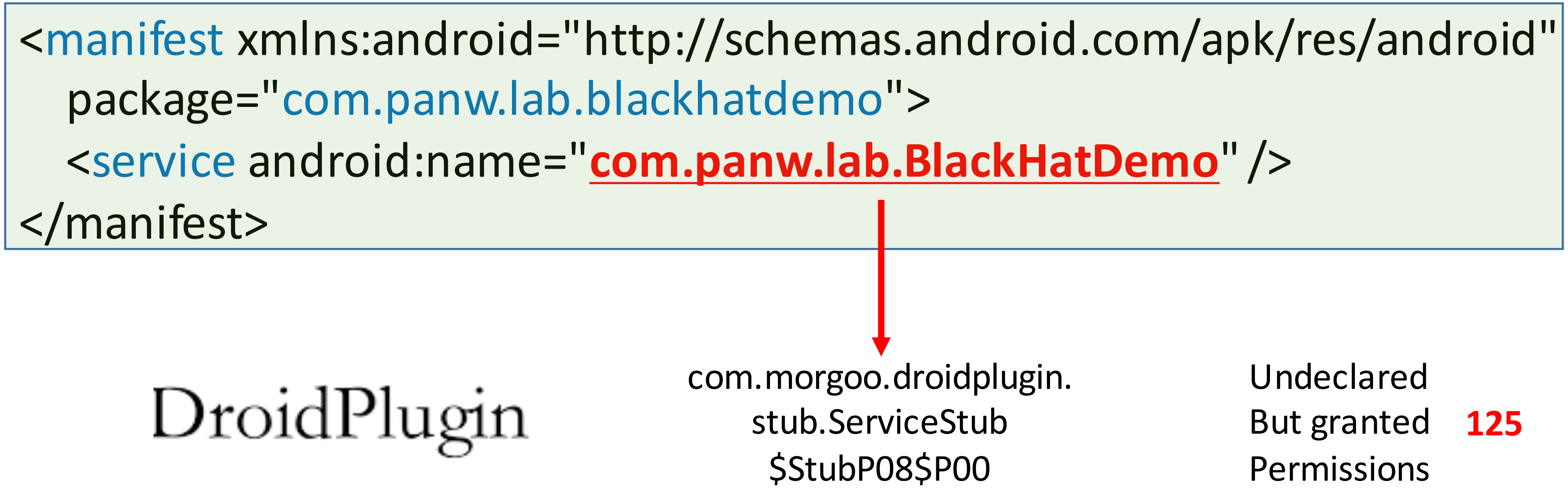
Similar to FrameBuster JavaScript code used in browser.

Detect Virtual Environment

- Mismatch in the Manifest Info
  - Service/Activity Name.
  - Permissions.
- Detect from Runtime Info
  - Process with same UID.
  - Working Directory.
  - Process Name.
- Runtime Change component Features.
  - Enable a broadcast Receiver declared as Disabled in manifest.
- Broadcast Receiver
  - unregister all dynamic receivers and try to trigger static receivers.

Mismatch in the Manifest Info

Plugin's Manifest File



```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.panw.lab.blackhatdemo">
    <service android:name="com.panw.lab.BlackHatDemo" />
</manifest>
```

DroidPlugin

com.morgoo.droidplugin.
stub.ServiceStub
$StubP08$P00

Undeclared
But granted **125**
Permissions

Detect from Runtime Info

	Process Name With Same UID	DataDir: *Directory assigned to the package for its persistent data*
	com.panw.lab.blackhatdemo	/data/data/com.panw.lab.blackhatdemo
DroidPlugin	com.droidplugin.demo **com.droidplugin.demo:PluginP02**	/data/data/**com.droidplugin.demo/Plugin/com.panw.lab.blackhatdemo/data**/com.panw.lab.blackhatdemo

Detect from App Component Behavior

- Number of launched Activity and Service.
  - DroidPlugin defined 10 stub activities and 10 stub services.
  - Launch more than 10 services.
- Static Broadcast Receiver.
  - DroidPlugin converts Static Receiver to Dynamic.
  - Define a Static Receiver, Unregister all Dynamic.
  - In DroidPlugin, no receiver is alive.

Runtime Change component Property

- Enable Broadcast Receiver with static intent-filter.

```
<receiver android:name=”.AntiReceiver"
   android:enabled = ” false ”>
   <intent-filter>
     <action android:name="ANTI_STATIC" />
   </intent-filter>
</receiver>
```

→ Whether or not the broadcast receiver can be instantiated by the system

```
ctx.getPackageManager().setComponentEnabledSetting(
       ComponentName, COMPONENT_ENABLED_STATE_ENABLED, …
)
```

→ Fail to Enable it at Runtime

Test Environments

Parallel Space
by LBE Tech

Go-Multiple
By GO Dev Team X

Parallel Accounts
By ImaTech
Innovations

Parallel Box
By ParallelBoxTeam

Gemini
Multi Accounts

DroidPlugin

VirtualApp

Anti Plugin SDK Evaluation

	Droid Plugin	Go Multiple	Multiple Accounts	Parallel Space	Parallel Accounts	Parallel Box	Gemini
ServiceName Check	DETECTED				DETECTED	DETECTED	DETECTED
Undeclared Permission	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED
SharedUID ProcessCheck	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED
AppRuntimeDir Check	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED	DETECTED
ReceiverFilter Check	DETECTED			DETECTED	DETECTED		
EnabledComp Check	DETECTED			DETECTED	DETECTED		

Three Takeaways

- Android Plugin Technology.
- Abusing of Plugin Technology by malware.
- Lightweight Solution to protect your app.

Q & A

- Looking for collaboration on New detection mechanism.