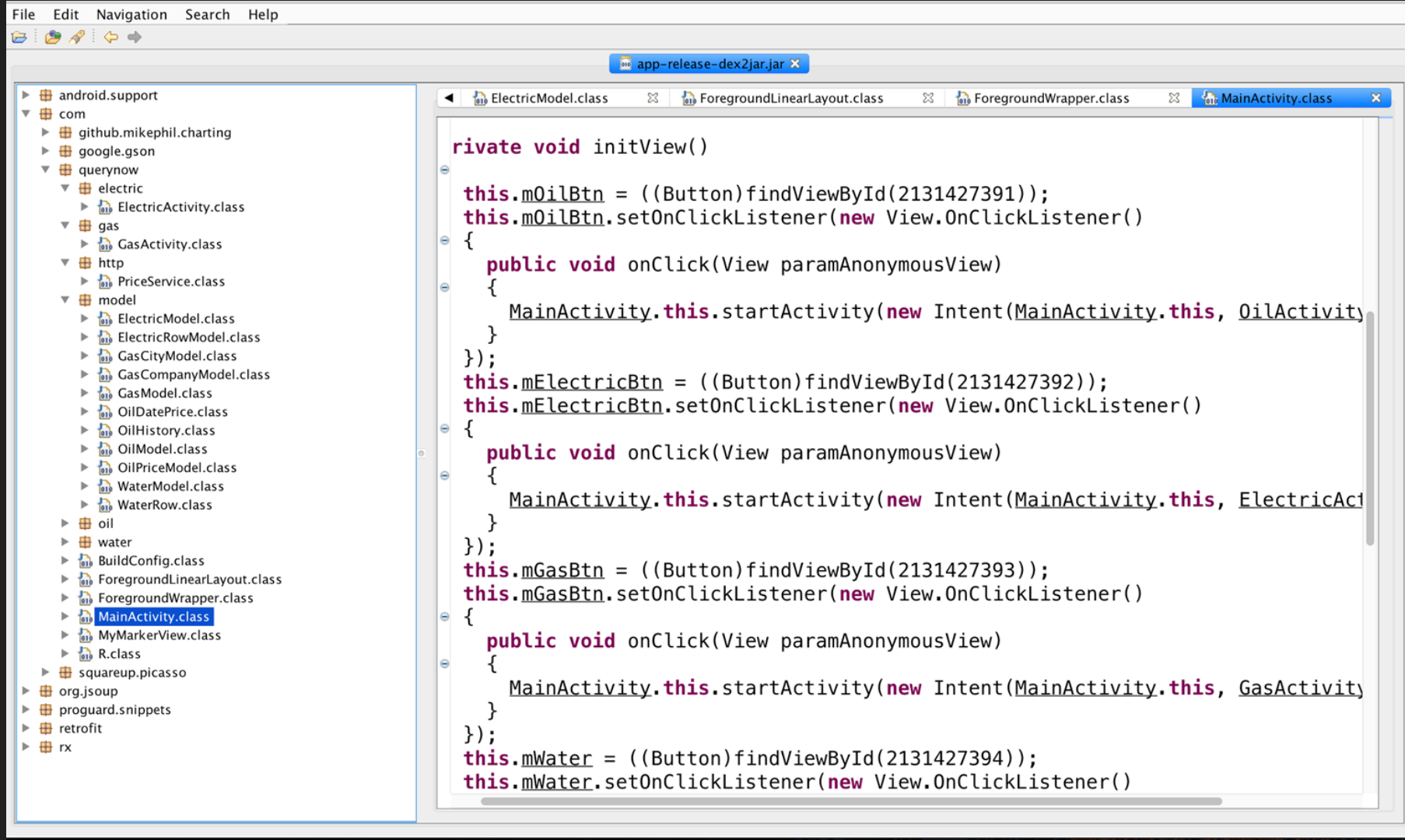
Android App 資安實務 過去/現在/未來

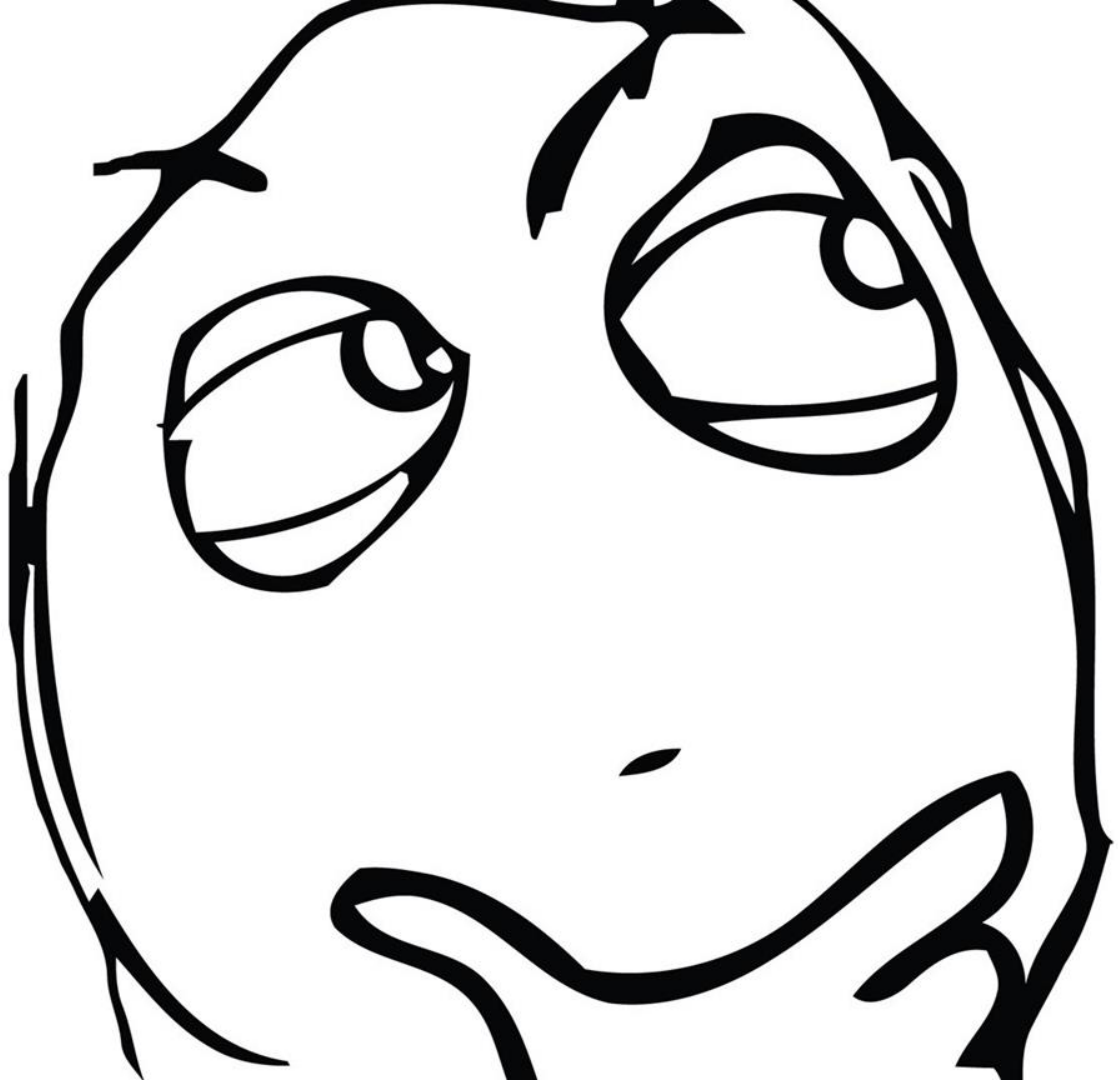
Ted

大綱

- past
  - early version of app without proguard (明碼)
  - dexguard with encrypt class string
- now
  - dexguard with encrypt class string + app 弱掃
  - https
- future
  - sign app 安全防護

```
public class GasActivity extends Activity {
    private ArrayList<GasModel> mList = new ArrayList<>();
    private TextView mFirstTitle;
    private TextView mSecTitle;
    private LinearLayout mFirstRoot;
    private LinearLayout mSecRoot;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.gas_main);
        mFirstTitle = (TextView) findViewById(R.id.first_title
        mSecTitle = (TextView) findViewById(R.id.sec_title);

        mFirstRoot = (LinearLayout) findViewById(R.id.first_Ro
        mSecRoot = (LinearLayout) findViewById(R.id.sec_Root);

        RestAdapter restAdapter = new RestAdapter.Builder()
```


File Edit Navigation Search Help
app-release-dex2jar.jar
android.support
com
github.mikephil.charting
google.gson
querynow
electric
ElectricActivity.class
gas
GasActivity.class
http
PriceService.class
model
ElectricModel.class
ElectricRowModel.class
GasCityModel.class
GasCompanyModel.class
GasModel.class
OilDatePrice.class
OilHistory.class
OilModel.class
OilPriceModel.class
WaterModel.class
WaterRow.class
oil
water
BuildConfig.class
ForegroundLinearLayout.class
ForegroundWrapper.class
MainActivity.class
MyMarkerView.class
R.class
squareup.picasso
org.jsoup
proguard.snippets
retrofit
rx
ElectricModel.class
ForegroundLinearLayout.class
ForegroundWrapper.class
MainActivity.class
rivate void initView()
this.mOilBtn = ((Button)findViewById(2131427391));
this.mOilBtn.setOnClickListener(new View.OnClickListener()
{
public void onClick(View paramAnonymousView)
{
MainActivity.this.startActivity(new Intent(MainActivity.this, OilActivity
}
});
this.mElectricBtn = ((Button)findViewById(2131427392));
this.mElectricBtn.setOnClickListener(new View.OnClickListener()
{
public void onClick(View paramAnonymousView)
{
MainActivity.this.startActivity(new Intent(MainActivity.this, ElectricAct
}
});
this.mGasBtn = ((Button)findViewById(2131427393));
this.mGasBtn.setOnClickListener(new View.OnClickListener()
{
public void onClick(View paramAnonymousView)
{
MainActivity.this.startActivity(new Intent(MainActivity.this, GasActivity
}
});
this.mWater = ((Button)findViewById(2131427394));
this.mWater.setOnClickListener(new View.OnClickListener()



ProGuard

ProGuard manual

```
}
buildTypes {
    release {
        minifyEnabled true
        proguardFiles getDefaultProguardFile('proguard-android.txt'), 'proguard-rules.pro'
    }
}
```

```
-keep class org.apache.commons.** {
    *;
}

-keep class android.support.** {
    *;
}

-keep class com.actionbarsherlock.** {
    *;
}

-keep class com.foxykeep.datadroid.** {
    *;
}


-keep class com.facebook.** { *; }
-keepattributes Signature


-keep class com.jeremyfeinstein.slidingmenu.lib.** {
    *;
}

-keep class com.viewpagerindicator.** {
    *;
}

-keep class com.nostra13.universalimageloader.** {
    *;
}

-keep class com.tonicartos.widget.stickygridheaders.** {
    *;
}
```

Retrofit

```
  <version>2.3.0</version>
</dependency>
```

GRADLE

```
compile 'com.squareup.retrofit2:retrofit:2.3.0'
```

Retrofit requires at minimum Java 7 or Android 2.3.

PROGUARD

If you are using ProGuard in your project add the following lines to your configuration:

```
# Platform calls Class.forName on types which do not exist on Android to determine platform.
-dontnote retrofit2.Platform
# Platform used when running on Java 8 VMs. Will not be used at runtime.
-dontwarn retrofit2.Platform$Java8
# Retain generic type information for use by reflection by converters and adapters.
-keepattributes Signature
# Retain declared checked exceptions for use by a Proxy instance.
-keepattributes Exceptions
```

Retrofit uses Okio under the hood, so you may want to look at its ProGuard rules as well.


A THEORY OF JUSTICE Rawls
The Cambridge Companion to RAWLS
Objective Knowledge
GREAT DIALOGUES
FRIEDRICH NIETZSCHE
Nietzsche Beyond Good and Evil
Academic Repression
STEAL THIS UNIVERSITY
THE MODERN SCHOOL MOVEMENT
DEMOCRACY AND EDUCATION
GLOBALIZATION
CAPITALISM & ITS ALTERNATIVES
ACCUMULATION
FREEDOM
HOW THE ECONOMY WAS LOST
ROBERTS
WE BUILD THE ROAD AS WE TRAVEL
Mondragon, A Cooperative Social System
MARX • CAPITAL: VOLUME I
NAOMI KLEIN
THE SHOCK DOCTRINE
FIELD GUIDE TO THE U.S. ECONOMY
NEOLIBERALISM
THE ABCs OF POLITICAL ECONOMY
DEBT
THE FIRST 5,000 YEARS
DAVID GRAEBER
GREAT DEBATES IN ECONOMICS
ROBIN HAHNEL
DOLLARS SENSE
CAPITALISM AND ITS ECONOMICS
Reading Capital Politically
THE EVOLUTION OF ECONOMIC THOUGHT
A GUIDE TO MARX'S CAPITAL
MONOPOLY CAPITAL
PARECON MICHAEL ALBERT
THE Marx-Engels READER
The Permanent Revolution
MARXIST THOUGHT
AUTONOMY, SOLIDARITY, POSSIBILITY
THE EGO AND HIS OWN
An Anarchist FAQ
VISION ON FIRE
Wobblies & Zapatistas
ANARCHISM
RACE, IDENTITY & CITIZENSHIP
VALUE WARS The Global Market Versus the Life
PHILOSOPHY BEFORE
MacKinnon
ETHIC
Alfred North W
KIERKEGAARD
IMMANUEL KANT
Kant Prolegomena to

android-proguard-snippets

Example Proguard configurations for common Android libraries.

This project assumes that your ProGuard configuration is based off of the latest official proguard-android.txt config as shown below. Each library configuration should only be the rules required for that specific library, not a complete Android ProGuard configuration. The various library configurations are combined by the Gradle build system. The library rules should be universal, any app specific rules (such as preserving model classes) should be added in a custom `proguard-project.pro` file.

Request additional libraries through issues. Pull requests are welcome.

- ActionBarSherlock 4.4.0
- ActiveAndroid
- Adjust
- Amazon Web Services 1.6.x / 1.7.x
- Amazon Web Services 2.1.x
- AndroidAnnotations
- android-gif-drawable
- Apache Avro
- Alibaba Fastjson
- Butterknife 5.1.2
- Crashlytics 1.+ / 2.+
- Crittercism
- EventBus 2.0.2
- Facebook 3.2.0
- Facebook Conceal
- Facebook Stetho
- Facebook Fresco
- Flurry 3.4.0
- Google Analytics 3.0+
- Google Guava
- Google Play Services 4.3.23
- GreenDao 1.3.x
- GSON 2.2.4
- Jackson 2.x
- Joda-Convert 1.6
- Joda-Time 2.3
- Jsoup
- LoganSquare
- New Relic
- Parse

Is it enough

?

```
protected void onCreate(Bundle paramBundle)
{
  super.onCreate(paramBundle);
  setContentView(2130903065);
  this.b = ((TextView)findViewById(2131427397));
  this.b.setText("123456");
  this.c = ((TextView)findViewById(2131427399));
  this.d = ((LinearLayout)findViewById(2131427396));
  this.e = ((LinearLayout)findViewById(2131427398));
  ((PriceService)new RestAdapter.Builder().setEndpoint("http://price.nat.gov.tw").setErrorHandler(new E
  {
    public Throwable handleError(RetrofitError paramAnonymousRetrofitError)
    {
      Log.d("Ted", "cause " + paramAnonymousRetrofitError.getMessage());
      return null;
    }
  }).build().create(PriceService.class)).getPower("E4").a(AndroidSchedulers.a()).a(new Action1()
  {
    public void a(Response paramAnonymousResponse)
    {
      int i = 0;
      Object localObject1 = new StringBuilder();
      for (;;)
      {
        try
        {
          paramAnonymousResponse = new BufferedReader(new InputStreamReader(paramAnonymousResponse.getB
        }
        catch (IOException paramAnonymousResponse)
        {
          Object localObject2;
```

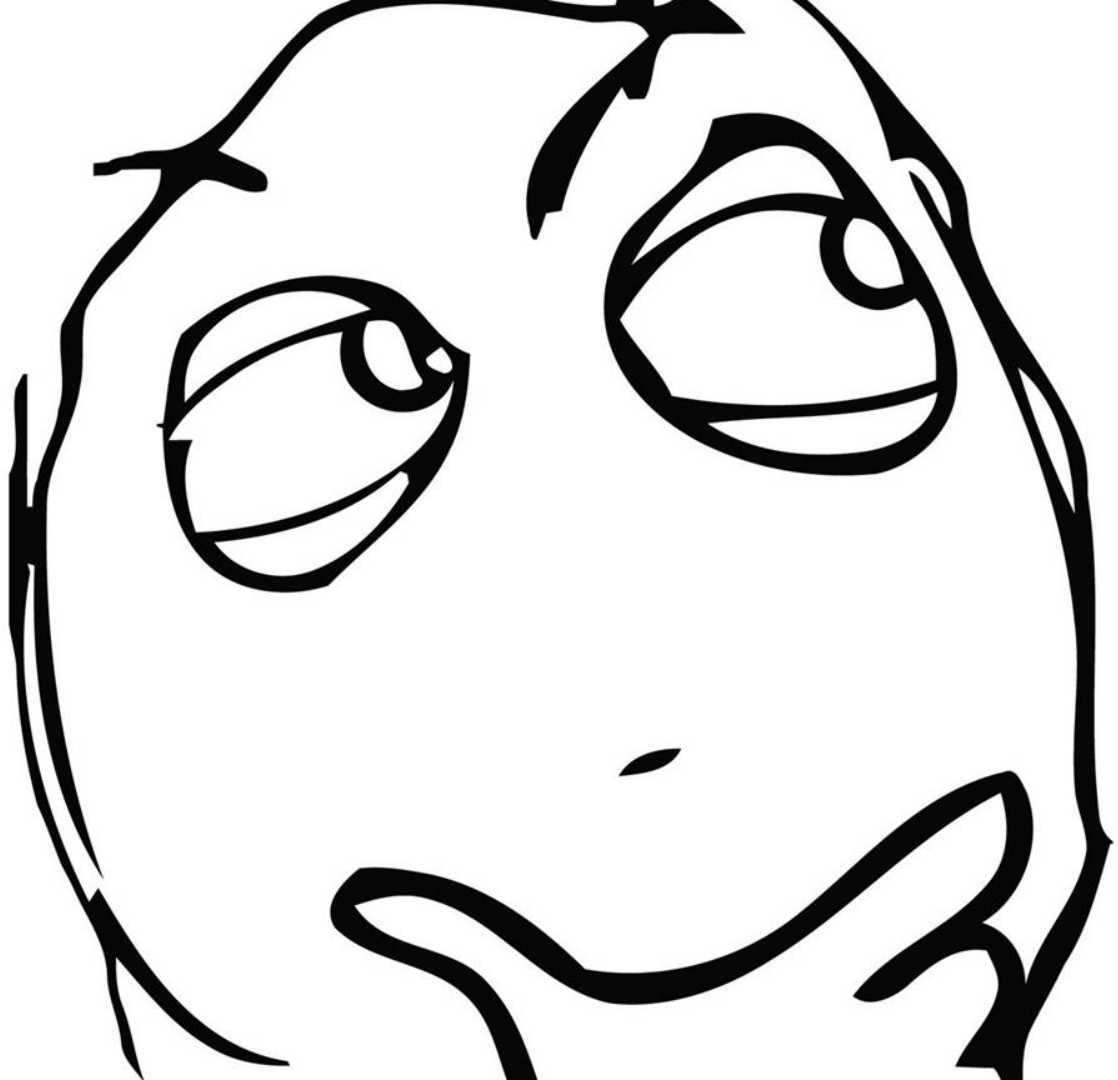
```java
public class ElectricActivity extends Activity {
    private ArrayList<ElectricModel> mList = new ArrayList<>();
    private TextView mFirstTitle;
    private TextView mSecTitle;
    private LinearLayout mFirstRoot;
    private LinearLayout mSecRoot;
    private static final String KEY = "123456";

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.electric_layout);
        mFirstTitle = (TextView) findViewById(R.id.first_title);
        mFirstTitle.setText(KEY);
        mSecTitle = (TextView) findViewById(R.id.sec_title);

        mFirstRoot = (LinearLayout) findViewById(R.id.first_Root);
        mSecRoot = (LinearLayout) findViewById(R.id.sec_Root);

        RestAdapter restAdapter = new RestAdapter.Builder()
                .setEndpoint("http://price.nat.gov.tw")
                .setErrorHandler((cause) -> {
                        Log.d("Ted", "cause " + cause.getMessage());
                        return null;
                })
```

```
protected void onCreate(Bundle paramBundle)
{
  super.onCreate(paramBundle);
  setContentView(2130903065);
  this.b = ((TextView)findViewById(2131427397));
  this.b.setText("123456");
  this.c = ((TextView)findViewById(2131427399));
  this.d = ((LinearLayout)findViewById(2131427396));
  this.e = ((LinearLayout)findViewById(2131427398));
  ((PriceService)new RestAdapter.Builder().setEndpoint("http://price.nat.gov.tw").setErrorHandler(new E
  {
    public Throwable handleError(RetrofitError paramAnonymousRetrofitError)
    {
      Log.d("Ted", "cause " + paramAnonymousRetrofitError.getMessage());
      return null;
    }
  }).build().create(PriceService.class)).getPower("E4").a(AndroidSchedulers.a()).a(new Action1()
  {
    public void a(Response paramAnonymousResponse)
    {
      int i = 0;
      Object localObject1 = new StringBuilder();
      for (;;)
      {
        try
        {
          paramAnonymousResponse = new BufferedReader(new InputStreamReader(paramAnonymousResponse.getE
        }
        catch (IOException paramAnonymousResponse)
```

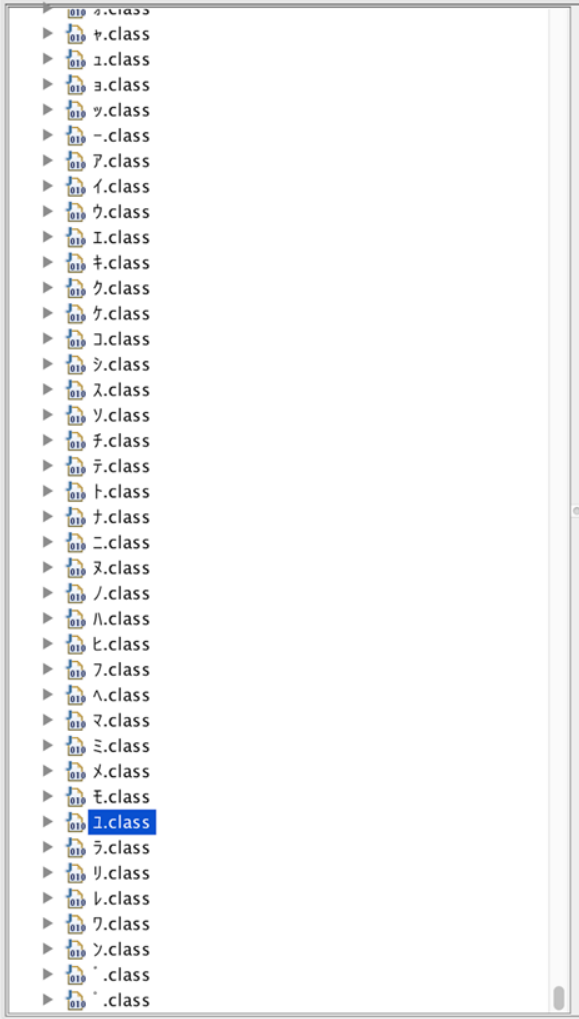
```
new CipherKey(1234, 456, 789);
```

```
paramAnonymousResponse.printStackTrace();
paramAnonymousResponse = new CipherKey(1234, 456, 789);
Log.d("Ted", "key " + paramAnonymousResponse);
```


PASSWORD




DexGuard
Get your quote




```
package o;

import android.content.DialogInterface;
import android.content.DialogInterface.OnClickListener;

final class \uFF95
  implements DialogInterface.OnClickListener
{
  \uFF95(\uFF7F param\uFF7F) {}

  public final void onClick(DialogInterface paramDialogInterface, int paramInt)
  {
    paramDialogInterface.dismiss();
    paramDialogInterface = \u2148.\u0971();
    String str = \uFF84.\u02CA\u0971(this.\u02CA.\u02CF);
    if (paramDialogInterface.\u02CE != null) {
      paramDialogInterface.\u02CE.\u02CE(str);
    }
  }
}
```

Dexguard

- EncryptStrings

```
new CipherKey(1234, 456, 789);
```

```
false;
new \uFBE9((byte[])\u02C9.coN.\u02CA("o.¯").getField("´").get(null)
new \uFB54(3);
```

Is It Safe?


YES!!

App安全檢測結果總覽

威脅	風險等級	檢測結果
Java程式碼反編譯風險	高	存在風險
Java程式動態偵錯風險	低	安全
元件導出風險	中	存在風險
敏感函數呼叫風險	中	存在風險
偵錯日誌函數呼叫風險	高	存在風險
原生函式庫動態偵錯風險	中	存在風險
程式碼動態植入風險	高	存在風險
Apk修改/重新打包風險	高	安全
數位憑證竊取風險	高	安全
App資料任意備份風險	中	存在風險
HTTP傳輸資料風險	高	安全
Webview儲存密碼遭竊風險	中	存在風險
內網測試資訊殘留漏洞	低	安全
下載任意apk漏洞	中	安全
HTTPS未檢查伺服器憑證漏洞	中	安全
Content provider資料洩露漏洞	高	安全
資料庫SQL指令植入漏洞	高	安全
App內部檔案可任意讀寫漏洞	中	安全
Webview遠端程式碼執行漏洞	高	存在漏洞
Webview繞過憑證校驗漏洞	低	存在漏洞

檢測目的	檢測App是否存在資料被任意備份的風險。
風險等級	中
威脅	使用Android API 8以上的Android系統為app提供了應用程式資料的備份和還原功能，此功能由AndroidMainfest.xml檔中的allowBackup值控制，預設值為true。如果開發者沒有特別將該屬性設為false，攻擊者可以使用adb backup和adb restore對App的應用程式資料進行備份和還原，然後對應用程式資料進行分析，可能會獲得一些機敏資訊，如使用者帳號、密碼、手機號碼、CVN、交易密碼、token、交易記錄等。
檢測結果	存在風險
檢測結果描述	此App中的應用程式資料存在被任意備份的風險
檢測詳細資訊	android:allowBackup=true

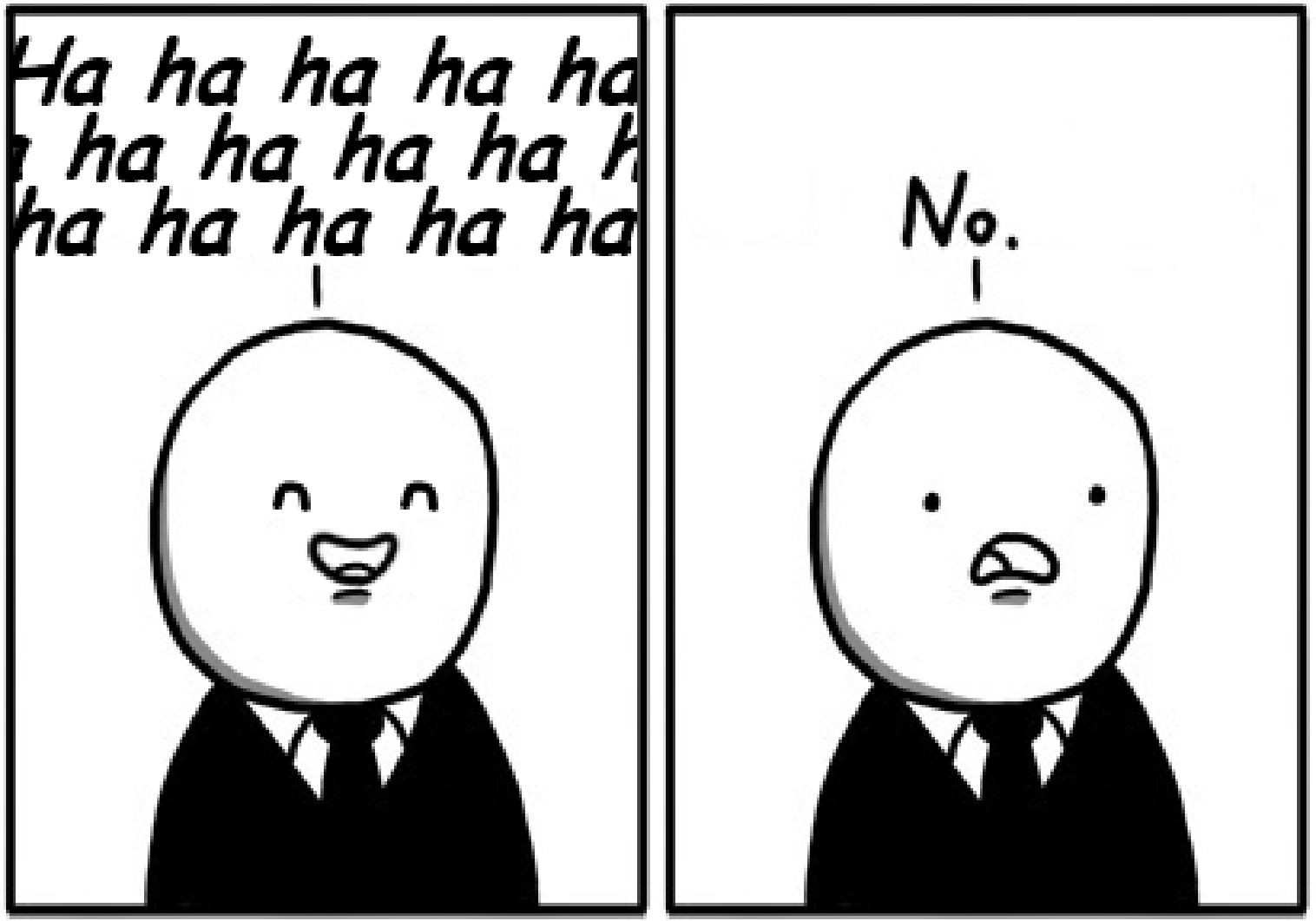
檢測目的	檢測App是否使用了未加密儲存帳號密碼的Webview元件。
風險等級	中
威脅	Android的Webview元件中預設開啟提示使用者是否儲存密碼的功能，如果使用者選擇儲存，則帳號和密碼將被以明文型式儲存到此app目錄databases/webview.db中。明文儲存的帳號和密碼，可能會被其他惡意程式存取，進而竊取使用者記錄的帳號、密碼。
檢測結果	存在風險
檢測結果描述	此App Webview元件中自動儲存密碼功能未被關閉，密碼有遭竊取的風險。
檢測詳細資訊	["o.bR.onCreateView:(Landroid/view/LayoutInflater;Landroid/view/ViewGroup;Landroid/os/Bundle;)Landroid/view/View;"]
解決方案	設定WebView.getSettings().setSavePassword(false)以關閉webview元件的儲存密碼功能。

檢測目的	檢測App的webview元件是否在https網頁憑證錯誤時依舊繼續載入網頁的漏洞。
風險等級	低
威脅	客戶端的Webview元件存取HTTPS協定加密的url時，如果伺服器憑證錯誤時，客戶端應該拒絕繼續載入頁面。但如果在WebView的onReceivedSslError()函數裡執行handler.proceed()函數，客戶端可以繞過憑證校驗錯誤繼續存取此非法URL。如此將會導致“中間人攻擊”，攻擊者可以冒充伺服器與銀行客戶端程式傳輸資料，同時又冒充銀行客戶端程式與銀行伺服器傳輸資料，然後在中間轉發資料的時候，趁機竊取使用者重要資料。
檢測結果	存在漏洞
檢測結果描述	此App的webview元件存在忽略憑證校驗錯誤的漏洞。
檢測詳細資訊	["o.bR.`.onReceivedSslError:(Landroid/webkit/WebView;Landroid/webkit/SslErrorHandler;Landroid/net/http/SslError;)V"]
解決方案	禁止在onReceivedSslError()函數裡執行handler.proceed()函數。

```
HostnameVerifier hostnameVerifier = new HostnameVerifier() {
    @Override
    public boolean verify(String hostname, SSLSession session) {
        return true;
    }
};
```

```
TrustManager[] trustAllCerts = new TrustManager[]{new X509TrustManager() {
    public X509Certificate[] getAcceptedIssuers() {
        return new X509Certificate[]{};
    }

    public void checkClientTrusted(X509Certificate[] certs, String authType) {
    }

    public void checkServerTrusted(X509Certificate[] certs, String authType) {
    }
}};

SSLContext sc = SSLContext.getInstance("SSL");
sc.init(null, trustAllCerts, new java.security.SecureRandom());

builder.sslSocketFactory(sc.getSocketFactory());
```

這樣應該夠了吧?


Ha ha ha ha ha
ha ha ha ha h
ha ha ha ha ha
No.

future

1. sign app 安全防護
2. 上架 app 安全防護
3. android N 之後的network security config
4. 2017 google io security 功能

```
signingConfigs {
    config {
        keyAlias 'android'
        keyPassword '123456'
        storeFile file('/Users/tedliang/Bitbucket/QueryNow/release.jks')
        storePassword '123456'
    }
}
```

gradle.properies

```
ANDROID_PASSWORD=123456
```

build.gradle

```
signingConfigs {
    release {
        keyAlias 'android'
        keyPassword ANDROID_PASSWORD
        storeFile file('/Users/tedliang/Bitbucket/QueryNow/release.jks')
        storePassword ANDROID_PASSWORD
    }
}
```

自動上架程式

p12 控管

應該只有特定的機器可以取得

CI/CD server

你的p12 權限是什麼

絕對不要設定成最大權限

調整成你需要的權限

network security config

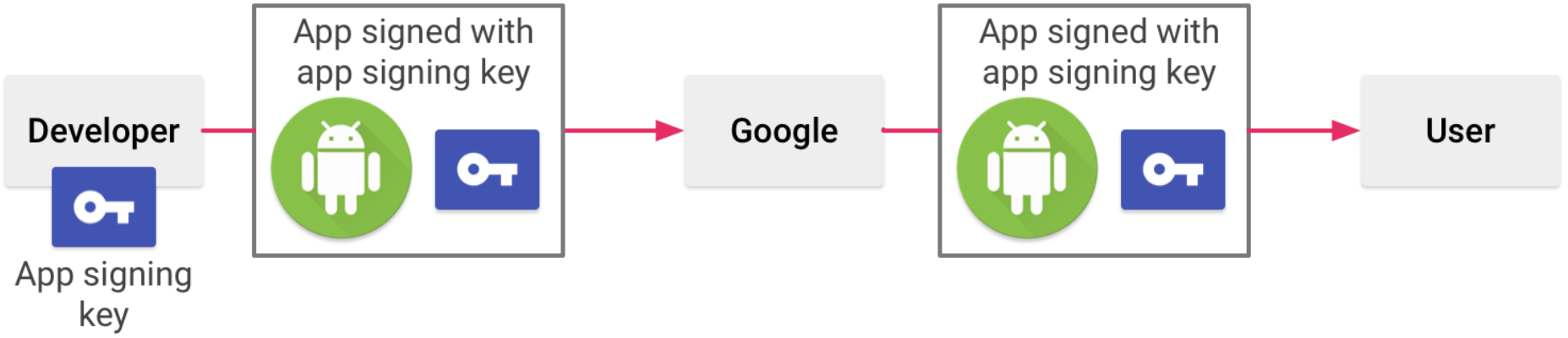
```xml
<?xml version="1.0" encoding="utf-8"?>
<network-security-config>
    <base-config cleartextTrafficPermitted="false" />
    <domain-config cleartextTrafficPermitted="true">
        <domain includeSubdomains="true">localhost</domain>
    </domain-config>
</network-security-config>
```

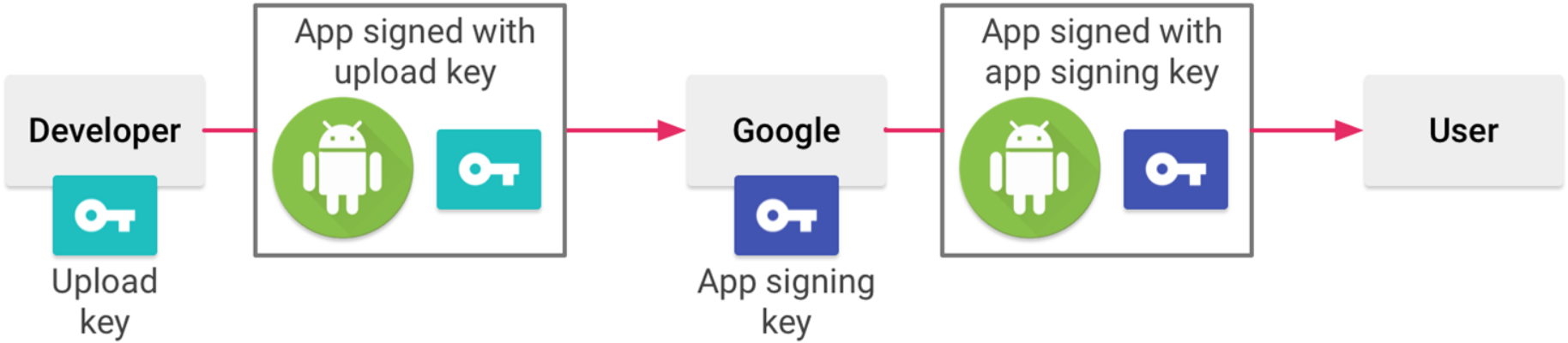
Custom trust anchors: Customize which Certificate Authorities (CA) are trusted for an app's secure connections. For example, trusting particular self-signed certificates or restricting the set of public CAs that the app trusts.

Debug-only overrides: Safely debug secure connections in an app without added risk to the installed base.

Cleartext traffic opt-out: Protect apps from accidental usage of cleartext traffic.

Certificate pinning: Restrict an app's secure connection to particular certificates.

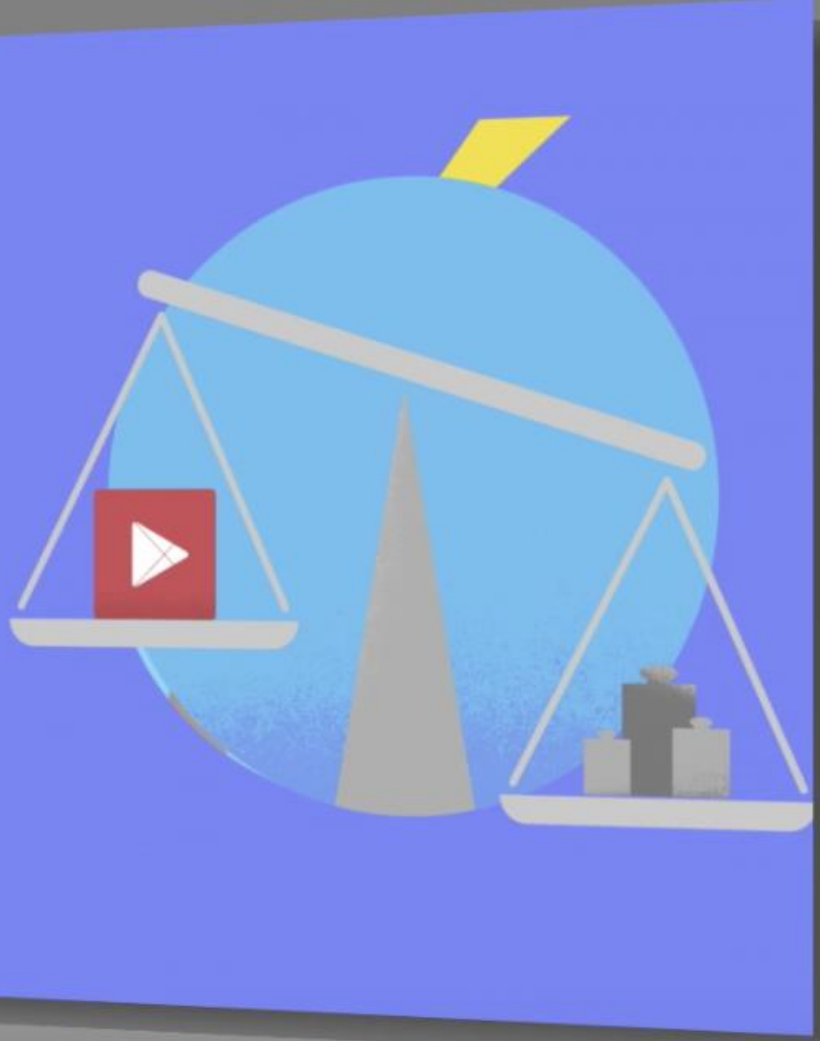

Developer
App signing key
App signed with app signing key
Google
App signed with app signing key
User


Developer
Upload key
App signed with upload key
Google
App signing key
App signed with app signing key
User

App signing + optimizations

We know developers care about APK size, particularly those targeting emerging markets.

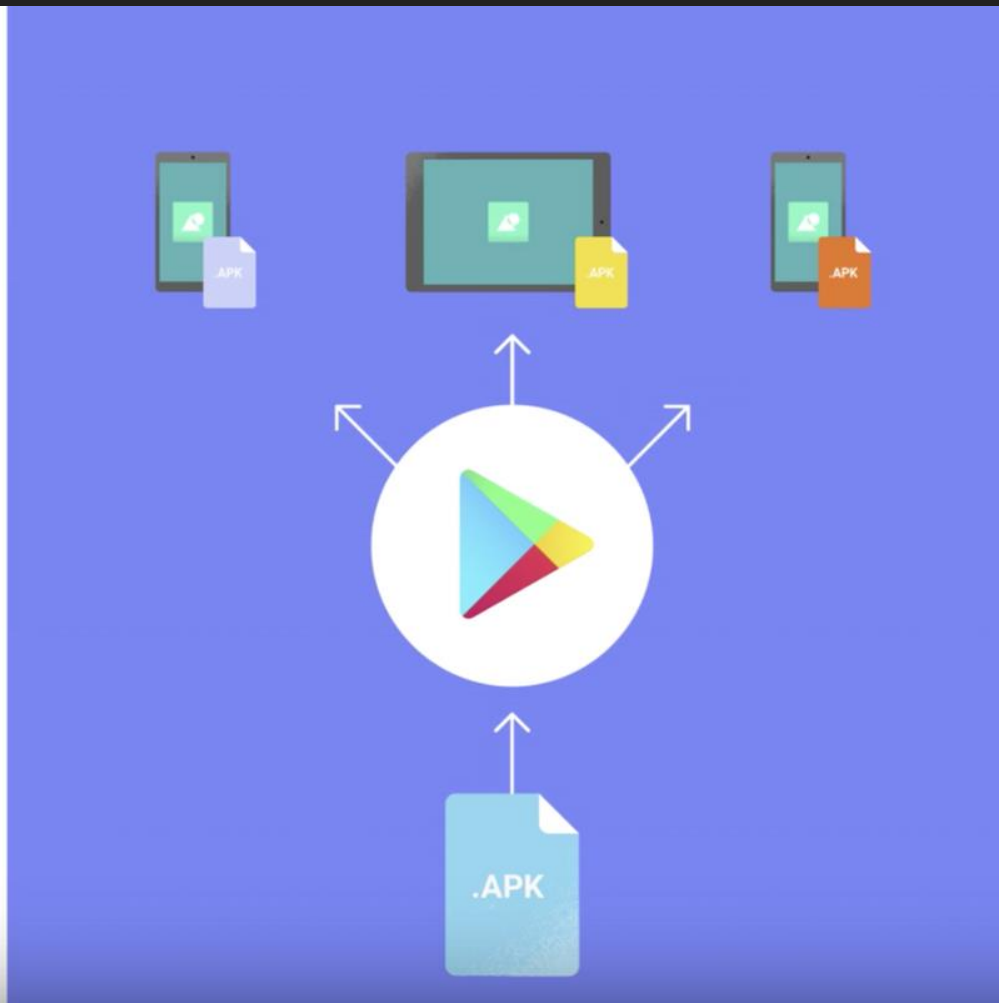
With app optimizations, Play will be able to deliver an APK optimized for screen density and native architecture.




APK
Other Stuff
(dex files, strings, assets)
Native Libraries (.so)
armeabi_v7a
x86
armeabi
mips
Drawables (.png, .jpg, ...)
xxxhdpi
xxhdpi
xhdpi
hdpi
mdpi
ldpi
Required
Not required

Auto multiple-APK

With this app optimization, we deliver just what is needed by each device.



資訊主頁
統計資訊
Android Vitals
開發工具
版本管理
版本資訊主頁
應用程式版本
Android 免安裝應用程式
成果庫
裝置目錄
應用程式登錄
正式發佈前測試報告
商店發佈
商店資訊
商店資訊實驗功能
定價與發佈
內容分級
應用程式內商品
翻譯服務

在你使用上傳金鑰簽署應用程式後，Google 會進行驗證並移除上傳金鑰簽章。最後，Google 會使用你提供的原始應用程式簽署金鑰重新簽署應用程式，再將應用程式提供給使用者。

重要須知：不論註冊前後，為使用者提供的應用程式都是使用同一個應用程式簽署金鑰進行簽署。

匯出、加密及上傳你的應用程式簽署私密金鑰。
如果你不是將私密金鑰儲存在 Java Keystore，請按照進階操作說明進行。

1. 下載 PEPK 工具，以匯出並加密處理你的私密金鑰。

   PEPK 工具

2. 使用下方的指令執行工具，匯出並加密處理你的私密金鑰。請務必取代以粗體醒目顯示的引數。然後在系統提示時輸入你的金鑰庫和金鑰密碼。

   ```
   [redacted]
   ```

3. 上傳你的加密應用程式簽署私密金鑰。

   應用程式簽署私密金鑰

產生並註冊新的上傳公開金鑰

4. 產生新的上傳金鑰。請按照這些操作說明產生新的金鑰。
5. 為新產生的上傳金鑰匯出憑證 (PEM 格式)。請務必取代以粗體醒目顯示的引數。

   ```
   [redacted]
   ```

6. 上傳你的上傳金鑰憑證，向 Google 註冊該憑證。

   上傳公開金鑰憑證

註冊使用 Google Play 應用程式簽署服務
點擊 [註冊] 會產生下列影響：

- 在 Google Play 應用程式簽署服務中註冊應用程式後便無法取消。
- 你的應用程式簽署金鑰會轉移至 Google。
- 註冊新的上傳金鑰後，系統會利用該金鑰處理後續所有 APK 上傳作業

注意：如果你的應用程式含有嵌入的 Android Wear APK (適用於 Android Wear 1.0)，則必須繼續使用原本的應用程式簽署金鑰簽署嵌入的 APK。

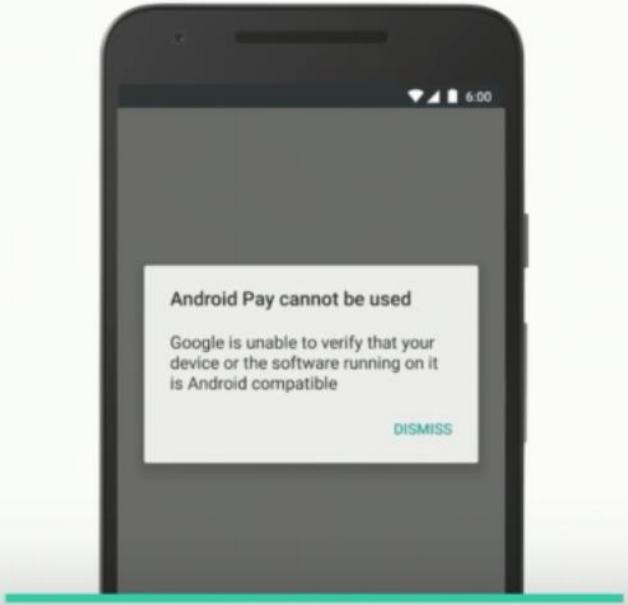
註冊

Verify Apps API

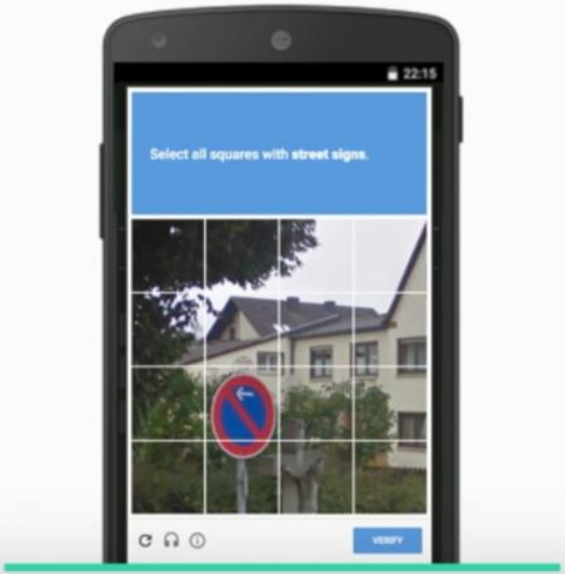
Query for the state of Verify Apps, and any harmful apps installed

```
isVerifyAppsEnabled()

enableVerifyApps()

listHarmfulApps()
```

SafetyNet attestation



reCAPTCHA for Android



last and the
most important

Summary

- 沒有絕對安全的處理方式
  - 不管是dexguard & 各種加密演算法都只是增加破解難度而已
- 如果可以不要把重要資料存在local
- 如果可以不要把重要資料利用網路傳輸
- 如果要盡可能安全
  - 可以偵測root device
    - root 就不能用
  - SafetyNet api