Безопасность Android приложений

2020 (c) Лукьяненко Дмитрий

Обо мне

- Бывший Android разработчик (TUT.by, Yandex, Aimatter)
- Поиск уязвимостей - 2 года хобби + 2 года основная работа
- CVE-2016-1656, CVE-2016-2458, CVE-2017-0414, CVE-2017-0703... (>10)
- TOP 20 Google BugHunters https://bughunter.withgoogle.com/rank/hof (2020)
- TOP 5 Facebook BugHunters https://web.facebook.com/whitehat/thanks/ (2019)

https://www.vulnano.com

Содержание

1. Подпись приложения
2. Permissions
3. Permissions в ContentProvider
4. Передача файлов
5. SSL Pinning
6. Exported компоненты
7. Неявный Intent
8. WebView
9. Доверие к параметрам

Подпись приложения

Подпись приложения

1. Сгенерировать сертификат
2. Держать в секрете
3. Подписать .apk файл

Подробнее
https://developer.android.com/studio/publish/app-signing

Подпись приложения

Сертификат

1. Приложение -> подпись сертификата
2. Одинаковый сертификат -> одинаковые подписи
3. Одинаковые подписи -> “доверенное” взаимодействие

Подпись приложения: PackageManager

Получение подписи

- getPackageInfo
- checkSignatures

Подробнее
https://developer.android.com/reference/android/content/pm/PackageManager

Permissions

Permissions: объявление

```
<permission
    android:name="com.facebook.orca.permission.CROSS_PROCESS_BROADCAST_MANAGER"
    android:protectionLevel="signature"/>

<uses-permission
    android:name="com.facebook.orca.permission.CROSS_PROCESS_BROADCAST_MANAGER"/>
```

https://developer.android.com/guide/topics/permissions/defining

Permissions: android:protectionLevel

- normal (default)
- dangerous
- signature
- signatureOrSystem

https://developer.android.com/guide/topics/manifest/permission-element#plevel

Permissions: AndroidManifest.xml

```
<uses-permission android:name="android.permission.READ_CONTACTS" />

<uses-permission android:name="android.permission.CAMERA" />

<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

Permissions: проверка наличия

```
if (checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
    requestPermissions(new String[]{Manifest.permission.CAMERA}, requestCode: 100);
} else {
    disableSelf();
}
```

Permissions: запрос

```
if (checkSelfPermission(Manifest.permission.CAMERA) != PackageManager.PERMISSION_GRANTED) {
    requestPermissions(new String[]{Manifest.permission.CAMERA}, requestCode: 100);
} else {
    disableSelf();
}
```

Permissions: проверка результата

```
@Override
public void onRequestPermissionsResult(int requestCode,
        String[] permissions, int[] grantResults) {
    switch (requestCode) {
        case MY_PERMISSIONS_REQUEST_READ_CONTACTS: {
            // If request is cancelled, the result arrays are empty.
            if (grantResults.length > 0
                && grantResults[0] == PackageManager.PERMISSION_GRANTED) {
                // permission was granted, yay! Do the
                // contacts-related task you need to do.
            } else {
                // permission denied, boo! Disable the
```

https://developer.android.com/training/permissions/requesting

Permissions: пример

```
<activity android:configChanges="keyboardHidden|orientation|screenSize"
    android:enabled="false"
    android:excludeFromRecents="true"
    android:name="com.facebook.messenger.intents.SecureSameTaskIntentHandlerActivity"
    android:permission="com.facebook.permission.prod.FB_APP_COMMUNICATION"
    android:theme="@style/(name removed)_APKTOOL_DUPLICATENAME_0x7f12001b">
    <intent-filter>
        <action android:name="com.facebook.orca.notify.SECURE_VIEW"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:scheme="fb-messenger-sametask"/>
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.SENDTO"/>
        <category android:name="android.intent.category.DEFAULT"/>
        <data android:scheme="fb-messenger-sametask"/>
    </intent-filter>
</activity>
```

Permissions в ContentProvider

Permissions в ContentProvider

android:grantUriPermissions = true/false

Поддерживается выдача доступа к определенному URI на чтение/запись

Permissions в ContentProvider

android:readPermission=<permission>

android:writePermission=<permission>

Поддерживается выдача доступа к определенному URI на чтение или запись по такому-то разрешению

Подробнее

https://developer.android.com/guide/topics/manifest/provider-element#gprmsn

Permissions в ContentProvider: пример

```
<provider android:authorities="com.whatsapp.w4b.provider.media"
          android:exported="false"
          android:grantUriPermissions="true"
          android:name="com.whatsapp.contentprovider.MediaProvider"/>
```

Permissions в ContentProvider: пример

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.addFlags(Intent.FLAG_GRANT_WRITE_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_PERSISTABLE_URI_PERMISSION);

Uri uri = Uri.parse("content://com.whatsapp.provider.media/item/5");
intent.putExtra(Intent.EXTRA_STREAM, uri);
intent.setType("*/*");

startActivity(intent);
```

Permissions в ContentProvider: пример

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.addFlags(Intent.FLAG_GRANT_WRITE_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_PERSISTABLE_URI_PERMISSION);

Uri uri = Uri.parse("content://com.whatsapp.provider.media/item/5");
intent.putExtra(Intent.EXTRA_STREAM, uri);
intent.setType("*/*");

startActivity(intent);
```

Permissions в ContentProvider: пример

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.addFlags(Intent.FLAG_GRANT_WRITE_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_PERSISTABLE_URI_PERMISSION);

Uri uri = Uri.parse("content://com.whatsapp.provider.media/item/5");
intent.putExtra(Intent.EXTRA_STREAM, uri);
intent.setType("*/*");

startActivity(intent);
```

Permissions в ContentProvider: пример

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.addFlags(Intent.FLAG_GRANT_WRITE_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_PERSISTABLE_URI_PERMISSION);

Uri uri = Uri.parse("content://com.whatsapp.provider.media/item/5");
intent.putExtra(Intent.EXTRA_STREAM, uri);
intent.setType("*/*");

startActivity(intent);
```

Permissions в ContentProvider: пример

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.addFlags(Intent.FLAG_GRANT_WRITE_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_PERSISTABLE_URI_PERMISSION);

Uri uri = Uri.parse("content://com.whatsapp.provider.media/item/5");
intent.putExtra(Intent.EXTRA_STREAM, uri);
intent.setType("*/*");

startActivity(intent);
```

Permissions в ContentProvider: пример

```
Intent intent = new Intent(Intent.ACTION_SEND);
intent.addFlags(Intent.FLAG_GRANT_WRITE_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_READ_URI_PERMISSION);
intent.addFlags(Intent.FLAG_GRANT_PERSISTABLE_URI_PERMISSION);

Uri uri = Uri.parse("content://com.whatsapp.provider.media/item/5");
intent.putExtra(Intent.EXTRA_STREAM, uri);
intent.setType("*/*");

startActivity(intent);
```

Permissions в ContentProvider: зачем?

Безопасно:

- Передача данных (файлов) между приложениями
- Изменение данных между приложениями

Передача файлов

Передача файлов

- /data/data/<package_name> - внутренние данные приложения
- доступ только у этого приложения
- нельзя шарить напрямую file:/// из директории

Передача файлов: FileProvider

Использовать FileProvider для шаринга файлов

- Безопасно
- Готовая реализация

Подробнее
https://developer.android.com/reference/android/support/v4/content/FileProvider

SSL Pinning

Защита приложений в Android: SSL Pinning

- HTTP трафик легко перехватывается
- HTTPS трафик перехватывается если установлен доверенный сертификат
- Сертификат мог быть установлен пользователем ошибочно (вирус) / Подменен сертификат MITM (Man in the middle)

Защита приложений в Android: SSL Pinning

SSL Pinning

- HTTPS
- Проверка сертификатов
- Пропуск трафика только для доверенных сертификатов

Подробнее:

https://developer.android.com/training/articles/security-ssl

Защита приложений в Android: SSL Pinning

SSL Pinning

- HTTPS
- Проверка сертификатов
- Пропуск трафика только для доверенных сертификатов

Подробнее:

https://developer.android.com/training/articles/security-ssl

Exported компоненты

Exported компоненты: AndroidManifest

AndroidManifest.xml

Activity

Service

BroadcastReceiver

ContentProvider

Exported компоненты: явные

```
<activity
        android:exported="false"
        android:name=".browse.BrowseActivity"/>

<activity
        android:exported="true"
        android:name=".social.licenses.LicenseMenuActivity"/>
```

Exported компоненты: неявные

```
<activity
      android:name=".VulnanoAuthActivity">
      <intent-filter android:label="Test share">
            <data android:scheme="content" />
      </intent-filter>
</activity>
```

exported=true

Exported компоненты: проблема

- Выполнение действий от имени пользователя
- Доступ к данными приложения

Exported компоненты: приложение Mail.Ru

Год: 2016

Проблема: доступ к контактам приложения Mail.Ru

```
<provider
    android:authorities="ru.mail.mailbox.contacts"
    android:enabled="true"
    android:exported="true"
    android:name="ru.mail.mailbox.content.contact.ContactsProvider"
    android:syncable="false"/>
```

Exported компоненты: рекомендации

- Явно указывать exported=false

Exported компоненты: рекомендации

- Явно указывать exported=false
- Использовать доступ по android:permission

Exported компоненты: рекомендации

- Явно указывать exported=false
- Использовать доступ по android:permission
- Ручной контроль: getCallingPackage(), Binder.getCallingPid()... и т.п. + проверка на подпись

Неявный Intent

Неявный Intent

intent.getPackageName() == null

intent.getComponent() == null

Перехват

Неявный Intent: проблемы

Запуск Activity

- Подмена Activity (фишинг)
- Перехват информации в intent

Неявный Intent: проблемы

Запуск Activity

- Подмена Activity (фишинг)
- Перехват информации в intent

Отправка broadcast’a

- Перехват информации в Intent

Неявный Intent: рекомендации

- Использовать явные Intent (setPackageName, setClassName)

Неявный Intent: рекомендации

- Использовать явные Intent (setPackageName, setClassName)
- Использовать LocalBroadcastManager для отправки broadcast’ов

Неявный Intent: рекомендации

- Использовать явные Intent (setPackageName, setClassName)
- Использовать LocalBroadcastManager для отправки broadcast’ов
- Отправлять неявные broadcast’ы с permission

WebView

WebView: опасности

- Работа с JavascriptInterface
- Загрузка file://
- Проблемы валидации ссылок

WebView: JavascriptInterface

Мост для вызовов Android кода из JavaScript

Риски:

Вызов методов JavascriptInterface из вредоносного сайта

WebView: JavascriptInterface, Grab

Год: 2018

Автор: @bagipro,
https://hackerone.com/reports/401793

Проблема: Неправильная обработка URI ведет к загрузке стороннего сайта в WebView

```
<script type="text/javascript">
    var data;
    if(window.Android) { // Android
        data = window.Android.getGrabUser();
    }
    else if(window.grabUser) { // iOS
        data = JSON.stringify(window.grabUser);
    }

    if(data) {
        document.write("Stolen data: " + data);
    }
</script>
```

WebView: рекомендации

- JavascriptInterface -> проверять origin текущей страницы

WebView: рекомендации

- JavascriptInterface -> проверять origin текущей страницы
- WebSettings.setAllowFileAccess(false)

WebView: рекомендации

- JavascriptInterface -> проверять origin текущей страницы
- WebSettings.setAllowFileAccess(false)
- WebSettings.setJavascriptEnabled(false)

WebView: рекомендации

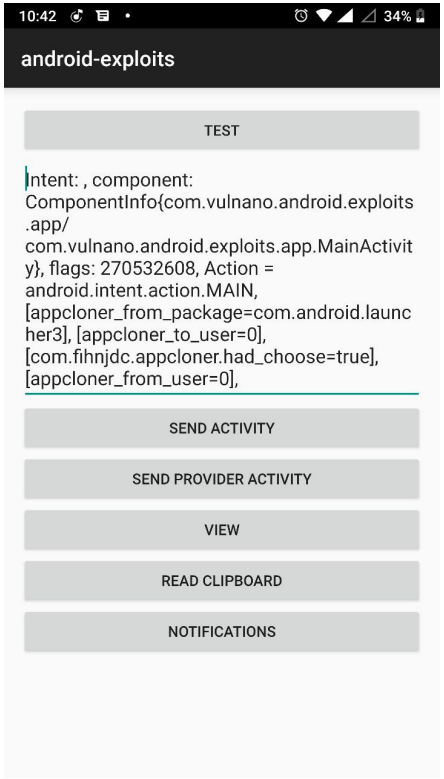
- JavascriptInterface -> проверять origin текущей страницы
- WebSettings.setAllowFileAccess(false)
- WebSettings.setJavascriptEnabled(false)
- Отключить все что не требуется

WebView: рекомендации

- JavascriptInterface -> проверять origin текущей страницы
- WebSettings.setAllowFileAccess(false)
- WebSettings.setJavascriptEnabled(false)
- Отключить все что не требуется
- Избегать доступа к WebView в exported activity

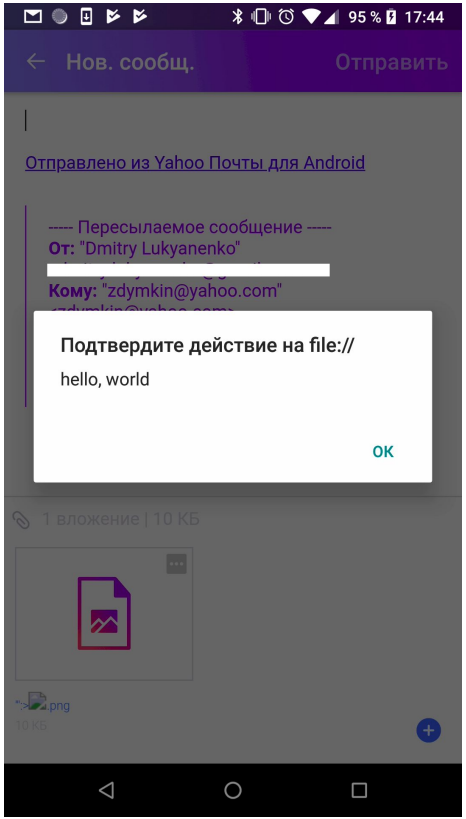
Доверие к параметрам

Параметры



intent.getExtras()

intent.getData()



Параметры

Кому можно доверять?

НИКОМУ! (даже Android)

Параметры: доверие к Android

Кто угодно:

- Чтение/запись любых (даже не exported) ContentProvider
- Версии ОС: Android 8.0, 8.1, 9.0

Параметры: доверие к Android

CVE-2018-9492

Автор: Michał Bednarski

Проблема: Доступ к любым (даже не exported) ContentProvider

Параметры: рекомендации ContentProvider

Нет доверия к Android:

- Проверка источника вызова через Binder.getCallingUid()

Параметры: запуск чужих Intent’ов

```
Intent nextIntent = (Intent)intent.getParcelableExtra("nextIntent") ;

startActivity(nextIntent);
```

Параметры: запуск чужих Intent’ов

```
Intent nextIntent = (Intent)intent.getParcelableExtra("nextIntent") ;

startActivity(nextIntent);
```

Параметры: запуск чужих Intent’ов

```
Intent nextIntent = (Intent)intent.getParcelableExtra("nextIntent") ;

startActivity(nextIntent);
```

- Запуск non exported компонент (например с WebView)
- Выдача доступа к android:grantUri=”true” ContentProvider’ам

Параметры: запуск чужих Intent’ов Uber

Год: 2016

Проблема: запуск Intent передаваемого из атакующего приложения ведет к запуску Activity с WebView и краже некоторых файлов

Параметры: запуск чужих Intent’ов рекомендации

- Не запускать Intent’ы из параметров без модификации

Параметры: запуск чужих Intent’ов рекомендации

- Не запускать Intent’ы из параметров без модификации
- Не клонировать текущий intent для запуска без модификации

Параметры: запуск чужих Intent’ов рекомендации

- Не запускать Intent’ы из параметров без модификации
- Не клонировать текущий intent для запуска без модификации
- Создавать Intent для запуска самостоятельно заполняя его

Параметры: запуск чужих Intent’ов рекомендации

- Не запускать Intent’ы из параметров без модификации
- Не клонировать текущий intent для запуска без модификации
- Создавать Intent для запуска самостоятельно заполняя его
- Явные Intent’ы

Практика apktool

Практика apktool

Установить приложения на телефон

- Facebook (com.facebook.katana)
- Vkontakte (com.vkontakte.android)
- Мой МТС (by.mts.client)

На PC

- apktool
- adb tools
- ByteCodeViewer (https://bytecodeviewer.com/)

Практика apktool

adb shell pm list packages | grep <packageName>

adb shell pull <path_to_package>

apktool d <file>.apk

Контакты

Лукьяненко Дмитрий
www.vulnano.com
vulnano.research@gmail.com
<версия не для публичного распространения>