

# DevFest

## Проблемы безопасности Android-приложений или как обезопасить ваше приложение



Сосновский Юрий

Telegram @JurySosnovsky vk.com/sosnoffsky



# Модели угроз

# Модели угроз

## Локальная атака

1. Измененное приложение
2. Вредоносные приложение



# Модели угроз

## Удаленные атаки

1. С другого сервера
2. С другого клиента
3. В канале связи



# Модели угроз

## Доступ к устройству

1. Кража или потеря
2. Кратковременный доступ

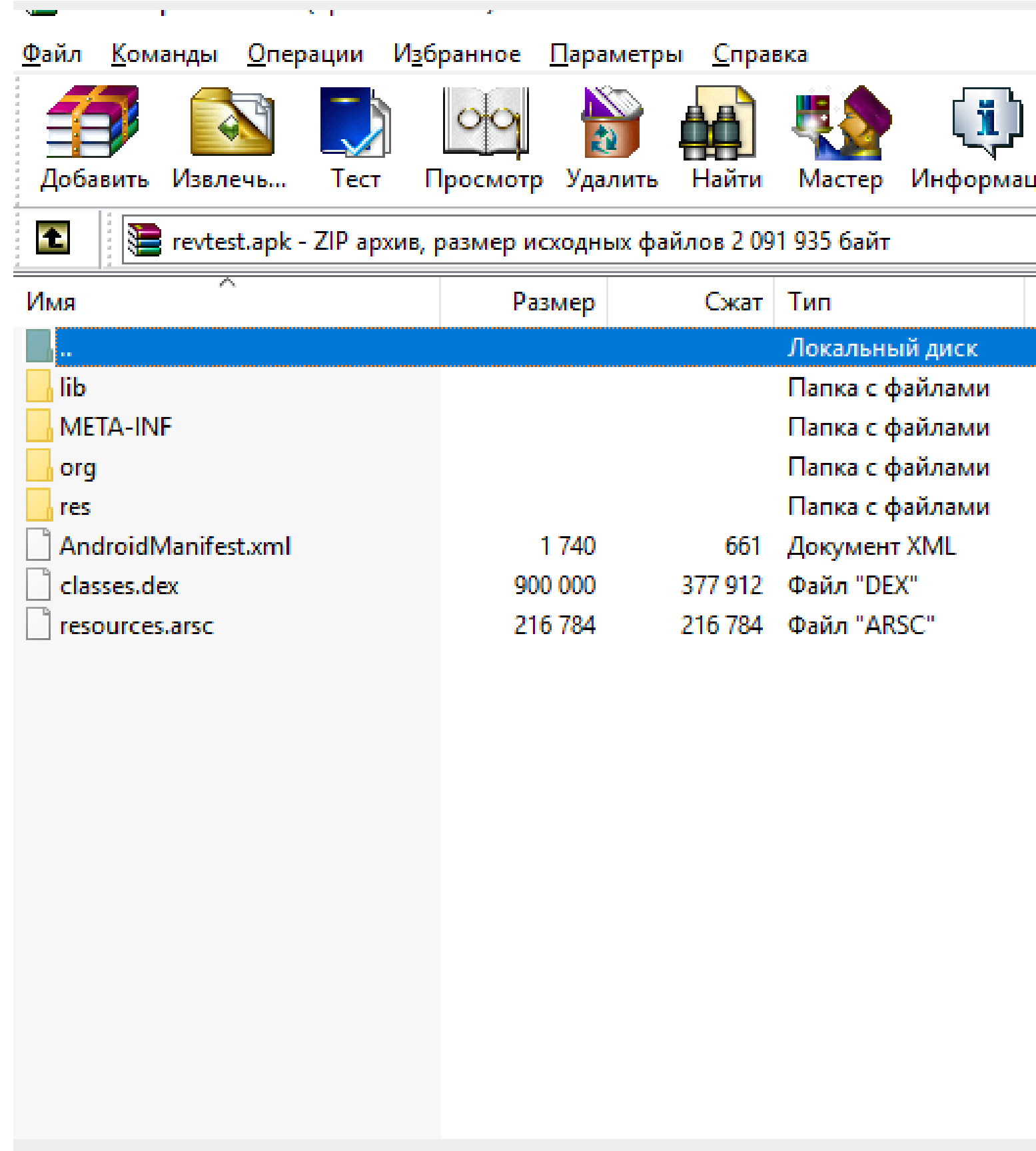


# Локальные атаки

# Локальные атаки

## Устройство APK файла

1. Обычный zip-архив
2. DEX – файлы с байткодом
3. ARSC – ресурсы
4. lib – системные библиотеки
5. Прочее (манифесты, конфигурации)



# Локальные атаки

## Декомпиляция APK

1. Получение APK (Google Play или с устройства)
2. Извлечение ресурсов
3. Извлечение байткода
4. ...
5. PROFIT!





## Files

- a\$.class
- a\$.class
- a\$.class
- a\$.class
- a\$.class
- a\$.class
- a.class

v4

v7

## Decoded Resources

res

anim

anim-v21

animator-v21

color

color-v11

color-v23

drawable

drawable-anydpi-v21

drawable-hdpi-v4

drawable-ldrtl-hdpi-v17

drawable-ldrtl-mdpi-v17

drawable-ldrtl-xhdpi-v17

drawable-ldrtl-xxhdpi-v17

Quick file search (no file extension)

☐ Exact

## Search

Search from All\_Classes

LDC

Search String:

☐ Exact

Search

Root

## Work Space

☐ Exact

Smali Decompiler - Editable: false

```
28
29
30 # direct methods
31 .method public constructor <init>(Landroid/view/View;)V
32     .registers 3
33
34     invoke-direct {p0, p1}, Landroid/support/v7/widget/
35
36     const v0, 0x7f0d007d
37
38     invoke-virtual {p1, v0}, Landroid/view/View;->find<
39
40     move-result-object v0
41
42     check-cast v0, Landroid/widget/LinearLayout;
43
44     iput-object v0, p0, Landroid/widget/LinearLayout;
45
46     const v0, 0x7f0d007e
47
48     invoke-virtual {p1, v0}, Landroid/view/View;->find<
49
50     move-result-object v0
51
52     check-cast v0, Landroid/widget/TextView;
53
54     iput-object v0, p0, Landroid/widget/TextView;
55
56     const v0, 0x7f0d007e
57
58     invoke-virtual {p1, v0}, Landroid/view/View;->find<
59
60     move-result-object v0
61
62     check-cast v0, Landroid/widget/TextView;
63
64     iput-object v0, p0, Landroid/widget/TextView;
65
66     const v0, 0x7f0d0049
67
68     invoke-virtual {p1, v0}, Landroid/view/View;->find<
69
70     move-result-object v0
71
72     check-cast v0, Landroid/widget/ImageView;
```

☐ Exact

Bytecode Decompiler - Editable: false

```
9
10
11 public a$a(android.view.View arg0) { // <init> // (Landroid/view/View;)V
12     aload0 // reference to self
13     aload1
14     invokespecial android/support/v7/widget/RecyclerView$w.<init>
15     aload0 // reference to self
16     aload1
17     ldc 2131558525 (java.lang.Integer)
18     invokevirtual android/view/View.findViewById(I)Landroid/view/View;
19     checkcast android/widget/LinearLayout
20     putfield r/a$a.s:android.widget.LinearLayout
21     aload0 // reference to self
22     aload1
23     ldc 2131558474 (java.lang.Integer)
24     invokevirtual android/view/View.findViewById(I)Landroid/view/View;
25     checkcast android/widget/TextView
26     putfield r/a$a.o:android.widget.TextView
27     aload0 // reference to self
28     aload1
29     ldc 2131558526 (java.lang.Integer)
30     invokevirtual android/view/View.findViewById(I)Landroid/view/View;
31     checkcast android/widget/TextView
32     putfield r/a$a.n:android.widget.TextView
33     aload0 // reference to self
34     aload1
35     ldc 2131558473 (java.lang.Integer)
36     invokevirtual android/view/View.findViewById(I)Landroid/view/View;
37     checkcast android/widget/ImageView
38     putfield r/a$a.r:android.widget.ImageView
39     aload0 // reference to self
40     aload1
41     ldc 2131558528 (java.lang.Integer)
42     invokevirtual android/view/View.findViewById(I)Landroid/view/View;
43     checkcast android/widget/TextView
44     putfield r/a$a.q:android.widget.TextView
45     aload0 // reference to self
46     aload1
47     ldc 2131558527 (java.lang.Integer)
48     invokevirtual android/view/View.findViewById(I)Landroid/view/View;
49     checkcast android/widget/TextView
50     putfield r/a$a.p:android.widget.TextView
51     return
52 }
53 }
```

Refresh

Bytecode Viewer

# Локальные атаки

## Вектор атаки

1. Внедрение вредоносного кода
2. Фишинг, реклама, рескин



# Локальные атаки

## Способы защиты

Смириться, нет абсолютной защиты, можно усложнить процесс анализа приложение.

1. Обфусцирование кода (proguard)
2. Шифрование байткода
3. Утончение клиента и вынос важной логики на сервер
4. Реализация логики в нативных библиотеках
5. Контроль и защита «чувствительных» данных

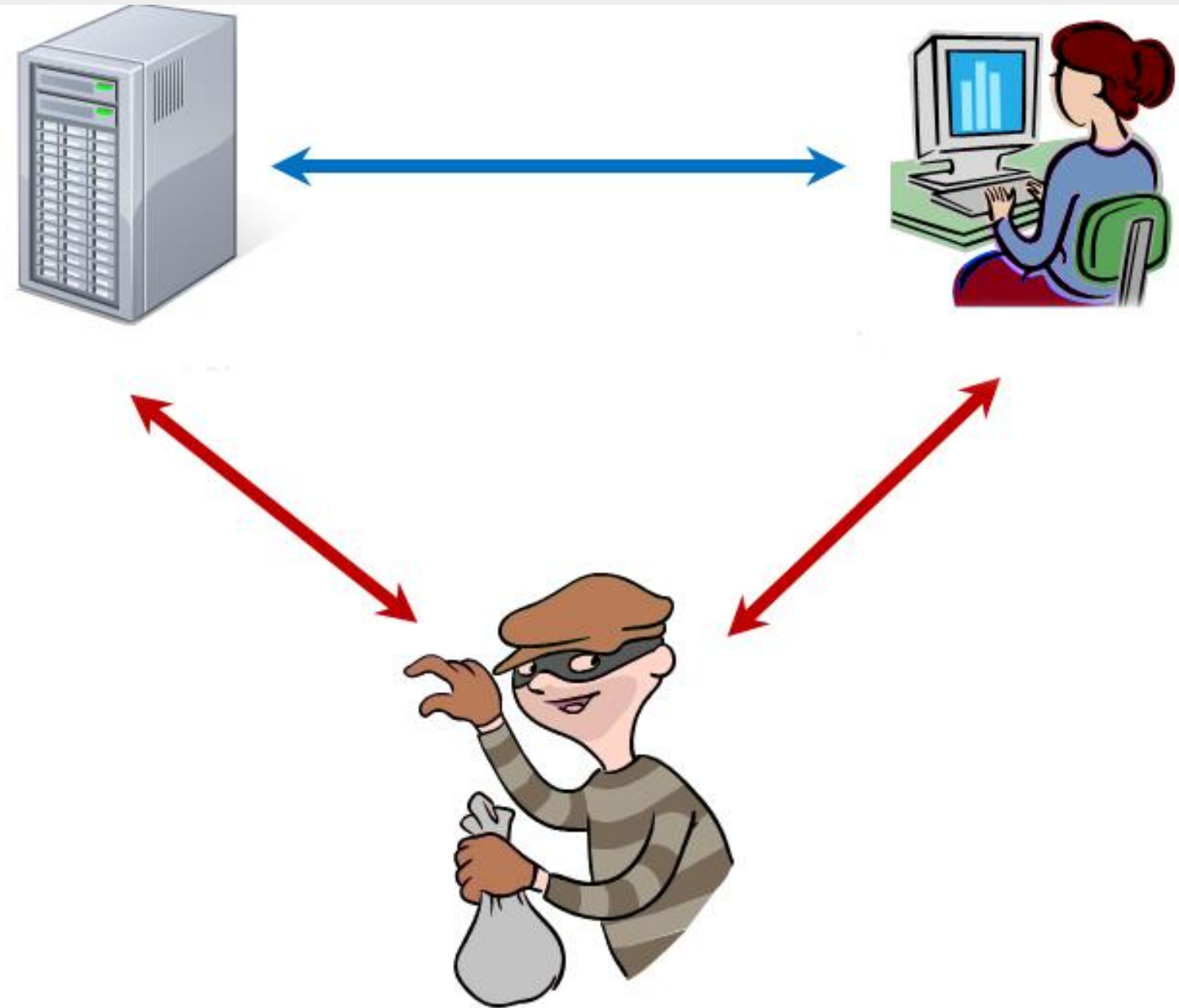


# Удаленные атаки

# Удаленные атаки

## Основные виды атак

1. Man in the middle
2. SSL unpinning



# Удаленные атаки

## Вектор атак

Атака MitM

1. Внедрение доверенного сертификата
2. Перехват или изменение трафика

Атака ssl unpinning

1. Переопределение методов java для работы с сетью и сертификатами
2. Перехват трафика



# MitM на Android с Fiddler

Progress Telerik Fiddler Web Debugger

File Edit Rules Tools View Help

WinConfig Replay X Go Stream Decode Keep: All sessions Any Process Find Save Browse Clear Cache TextWizard Tearoff MSDN Search... Online

#	Result	Protocol	Host	URL	Body	Caching	Content-Type	Process
42	200	HTTP	Tunnel to	lh3.googleusercontent.co...	1 554			
43	200	HTTPS	lh5.ggpht.com	/jZ8XCjpCQWWZ5GLhbJR...	3 929	public, ...	image/png	
44	200	HTTP	Tunnel to	lh3.googleusercontent.co...	1 554			
45	200	HTTPS	lh3.googleusercontent...	/i1-X1jPtSLgB7oPbtr-1dLt...	16 972	public, ...	image/png	
46	200	HTTPS	lh3.googleusercontent...	/P5mM4TgDYGeWXkV_ZG...	16 367	public, ...	image/png	
47	200	HTTPS	lh3.googleusercontent...	/8WsuI2NpvYJljZshzCgIX...	11 371	public, ...	image/png	
48	200	HTTPS	lh3.googleusercontent...	/TbsMAzKWLM93FK5hzFV...	37 354	public, ...	image/png	
49	200	HTTPS	lh3.googleusercontent...	/arhybrDSDtJf1qdUQdcnL...	34 800	public, ...	image/png	
50	200	HTTPS	android.clients.goo...	/fdfe/details?doc=com.ali...	66 539	private...	application/...	
51	200	HTTPS	lh3.googleusercontent...	/i1-X1jPtSLgB7oPbtr-1dLt...	14 902	public, ...	image/png	
52	200	HTTPS	android.clients.goo...	/fdfe/rev?doc=com.alibab...	21 702	private...	application/...	
53	200	HTTPS	lh3.googleusercontent...	/O3DWSC8VTipCLdbCpI9r...	14 995	public, ...	image/jpeg	
54	200	HTTPS	lh3.googleusercontent...	/vGHRf18dWJNkPtObzVa...	12 361	public, ...	image/jpeg	
55	200	HTTPS	lh3.googleusercontent...	/9fZV5omVUXGWT-wgDuV...	9 752	public, ...	image/jpeg	
56	200	HTTPS	lh3.googleusercontent...	/f2MThcJ67DQ9qYs8kFjry...	10 434	public, ...	image/jpeg	
57	200	HTTPS	lh3.googleusercontent...	/uK7bbtCLMchleKkwoQML...	10 772	public, ...	image/jpeg	
58	200	HTTPS	lh3.googleusercontent...	/56bmEMqn32oqsJ482hz6...	9 447	public, ...	image/jpeg	
59	200	HTTPS	lh3.googleusercontent...	/KhXbXihELb9A_7Lcql9M...	28 586	public, ...	image/jpeg	
60	200	HTTPS	lh3.googleusercontent...	/grioODSgJNqM75YGcVLV...	23 790	public, ...	image/jpeg	
61	200	HTTPS	lh3.googleusercontent...	/EYR8bHUs7Ocdl312nG5...	17 478	public, ...	image/jpeg	
62	200	HTTPS	lh3.googleusercontent...	/v34Dp5ft18-F56acw8...	20 878	public, ...	image/jpeg	
63	200	HTTPS	lh3.googleusercontent...	/8yLzYS0zAJp2vY...	18 731	public, ...	image/jpeg	
64	200	HTTPS	lh3.googleusercontent...	/yhfrEXW58k47Y...	73 173	public, ...	image/jpeg	
65	200	HTTPS	lh3.googleusercontent...	/usdJmSLDDZnrPuuG3...	32 033	public, ...	image/jpeg	
66	200	HTTPS	lh3.googleusercontent...	/1RC0pt5epHhy9SSxuk7...	25 427	public, ...	image/jpeg	
67	200	HTTPS	lh3.googleusercontent...	/s83MchnrUSB4-WEmCB-X...	19 117	public, ...	image/jpeg	
68	200	HTTPS	lh3.googleusercontent...	/hr5t4tnjV8IZGyQINPama...	22 128	public, ...	image/jpeg	
69	200	HTTPS	lh3.googleusercontent...	/KTKyhbDaO8UciIhkJUffV...	19 228	public, ...	image/jpeg	
70	200	HTTPS	lh3.googleusercontent...	/x208R6aOT-SCLPVyKdGI...	17 266	public, ...	image/jpeg	
71	200	HTTP	Tunnel to	lh5.googleusercontent.co...	1 554			
72	200	HTTP	Tunnel to	lh4.googleusercontent.co...	1 554			
73	200	HTTP	Tunnel to	lh6.googleusercontent.co...	1 554			
74	200	HTTPS	lh4.googleusercontent...	/rgRuCNxqtYQ/AAAAAA...	653	public, ...	image/png	
75	200	HTTPS	lh6.googleusercontent...	/dxde4tUoIoY/AAAAAA...	331	public, ...	image/png	
76	200	HTTPS	lh5.googleusercontent...	/-AJdMnus3_ZQ/AAAAAA...	888	public, ...	image/png	
77	200	HTTPS	android.clients.goo...	/fdfe/plusProfile	48	no-cac...	application/...	
78	200	HTTPS	android.clients.goo...	/fdfe/purchase	974	no-cac...	application/...	
79	200	HTTPS	android.clients.goo...	/fdfe/delivery?doc=com.a...	674	no-cac...	application/...	
80	200	HTTPS	android.clients.goo...	/fdfe/log	47	no-cac...	application/...	
81	200	HTTP	Tunnel to	android.clients.google.co...	1 658			
82	302	HTTPS	android.clients.goo...	/market/download/Downlo...	723	no-cache	text/html; c...	
83	200	HTTP	Tunnel to	r2---sn-ug5onuxaxjvh-guf...	838			
84	200	HTTPS	r2---sn-ug5onuxaxj...	/market/GetBinary/GetBin...	28 96...	public, ...	application/...	
85	200	HTTPS	android.clients.goo...	/market/api/ApiRequest	33	private...	application/...	
86	200	HTTP	Tunnel to	array405-prod.do.dsp.mp...	873			svchos...
87	200	HTTPS	array405-prod.do.d...	/join/	288	no-cac...		svchos...
88	200	HTTP	Tunnel to	e.crashlytics.com:443	794			
89	200	HTTPS	e.crashlytics.com	/spi/v2/events	0		text/plain; ...	
90	304	HTTP	ctdl.windowsupdat...	/msdownload/update/v3/s...	0	max-aq...	application/...	svchos...

[QuickExec] ALT+Q > type HELP to learn more

Capturing All Processes 1 / 159 https://android.clients.google.com/fdfe/plusProfile

Fiddler Orchestra Beta FiddlerScript Log Filters Timeline

Statistics Inspectors AutoResponder Composer

Headers TextView SyntaxView WebForms HexView Auth Cookies Raw JSON

XML

Request Headers [Raw] [Header Definitions]

GET /fdfe/plusProfile HTTP/1.1

Client

Accept-Language: ru-RU  
User-Agent: Android-Finsky/4.6.17 (api=3,versionCode=80260017, sdk=19,device=Zera%20S,hardware=mt65

Miscellaneous

X-DFE-Client-Id: am-unknown  
X-DFE-Device-Id: 32c824b90442ec04  
X-DFE-Enabled-Experiments: d:details.double\_fetch\_social\_data  
X-DFE-Filter-Level: 3  
X-DFE-Logging-Id: -28704adaec552168

Response body is encoded. Click to decode.

Transformer Headers TextView SyntaxView ImageView HexView WebView Auth

Caching Cookies Raw JSON XML

Response Headers [Raw] [Header Definitions]

HTTP/1.1 200 OK

Caching

Cache-Control: no-store, no-cache, must-revalidate  
Date: Sat, 22 Jul 2017 09:05:50 GMT  
Expires: Mon, 01 Jan 1990 00:00:00 GMT  
Pragma: no-cache  
Vary: Accept-Encoding

Entity

Content-Type: application/protobuf

Miscellaneous

Accept-Ranges: none  
Server: GSE  
X-DFE-Content-Length: 58

Security

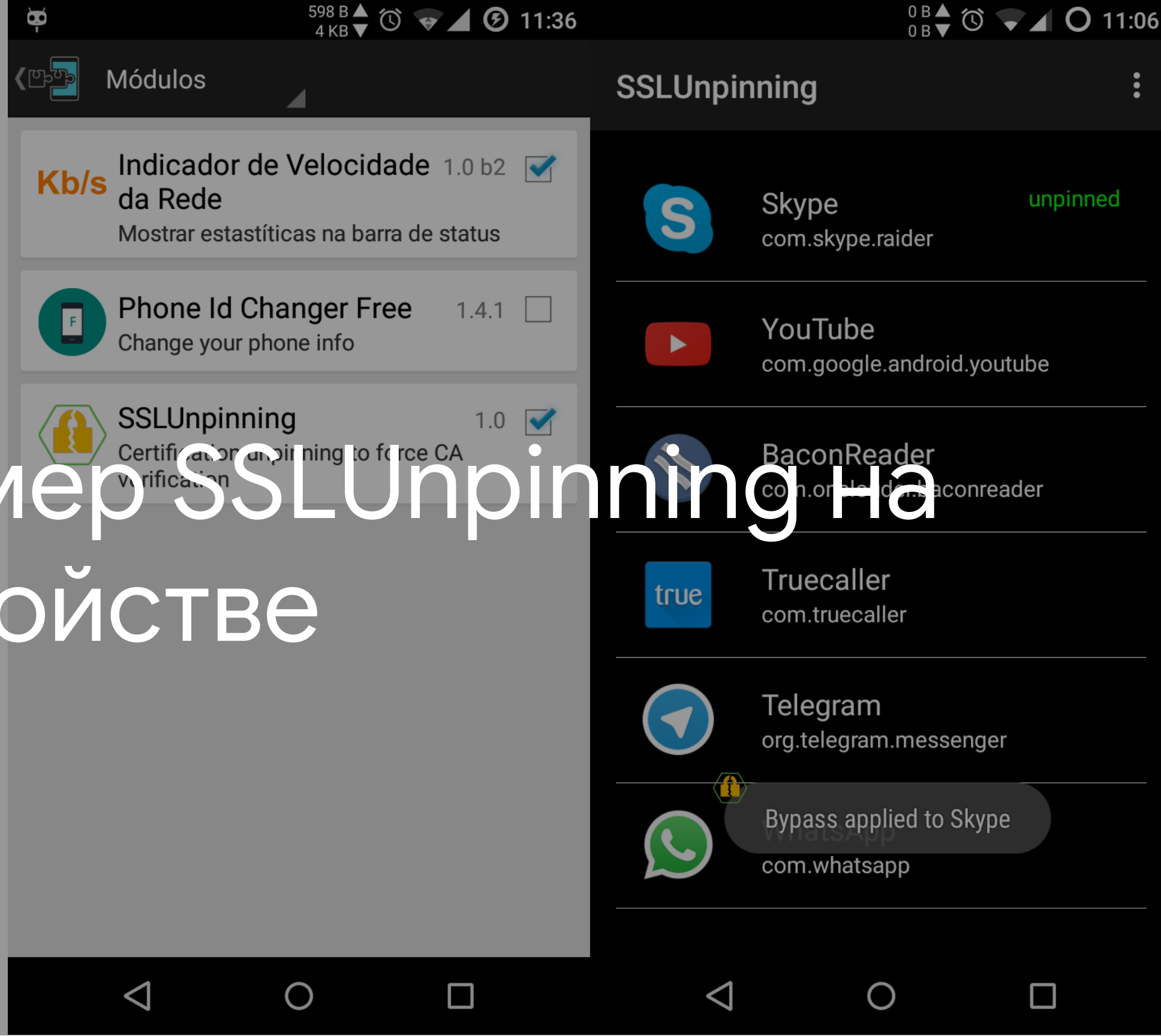
X-Content-Type-Options: nosniff  
X-Frame-Options: SAMEORIGIN  
X-XSS-Protection: 1; mode=block

Transport

Alt-Svc: quic=":443"; ma=2592000; v="39,38,37,36,35"  
Transfer-Encoding: chunked



# Пример SSLUnpinning на устройстве





# Удаленные атаки

## Способы защиты

1. SSL pinning
2. Проверка цепочки сертификата
3. SSL unpinning работает на рутованных устройствах – проверка на root
4. Дополнительное шифрование
5. Усложнение протокола обмена



# Доступ к устройству

При данных видах атаки, злоумышленник заполучает устройство целиком. Если вышеописанные методы атак безуспешны применяется анализ на наличие уязвимостей в программном коде или операционной системы Android.



# Эксплуатация уязвимостей

# Ключи авторизации в коде

## Примеры уязвимости

1. Логины пароли в коде или манифесте
2. Ключи шифрования в открытом виде
3. Слабые ключи шифрования





```
6      <!--
7          <meta-data>
8              <user>admin</user>
9              <password>p@$sW0rd</password>
10         </meta-data>
11     -->
12
13     <application
14         android:allowBackup="true"
15         android:debuggable="true"
16         android:icon="@mipmap/ic_launcher"
17         android:label="Security App"
18         android:roundIcon="@mipmap/ic_launcher_round"
19         android:supportsRtl="true"
20         android:theme="@style/AppTheme"
21
22
23
24
```

# Ключи авторизации в коде

## Защита

1. Не хранить важные данные в коде или настройках
2. Генерировать ключи шифрования на основе логина/пароля пользователя на лету



# Атака на WebView

## Вектор атаки

1. Неверный ssl-pinning или его отсутствие
2. Эксплуатация кеша WebView
3. Доступ к файлам через WebView





```
// Защита от кеширования
```

```
mWebView.getSettings().setAppCacheEnabled(false);
```

```
mWebView.getSettings().setCacheMode(WebSettings.LOAD_NO_CACHE);
```

```
// Пример доступа к файлам через WebView
```

```
webView.getSettings().setJavaScriptEnabled(true);
```

```
webView.getSettings().setAllowFileAccessFromFile(true);
```

```
WebView.loadUrl(url);
```

```
// url = "file:///data/data/bankclient_android/shared_prefs/prefs.xml"
```

# Атака на WebView

## Защита

1. Использование ssl-pinning для webview
2. Запрещать кеширование данных
3. Запрет на загрузку файлов



# Атака через allowBackup и debuggable

## Вектор атаки

1. Позволяет сделать дамп всей информации приложения
2. Позволяет выполнить отладку приложения



// Пример запуска полного дампа устройства

```
$ adb backup -all
```

// Отладка приложения

```
$ adb forward tcp:54321 jdwp:543
```

```
$ jdb -attach localhost:54321
```

# Атака через отправку сообщений

## Вектор атаки

1. Экспорт компонентов
2. Отправка широковещательных сообщений
3. Отсутствие проверки источника сообщения
4. Отсутствие фильтрации во входящем intent
5. Вложенные intent



```
<receiver android:name=".SecurityReceiver"

    android:enabled="true"

    android:exported="true">

    <intent-filter>

        <data

            android:scheme="https"

            android:host="securehost.com"

            android:pathPattern="/auth/.*/"

            />

            <category android:name="android.intent.category.DEFAULT" />

        </intent-filter>

    </receiver>
```

# Пример перенаправления Intent



# Атака через отправку сообщений

## Защита

1. Правильно экспортировать компоненты
2. Проверка источника сообщения
3. Фильтрация входных сообщений





# Статистика

Соотношение кода ~82% - библиотеки ~18%  
собственный код

Вторая по популярности библиотека – facebook

Класс «а» - 257 место по использованию.

Класс «о» - 81 по использованию

99% приложений используют доступ в сети

Интернет



# Спасибо!



Сосновский Юрий

Telegram @JurySosnovsky, vk.com/sosnoffsky



# DevFest

## Вопросы

Вопросы можно направить на [qa@sosnoffsky.com](mailto:qa@sosnoffsky.com)

