# Безопасность

Червяков Алексей

# Наш план

- Проблемы с сетью
- Проблемы с хранением
- Проблемы в приложении

# Организационные моменты

# Напоминание отметиться на портале

# Безопасность сети

# Безопасность сети

**Проблема:**

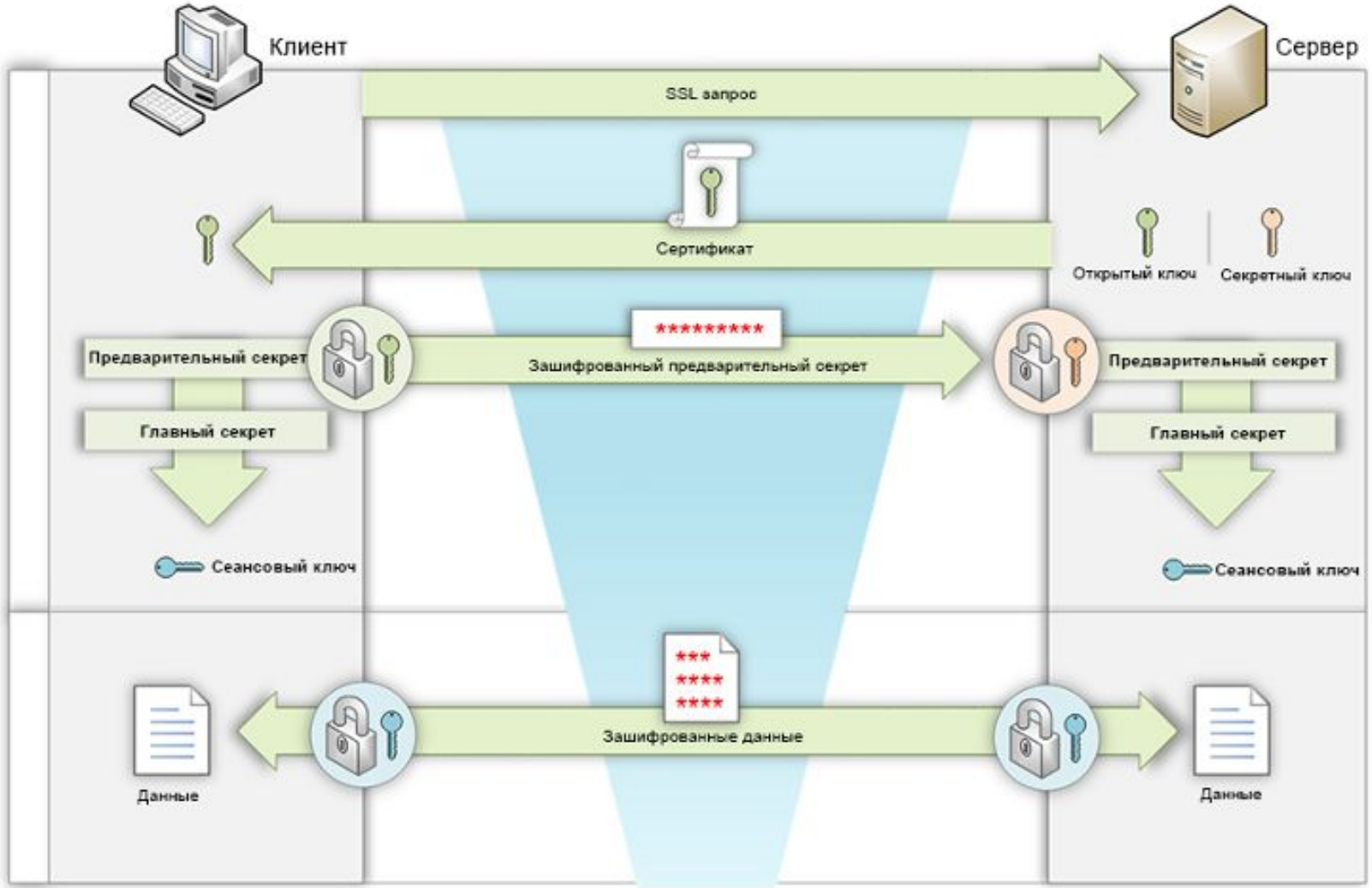
HTTP — широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (тех, которые могут содержать ссылки, позволяющие организовать переход к другим документам).

**Решение:**

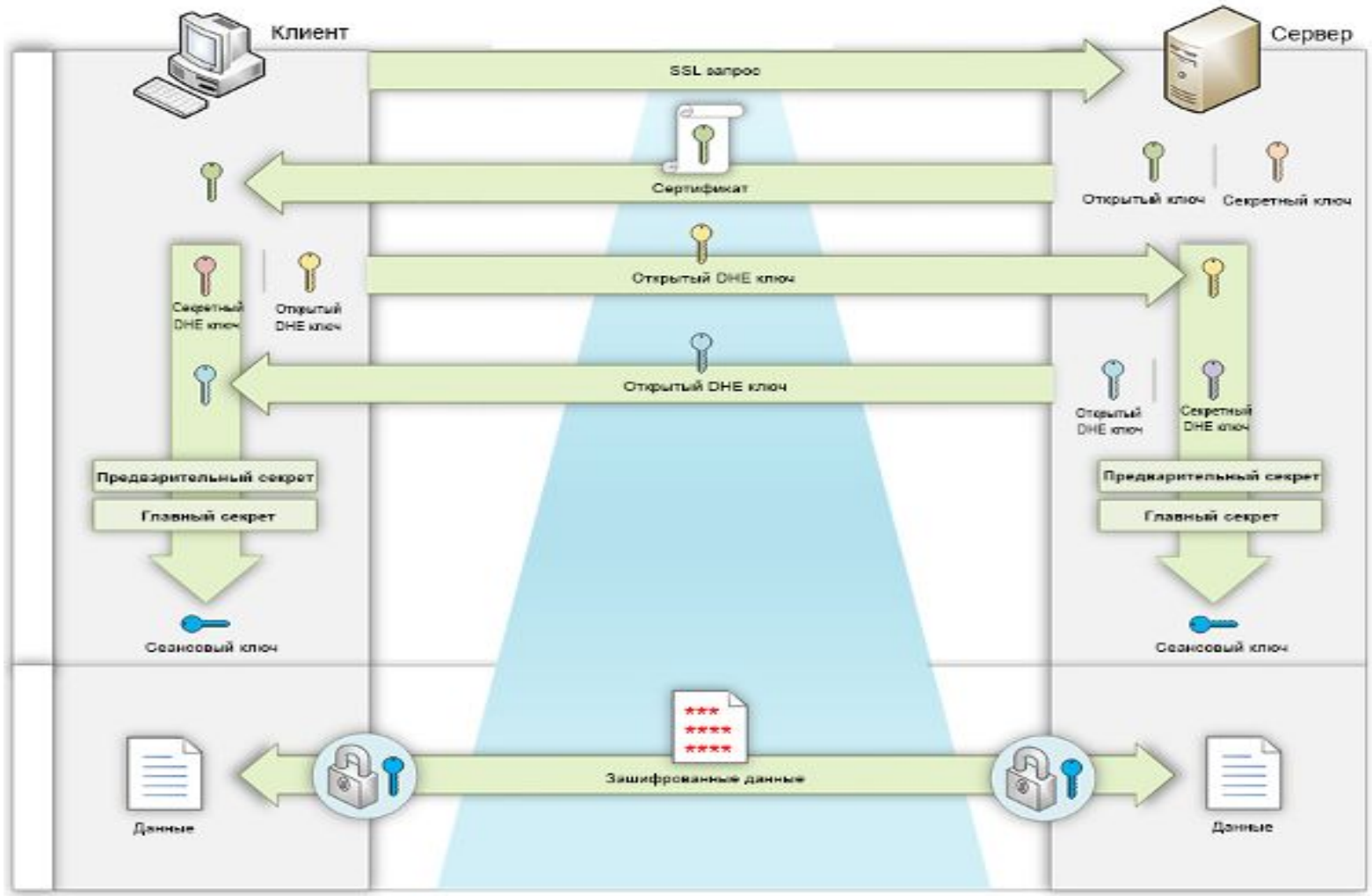
Для того, чтобы предотвратить возможность чтения и модификации запросов, поверх обычного HTTP добавили SSL и появился протокол HTTPS, где буква S расшифровывается как Secure

использовать SSL/TLS сертификат

# TLS handshake (RSA)



# TLS handshake (DHE)



# MITM (Man in the middle)

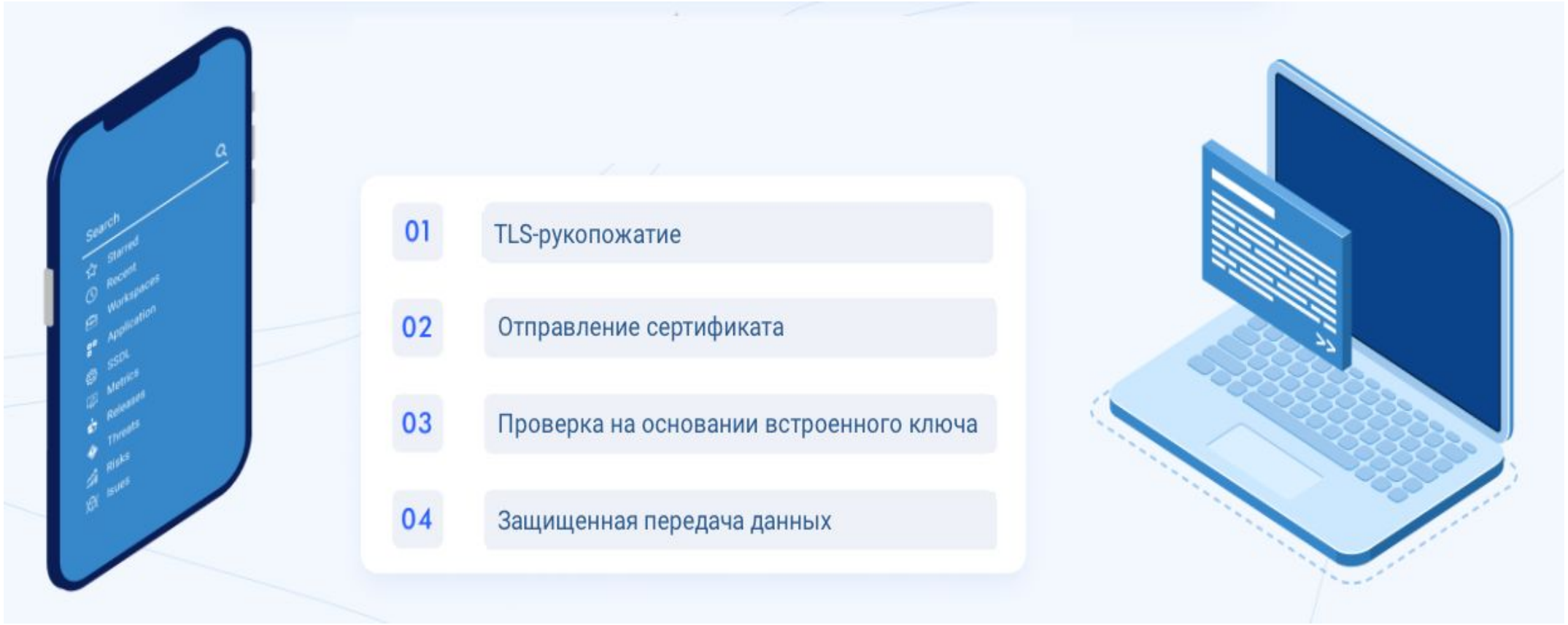
**Проблема:**
Атака на канал связи, при которой злоумышленник находится в одной сети с вами и обладает контролем над точкой доступа, или каким-то образом может перенаправить вас на свой прокси-сервер внутри сети. Злоумышленник для клиента представляется конечным сервером, а для сервера - клиентом

**Решение:**
В качестве защиты мобильного приложения от подобных атак применяют механизм, который называется SSL Pinning

# MITM


Злоумышленник может получить доступ к проходящей информации в открытом виде
Пользователь
Фальшивые сертификаты
Мошенническая точка беспроводного доступа
Сертификаты
Корпоративный магазин приложений
Безопасный веб-сайт

# SSL Pinning


Search
Starred
Recent
Workspaces
Application
SSDL
Metrics
Releases
Threats
Risks
Issues
01
TLS-рукопожатие
02
Отправление сертификата
03
Проверка на основании встроенного ключа
04
Защищенная передача данных

# Настройка в android (стандартный механизм)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest ... >
    <application android:networkSecurityConfig="@xml/network_security_config"
                    ... >
        ...
    </application>
</manifest>
```

# Настройка в android (стандартный механизм)

```
<network-security-config>
    <domain-config>
        <domain includeSubdomains="true">example.com</domain>
        <pin-set>
            <pin digest="SHA-256">7HIpactkIAq2Y49orFOOQKurWxmmSFZhBCoQYcRhJ3Y=</pin>
        </pin-set>
    </domain-config>
</network-security-config>
```

# Okhttp client

```
CertificatePinner certPinner = new CertificatePinner.Builder()
        .add("appmattus.com",
                "sha256/4hw5tz+scE+TW+mlai5YipDfFWn1dqvfLG+nU7tq1V8=")
        .build();

OkHttpClient okHttpClient = new OkHttpClient.Builder()
        .certificatePinner(certPinner)
        .build();
```

# Безопасность хранения данных

# Хранение данных

**Проблема:**
Данные хранятся на устройстве

**Решение:**
Шифрование данных android > 6 версии

```
SharedPreferences.Editor editor = getSharedPreferences("preferenceName",
MODE_PRIVATE).edit();

 editor.putString("key", "value");

 editor.commit();
```

`MODE_PRIVATE` - означает, что доступ к данным может получить только ваше приложение.

# Хранение токена

**Проблема:**
В shared preferences можно хранить данные, но как с ними взаимодействовать и обновлять. А еще есть кнопка очистить данные

**Решение:**
Использовать account manager android > 7 версии либо использовать БД с шифрованием, например realm

# Account manager

```
<service
            android:name=".account.AuthenticatorService"
            android:exported="false">
            <intent-filter>
                <action android:name="android.accounts.AccountAuthenticator" />
            </intent-filter>

            <meta-data
                android:name="android.accounts.AccountAuthenticator"
                android:resource="@xml/authenticator" />
        </service>
```

# Account manager

```
<account-authenticator
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:accountType="@string/account_type"
    android:label="@string/app_name" />
```

# Рутованные девайсы

Проблема:
данные хранятся в xml shared preferences

Решение:
Использовать шифрование дополнительное, например javax.crypto

# Безопасность приложения

# Обфускация

**Проблема:**
Можно декомпилировать apk и посмотреть, что в нем есть

**Решение:**

Использовать proguard или проприетарный обфускатор. Можно использовать JNI

# Proguard

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

# Уязвимость компонентов

**Проблема:**
Экспортированные компоненты. Компоненты в которых настроены intent фильтры. Неявные интенты

**Решение:**

Закрыть дополнительным permission. Сделать не экспортируемый компонент

# Exported

```
<activity
    android:name="com.companyX.activities.ActivityA"
    android:exported="false"/>
```

# Permission

```
<permission
   android:name="com.companyX.permission.custom"
   android:description="@string/custom_permission_description"
   android:icon="R.drawable.ic_custom_permission
   android:label="@string/custom_permission_label"
   android:protectionLevel="signature"/>
```

```
<receiver
    android:name="com.companyX.receivers.ReceiverA"
    android:permission="com.companyX.permission.custom">
    <intent-filter android:priority="999">
      <action android:name="com.companyX.action.ACTION_A"/>
    </intent-filter>
</receiver>
```

# WebView

**Проблема:**

Открыть файл локальный а в нем есть вредоносный скрипт, позволяющий получить содержимое приватного файла

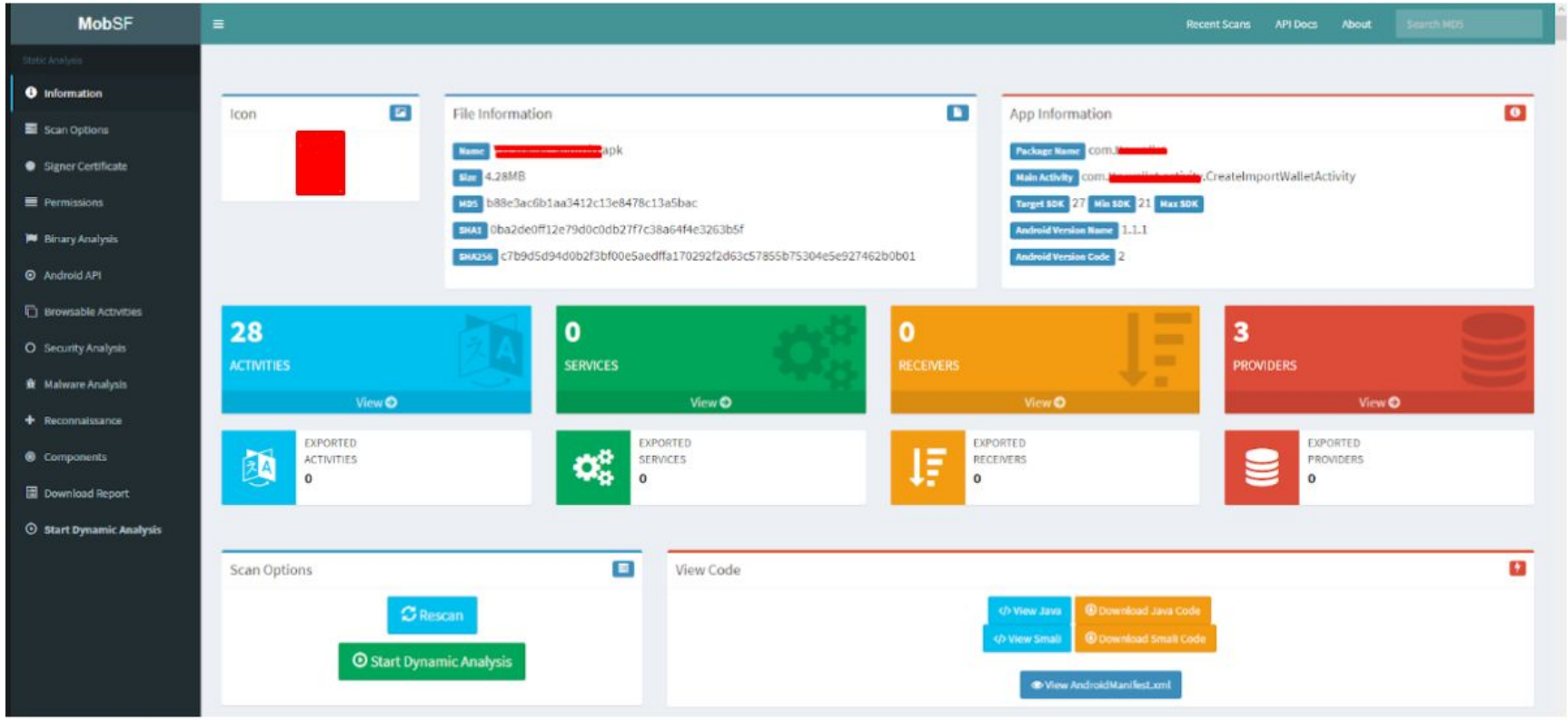
**Решение:**

использовать Android > 6. Не разрешать открывать локальные файлы. Закрыть доступ к webview извне

# Вспомогательный софт

MobSF

bytecode-viewer



# Оставьте отзыв!

# Спасибо за внимание!

Червяков Алексей