CI/CD и Безопасность

Червяков Алексей

Организационные моменты

Напоминание отметиться на портале

Наш план

- CI/CD
  - Что это за зверь?
  - CI - continuous integration
  - CD - continuous delivery
- Безопасность
  - Проблемы с сетью
  - Проблемы с хранением
  - Проблемы в приложении

Что за зверь CI/CD

Что это?

CI/CD - одна из практик DevOps, которая заключается в комбинации непрерывной интеграции и непрерывной доставке

В настоящий момент DevOps стремятся применять CI/CD практически для всех задач

Расшифровка

CI - continuous integration

CD - continuous delivery

CD - continuous deployment

Что это?

CI — практика разработки ПО, которая заключается в постоянном слиянии рабочих копий в общую основную ветвь разработки и выполнении частых автоматизированных сборок проекта для скорейшего выявления потенциальных дефектов и решения интеграционных проблем

CD — практика разработки ПО, которое производится короткими итерациями, гарантируя, что сборка является стабильным и может быть передано в эксплуатацию в любое время, а передача его не происходит вручную

CI

Плюсы

- проблемы интеграции выявляются и исправляются быстро, что оказывается дешевле
- немедленный прогон модульных тестов для свежих изменений
- постоянное наличие текущей стабильной версии вместе с продуктами сборок — для тестирования, демонстрации, и т. п.
- немедленный эффект от неполного или неработающего кода приучает разработчиков к работе в итеративном режиме с более коротким циклом.

CI

Плюсы

- проблемы интеграции выявляются и исправляются быстро, что оказывается дешевле
- немедленный прогон модульных тестов для свежих изменений
- постоянное наличие текущей стабильной версии вместе с продуктами сборок — для тестирования, демонстрации, и т. п.
- немедленный эффект от неполного или неработающего кода приучает разработчиков к работе в итеративном режиме с более коротким циклом.

Минусы

- значительные затраты на поддержку работы непрерывной интеграции
- необходимость в дополнительных вычислительных ресурсах под нужды непрерывной интеграции

CD

Плюсы

- Уменьшение стоимости, времени и риска внесения изменений
- Уменьшение человеческого фактора

CD

Плюсы

- Уменьшение стоимости, времени и риска внесения изменений
- Уменьшение человеческого фактора

Минусы

- значительные затраты на поддержку работы непрерывной интеграции
- необходимость в дополнительных вычислительных ресурсах под нужды непрерывной интеграции

Как это работает?

Зачем это работает?

Уменьшение Time to Market

Зачем это работает?

Уменьшение Time to Market

- Уменьшение риска человеческого фактора
- Уменьшение времени лага на сборках
- Улучшение качества кода
- Обкатанная доставка в продакшен
- Совместная работа

Мне это нужно?

Возможно

- Стартап из 2х человек
- Проект очень старый и мы не знаем как он работает
- Ты единственный пользователь

Мне это нужно?

Возможно

- Стартап из 2х человек
- Проект очень старый и мы не знаем как он работает
- Ты единственный пользователь

Во всех остальных случаях лучше его реализовать

И снова git

Чтобы CI/CD работал хорошо, нужно выбрать подход ведения git

- Git flow
- Trunk based development

Git flow

- Ветки живут на протяжении всей фичи
- Данная модель отлично подходит для организации рабочего процесса на основе релизов
- Gitflow предлагает создание отдельной ветки для исправлений ошибок в продуктовой среде
- Мастер всегда готов к деплою

Trunk based development

- Короткоживущие ветки. Любая ветка, которая создается, живет не больше 2х дней
- Feature Flags и Branch By Abstraction
- Continuous Code Review
- Мастер всегда готов к деплою, даже если в нем есть недописанные фичи

Процесс в android

Процесс выполнения в android

./gradlew lint

./gradlew build

./gradlew test

Процесс выполнения в android

./gradlew lint

./gradlew build

- ./gradlew assembleDebug
- ./gradlew assembleRelease

./gradlew test

- ./gradlew assembleAndroidTest

Linter

инструмент для статического анализа кода, который помогает разработчикам изловить потенциальные проблемы еще до того, как код скомпилируется

- неиспользуемые переменные
- неиспользуемые аргументы функций
- упрощение условий
- неправильные области видимости
- неопределенные переменные или функции
- плохо оптимизированный код

Инструменты

Инструменты CI

- GitLab - для inhouse
- GitHub - для open source
- BitBucket - для inhouse
- Jenkins - для inhouse
- TeamCity - для больших компаний

Пример

Как сделать хорошо

- Стабильность CI
- Скорость сборки
- Герметичность сборки
- Регулярные сборки
- Документация сборки

Как сделать хорошо

- Стабильность CI
  - Если сломан CI, часть команды не работает
- Скорость сборки
- Герметичность сборки
- Регулярные сборки
- Документация сборки

Как сделать хорошо

- Стабильность CI
  - Если сломан CI, часть команды не работает
- Скорость сборки
  - Влияет на доставку артефактов
- Герметичность сборки
- Регулярные сборки
- Документация сборки

Как сделать хорошо

- Стабильность CI
  - Если сломан CI, часть команды не работает
- Скорость сборки
  - Влияет на доставку артефактов
- Герметичность сборки
  - Избавляемся от внешних зависимостей
- Регулярные сборки
- Документация сборки

Как сделать хорошо

- Стабильность CI
  - Если сломан CI, часть команды не работает
- Скорость сборки
  - Влияет на доставку артефактов
- Герметичность сборки
  - Избавляемся от внешних зависимостей
- Регулярные сборки
  - Запускаем регулярно и отслеживаем
- Документация сборки

Как сделать хорошо

- Стабильность CI
  - Если сломан CI, часть команды не работает
- Скорость сборки
  - Влияет на доставку артефактов
- Герметичность сборки
  - Избавляемся от внешних зависимостей
- Регулярные сборки
  - Запускаем регулярно и отслеживаем
- Документация сборки
  - Оставляем всем знания

Вопросы

Безопасность сети

Безопасность сети

Проблема:

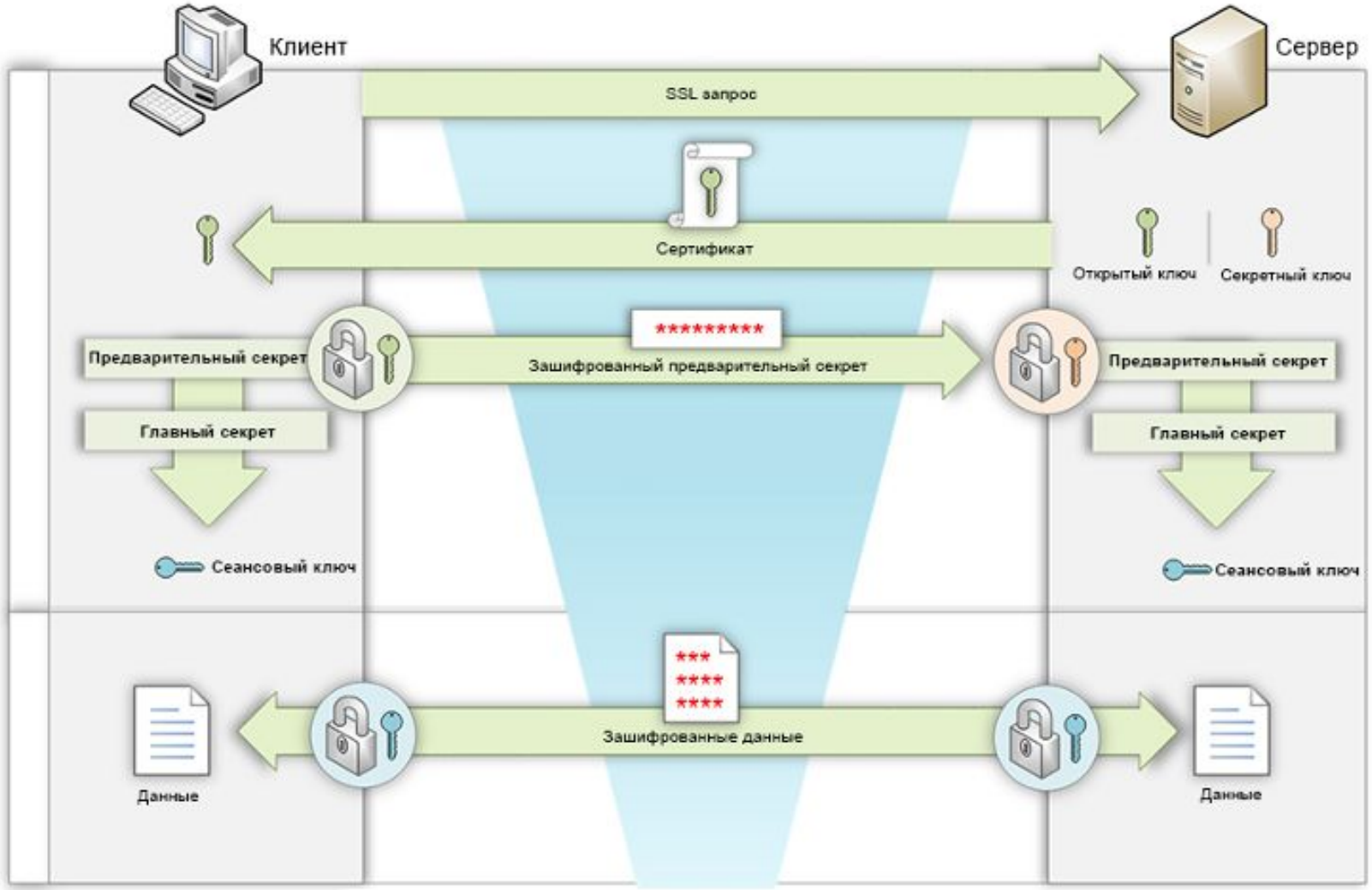
HTTP — широко распространённый протокол передачи данных, изначально предназначенный для передачи гипертекстовых документов (тех, которые могут содержать ссылки, позволяющие организовать переход к другим документам).

Решение:

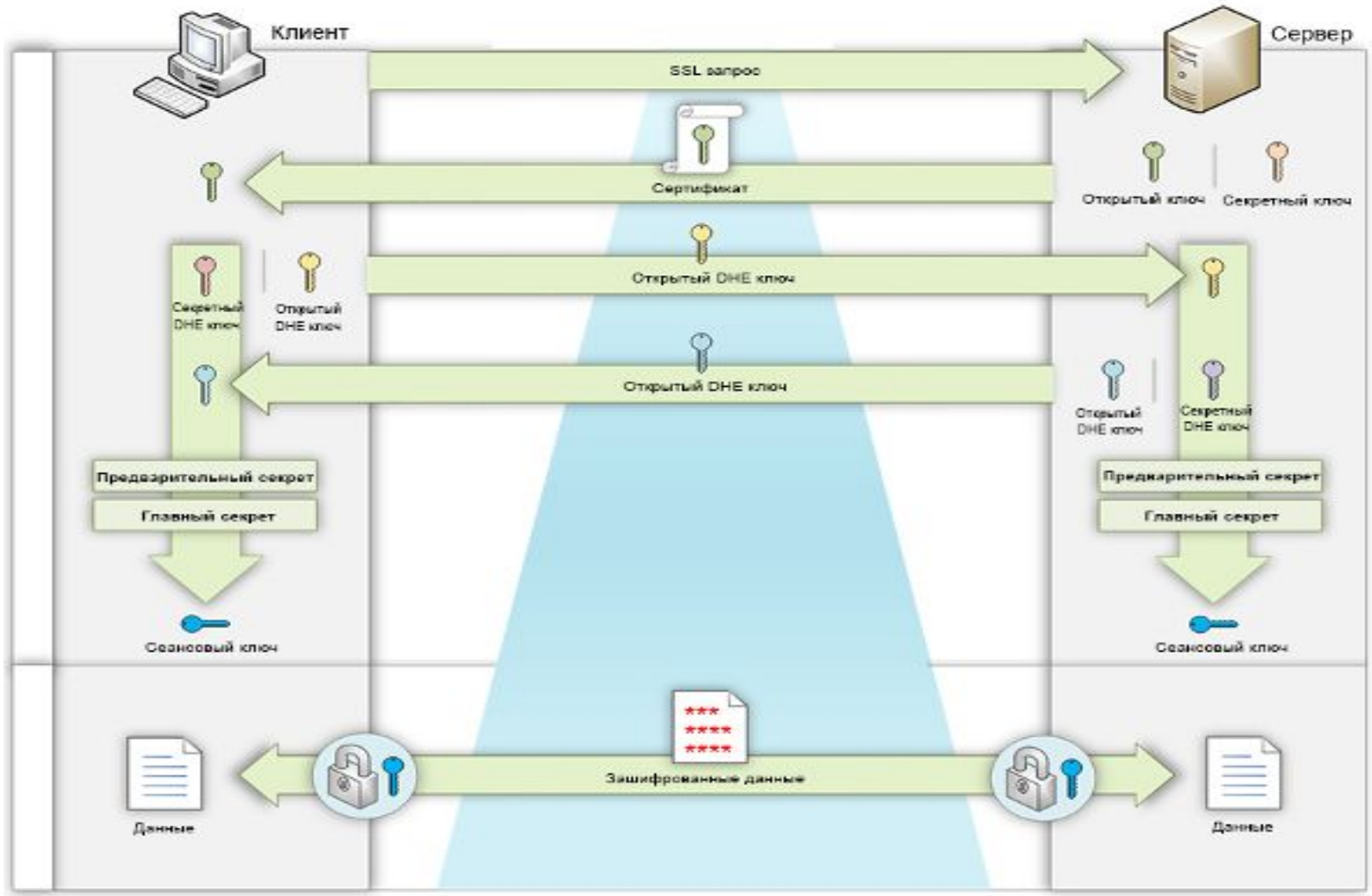
Для того, чтобы предотвратить возможность чтения и модификации запросов, поверх обычного HTTP добавили SSL и появился протокол HTTPS, где буква S расшифровывается как Secure

использовать SSL/TLS сертификат

TLS handshake (RSA)



TLS handshake (DHE)



MITM (Man in the middle)

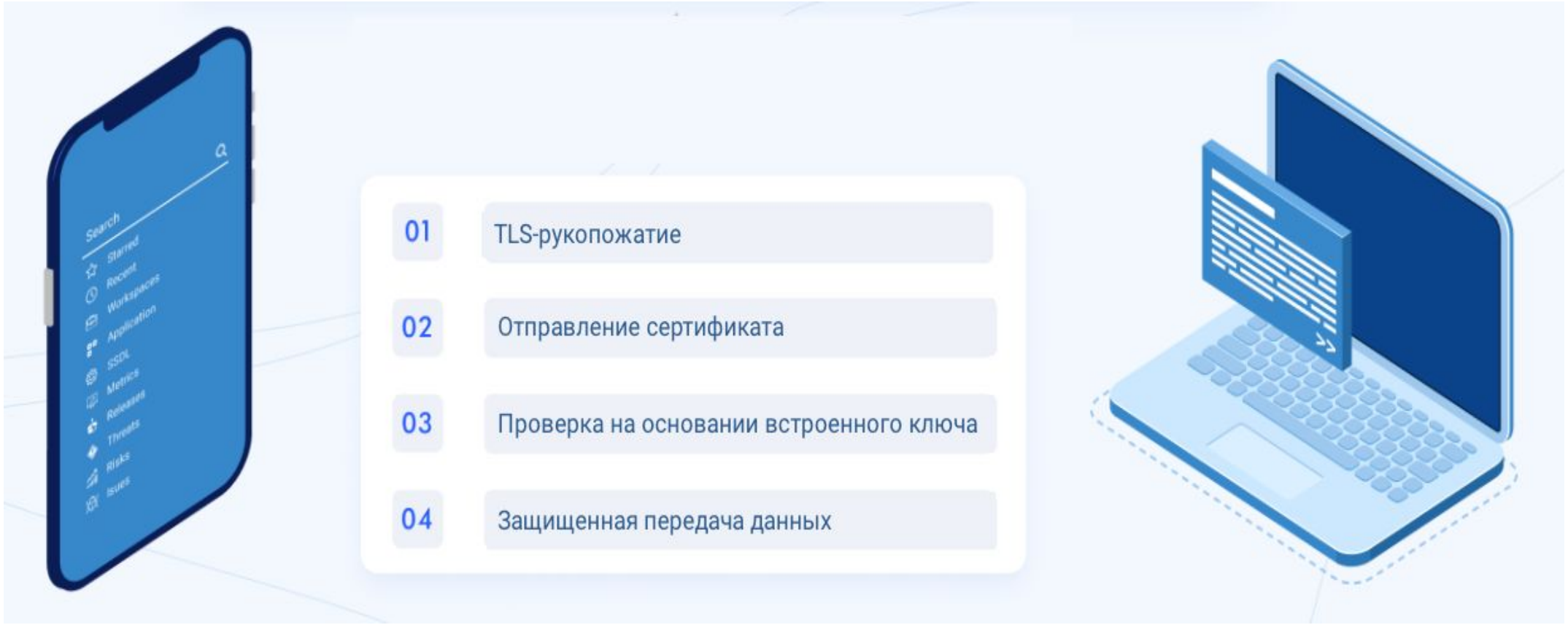
Проблема:
Атака на канал связи, при которой злоумышленник находится в одной сети с вами и обладает контролем над точкой доступа, или каким-то образом может перенаправить вас на свой прокси-сервер внутри сети. Злоумышленник для клиента представляется конечным сервером, а для сервера - клиентом

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

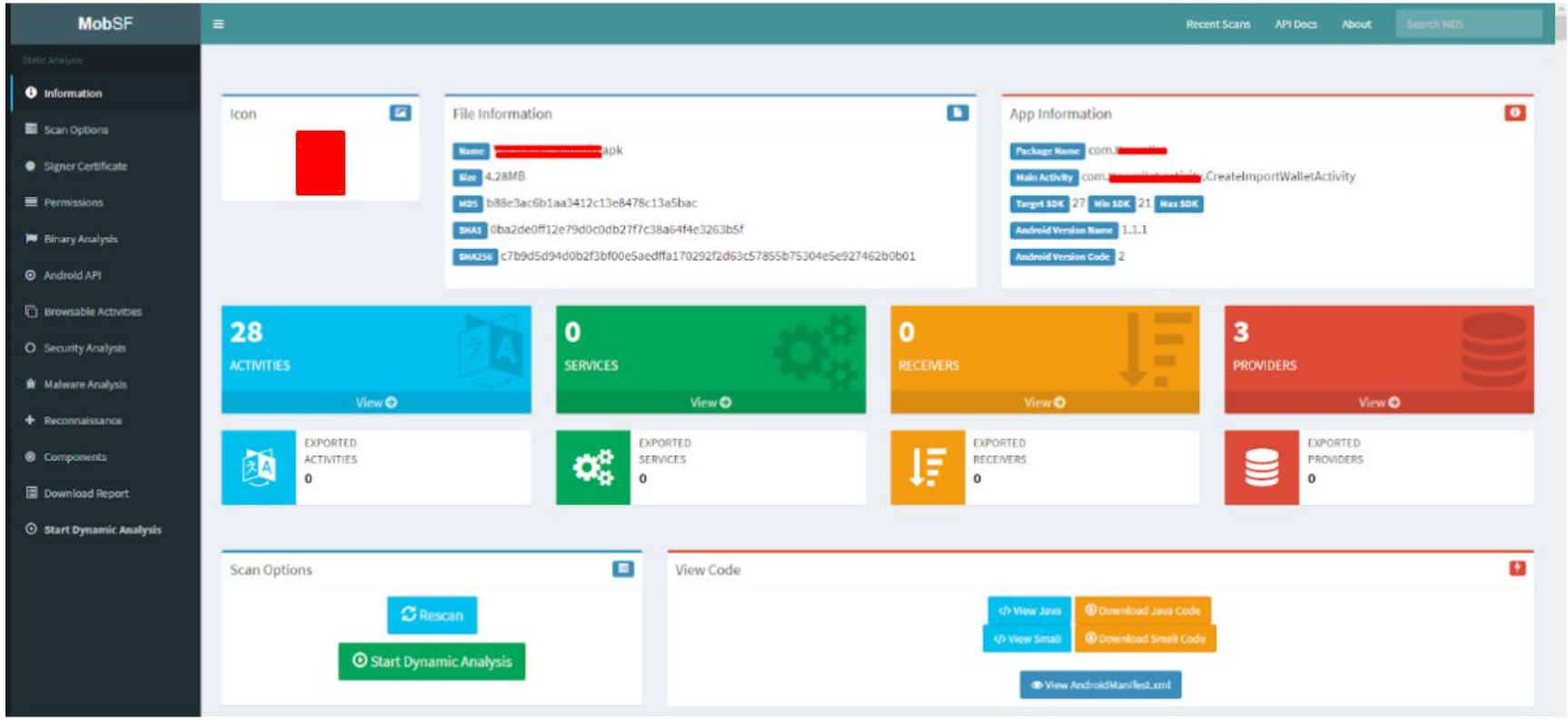
Решение:

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