Mobile security testing starter kit

Александра Сватикова

Heisenbug, 2022

HEISENBUG

// 2018 Piter

Андрей Леонов
SEMrush

Web security testing starter kit

Обо мне

- Лидер Продуктовой безопасности в Одноклассниках (Экосистема VK)

- alexandra.svatikova@vk.team

О чем сегодня поговорим

О чем сегодня поговорим

Чего мы хотим?

1. smoke / sanity план тестирования безопасности

2. принцип 80/20

О чем сегодня поговорим

1. Вступление
2. Мобильная безопасность 101
3. OWASP Mobile *
4. Задачи и инструменты
5. Чеклист

1 Вступление

Что мы знаем о мобильной безопасности

- Первый релиз Android — 23.09.2008
- Первый релиз iOS — 29.06.2007

Что мы знаем о мобильной безопасности

WWW — 12.03.1989

Что мы знаем о мобильной безопасности

Что знали про безопасность веб в 2004?

1. первая SQL инъекция описана в 1998

2. первая XSS обнаружена в 2000

3. первый OWASP Top 10 - 2003

про мобильные знаем намного больше!

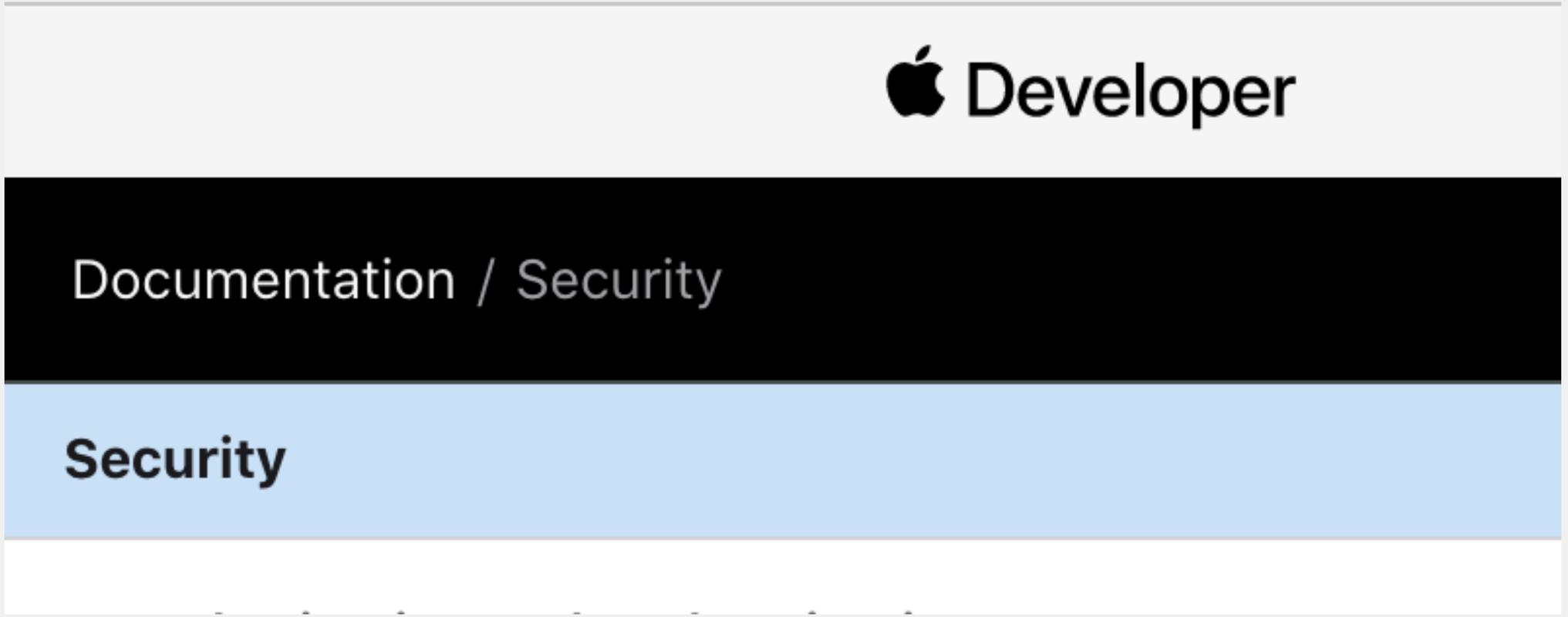
Что мы знаем о мобильной безопасности

Android Developers > Docs > Guides

App security best practices

By making your app more secure, you help preserve user trust and device integrity.

Что мы знаем о мобильной безопасности



Что мы знаем о мобильной безопасности

2

Мобильная безопасность 101

Что такое безопасность

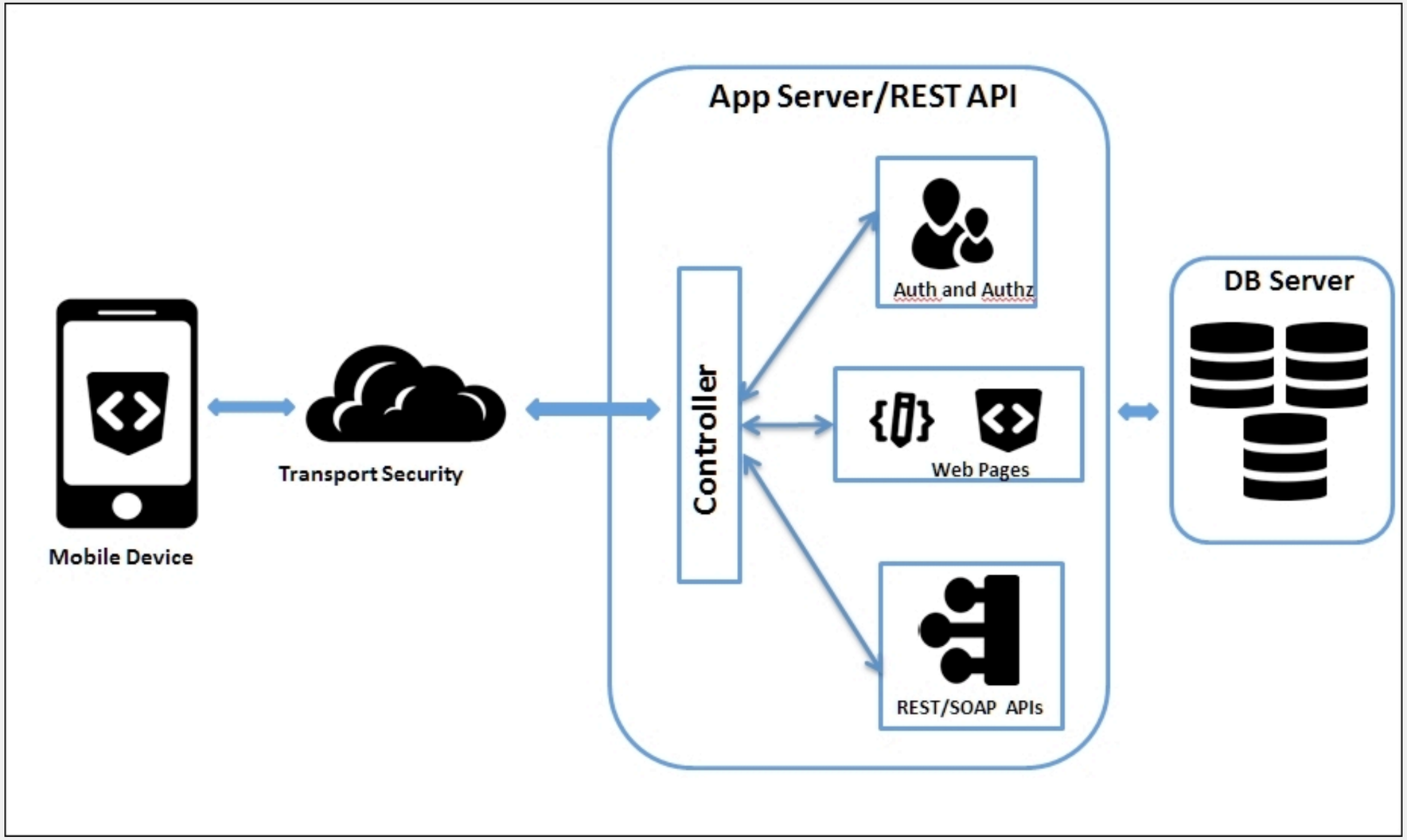
Безопасность =

конфиденциальность +

доступность +

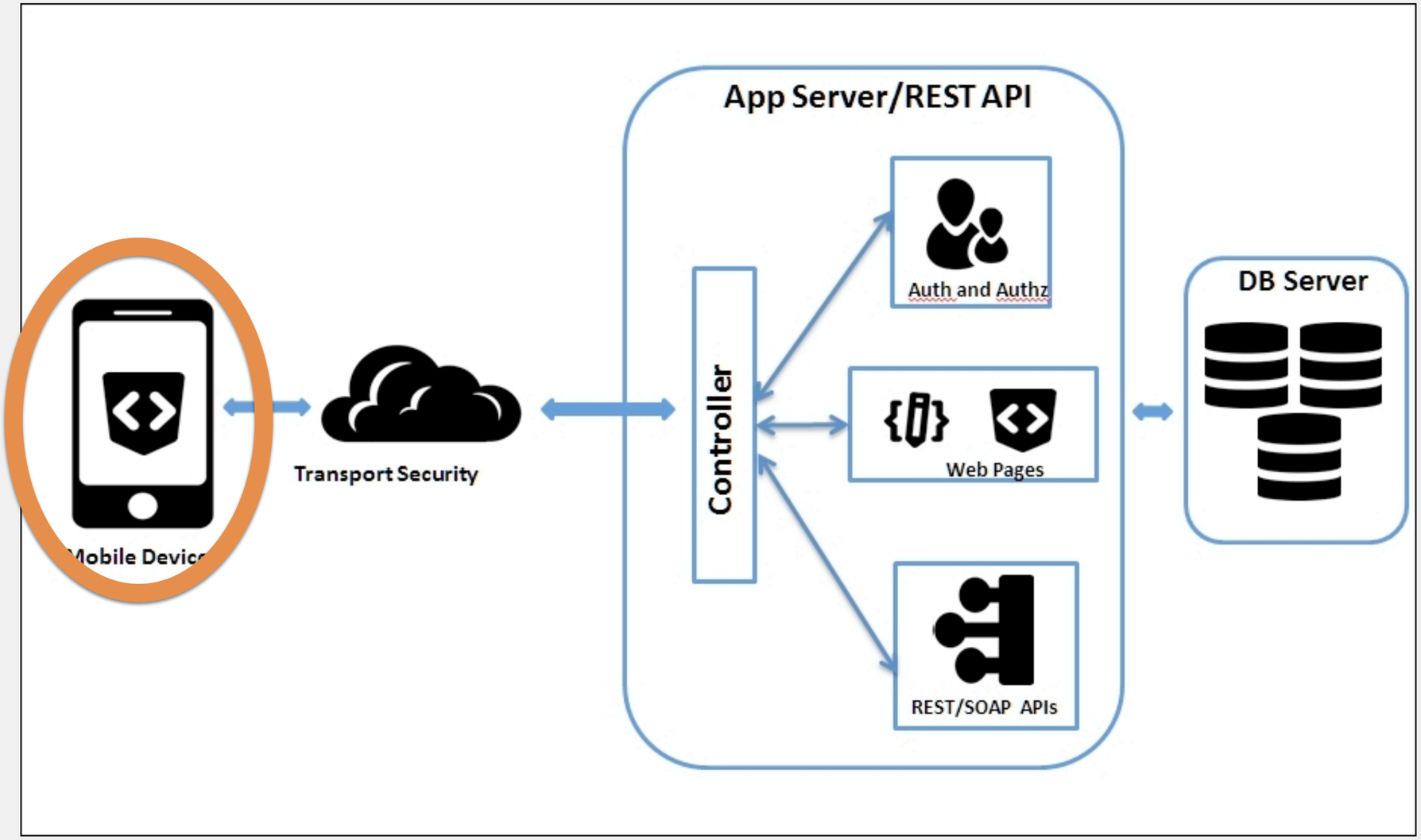
целостность

Что такое мобильная безопасность



https://www.oreilly.com/library/view/hacking-android/9781785883149/ch06s02.html

Что такое мобильная безопасность



https://www.oreilly.com/library/view/hacking-android/9781785883149/ch06s02.html

Типичные последствия мобильных уязвимостей

1. Разглашение конфиденциальных данных

Типичные последствия мобильных уязвимостей

1. Разглашение конфиденциальных данных

2. Повышение привилегий / несанкционированный доступ

Типичные последствия мобильных уязвимостей

1. Разглашение конфиденциальных данных
2. Повышение привилегий / несанкционированный доступ
3. Выполнение произвольного кода на устройстве

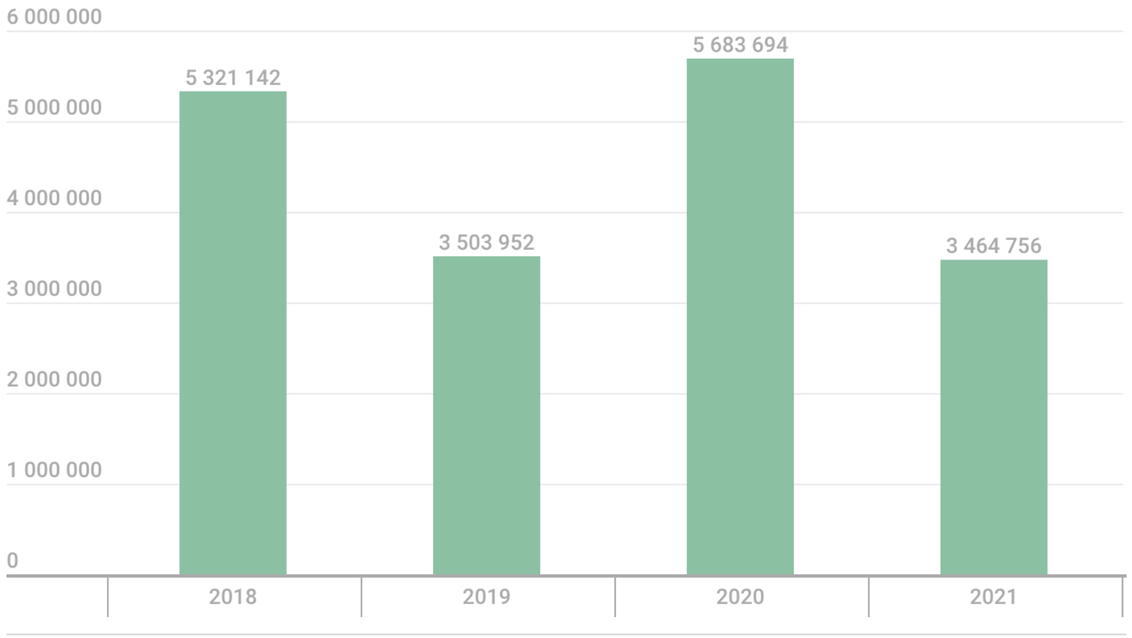
Типичные сценарии атак

1. Пользователь устанавливает вредоносное приложение

Типичные сценарии атак

Number of installation package

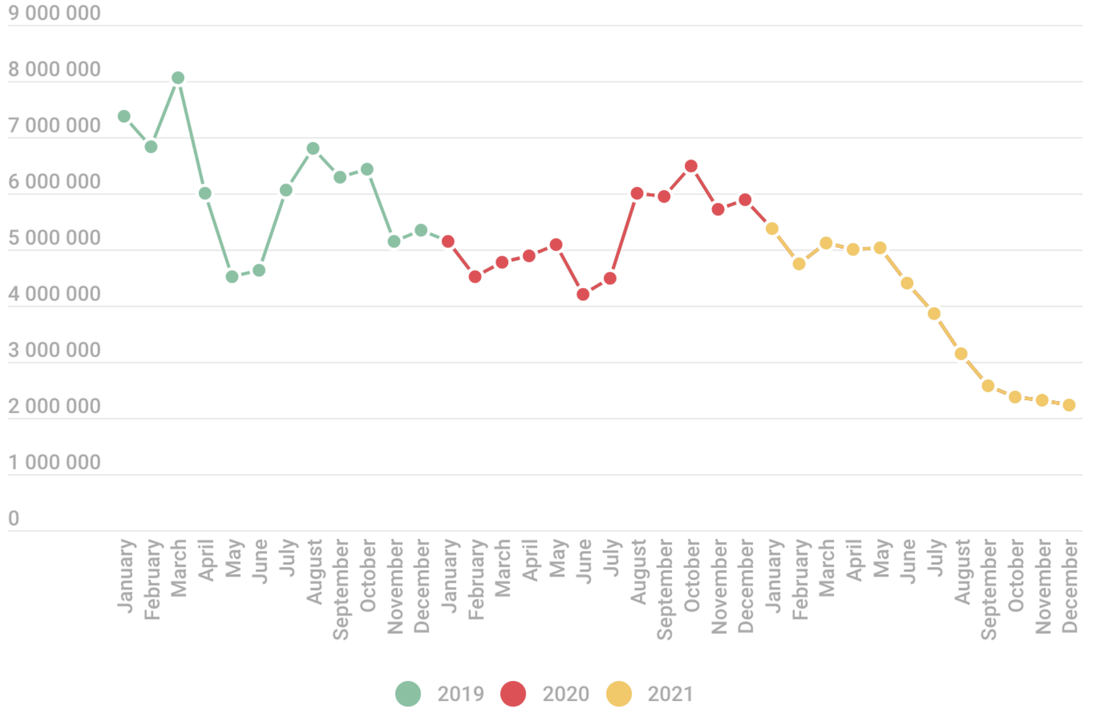
In 2021, we detected 3,464,756 mobile malicious installation packages, down 2,218,938 from the previous year. Overall, the number of mobile malware installation packages dropped to around 2019 levels.



kaspersky

Number of attacks on mobile users

The number of attacks fell smoothly throughout the reporting period, reaching in H2 2021 the lowest monthly average in the past two years.



kaspersky

https://securelist.com/mobile-malware-evolution-2021/105876/

Типичные сценарии атак

1. Пользователь устанавливает вредоносное приложение
2. Пользователь открывает вредоносную веб страницу

Типичные сценарии атак

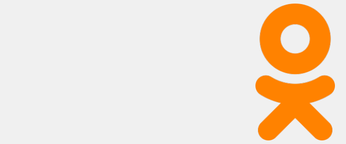
1. Пользователь устанавливает вредоносное приложение
2. Пользователь открывает вредоносную веб страницу
3. Пользователь теряет устройство

3

OWASP Mobile *

OWASP Mobile Application Security Verification Standard

OWASP Mobile Application Security Verification Standard

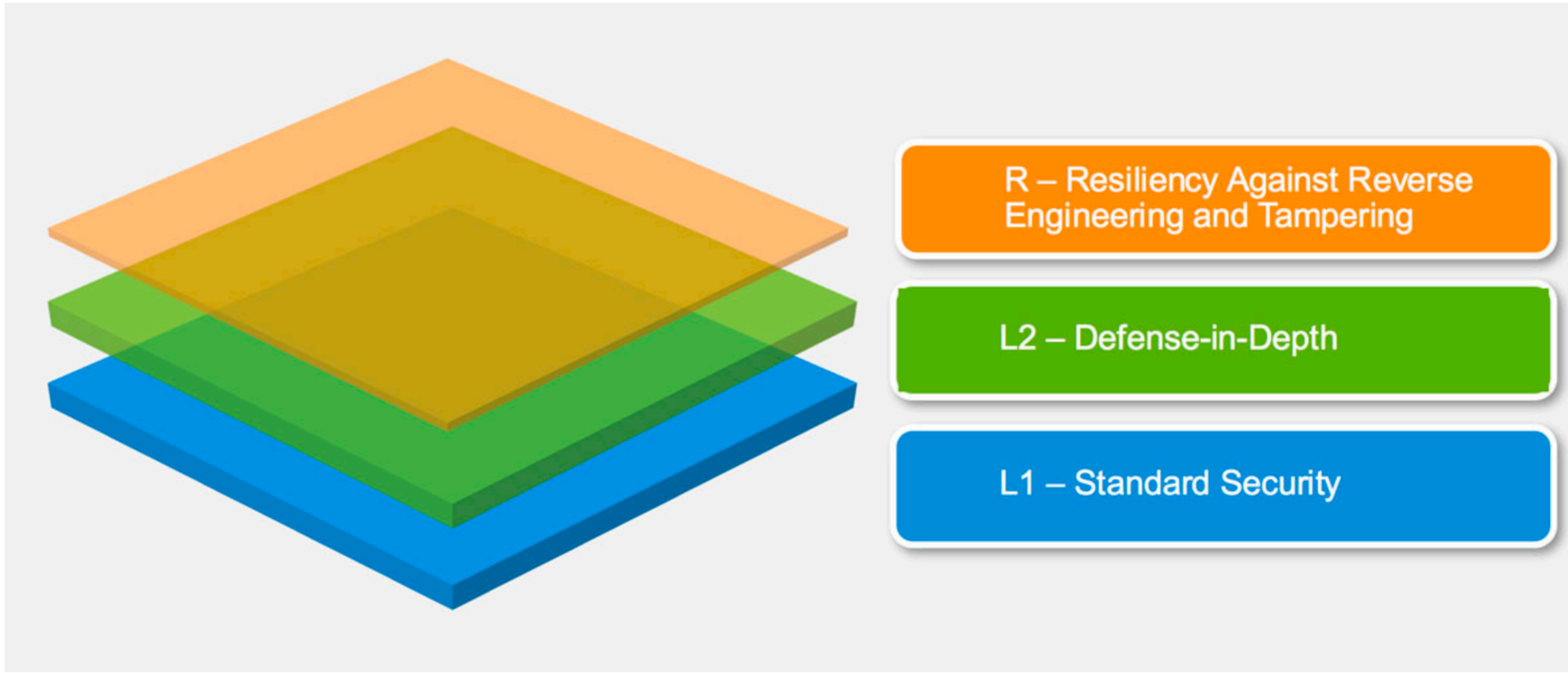


являются обязательными для всех уровней проверки.

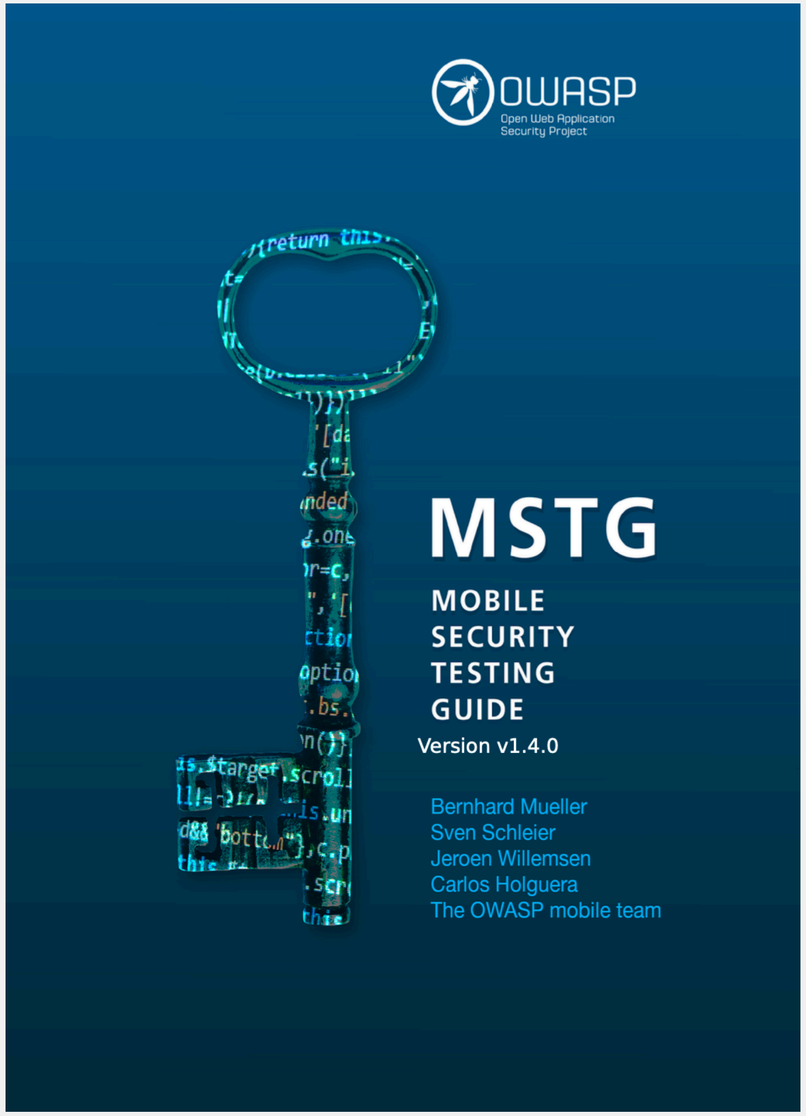
#	MSTG-ID	Описание	L1	L2
2.1	MSTG-STORAGE-1	Хранилище учетных данных системы должно использоваться надлежащим образом для хранения чувствительных данных, таких как персональные данные, данные пользователя для авторизации и криптографические ключи.	x	x
2.2	MSTG-STORAGE-2	Чувствительные данные хранятся только во внутреннем хранилище приложения, либо в системном хранилище авторизационных данных.	x	x
2.3	MSTG-STORAGE-3	Чувствительные данные не попадают в логи приложения.	x	x
2.4	MSTG-STORAGE-4	Никакие чувствительные данные не передаются третьей стороне, если это не является необходимой частью архитектуры.	x	x
2.5	MSTG-STORAGE-5	Кэш клавиатуры выключен для полей ввода чувствительных данных.	x	x
2.6	MSTG-STORAGE-6	Чувствительные данные недоступны для механизмов межпроцессного взаимодействия (IPC).	x	x

OWASP Mobile Application Security Verification Standard

OWASP Mobile Application Security Verification Standard v1.4.2



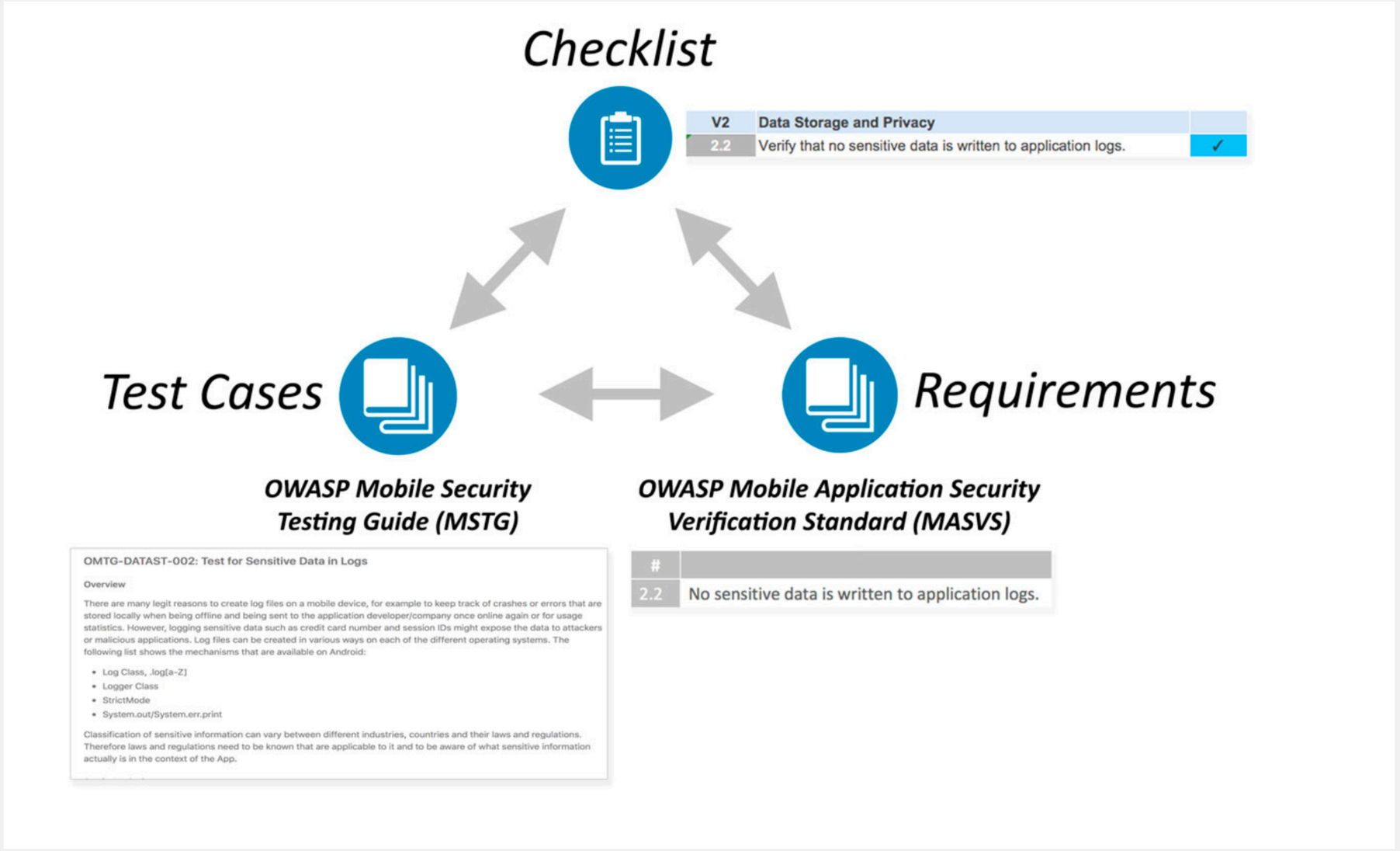
OWASP Mobile Application Security Testing Guide



OWASP Mobile Application Security Testing Guide

1. Общие тест кейсы
2. Специфика для android
3. Специфика для iOS

OWASP Mobile Application Security Checklist



OWASP Mobile Application Security Checklist

Data Storage and Privacy Requirements

ID	MSTG-ID	Detailed Verification Requirement	L1	L2	R	Android	iOS	Status
2.1	MSTG-STORAGE-1	Хранилище учетных данных системы должно использоваться надлежащим образом для хранения чувствительных данных, таких как персональные данные, данные пользователя для авторизации и криптографические ключи.	■	■		Test Case	Test Case	
2.2	MSTG-STORAGE-2	Чувствительные данные хранятся только во внутреннем хранилище приложения, либо в системном хранилище авторизационных данных.	■	■		Test Case	Test Case	
2.3	MSTG-STORAGE-3	Чувствительные данные не попадают в логи приложения.	■	■		Test Case	Test Case	
2.4	MSTG-STORAGE-4	Никакие чувствительные данные не передаются третьей стороне, если это не является необходимой частью архитектуры.	■	■		Test Case	Test Case	
2.5	MSTG-STORAGE-5	Кэш клавиатуры выключен для полей ввода чувствительных данных.	■	■		Test Case	Test Case	
2.6	MSTG-STORAGE-6	Чувствительные данные недоступны для механизмов межпроцессного взаимодействия (IPC).	■	■		Test Case	Test Case	
2.7	MSTG-STORAGE-7	Никакие чувствительные данные, такие как пароли или пин-коды, не видны через пользовательский интерфейс.	■	■		Test Case	Test Case	
2.8	MSTG-STORAGE-8	Никакие чувствительные данные не попадают в бэкапы, создаваемые операционной системой.		■		Test Case	Test Case	
2.9	MSTG-STORAGE-9	Приложение скрывает чувствительные данные с экрана, когда находится в фоновом режиме.		■		Test Case	Test Case	

OWASP Mobile Application Security Checklist

Data Storage and Privacy Requirements

ID	MSTG-ID	Detailed Verification Requirement	L1	L2	R	Android	iOS	Status
2.1	MSTG-STORAGE-1	Хранилище учетных данных системы должно использоваться надлежащим образом для хранения чувствительных данных, таких как персональные данные, данные пользователя для авторизации и криптографические ключи.	■	■		Test Case	Test Case	
2.2	MSTG-STORAGE-2	Чувствительные данные хранятся только во внутреннем хранилище приложения, либо в системном хранилище авторизационных данных.	■	■		Test Case	Test Case	
2.3	MSTG-S							
2.4	MSTG-S							
2.5	MSTG-STORAGE-5	Кэш клавиатуры выключен для полей ввода чувствительных данных.	■	■		Test Case	Test Case	
2.6	MSTG-STORAGE-6	Чувствительные данные недоступны для механизмов межпроцессного взаимодействия (IPC).	■	■		Test Case	Test Case	
2.7	MSTG-STORAGE-7	Никакие чувствительные данные, такие как пароли или пин-коды, не видны через пользовательский интерфейс.	■	■		Test Case	Test Case	
2.8	MSTG-STORAGE-8	Никакие чувствительные данные не попадают в бэкапы, создаваемые операционной системой.		■		Test Case	Test Case	
2.9	MSTG-STORAGE-9	Приложение скрывает чувствительные данные с экрана, когда находится в фоновом режиме.		■		Test Case	Test Case	

2.3 MSTG-STORAGE-3 Чувствительные данные не попадают в логи приложения.

OWASP Mobile Application Security Checklist

Testing Logs for Sensitive Data (MSTG-STORAGE-3)

Overview

This test case focuses on identifying any sensitive application data within both system and application logs. The following checks should be performed:

- Analyze source code for logging related code.
- Check application data directory for log files.
- Gather system messages and logs and analyze for any sensitive data.

As a general recommendation to avoid potential sensitive application data leakage, logging statements should be removed from production releases unless deemed necessary to the application or explicitly identified as safe, e.g. as a result of a security audit.

Static Analysis

Applications will often use the Log Class and Logger Class to create logs. To discover this, you should audit the application's source code for any such logging classes. These can often be found by searching for the following keywords:

- Functions and classes, such as:
  - `android.util.Log`
  - `Log.d` | `Log.e` | `Log.i` | `Log.v` | `Log.w` | `Log.wtf`
  - `Logger`

OWASP Mobile Application Security Checklist

Testing Logs for Sensitive Data (MSTG-STORAGE-3)

Overview

This test case focuses on identifying any sensitive application data within both system and application logs. The following checks should be performed:

- Analyze source code for logging related code.
- Check application data directory for log files.
- Gather system messages and logs and analyze for any sensitive data.

As a general recommendation to avoid potential sensitive application data leakage, logging statements should be removed from production releases unless deemed necessary to the application or explicitly identified as safe, e.g. as a result of a security audit.

Static Analysis

Applications will often use the Log Class and Logger Class to create logs. To discover this, you should audit the application's source code for any such logging classes. These can often be found by searching for the following keywords:

- Functions and classes, such as:
  - `android.util.Log`
  - `Log.d` | `Log.e` | `Log.i` | `Log.v` | `Log.w` | `Log.wtf`
  - `Logger`

OWASP Mobile Application Security Checklist

Dynamic Analysis

Use all the mobile app functions at least once, then identify the application's data directory and look for log files (`/data/data/<package-name>`). Check the application logs to determine whether log data has been generated; some mobile applications create and store their own logs in the data directory.

Many application developers still use `System.out.println` or `printStackTrace` instead of a proper logging class. Therefore, your testing strategy must include all output generated while the application is starting, running and closing. To determine what data is directly printed by `System.out.println` or `printStackTrace` , you can use `Logcat` as explained in the chapter "Basic Security Testing", section "Monitoring System Logs".

Remember that you can target a specific app by filtering the Logcat output as follows:

```
$ adb logcat | grep "$(adb shell ps | grep <package-name> | awk '{print $2}')"
```

> If you already know the app PID you may give it directly using `--pid` flag.

You may also want to apply further filters or regular expressions (using `logcat` 's regex flags `-e <expr>, --regex=<expr>` for example) if you expect certain strings or patterns to come up in the logs.

Чего мы хотим?

1. smoke / sanity план тестирования безопасности

2. принцип 80/20

Чего мы хотим?

1. smoke / sanity план тестирования безопасности (короткий)

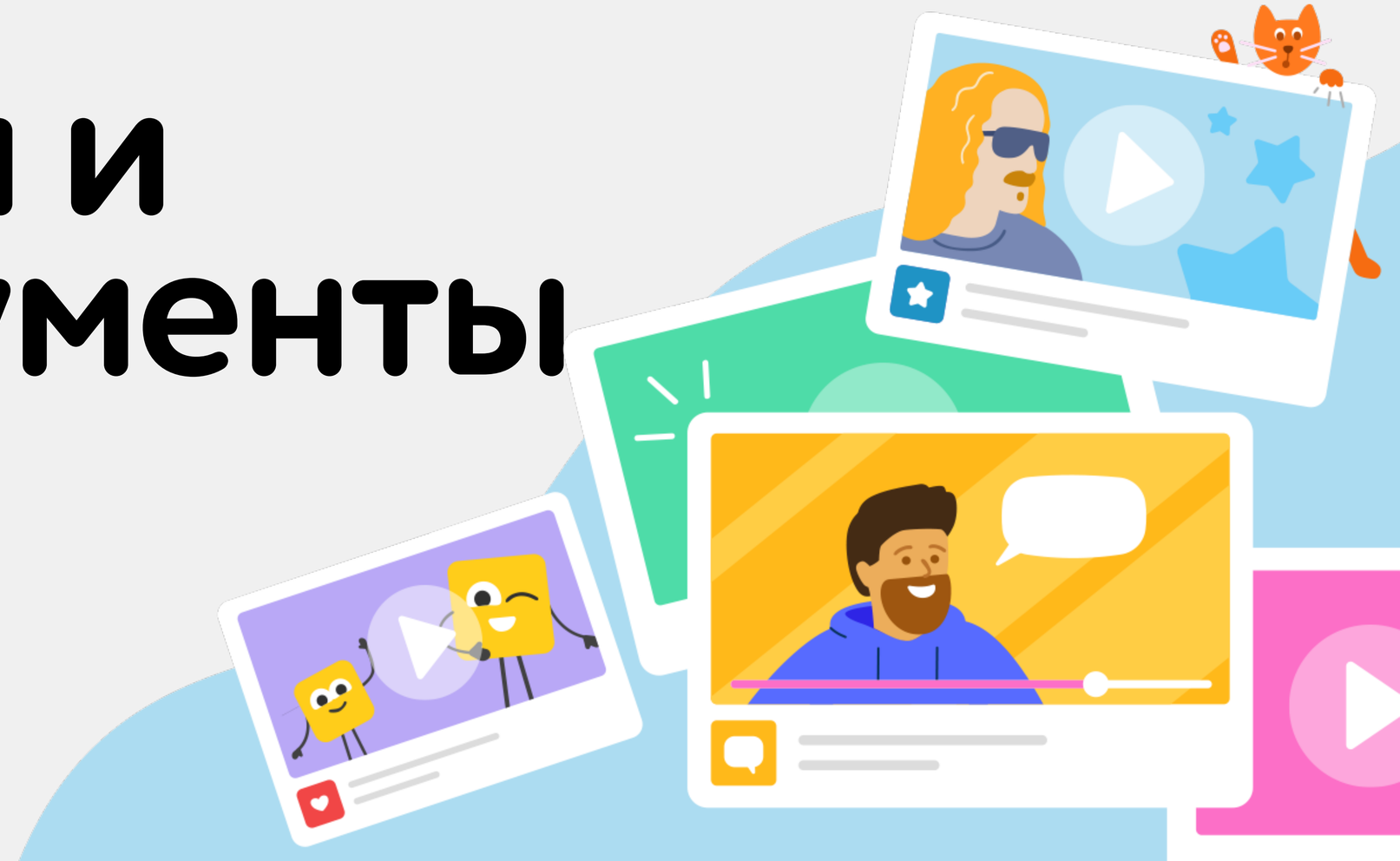
2. принцип 80/20

Чего мы хотим?

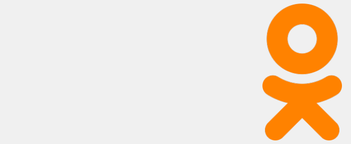
1. smoke / sanity план тестирования безопасности (короткий)

2. принцип 80/20 (автоматизированный)

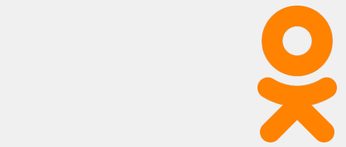
Задачи и инструменты



Задачи



Задачи



1. Получить исходный код

Задачи

1. Получить исходный код

2. Исследовать трафик

Задачи

1. Получить исходный код
2. Исследовать трафик
3. Исследовать файловую систему

Задачи

1. Получить исходный код
2. Исследовать трафик
3. Исследовать файловую систему
4. Исследовать компоненты приложения и взаимодействие с ОС

Как тестировать

Как тестировать

1. Изучать исходный код (статический анализ)

Как тестировать

1. Изучать исходный код (статический анализ)
2. Изучать поведение приложения (динамический анализ)

Автоматизируем

1. Статический анализ (Static Application Security Testing, SAST)

2. Динамический анализ (Dynamic Appliciton Security Testing, DAST)

Дальше — все про Android

(но для iOS аналогично)

Автоматизируем

1. Статический анализ (Static Application Security Testing, SAST)

2. Динамический анализ (Dynamic Appliciton Security Testing, DAST)

Автоматизируем

1. Статический анализ (Static Application Security Testing, SAST)

QARK, MobSF, find-security-bugs

2. Динамический анализ (Dynamic Appliciton Security Testing, DAST)

Автоматизируем

1. Статический анализ (Static Application Security Testing, SAST)

QARK, MobSF, find-security-bugs

2. Динамический анализ (Dynamic Appliciton Security Testing, DAST)

drozer, MobSF

Автоматизируем

1. Статический анализ (Static Application Security Testing, SAST)

QARK, **MobSF**, find-security-bugs

2. Динамический анализ (Dynamic Appliciton Security Testing, DAST)

drozer, MobSF

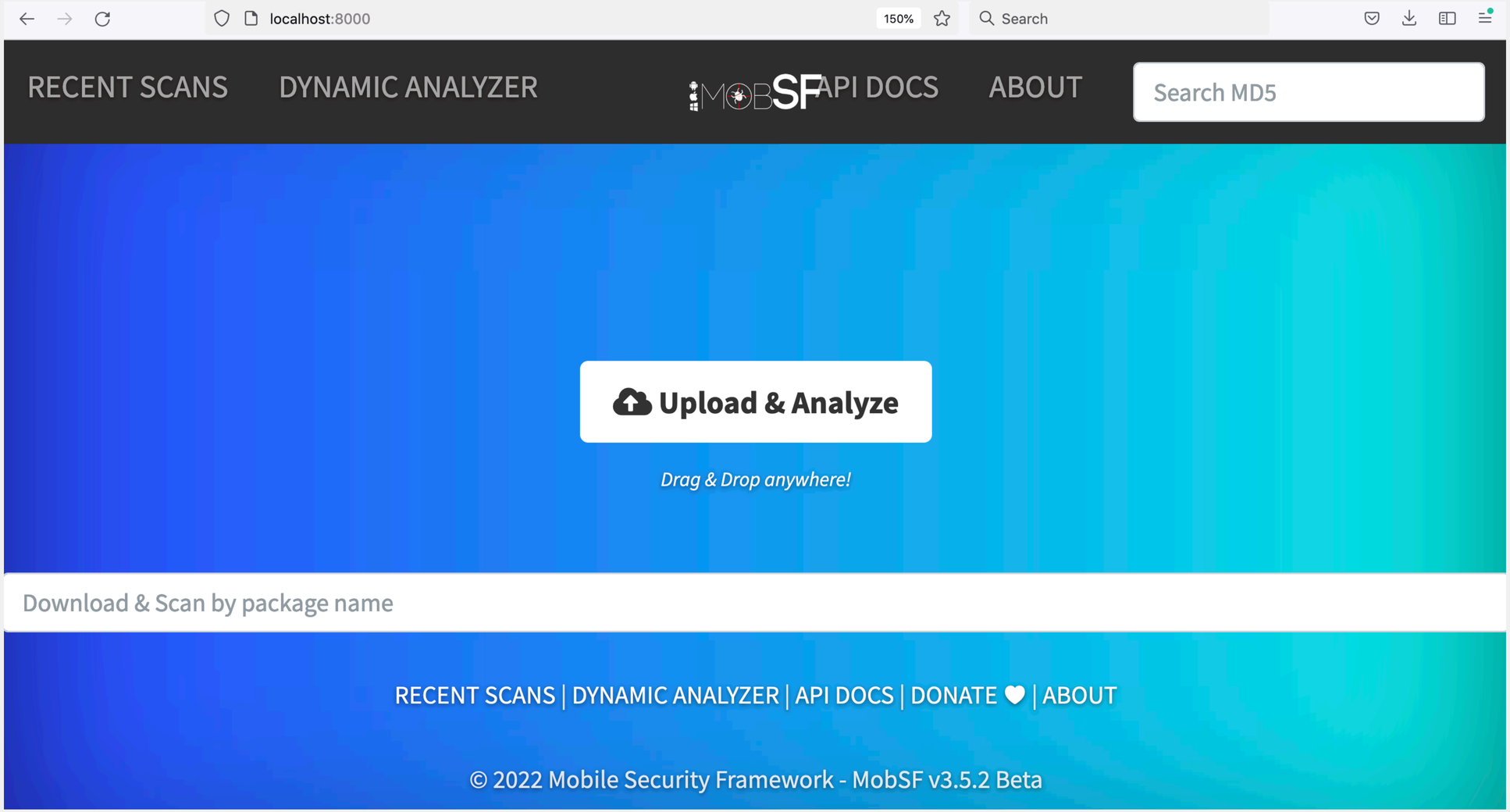
Mobile Security Framework (MobSF)

https://github.com/MobSF/Mobile-Security-Framework-MobSF

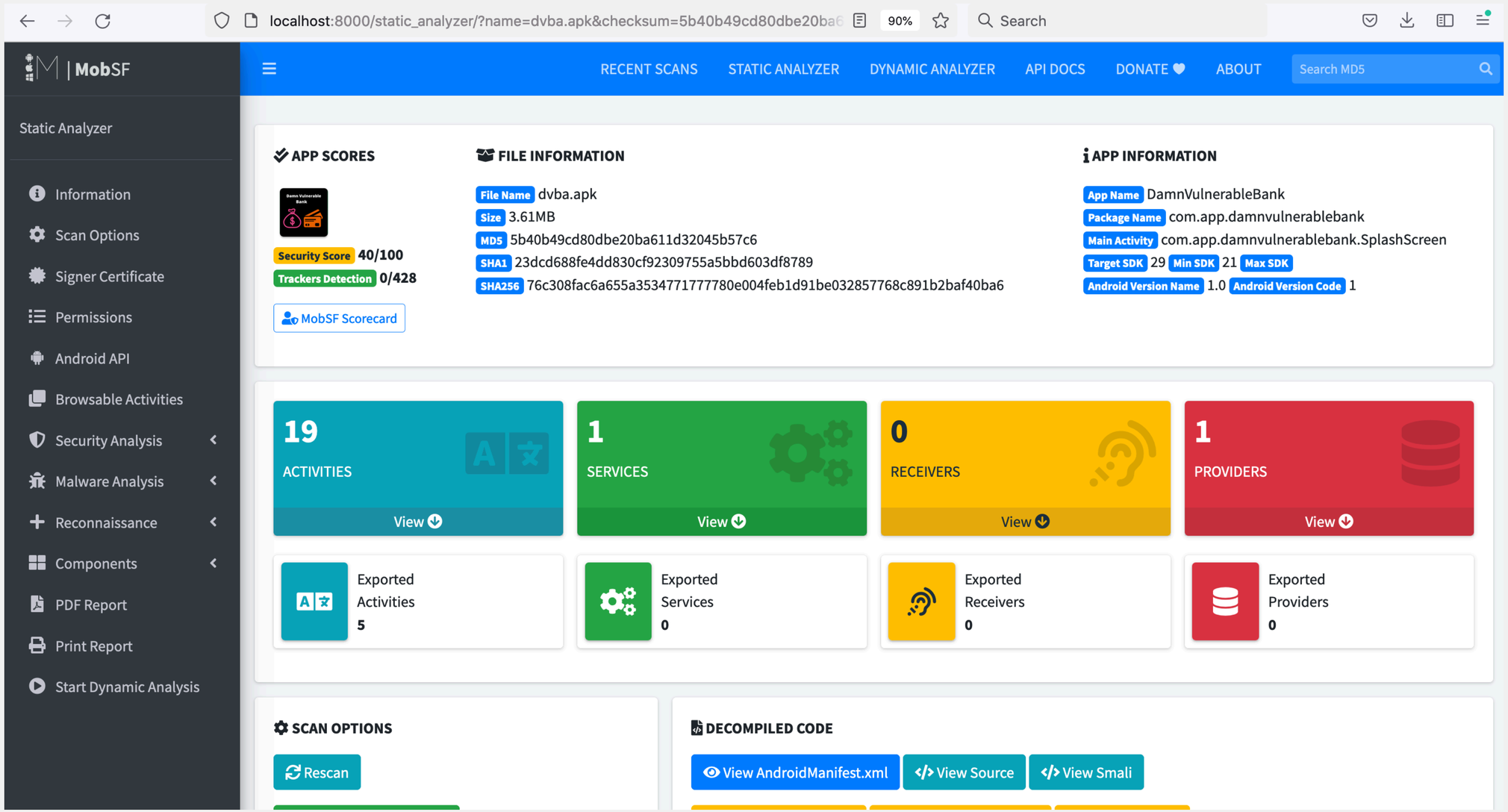
Mobile Security Framework (MobSF)

```
asvatikova:heisenbug alexandra.svatikova$ docker run -it --rm -p 8000:8000 opensecurity/mobile-security-framework-mobsf:latest
[INFO] 14/Jun/2022 05:03:54 -

  __  __       _     ____  _____        _____  ____
 |  \/  | ___ | |__/ ___||  ___|_   _|___ / | ___|
 | |\/| |/ _ \| '_ \___ \| |_  \ \ / / |_ \ |___ \
 | |  | | (_) | |_) |__) |  _|  \ V / ___) | ___) |
 |_|  |_|\___/|_.__/____/|_|     \_/ |____(_)____/

[INFO] 14/Jun/2022 05:03:54 - Mobile Security Framework v3.5.2 Beta
REST API Key: 674082cf545179ce1f5a2ca77d2aceaca5d451a7894cf89d701fbcd790a8e0da
[INFO] 14/Jun/2022 05:03:54 - OS: Linux
[INFO] 14/Jun/2022 05:03:54 - Platform: Linux-5.10.104-linuxkit-x86_64-with-glibc2.29
[INFO] 14/Jun/2022 05:03:54 - Dist: ubuntu 20.04 Focal Fossa
[INFO] 14/Jun/2022 05:03:54 - MobSF Basic Environment Check
No changes detected
[INFO] 14/Jun/2022 05:03:54 - Checking for Update.
[INFO] 14/Jun/2022 05:03:55 - No updates available.
[INFO] 14/Jun/2022 05:03:55 -
```

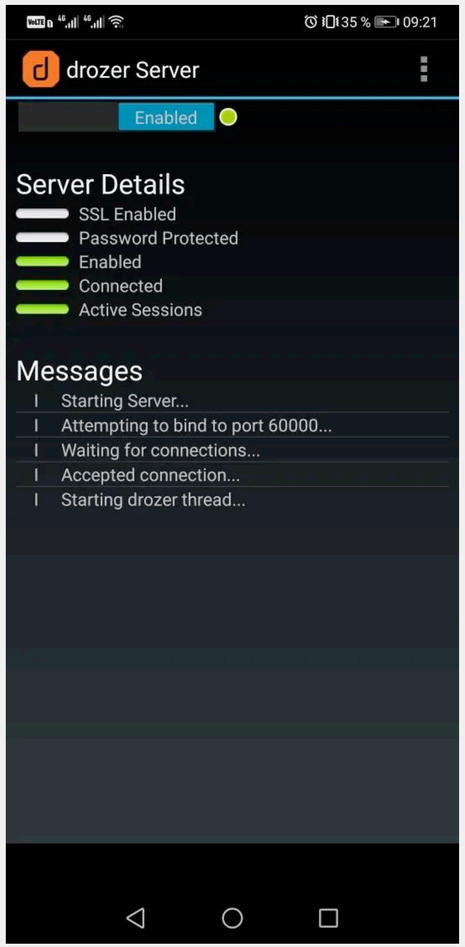
Mobile Security Framework (MobSF)



Mobile Security Framework (MobSF)



drozer




```
Selecting 35ca79a1e562f09d (HUAWEI CLT-L29 10)

            ..                    ..:.
           ..o..                  .r..
            ..a..  . ....... .  ..nd
              ro..idsnemesisand..pr
              .otectorandroidsneme.
           .,sisandprotectorandroids+.
         ..nemesisandprotectorandroidsn:.
        .emesisandprotectorandroidsnemes..
      ..isandp,..,rotectorandro,..,idsnem.
      .isisandp..rotectorandroid..snemisis.
      ,andprotectorandroidsnemisisandprotec.
     .torandroidsnemesisandprotectorandroid.
     .snemisisandprotectorandroidsnemesisan:
     .dprotectorandroidsnemesisandprotector.

drozer Console (v2.4.2)
dz>
dz>
```

https://github.com/FSecureLABS/drozer

5 Чеклист

Тестовое приложение

https://github.com/rewanthtammana/Damn-Vulnerable-Bank

Data Storage and Privacy Requirements

Data Storage and Privacy Requirements

ID	MSTG-ID	Detailed Verification Requirement	L1	L2	R	Android
2.1	MSTG-STORAGE-1	Хранилище учетных данных системы должно использоваться надлежащим образом для хранения чувствительных данных, таких как персональные данные, данные пользователя для авторизации и криптографические ключи.				Test Case
2.2	MSTG-STORAGE-2	Чувствительные данные хранятся только во внутреннем хранилище приложения, либо в системном хранилище авторизационных данных.				Test Case
2.3	MSTG-STORAGE-3	Чувствительные данные не попадают в логи приложения.				Test Case
2.4	MSTG-STORAGE-4	Никакие чувствительные данные не передаются третьей стороне, если это не является необходимой частью архитектуры.				Test Case
2.5	MSTG-STORAGE-5	Кэш клавиатуры выключен для полей ввода чувствительных данных.				Test Case
2.6	MSTG-STORAGE-6	Чувствительные данные недоступны для механизмов межпроцессного взаимодействия (IPC).				Test Case
2.7	MSTG-STORAGE-7	Никакие чувствительные данные, такие как пароли или пин-коды, не видны через пользовательский интерфейс.				Test Case

Data Storage and Privacy Requirements

Data Storage and Privacy Requirements

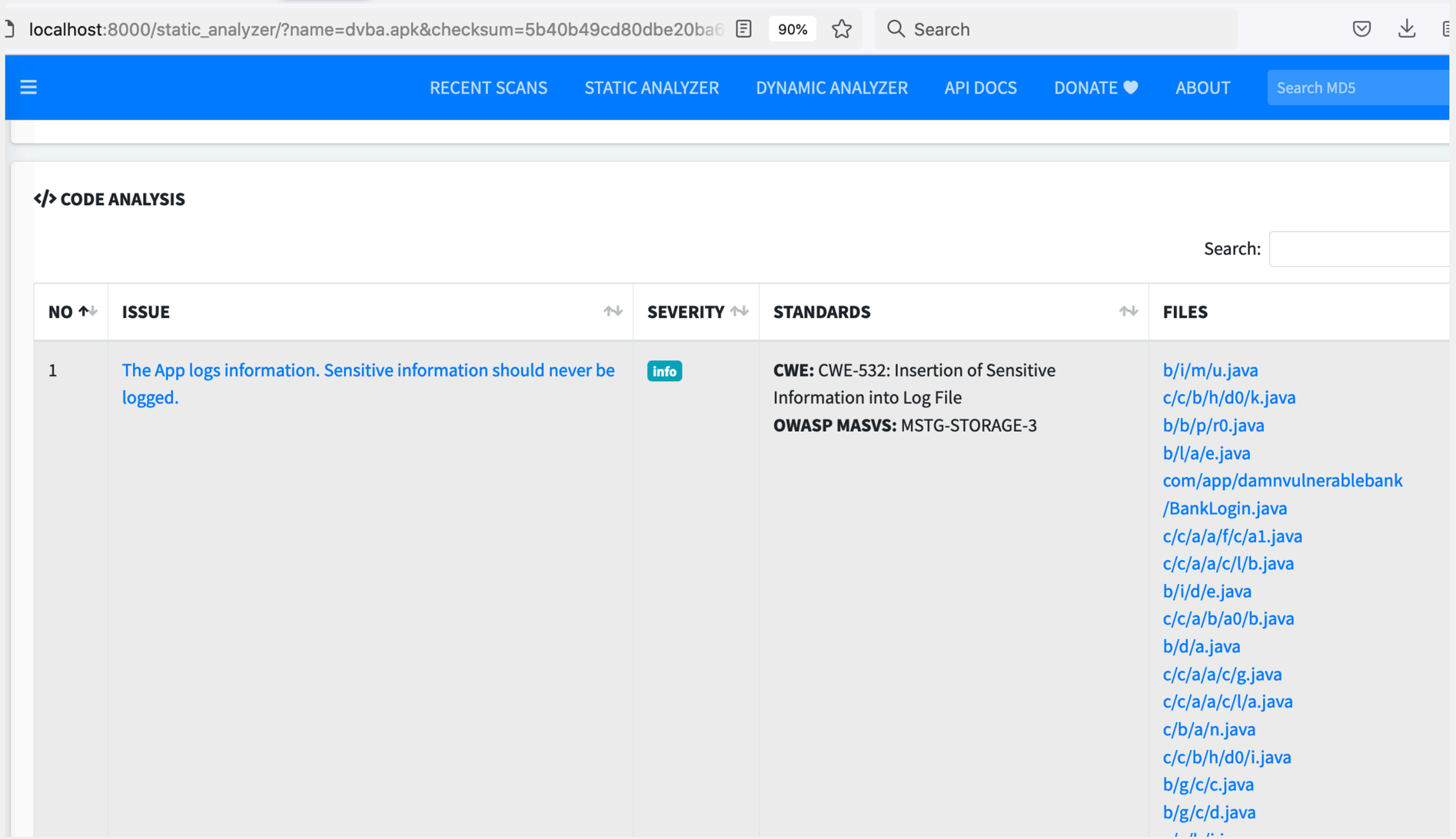
ID	MSTG-ID	Detailed Verification Requirement	L1	тест
2.1	MSTG-STORAGE-1	Хранилище учетных данных системы должно использоваться надлежащим образом для хранения чувствительных данных, таких как персональные данные, данные пользователя для авторизации и криптографические ключи.	■	
2.2	MSTG-STORAGE-2	Чувствительные данные хранятся только во внутреннем хранилище приложения, либо в системном хранилище авторизационных данных.	■	
2.3	MSTG-STORAGE-3	Чувствительные данные не попадают в логи приложения.	■	
2.4	MSTG-STORAGE-4	Никакие чувствительные данные не передаются третьей стороне, если это не является необходимой частью архитектуры.	■	
2.5	MSTG-STORAGE-5	Кэш клавиатуры выключен для полей ввода чувствительных данных.	■	
2.6	MSTG-STORAGE-6	Чувствительные данные недоступны для механизмов межпроцессного взаимодействия (IPC).	■	
2.7	MSTG-STORAGE-7	Никакие чувствительные данные, такие как пароли или пин-коды, не видны через пользовательский интерфейс.	■	

Data Storage and Privacy Requirements

Data Storage and Privacy Requirements

ID	MSTG-ID	Detailed Verification Requirement	L1	тест
2.1	MSTG-STORAGE-1	Хранилище учетных данных системы должно использоваться надлежащим образом для хранения чувствительных данных, таких как персональные данные, данные пользователя для авторизации и криптографические ключи.		
2.2	MSTG-STORAGE-2	Чувствительные данные хранятся только во внутреннем хранилище приложения, либо в системном хранилище авторизационных данных.		
2.3	MSTG-STORAGE-3	Чувствительные данные не попадают в логи приложения.		
2.4	MSTG-STORAGE-4	Никакие чувствительные данные не передаются третьей стороне, если это не является необходимой частью архитектуры.		
2.5	MSTG-STORAGE-5	Кэш клавиатуры выключен для полей ввода чувствительных данных.		
2.6	MSTG-STORAGE-6	Чувствительные данные недоступны для механизмов межпроцессного взаимодействия (IPC).		
2.7	MSTG-STORAGE-7	Никакие чувствительные данные, такие как пароли или пин-коды, не видны через пользовательский интерфейс.		

Data Storage and Privacy Requirements



Data Storage and Privacy Requirements

2	App can read/write to External Storage. Any App can read data written to External Storage.	warning	**CWE:** CWE-276: Incorrect Default Permissions **OWASP Top 10:** M2: Insecure Data Storage **OWASP MASVS:** MSTG-STORAGE-2	com/app/damnvulnerablebank/MainActivity.java

Data Storage and Privacy Requirements

```
    }
    if (i == 0 && constraintLayout2 != null) {
        i = d(constraintLayout2, trim);
    }
    if (i == 0) {
        try {
            i = j.class.getField(trim).getInt(null);
        } catch (Exception unused) {
        }
    }
    if (i == 0) {
        i = this.d.getResources().getIdentifier(trim, "id", this.d.getPackageName());
    }
    if (i != 0) {
        this.h.put(Integer.valueOf(i), trim);
        b(i);
        return;
    }
    Log.w("ConstraintHelper", "Could not find id of \"" + trim + "\"");
```

Data Storage and Privacy Requirements

```
Intent intent;
try {
    JSONObject jSONObject2 = new JSONObject(e.a(jSONObject.get("enc_data").toString()));
    if (jSONObject2.getJSONObject("status").getInt("code") != 200) {
        Context applicationContext = BankLogin.this.getApplicationContext();
        Toast.makeText(applicationContext, "Error: " + jSONObject2.getJSONObject("data").getString("message"), 0).show();
        bankLogin = BankLogin.this;
        intent = new Intent(BankLogin.this, BankLogin.class);
    } else {
        String string = jSONObject2.getJSONObject("data").getString("accessToken");
        SharedPreferences sharedPreferences = BankLogin.this.getSharedPreferences("jwt", 0);
        Log.d("accesstoken", string);
        sharedPreferences.edit().putString("accesstoken", string).apply();
        sharedPreferences.edit().putBoolean("isloggedin", true).apply();
        bankLogin = BankLogin.this;
        intent = new Intent(BankLogin.this, Dashboard.class);
    }
    bankLogin.startActivity(intent);
} catch (JSONException e) {
    e.printStackTrace();
}
```

Data Storage and Privacy Requirements

Data Storage and Privacy Requirements

ID	MSTG-ID	Detailed Verification Requirement	L1	тест
2.1	MSTG-STORAGE-1	Хранилище учетных данных системы должно использоваться надлежащим образом для хранения чувствительных данных, таких как персональные данные, данные пользователя для авторизации и криптографические ключи.		MobSF
2.2	MSTG-STORAGE-2	Чувствительные данные хранятся только во внутреннем хранилище приложения, либо в системном хранилище авторизационных данных.		MobSF
2.3	MSTG-STORAGE-3	Чувствительные данные не попадают в логи приложения.		MobSF
2.4	MSTG-STORAGE-4	Никакие чувствительные данные не передаются третьей стороне, если это не является необходимой частью архитектуры.		MobSF
2.5	MSTG-STORAGE-5	Кэш клавиатуры выключен для полей ввода чувствительных данных.		манифест
2.6	MSTG-STORAGE-6	Чувствительные данные недоступны для механизмов межпроцессного взаимодействия (IPC).		
2.7	MSTG-STORAGE-7	Никакие чувствительные данные, такие как пароли или пин-коды, не видны через пользовательский интерфейс.		тест

Authentication and Session Management Requirements

Authentication and Session Management Requirements

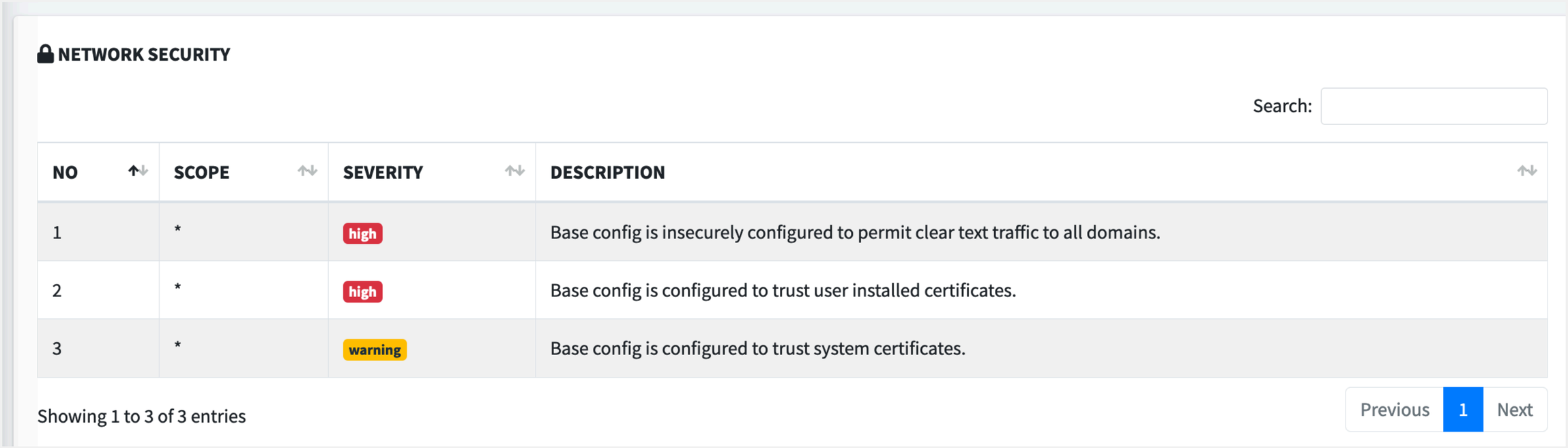
ID	MSTG-ID	Detailed Verification Requirement	L1	тест
4.1	MSTG-AUTH-1	Если приложение предоставляет пользователям доступ к удалённым сервисам, на бэкенде должна быть реализована аутентификация, например, по логину и паролю.		тест
4.2	MSTG-AUTH-2	Если используются сессии, бекэнд случайно генерирует идентификаторы сессии для аутентификации клиентских запросов без отправки данных учётной записи.		тест
4.3	MSTG-AUTH-3	Если используется аутентификация на основе токена, сервер предоставляет токен, подписанный с использованием безопасного криптоалгоритма.		тест
4.4	MSTG-AUTH-4	Бэкенд удаляет существующую сессию, когда пользователь выходит из системы.		тест
4.5	MSTG-AUTH-5	На сервере реализована парольная политика.		тест
4.6	MSTG-AUTH-6	На сервере реализован механизм защиты от перебора авторизационных данных.		тест
4.8	MSTG-AUTH-8	Сессии становятся невалидными на бэкенде после определенного периода бездействия, срок действия токена истекает.		тест

Network Communication Requirements

Network Communication Requirements

ID	MSTG-ID	Detailed Verification Requirement	L1	тест
5.1	MSTG-NETWORK-1	Данные, передаваемые по сети, шифруются с использованием TLS. Безопасный канал используется для всех сервисов приложения.		
5.2	MSTG-NETWORK-2	Настройки TLS соответствуют современным лучшим практикам, или максимально приближены к ним, если операционная система не поддерживает рекомендуемые стандарты.		
5.3	MSTG-NETWORK-3	Приложение верифицирует X.509 сертификаты сервера во время установления защищённого канала. Принимаются только сертификаты, подписанные доверенным удостоверяющим центром (CA).		

Network Communication Requirements



NETWORK SECURITY

Search:

NO	SCOPE	SEVERITY	DESCRIPTION
1	*	high	Base config is insecurely configured to permit clear text traffic to all domains.
2	*	high	Base config is configured to trust user installed certificates.
3	*	warning	Base config is configured to trust system certificates.

Showing 1 to 3 of 3 entries

Previous 1 Next

Network Communication Requirements

URLS

Search:

URL	FILE
http://localhost	c/c/a/a/f/c/n1.java
http://schemas.android.com/apk/res/android	a/a/a/a/a.java
https://damn-vulnerable-bank.firebaseio.com	Android String Resource
https://plus.google.com/	c/c/a/a/c/l/f0.java
https://www.xe.com/	com/app/damnvulnerablebank/CurrencyRates.java

Showing 1 to 5 of 5 entries

Previous 1 Next

Network Communication Requirements

Network Communication Requirements

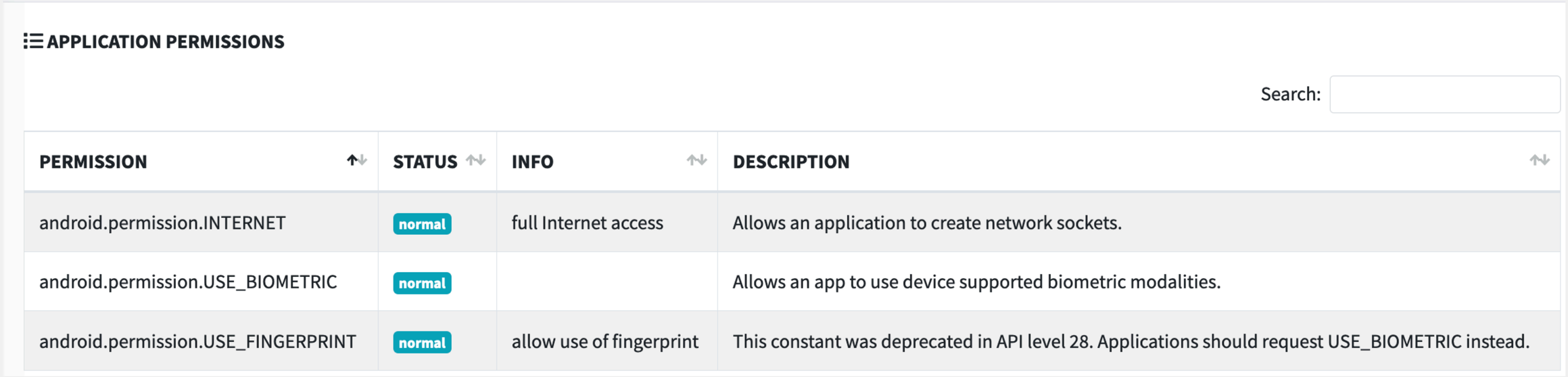
ID	MSTG-ID	Detailed Verification Requirement	L1	тест
5.1	MSTG-NETWORK-1	Данные, передаваемые по сети, шифруются с использованием TLS. Безопасный канал используется для всех сервисов приложения.		MobSF
5.2	MSTG-NETWORK-2	Настройки TLS соответствуют современным лучшим практикам, или максимально приближены к ним, если операционная система не поддерживает рекомендуемые стандарты.		—
5.3	MSTG-NETWORK-3	Приложение верифицирует X.509 сертификаты сервера во время установления защищённого канала. Принимаются только сертификаты, подписанные доверенным удостоверяющим центром (CA).		MobSF

Platform Interaction Requirements

Platform Interaction Requirements

ID	MSTG-ID	Detailed Verification Requirement	L1	тест
6.1	MSTG-PLATFORM-1	Приложение запрашивает минимально необходимый набор разрешений.		
6.2	MSTG-PLATFORM-2	Все данные, поступающие из внешних источников и от пользователя, валидируются и санитизируются. Сюда входят данные, полученные через пользовательский интерфейс, механизмы IPC (такие как intent-ы, кастомные URL-схемы) и из сети.		
6.3	MSTG-PLATFORM-3	Приложение не экспортирует чувствительные данные через кастомные URL-схемы, если эти механизмы не защищены должным образом.		
6.4	MSTG-PLATFORM-4	Приложение не экспортирует чувствительные данные через IPC механизмы без должной защиты.		
6.5	MSTG-PLATFORM-5	JavaScript отключен в компонентах WebView, если в нём нет необходимости.		
6.6	MSTG-PLATFORM-6	WebViews сконфигурирован с поддержкой минимального набора протоколов (в идеале только https). Поддержка потенциально опасных URL-схем (таких как: file, tel и app-id) отключена.		
6.7	MSTG-PLATFORM-7	Если нативные методы приложения используются WebView, верифицировать, что исполняются только Javascript объекты данного приложения.		
6.8	MSTG-PLATFORM-8	Десериализация объектов, если она есть, реализована с использованием безопасного API.		

Platform Interaction Requirements

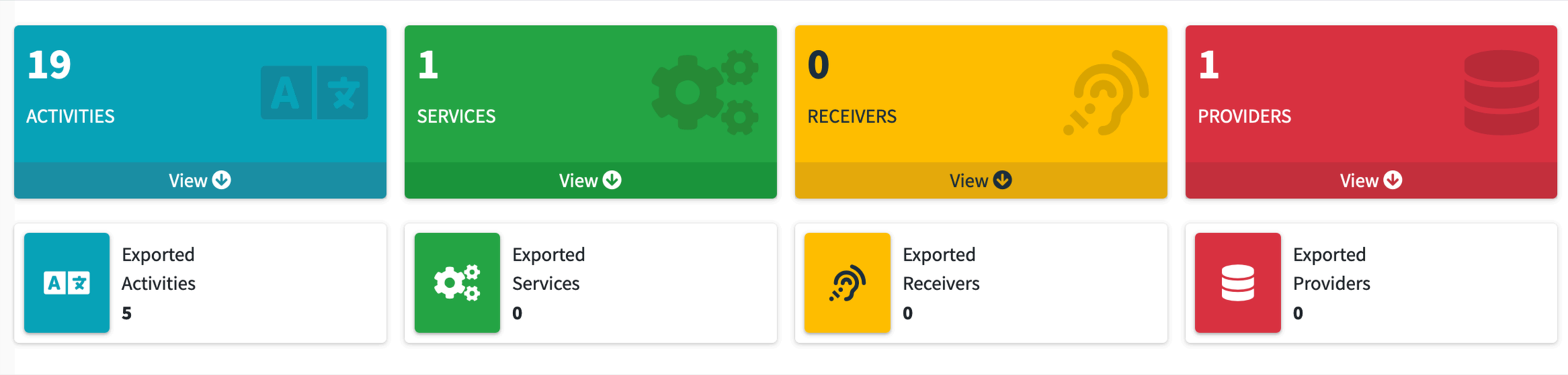


APPLICATION PERMISSIONS

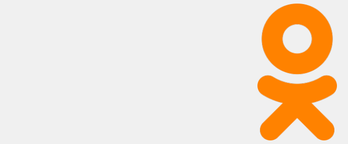
Search:

PERMISSION	STATUS	INFO	DESCRIPTION
android.permission.INTERNET	normal	full Internet access	Allows an application to create network sockets.
android.permission.USE_BIOMETRIC	normal		Allows an app to use device supported biometric modalities.
android.permission.USE_FINGERPRINT	normal	allow use of fingerprint	This constant was deprecated in API level 28. Applications should request USE_BIOMETRIC instead.

Platform Interaction Requirements



Platform Interaction Requirements



```
dz> run app.package.attacksurface com.app.damnvulnerablebank
Attack Surface:
  6 activities exported
  0 broadcast receivers exported
  0 content providers exported
  0 services exported
dz>
```

Platform Interaction Requirements

```
dz> run app.activity.info -a com.app.damnvulnerablebank
Package: com.app.damnvulnerablebank
  com.app.damnvulnerablebank.CurrencyRates
    Permission: null
  com.app.damnvulnerablebank.SendMoney
    Permission: null
  com.app.damnvulnerablebank.ViewBalance
    Permission: null
  com.app.damnvulnerablebank.SplashScreen
    Permission: null
  androidx.biometric.DeviceCredentialHandlerActivity
    Permission: null
  com.google.firebase.auth.internal.FederatedSignInActivity
    Permission: com.google.firebase.auth.api.gms.permission.LAUNCH_FEDERATED_SIGN_IN
```

Platform Interaction Requirements

```
dz> help app.activity.start

usage: run app.activity.start [-h] [--action ACTION] [--category CATEGORY]
                [--component PACKAGE COMPONENT] [--data-uri DATA_URI]
                [--extra TYPE KEY VALUE] [--flags FLAGS [FLAGS ...]]
                [--mimetype MIMETYPE]
```

Platform Interaction Requirements

Platform Interaction Requirements

ID	MSTG-ID	Detailed Verification Requirement	L1	тест
6.1	MSTG-PLATFORM-1	Приложение запрашивает минимально необходимый набор разрешений.		MobSF
6.2	MSTG-PLATFORM-2	Все данные, поступающие из внешних источников и от пользователя, валидируются и санитизируются. Сюда входят данные, полученные через пользовательский интерфейс, механизмы IPC (такие как intent-ы, кастомные URL-схемы) и из сети.		drozer
6.3	MSTG-PLATFORM-3	Приложение не экспортирует чувствительные данные через кастомные URL-схемы, если эти механизмы не защищены должным образом.		MobSF
6.4	MSTG-PLATFORM-4	Приложение не экспортирует чувствительные данные через IPC механизмы без должной защиты.		drozer
6.5	MSTG-PLATFORM-5	JavaScript отключен в компонентах WebView, если в нём нет необходимости.		MobSF
6.6	MSTG-PLATFORM-6	WebViews сконфигурирован с поддержкой минимального набора протоколов (в идеале только https). Поддержка потенциально опасных URL-схем (таких как: file, tel и app-id) отключена.		MobSF
6.7	MSTG-PLATFORM-7	Если нативные методы приложения используются WebView, верифицировать, что исполняются только Javascript объекты данного приложения.		MobSF
6.8	MSTG-PLATFORM-8	Десериализация объектов, если она есть, реализована с использованием безопасного API.		MobSF

Выводы

Выводы

1. Получили сравнительно короткий чеклист

Выводы

1. Получили сравнительно короткий чеклист

2. 2/3 можно автоматизировать

Выводы

1. Получили сравнительно короткий чеклист

2. 2/3 можно автоматизировать

3. Нужно понимать как устроено приложение

Выводы

1. Получили сравнительно короткий чеклист

2. 2/3 можно автоматизировать

3. Нужно понимать как устроено приложение

4. Начинать со статического анализа (MobSF)

https://github.com/alexandra-s/heisenbug-2022

Ссылки

1. OWASP Mobile Project https://owasp.org/www-project-mobile-security/
2. OWASP Mobile Application Security Verification Standard https://github.com/OWASP/owasp-masvs
3. OWASP Mobile Security Testing Guide https://github.com/OWASP/owasp-mstg

    + checklist https://github.com/OWASP/owasp-mstg/releases/tag/v1.4.0

4. Testing tools list https://github.com/OWASP/owasp-mstg/blob/master/Document/0x08-Testing-Tools.md
5. QARK https://github.com/linkedin/qark
6. MobSF https://github.com/MobSF/Mobile-Security-Framework-MobSF
7. Drozer https://github.com/FSecureLABS/drozer
8. Damn Vulnerable Bank https://github.com/rewanthtammana/Damn-Vulnerable-Bank


одноклассники