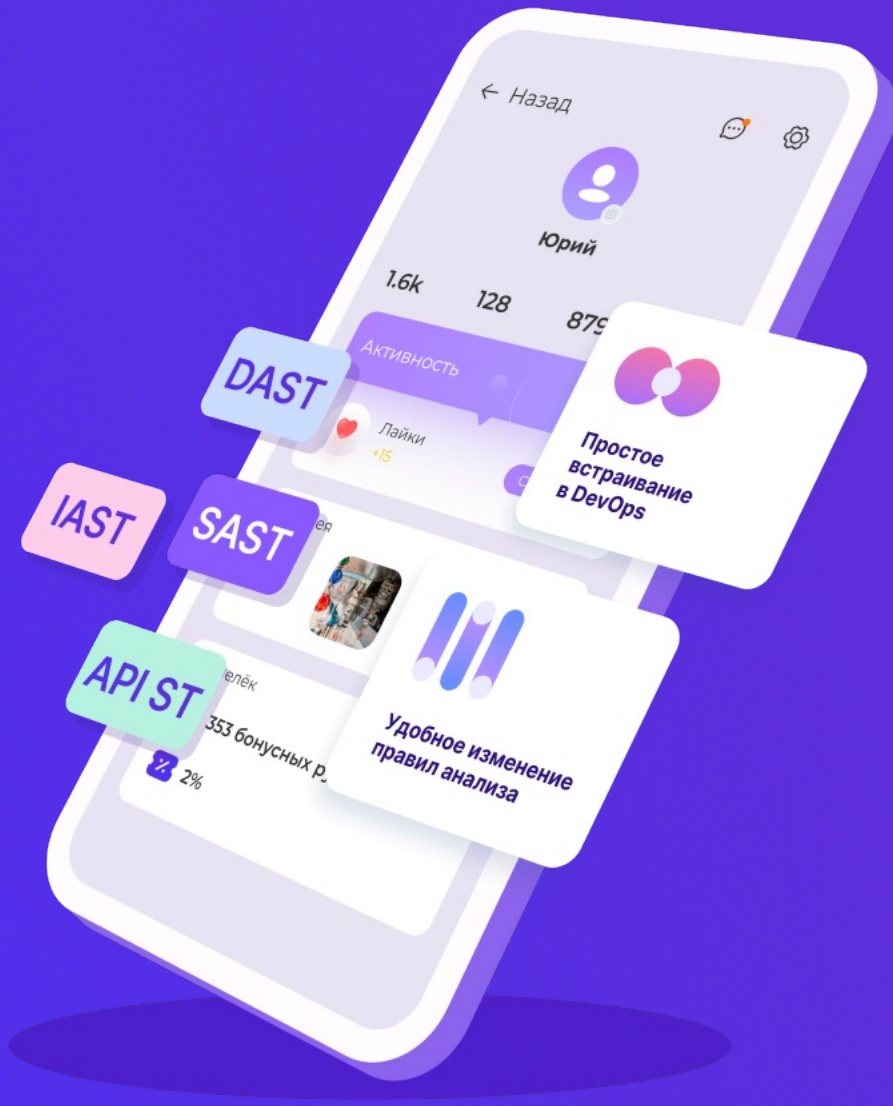
СТИНГРЕЙ

Безопасность — это просто.

Как улучшить защищенность мобильного приложения в процессе тестирования



Шабалин Юрий
Стингрей Технолоджиз,
Генеральный директор

СТИНГРЕЙ

Whoami

Юрий Шабалин

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

Тестирование и анализ защищенности, в чем отличие?

СТИНГРЕЙ

Not all **Software Quality** issues are **AppSec** issues.

But all **AppSec** issues are **Software Quality** issues.

Sherif Mansour, Expedia

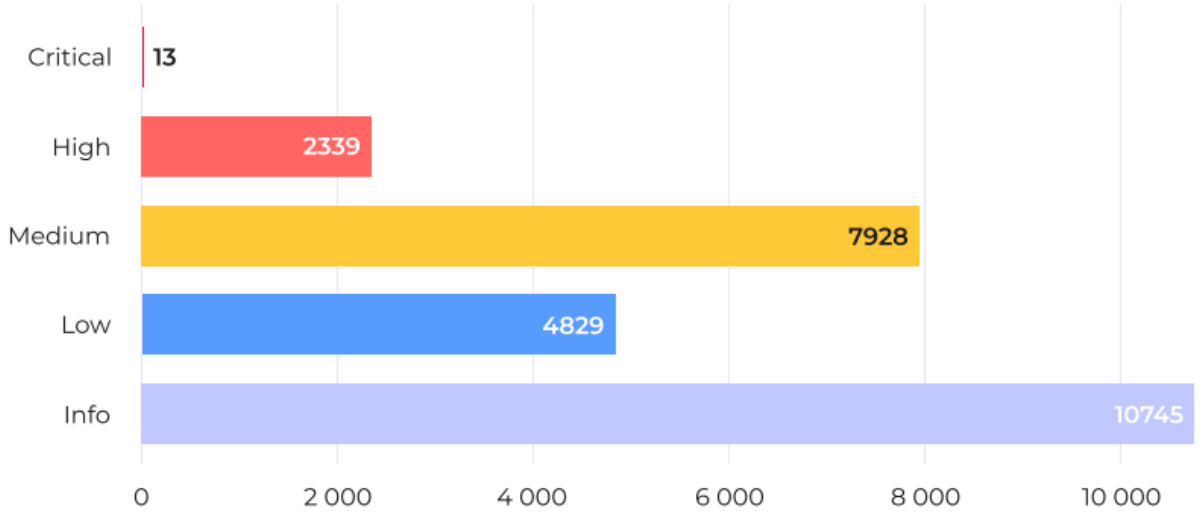
Почему это важно?

СТИНГРЕЙ

56%

мобильных приложений российских разработчиков содержат уязвимости уровней Critical и High

В 1816 приложениях было обнаружено 25854 уязвимости с разным уровнем критичности



Что можно уже сейчас включить в тестирование.

СТИНГРЕЙ

- Поиск секретов сторонних сервисов в собранном приложении
- Анализ правильной конфигурации приложения с точки зрения безопасности
- Анализ файла приложения на предмет лишних файлов
- Проверка обфускации в собранном приложении
- Список URL-адресов в собранном приложении
- Проверка Deeplink / Applink

Поиск секретов сторонних сервисов

СТИНГРЕЙ

- Мобильные приложения используют огромное количество внешних сервисов
- Для доступа к ним нужны токены/секреты
- Как правило они имеют намного больше привилегий, чем нужно

Примеры сервисов

- Firebase Cloud Messaging
- Firebase Remote Config
- Telegram (o_O)
- Facebook
- Twitter

Поиск секретов сторонних сервисов

СТИНГРЕЙ

Firebase Cloud Messaging

Чувствительная информация

AIzaSy[redacted]wo

Значение

AIzaSy[redacted]4wo

Путь

/data/data/ru[redacted]5b_0

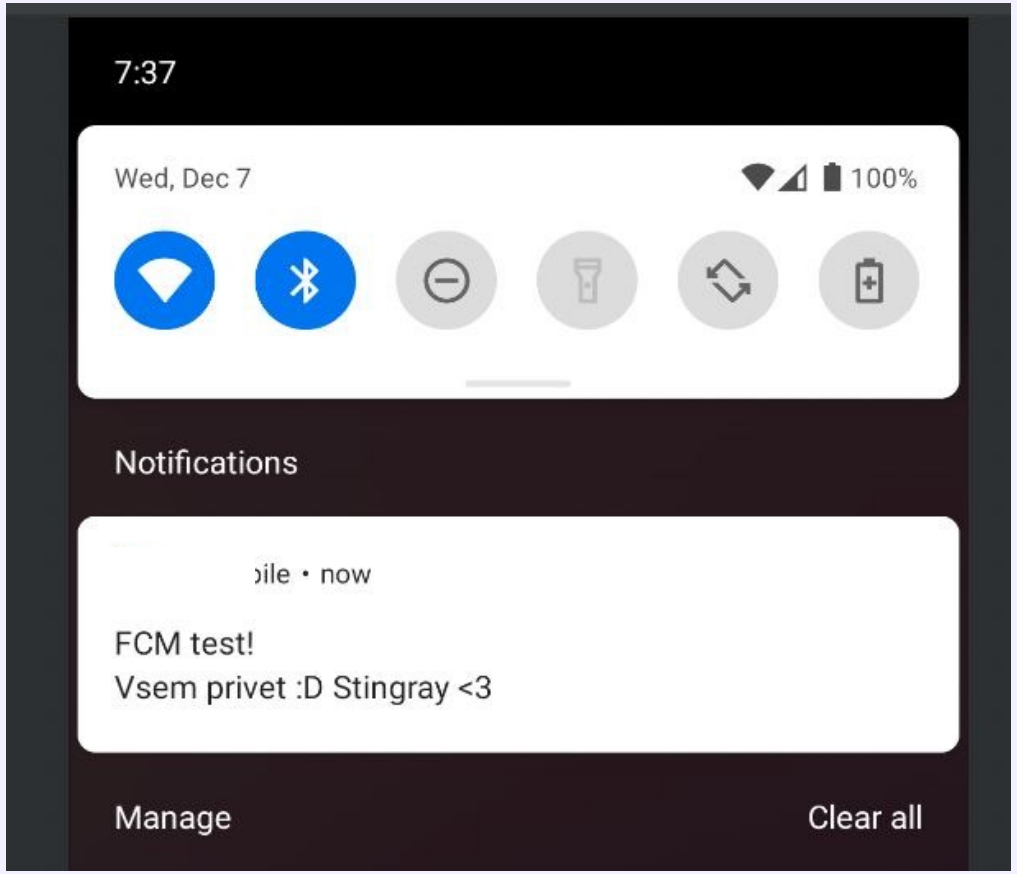
Результат проверки ключа для отправки Push-сообщений

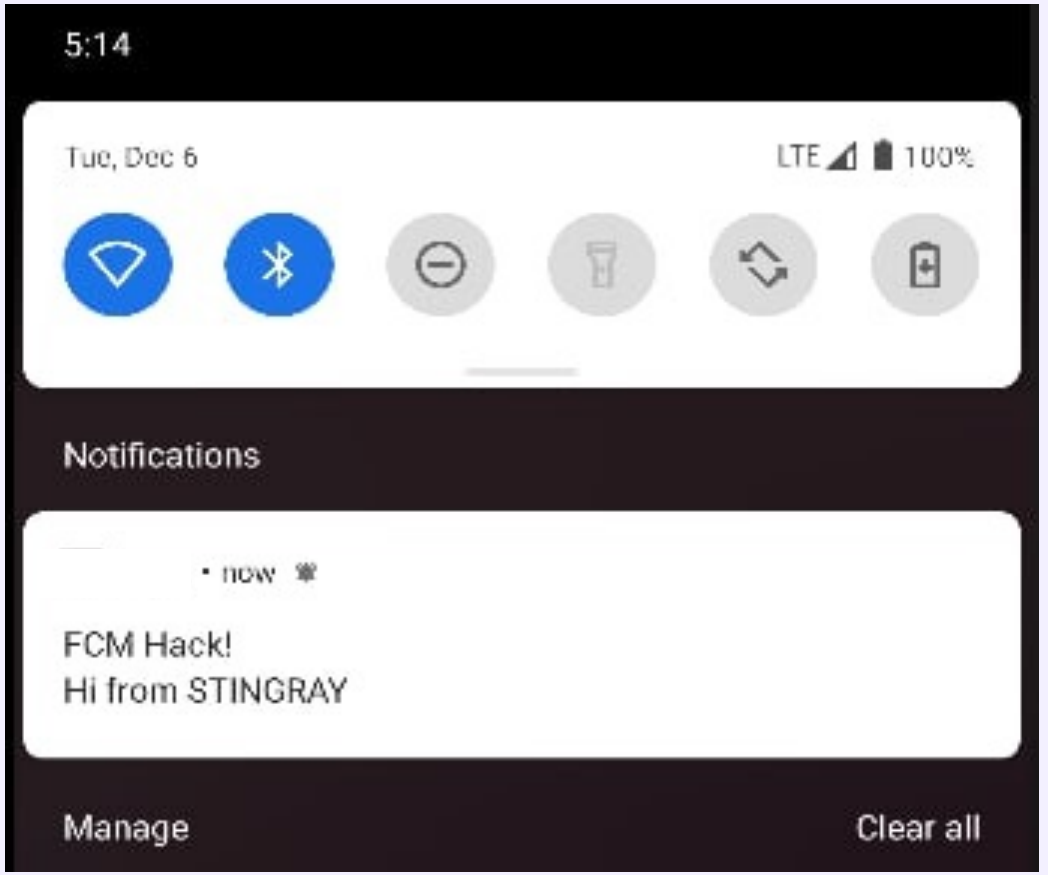

```
[
  {
    "url": "https://maps.googleapis.com/maps/api/staticmap?center=45%2C10&zoom=7&size=400x400&key=AIzaSyA
    "name": "Staticmap",
    "price": "$2/1.000",
    "status": "OK",
    "is_valid_key": true
  },
  {
    "url": "https://maps.googleapis.com/maps/api/streetview?size=400x400&location=40.720032,-73.988354&fo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

Поиск секретов сторонних сервисов

СТИНГРЕЙ

Возможность отправки произвольных PUSH-уведомлений





Поиск секретов сторонних сервисов

СТИНГРЕЙ

Как найти и проверить?

- Keyhacks
  https://github.com/streaak/keyhacks
- RegHex
  https://github.com/l4yton/RegHex
- Detect-secrets
  https://github.com/Yelp/detect-secrets
- DeepSecrets
  https://github.com/avito-tech/deepsecrets

Поиск секретов сторонних сервисов

СТИНГРЕЙ

$ jadx analyzed.apk

$ cd analyzed && detect-secrets scan --all-files


```
"results": {
  "resources/assets/priv_key": [
    {
      "type": "Private Key",
      "filename": "resources/assets/priv_key",
      "hashed_secret": "27c6929aef41ae2bcadac15ca6abcaff72cda9cd",
      "is_verified": false,
      "line_number": 1
    }
  ],
  "sources/androidx/autofill/HintConstants.java": [
    {
      "type": "Secret Keyword",
      "filename": "sources/androidx/autofill/HintConstants.java",
      "hashed_secret": "283d47a9338ed1100b5fe2a5aff2d1f7c799bfd0",
      "is_verified": false,
      "line_number": 20
    },
    {
      "type": "Secret Keyword",
      "filename": "sources/androidx/autofill/HintConstants.java",
      "hashed_secret": "5baa61e4c9b93f3f0682250b6cf8331b7ee68fd8",
      "is_verified": false,
      "line_number": 22
    }
  ],
  "sources/androidx/compose/ui/input/key/Key.java": [
    {
      "type": "Base64 High Entropy String",
      "filename": "sources/androidx/compose/ui/input/key/Key.java",
      "hashed_secret": "24d95a3a7d953eb30b43299d41478b55acbbcd13",
      "is_verified": false,
      "line_number": 345
    },
```

Анализ правильной конфигурации приложения с точки зрения безопасности

СТИНГРЕЙ

- Возможность создания резервной копии приложения
- Небезопасная конфигурация сетевого взаимодействия
- Небезопасная конфигурация в Info.plist
- Небезопасная навигация (Jetpack)

Анализ манифеста

СТИНГРЕЙ

Приложение Android, собранное с включенной опцией создания резервной копии (флаг android:allowBackup = True в ***AndroidManifest.xml***), может позволить злоумышленнику, имея физический доступ к устройству:

- Создать backup приложения со всеми данными, хранящимися во внутренней директории приложения.
- Возможность изменить информацию, содержащуюся в файлах и восстановить настройки из измененного backup

Это может повлечь за собой компрометацию пользовательских данных.

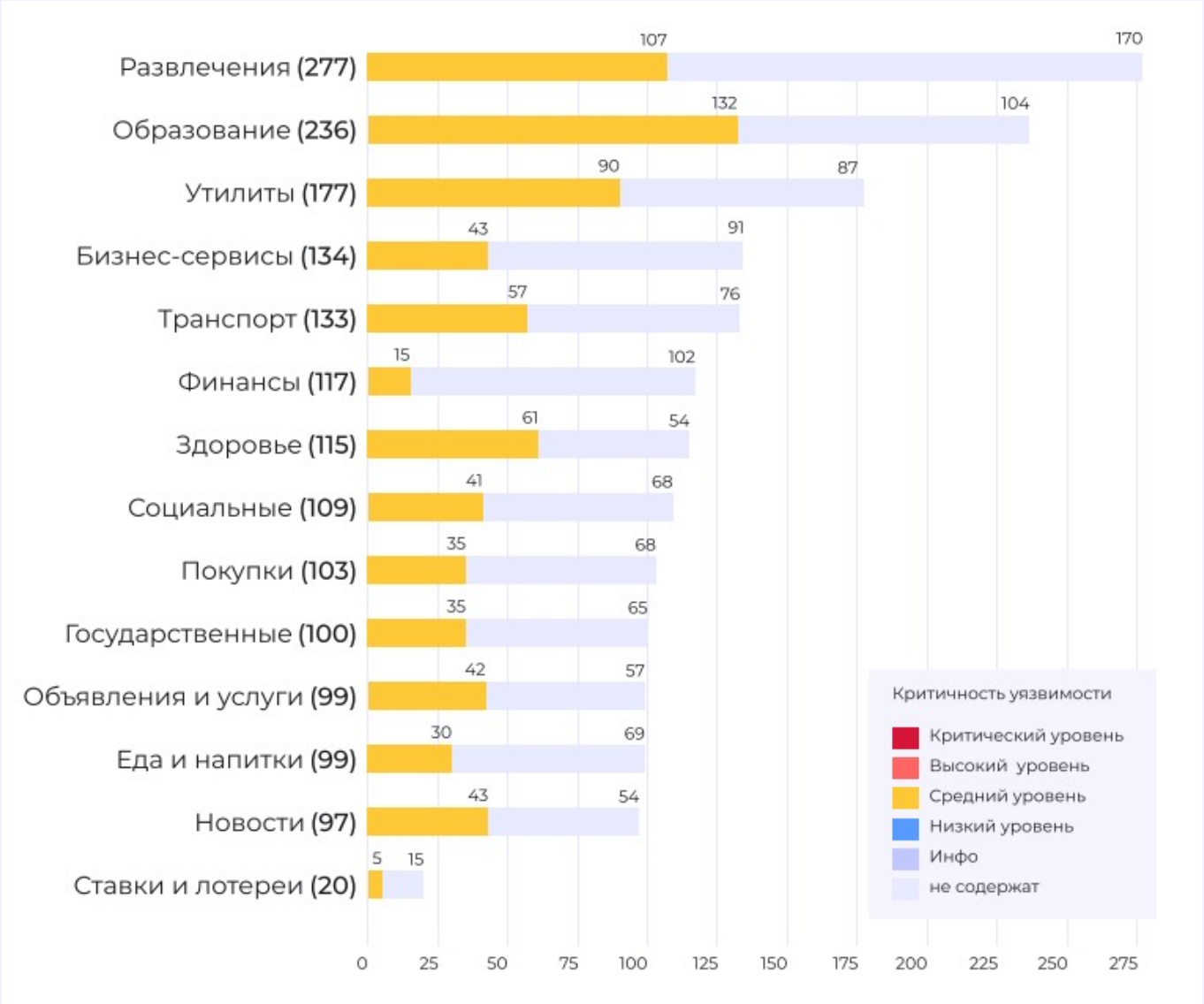
Анализ манифеста

СТИНГРЕЙ

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.appsec.android.activity.privateactivity" >
<application
    android:allowBackup="true"
    android:icon="@drawable/ic_launcher"
    android:label="@string/app_name" >
    <activity
        android:name=".PrivateActivity"
        android:label="@string/app_name"
        android:exported="false"/>
    </application>
</manifest>
```

Анализ манифеста

СТИНГРЕЙ



Анализ манифеста

СТИНГРЕЙ

$ apktool d your_app.apk -o output_folder

```python
import xml.etree.ElementTree as ET


def check_allow_backup(manifest_path):
    # Load and parse the AndroidManifest.xml file
    tree = ET.parse(manifest_path)
    root = tree.getroot()

    # Namespace needed for searching through the XML
    ns = {'android': 'http://schemas.android.com/apk/res/android'}

    # Find the <application> element and check 'allowBackup'
    application = root.find('application')
    if application is not None:
        allow_backup = application.get('{http://schemas.android.com/apk/res/android}allowBackup')
        if allow_backup is not None:
            return allow_backup.lower() == 'true'

    return False

# Replace 'path_to_manifest' with the actual path to your AndroidManifest.xml
manifest_path = 'output_folder/AndroidManifest.xml'
backup_allowed = check_allow_backup(manifest_path)
print(f'allowBackup is set to True: {backup_allowed}')
```

Небезопасная конфигурация сетевого взаимодействия

СТИНГРЕЙ

Недостатки могут привести к:

- Значительному ослаблению защиты передаваемой информации
- Перехвату трафика и получению контроля над данными пользователя.

Также неправильная настройка способна раскрывать внутренние адреса серверов и помогать злоумышленникам в осуществлении дальнейших атак.

Небезопасная конфигурация сетевого взаимодействия

СТИНГРЕЙ

Результат

Скачать результат

```
      <certificates src="@7F1201EA"/>
      <certificates src="system"/>
    </trust-anchors>
  </base-config>
  <domain-config cleartextTrafficPermitted="true">
    <domain includeSubdomains="false">lk.[redacted]ru</domain>
    <domain includeSubdomains="false">anket[redacted]ru</domain>
    <domain includeSubdomains="false">develop.ci[redacted]ink</domain>
    <domain includeSubdomains="false">preprod.ci[redacted]ink</domain>
    <domain includeSubdomains="false">test[redacted].ru</domain>
    <domain includeSubdomains="false">dfa.[redacted]u</domain>
  </domain-config>
</network-security-config>
"
```

Небезопасная конфигурация сетевого взаимодействия

СТИНГРЕЙ

Минимум, на что обращать внимание:

- Отсутствие тестовых/внутренних доменов
- Блок <trust-anchors> - не должно быть значения user
- Атрибут cleartextTrafficPermitted не должен быть выставлен в true

Небезопасная конфигурация сетевого взаимодействия

СТИНГРЕЙ

```python
import xml.etree.ElementTree as ET


def parse_network_security_config(config_path):
    # Load and parse the network_security_config.xml
    tree = ET.parse(config_path)
    root = tree.getroot()

    # Check for trust anchors without "user" value
    user_trust_anchor_found = False
    trust_anchors = root.findall(".//trust-anchors")
    for anchor in trust_anchors:
        for cert in anchor:
            if 'source' in cert.attrib and cert.attrib['source'] == "user":
                user_trust_anchor_found = True
                break

    # Check if cleartextTrafficPermitted is set to true
    cleartext_traffic_permitted = False
    base_configs = root.findall(".//base-config")
    for config in base_configs:
        cleartext = config.find('cleartextTrafficPermitted')
        if cleartext is not None and cleartext.attrib.get('android:value', 'false') == 'true':
            cleartext_traffic_permitted = True
            break

    return user_trust_anchor_found, cleartext_traffic_permitted

# Replace 'path_to_config' with the actual path to your network_security_config.xml
config_path = 'path_to_config/network_security_config.xml'
user_trust, cleartext_permitted = parse_network_security_config(config_path)
print(f'User trust anchor found: {user_trust}')
print(f'Cleartext traffic permitted: {cleartext_permitted}')
```

Небезопасная конфигурация в Info.plist

Info.plist – основной файл для iOS-приложений внутри него можно найти много интересной информации:

- Настройки App Transport Security
- Applinks / Deeplinks
- Токены от сторонних сервисов
- Все, что захочет туда положить разработчик

Небезопасная конфигурация в Info.plist

Настройки App Transport Security

- Разархивируем .ipa файл
- В директории находим Info.plist
- Изучаем его содержимое и секцию NSAppTransportSecurity, в частности параметр NSAllowsArbitraryLoads

Небезопасная конфигурация в Info.plist

СТИНГРЕЙ

```python
import plistlib


def check_ats_enabled(plist_path):
    """
    Checks if App Transport Security (ATS) is enabled in an iOS app's Info.plist file.

    Args:
    plist_path (str): Path to the Info.plist file.

    Returns:
    bool: True if ATS is enabled, False otherwise.
    """
    try:
        # Load the plist file
        with open(plist_path, 'rb') as plist_file:
            plist_data = plistlib.load(plist_file)

        # Check the ATS settings
        ats_settings = plist_data.get('NSAppTransportSecurity', {})
        # If NSAllowsArbitraryLoads is True, ATS is disabled
        if ats_settings.get('NSAllowsArbitraryLoads', False):
            return False

        # Additional checks for more fine-grained ATS settings
        if ats_settings.get('NSAllowsArbitraryLoadsForMedia', False) or \
                ats_settings.get('NSAllowsArbitraryLoadsInWebContent', False):
            return False

        return True
    except Exception as e:
        print(f"An error occurred: {e}")
        return False


# Usage
plist_path = 'path/to/your/Info.plist'
is_ats_enabled = check_ats_enabled(plist_path)
print("App Transport Security is enabled:", is_ats_enabled)
```

Небезопасная навигация (Jetpack)

СТИНГРЕЙ

В Андроид для навигации между фрагментами можно использовать NavController, которому передается ресурс с графом навигации (***/res/navigation/some_file.xml***).

Проблема в том, что если в теге <navigation> используется атрибут android:id, то к указанным фрагментам возможен доступ через диплинки.

Таким образом стороннее приложение может открыть **любой** фрагмент приложения, указанный в файле (-ах) навигации

```
<navigation xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:app="http://schemas.android.com/apk/res-auto"
    xmlns:tools="http://schemas.android.com/tools"
    android:id="@+id/mobile_navigation"
    app:startDestination="@+id/navigation_home">

    <fragment
        android:id="@+id/navigation_home"
        android:name="ru.ptsecurity.navigation_example.ui.home.HomeFragment"
        android:label="@string/title_home"
        tools:layout="@layout/fragment_home">
        <action android:id="@+id/toFirst"
            app:destination="@+id/navigation_first">
        </action>
    </fragment>
```

https://swarm.ptsecurity.com/android-jetpack-navigation-deep-links-handling-exploitation/

Небезопасная навигация (Jetpack)

СТИНГРЕЙ

Как найти:

1. Определяем, что в приложении используется Jetpack Navigation
    1. Декомпилируем приложение
    2. Ищем в ресурсах XML у которого есть тег <navigation>

2. Если в этом xml в тегах <fragment> указан android.id это уже повод указать на потенциальную проблему

3. Попробовать запустить приватный экран приложения

```
$ adb shell
$ am start -n com.example.app/.HostActivityClassName --eia "android-support-nav:controller:deepLinkIds"
0xA,0xB
```

Небезопасная навигация (Jetpack)

Эта уязвимость позволяет открывать любые экраны приложения в обход всех проверок безопасности.

Но только в случае, если **каждый фрагмент** не проверяет аутентификацию.

Анализ файла приложения на предмет лишних файлов

СТИНГРЕЙ

- Файлы зависимостей
- Файлы сборки
- Дебаг-сертификаты
- Mock-сервисы
- Вспомогательные файлы
- Read.me

Анализ файла приложения на предмет лишних файлов

СТИНГРЕЙ

Что это может дать:

- Получение информации о компонентах приложения – поиск уязвимостей в них
- Внутренняя информация (подготовка фишинга)
  - Тестовые учетные записи
  - Контакты команды разработки
  - Личные аккаунты
  - Корпоративные учетные записи
- Информация об API (Mock-сервисы)
- Переключение приложения в dev-режим

Анализ файла приложения на предмет лишних файлов

СТИНГРЕЙ

- Jadx
https://github.com/skylot/jadx

- Apktool
https://apktool.org/

- APKEditor
https://github.com/REAndroid/APKEditor

- ZIP :)

Анализ файла приложения на предмет лишних файлов

СТИНГРЕЙ

- Алгоритм анализа Android

```
$ apktool d your_app.apk -o output_folder
```

OR

```
$ jadx analyzed.apk
```

- Алгоритм анализа iOS

Unzip .ipa file :D

Анализ файла приложения на предмет лишних файлов

СТИНГРЕЙ

```
13  ## Установка
14
15  Для установки необходимо добавить репозиторий с спецификациями Pod'ов в Podfile:
16  ```ruby
17  source 'https://gitlab.serv                    le-sdk/ios/podspecs.git'
18  source 'https://cdn.cocoapods.org/'
19  # source 'https://github.com/CocoaPods/Specs.git' замените строку выше если не можете использовать CDN по какой-то причине
20  ```
```

```
74  ## Команда разработки
75
76  Te                    ей, at                    ru
77
    К                     й, aa                     ru
```

```
29      <dict>
30          <key>description</key>
31          <string>🛠 Разрешить навигацию для всех событий и площадок</string>
32          <key>enabled</key>
33          <false/>
34          <key>jiraTaskURL</key>
35          <string></string>
36          <key>key</key>
37          <string>forceAllowNavigation</string>
38          <key>kind</key>
39          <string>simple</string>
40      </dict>
```

Проверка обфускации в собранном приложении

СТИНГРЕЙ

Отсутствие обфускации приводит:

- К более простому анализу логики приложения
- Проще найти уязвимости
- Проще обойти локальные проверки

Проверка обфускации в собранном приложении

СТИНГРЕЙ

https://github.com/liansecurityOS/apk-obfucation-detection

APK Information

APK Name:	易购
Possible Malware:	No
Package:	com.adegvbaerdg.aedrgaedr
MD5:	fe9815413730a9dbdc629d108547006e
File Size:	79.76MB
Shell:	腾讯
TLSH:	T133081271C380BECDFAB365BBC4219A92E017AD31662352AED445F6A1C27B196CFC3D05
SSDEEP:	1572864:vy5lWtQgeLQ0dPonWlvnQKTN8XOJ8tYlsdF8lLTKA188mXhRKzw2CwfALdTgk:IjxLQ9nWlvnQvXOJYYfUKY88vCX93
TrID:	Java Archive(78.3%) ZIP compressed archive(21.6%)
Obfuscation:	Coverage 40%
Report Time:	2024-01-27 14:51:14

Проверка обфускации в собранном приложении

СТИНГРЕЙ

Список URL-адресов в собранном приложении

СТИНГРЕЙ

На что обратить внимание:

- Внутренние домены
- Тестовые сервера
- Утечки данных

Можно дополнительно проверить на зловредные домены:
https://github.com/stamparm/blackbook

Список URL-адресов в собранном приложении

СТИНГРЕЙ

- Данные о внутренних доменах
  Помогут для составления атаки внутри

- Данные о серверах доступных снаружи.
  Тестовые сервера могут оказаться внге ведения инструментов защиты – хуже защищены, проще взломать, иногда используют базы с прома

- Домены, к которым могут обращаться библиотеки
  Потенциально утечки информации

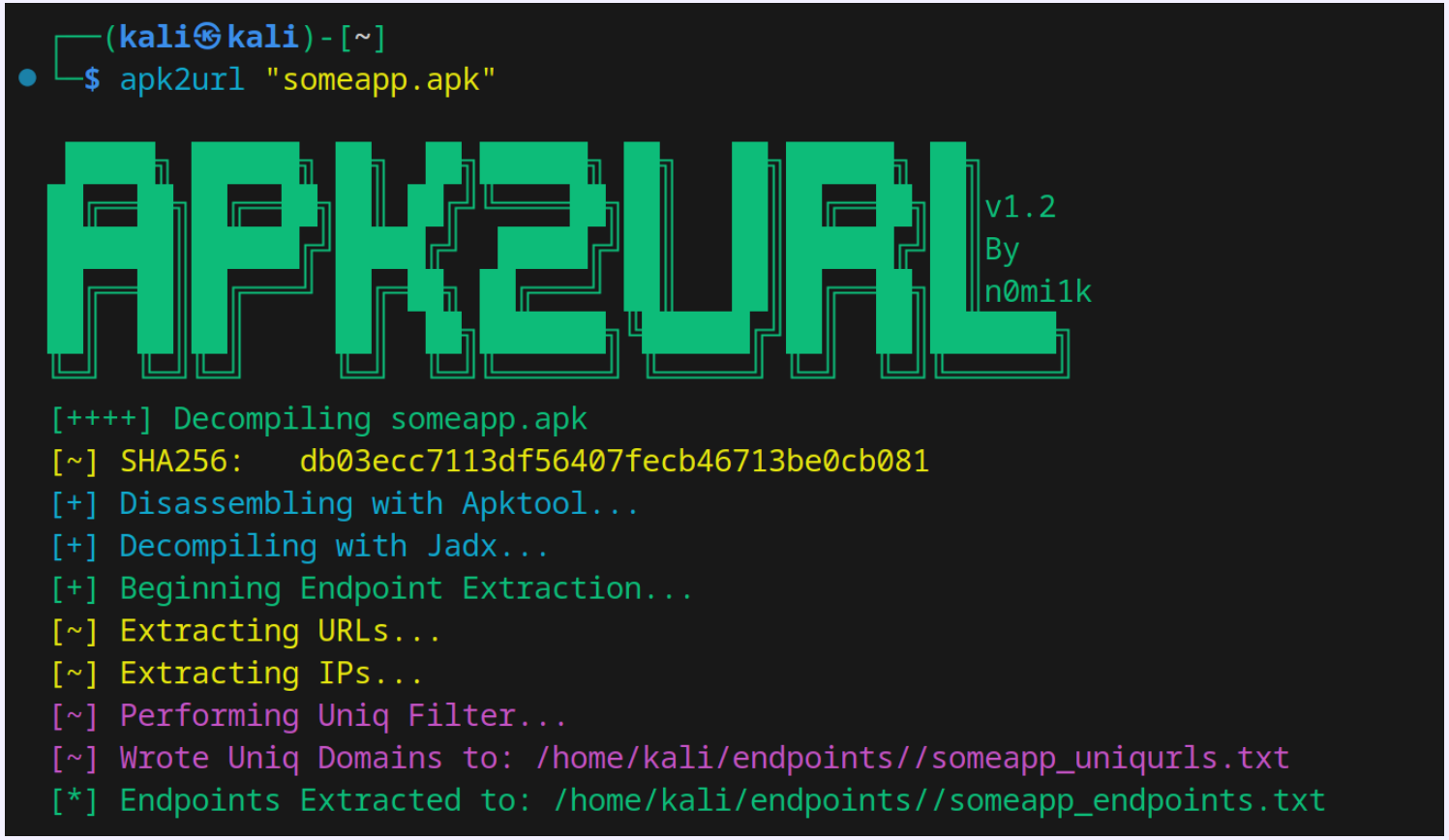
- Зловредные домены (атаки на цепочку поставок или атаки по геопозиции)
  Библиотеки могут использовать или содержать различные домены и обращаться к C&C серверам

Список URL-адресов в собранном приложении

СТИНГРЕЙ

Очень удобный инструмент apk2url

https://github.com/n0mi1k/apk2url



Список URL-адресов в собранном приложении

СТИНГРЕЙ

Для iOS алгоритм аналогичен, за исключением того, нужно самостоятельно распаковать приложение

```
$ unzip TEST.ipa

$ cd TEST/Payload/test.app

$ strings test > strings_from_binary.txt
```

После этого применить алгоритм из apk2url и поиск по всем файлам внутри директории приложения

Выводы

СТИНГРЕЙ

Безопасность мобильных приложений это задача всей команды, а не только безопасников

Интегрировать проверки безопасности в процесс тестирования просто и это дает колоссальный эффект.

Примеры в презентации только начало, можно сделать много чего еще, что поможет защитить ваших пользователей

Полезные материалы

СТИНГРЕЙ

- https://github.com/Swordfish-Security/awesome-android-security
- https://github.com/Swordfish-Security/awesome-ios-security
- https://t.me/mobile_appsec_world

СТИНГРЕЙ

Юрий Шабалин

https://t.me/mobile_appsec_world

https://stingray-mobile.ru